cause transaction enabler 160 to display $4300 (representing an increase in the present highest bid).

The user may select 'Bid History' to view the previous bidders and history. The relevant data may either be displayed based on data stored locally or the data may be retrieved from web site 130 in response to a user request. As is well known in the relevant arts, auction sites such as www.ebay.com provide such bid histories.

The user may specify her/his bid price in the box provided next to text 'Your Bid'. The user may then select the 'Submit' text to cause transaction enabler 160 to submit the bid. As noted above, the submission may be according to any mechanism. The bid can potentially be over a broadband interface to access a web site or to a server accepting over a telephone connection. Once the bid is submitted to a server at the access address, the auction item may be sold to a bidder in a known way. If the user of system 150 has the highest bid, the user may pay the bid amount and receive the auction item.

Thus, an interface such as the one above, a user (or television viewers) may bid for auction items in accordance with the present invention. The bid may be submitted according to any pre-specified protocol between transaction enabler 160 and an auction server (e.g., web site 130). The implementation of auction on web site 130 based on such received bid prices will be apparent to one skilled in the relevant arts.

8. Conclusion

While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

What Is Claimed Is:

1. A method of enabling a viewer of a television system to participate in auctions, said method comprising:

(a) encoding in a television signal a data describing an auction item and an access address of a server at which auction service for said auction item is provided; and

(b) transmitting said television signal,

wherein said data can be used to enable said viewer to bid for said auction item at said server.

2. The method of claim 1, wherein said method further comprises:

(c) receiving said television signal encoded with said data in a transaction enabler;

(d) recovering said data encoded in said television signal;

(e) displaying information describing said auction item on said television system;

(f) enabling said viewer to bid at said server specified by said access address.

3. The method of claim 2, further comprising:

(g) enabling said viewer to specify a bid price for said auction item.

4. The method of claim 3, wherein said enabling said viewer to specify said bid price comprises:

(h) enabling said viewer to indicate said bid price; and

(i) transmitting said bid price to said server at said access address.

5. The method of claim 4, wherein said access address comprises a telephone number

of said server, and said method further comprises:

(j) encoding a unique code identifying said auction item;

(k) recovering said unique code in said transaction enabler; and

(l) transmitting said unique code along with said bid price to said server,

whereby said server can easily associate said bid price with said auction item using said unique code.

6. The method of claim 4, wherein said access address comprises a universal resource locator (URL) of a web site, wherein said web site comprises said server, and wherein steps (h) and (i) comprise the further step of enabling said viewer to indicate said price on a web page provided by said web site.

7. The method of claim 1, further comprising:

(m) encoding a present highest bid in said television signal, wherein said present highest bid may be displayed to said viewer before said viewer decides to submit a bid.

8. The method of claim 7, wherein said server comprises a web site, and said method comprising the further step of retrieving said present highest bid from said web site.

9. The method of claim 1, wherein step (a) comprises the step of encoding said data in non-display portion of said television signal.

10. The method of claim 1, wherein step (a) comprises the further step of encoding said data in a non-display portion of said television signal.

11. The method of claim 10, wherein said non-display portion comprises vertical blanking interval (VBI).

12. The method of claim 1, further comprising:

transmitting an updated highest bid price in said television signal, wherein said updated highest bid price corresponds to a present highest bid for said auction item.

13. The method of claim 12, further comprising:

retrieving said updated bid price from said server,

wherein said step of transmitting said updated highest bid price is performed after said step of retrieving said updated bid price from said server.

14. The method of claim 13, further comprising:

enabling said viewer to request a bid history; and

displaying all of said updated bid prices to said viewer.

15. The method of claim 14, wherein said display corresponding to said bid history further comprises a description of the bidder corresponding to each of said present highest bid.

16. The method of claim 1, wherein said data further comprises a time at which auction for said auction item closes.

17. A method of enabling a viewer of a television system to participate in auctions,

said method comprising:

(a) receiving in a transaction enabler a television signal encoded with a data, said data including a description of an auction item and an access address of a server at which auction service for said auction item is provided;

(b) recovering said data encoded in said television signal;

(c) displaying said description of said auction item on said television system;

(d) enabling said viewer to bid at said server specified by said access address.

18. The method of claim 17, further comprising:

(e) enabling said viewer to indicate said bid price; and

(f) transmitting said bid price to said server at said access address.

19. The method of claim 4, wherein said access address comprises a telephone number of said server, and said method further comprises:

(g) encoding a unique code identifying said auction item;

(h) recovering said unique code in said transaction enabler; and

(i) transmitting said unique code along with said bid price to said server,

whereby said server can easily associate said bid price with said auction item using said said unique code.

20. An environment enabling a viewer of a television system to participate in auctions, said environment comprising:

encoding means for encoding in a television signal a data describing an auction item and an access address of a server at which auction service for said auction item is provided; and

transmission means for transmitting said television signal,

wherein said data can be used to enable said viewer to bid for said auction item at said server.

21. An environment enabling a viewer of a television system to participate in auctions, said environment comprising:

receiving means for receiving a television signal encoded with a data, said data including a description of an auction item and an access address of a server at which auction service for said auction item is provided;

recovery means for recovering said data encoded in said television signal;

displaying means for displaying said description of said auction item on said television system;

enabling means for enabling said viewer to bid at said server specified by said access address.

22. An environment enabling a viewer of a television system to participate in auctions, said environment comprising:

a broadcast system to encode in a television signal a data describing an auction item and an access address of a server at which auction service for said auction item is provided, said broadcast system being designed also to transmit said television signal,

wherein said data can be used to enable said viewer to bid for said auction item at said server.

23. The environment of claim 22, wherein said broadcast system comprises:

a production block to generate images to encode in a display data portion of said television signal;

an authoring block to encode said data in said television signal; and

a broadcast block to transmit said television signal containing said images and said data.

24. The environment of claim 23, further comprising an auction data interface to receive a present highest bid from a server, said auction data interface to provide said present highest bid to said authoring block, wherein said authoring block encodes said present highest bid in said television signal.

25. The environment of claim 24, further comprising a timing determination block to determine the time at which said authoring block encodes said data including said present highest bid in said television signal.

26. The environment of claim 22, further comprising:

a viewer bidding system to receive said television signal, and enabling said viewer to submit a bid and participate in said auction.

27. The environment of claim 26, wherein said viewer bidding system comprises:

a television system;

a remote control which enables said viewer to submit said bid; and a transaction enabler coupled to said television system and to receive said commands from said remote control, said transaction enabler to recover said data encoded in said television signal and display information contained in said data on said television,

wherein said viewer can submit said bid using said remote control.

28. The environment of claim 27, wherein said transaction enabler is integrated within said television system.

29. The environment of claim 27, wherein said transaction enabler is provided external to said television system, and wherein said transaction enabler overlays a window with information contained in said data on images encoded in the display data of said television signal.

30. The environment of claim 27, wherein said window is displayed in a transparent mode on said images.

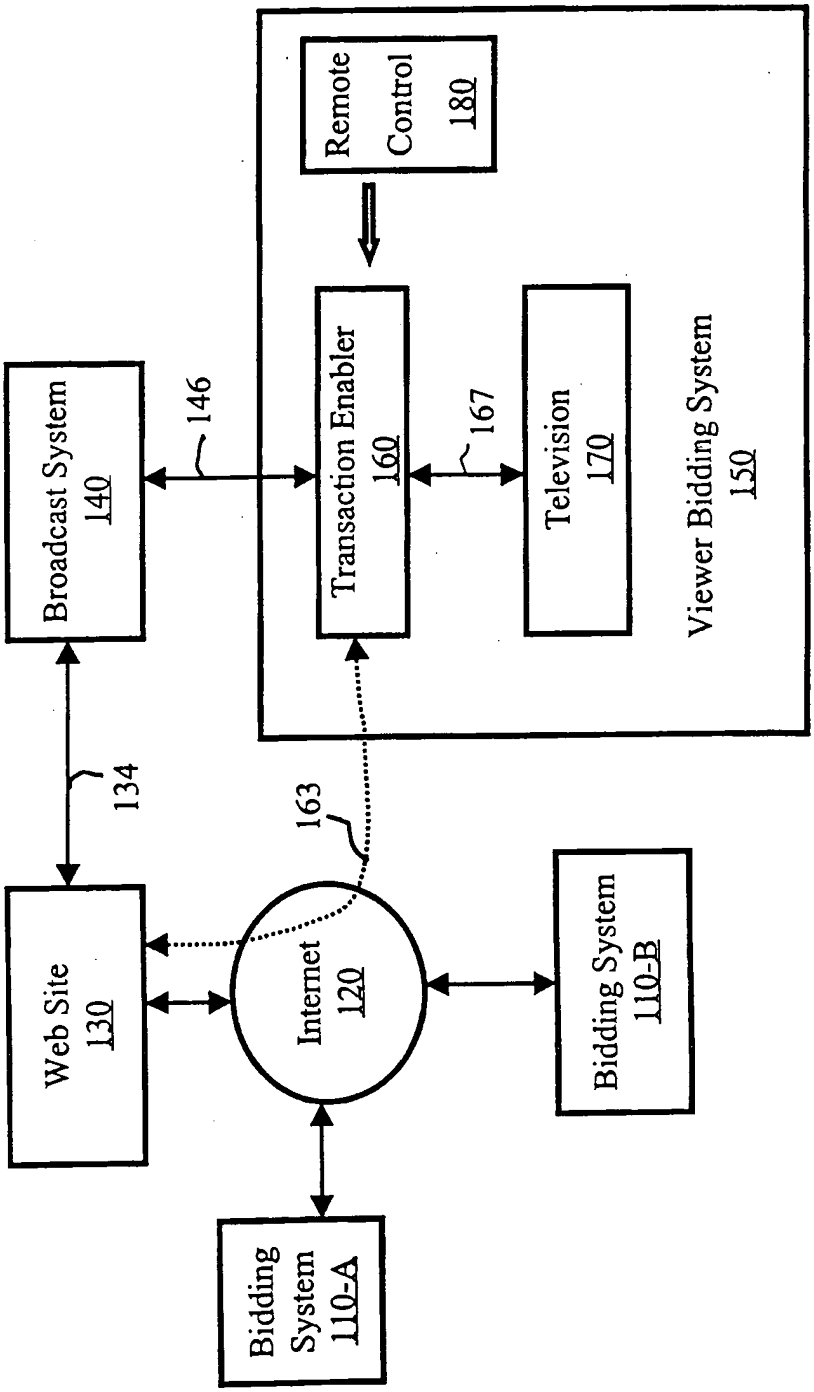

Remote Control 180
Broadcast System 140
146
Transaction Enabler 160
167
Television 170
Viewer Bidding System 150
134
163
Web Site 130
Internet 120
Bidding System 110-B
Bidding System 110-A

Figure 1

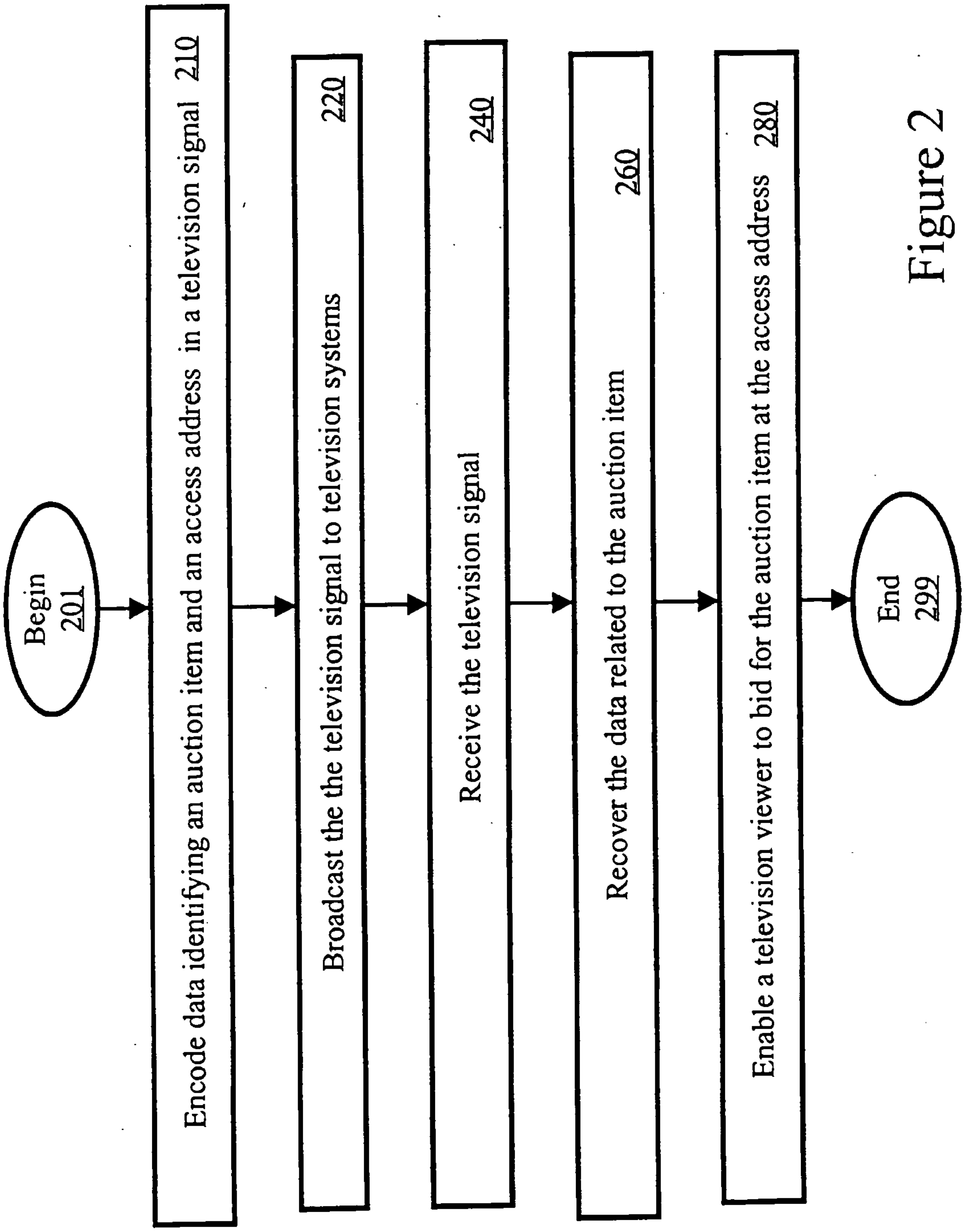

Begin
201
Encode data identifying an auction item and an access address in a television signal 210
Broadcast the the television signal to television systems 220
Receive the television signal 240
Recover the data related to the auction item 260
Enable a television viewer to bid for the auction item at the access address 280
End
299

Figure 2

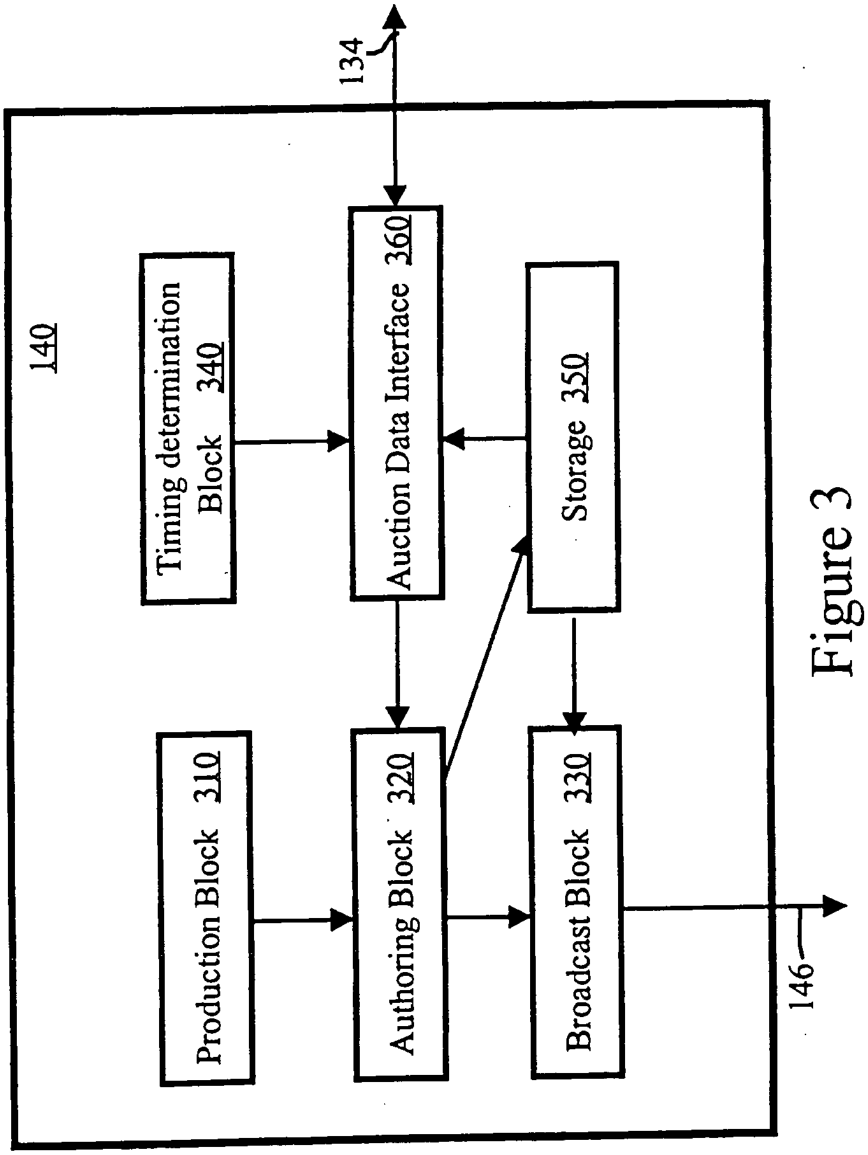

134
140
Timing determination Block 340
Auction Data Interface 360
Storage 350
Production Block 310
Authoring Block 320
Broadcast Block 330
146

Figure 3

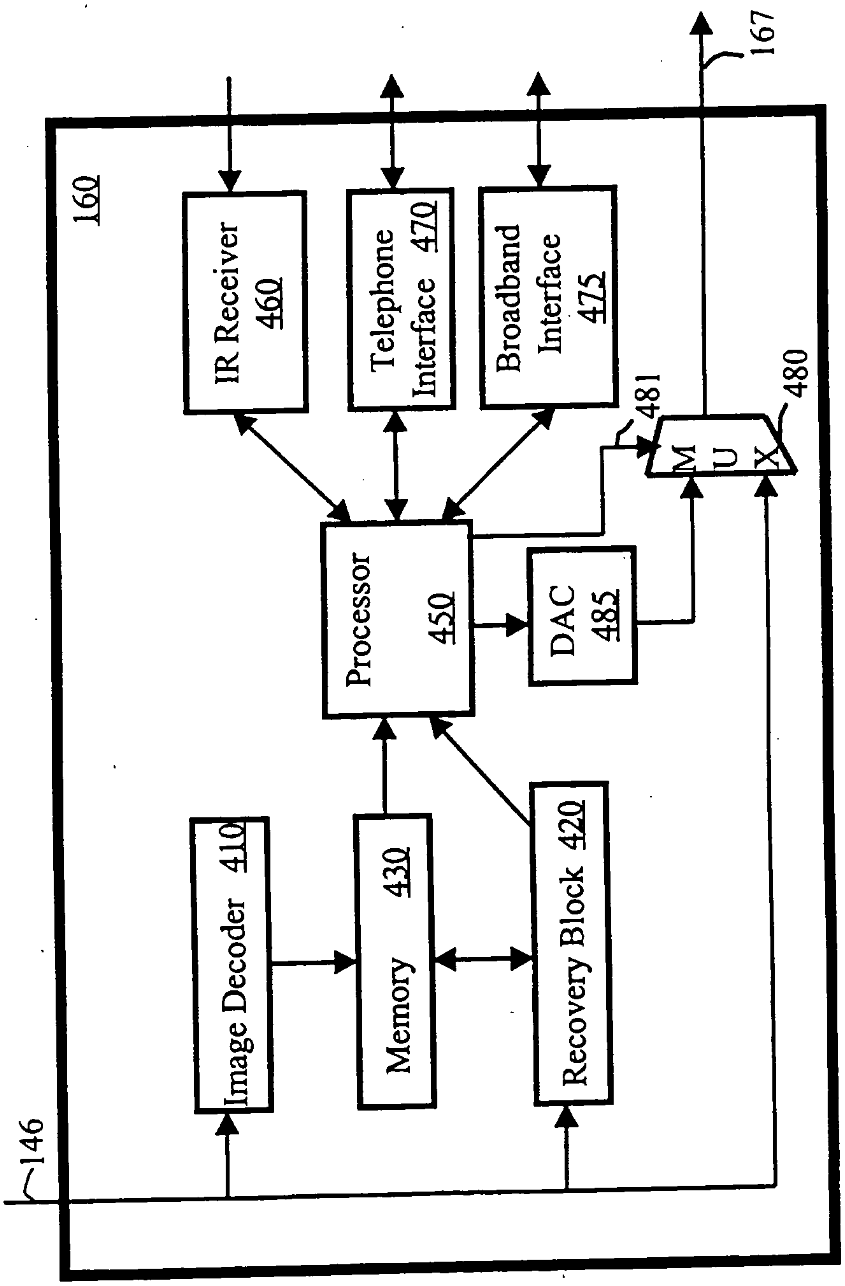

160
IR Receiver 460
Telephone Interface 470
Broadband Interface 475
167
481
480
MUX
Processor 450
DAC 485
Image Decoder 410
Memory 430
Recovery Block 420
146

Figure 4

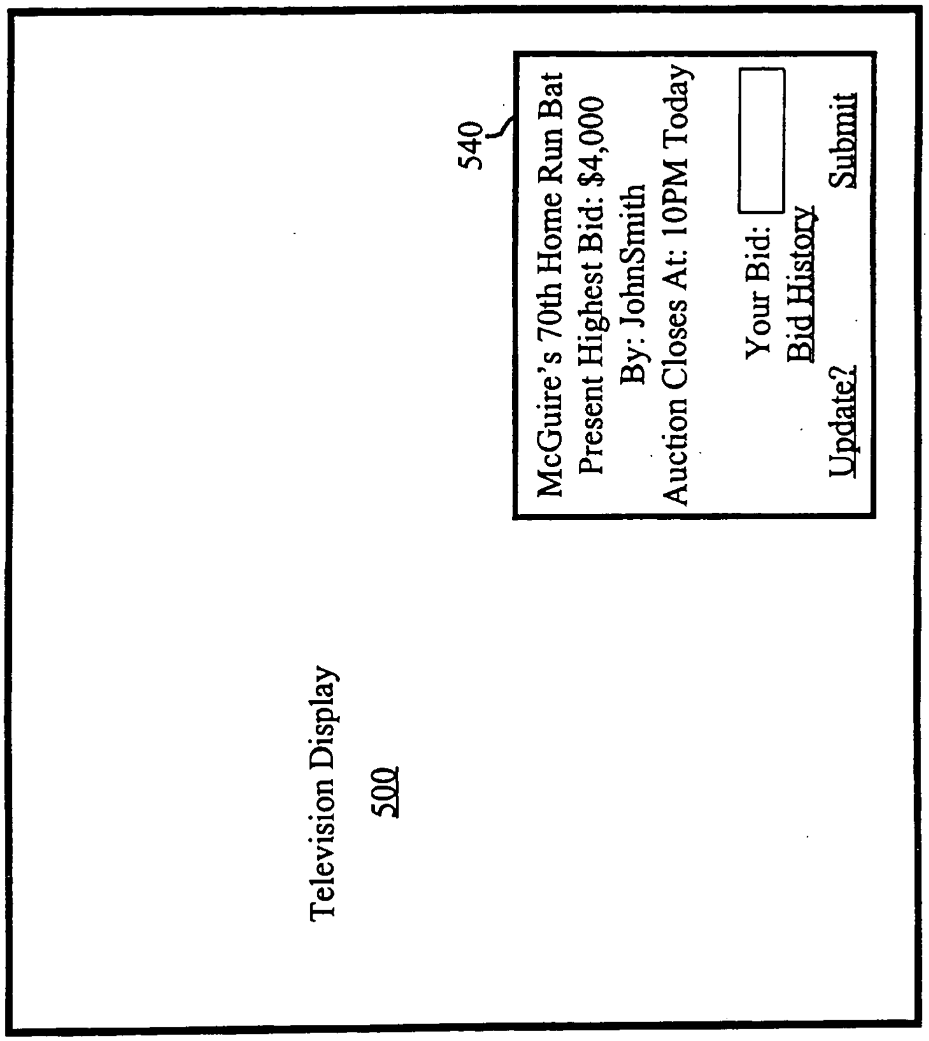

Television Display
500
540
McGuire's 70th Home Run Bat
Present Highest Bid: $4,000
By: JohnSmith
Auction Closes At: 10PM Today
Your Bid:
Bid History
Update?
Submit

Figure 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/18510

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 17/60
US CL : 705/26, 27, 37
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 705/26, 27, 37

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Please See Extra Sheet.

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EAST, CORPORATE RESOURCE NET

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	Auction Goes Upscale. Capital District Business Review. April 17, 1995. Vol. 22. Issue 1. page 43.	1-30
Y,E	Strategic Partnership Between ExtraLot.com and The Auction Channel. Business Wire. August 11, 2000.	1-30
Y	Auctioneer Onsale to Broadcast Live Commercials on ZDTV. Electronic Advertising and Marketplace Report. October 6, 1998. Vol 12. Issue 18. page 4.	1-30
Y	Philadelphia Business Journal. Auction Television Does $1 Million Stock Placement. January 29, 1999. Vol. 17. Issue 51. page 36.	1-30

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 22 AUGUST 2000	Date of mailing of the international search report 18 SEP 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer JAMES TRAMMELL Eugenia Logan Telephone No. (703) 305-[illegible]60

Form PCT/ISA/210 (second sheet) (July 1998)★

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/18510

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,905,975 A (AUSUBEL) 18 May 1999, col 3, lines 1-30.	1-30
Y	MARQUEZ, RACHELLE. New Dimension For Auction. 15 September 1997. Vol. 15. Issue 20. page 38.	1-30

Form PCT/ISA/210 (continuation of second sheet) (July 1998)★

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/18510

B. FIELDS SEARCHED
Documentation other than minimum documentation that are included in the fields searched:

NEWTON'S TELECOM DICTIONARY
McGRAW-HILL ENCYCLOPEDIA OF ELECTRONICS AND COMPUTERS

(12) UK Patent Application (19) GB (11) 2 354 102 (13) A

(43) Date of A Publication 14.03.2001

(21) Application No 9921227.6

(22) Date of Filing 08.09.1999

(71) Applicant(s)
Barron McCann Limited
(Incorporated in the United Kingdom)
BeMac House, Fifth Avenue, LETCHWORTH, Hertfordshire, SG6 2HF, United Kingdom

(72) Inventor(s)
Peter Alderson
Robert Andrew Edge

(74) Agent and/or Address for Service
Williams, Powell & Associates
4 St Paul's Churchyard, LONDON, EC4M 8AY, United Kingdom

(51) INT CL^7
G07F 7/10 , G06F 17/60

(52) UK CL (Edition S)
G4V VAK

(56) Documents Cited
EP 0813175 A2 WO 98/32260 A1 WO 97/50207 A1
WO 97/29416 A2 US 5809143 A

(58) Field of Search
UK CL (Edition R) **G4V VAK , H4P PDCSA**
INT CL^7 **G06F 17/60 , G07F 7/10**
Online: WPI, EPODOC, JAPIO

(54) Abstract Title
System for communicating over a public network

(57) A system for communicating with a remote service over a public network 18, such as the Internet, includes a client device 10 with a memory card 28 or the like, a card reader 26 and a public network communication device such as a personal computer or television, and a processor unit, such as a central gateway 12, which is located remotely from the client device. The memory card includes user details which are transmitted by the client device to the processor unit, and may be encrypted. The card reader may activate communication with the processor unit upon insertion of the memory card, which may be a smart card or magnetic card. The processor unit may determine which of a plurality of services 14,16 a user is authorised to access. The system provides for secure communication without burdening the user with encryption or authorisation tasks.

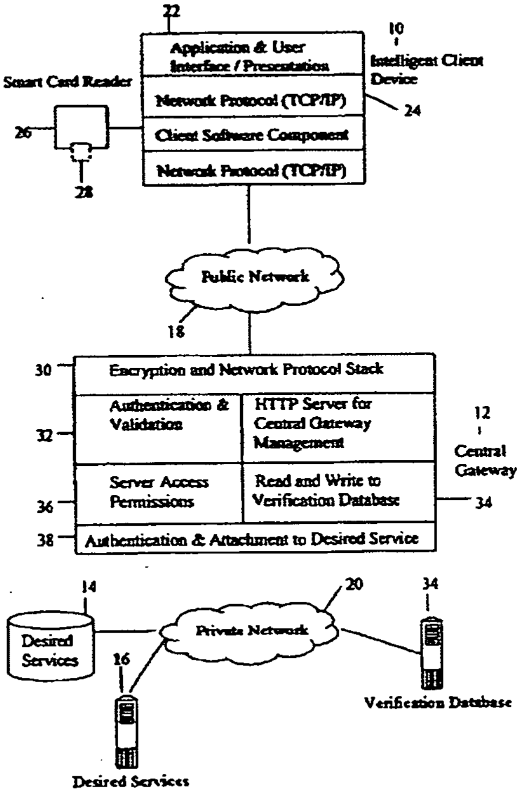

22
10
Application & User Interface / Presentation
Intelligent Client Device
Smart Card Reader
Network Protocol (TCP/IP)
24
26
Client Software Component
Network Protocol (TCP/IP)
28
Public Network
18
30
Encryption and Network Protocol Stack
Authentication & Validation
HTTP Server for Central Gateway Management
12
Central Gateway
32
Server Access Permissions
Read and Write to Verification Database
36
34
38
Authentication & Attachment to Desired Service
14
20
34
Desired Services
Private Network
16
Verification Database
Desired Services

Fig 1

GB 2 354 102 A

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

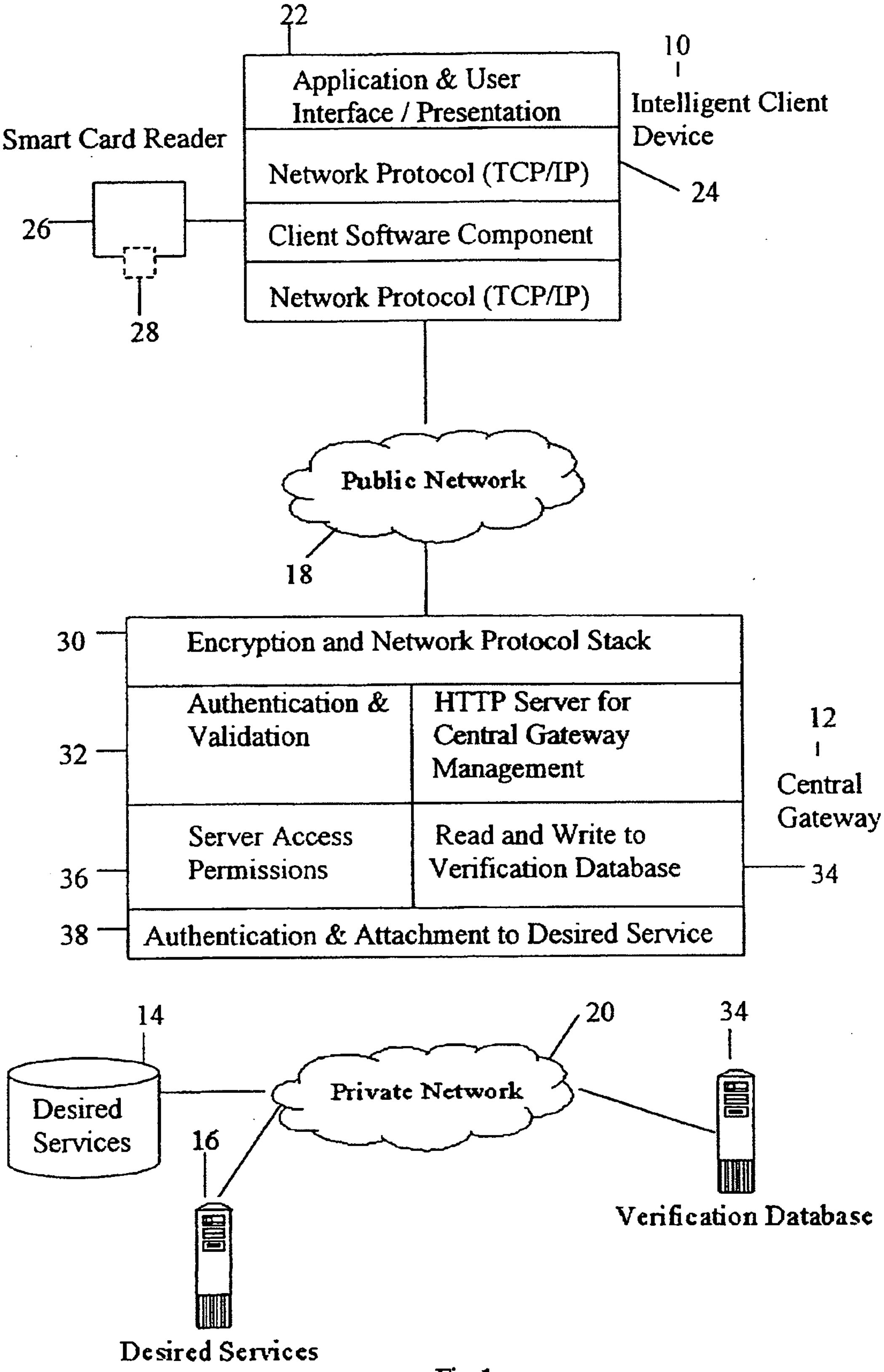

22
10
Application & User Interface / Presentation
Intelligent Client Device
Smart Card Reader
Network Protocol (TCP/IP)
24
26
Client Software Component
Network Protocol (TCP/IP)
28
Public Network
18
30
Encryption and Network Protocol Stack
Authentication & Validation
HTTP Server for Central Gateway Management
32
12
Central Gateway
Server Access Permissions
Read and Write to Verification Database
36
34
38
Authentication & Attachment to Desired Service
14
20
34
Desired Services
Private Network
16
Verification Database
Desired Services

Fig 1

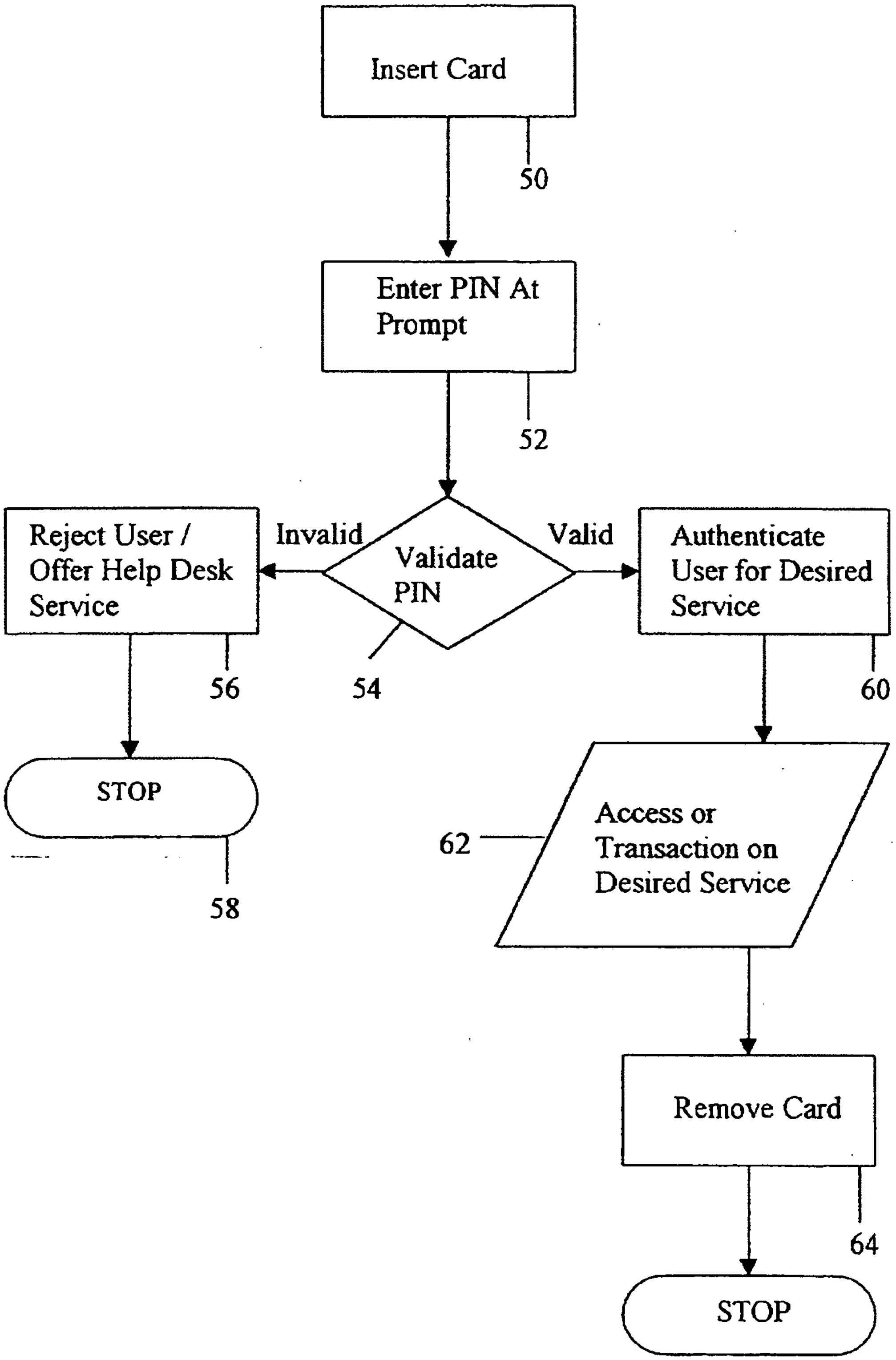

Insert Card
50
Enter PIN At Prompt
52
Invalid
Validate PIN
Valid
Reject User / Offer Help Desk Service
Authenticate User for Desired Service
56
54
60
STOP
58
62
Access or Transaction on Desired Service
Remove Card
64
STOP

Fig 2

SECURITY SYSTEM

The present invention relates to a security system, for use for example in accessing remote services such as on the Internet.

With the advent of modern technology, a growing number of transactions are being carried out by the user across insecure networks. These can be, for example, transactions involving confidential data and money for payment or investment. With such transactions there are problems with security, fraud and so on. Various security systems have been devised, such as use of personal identification numbers, encryption of transmissions. While these systems usually work well for the particular environment for which they have been designed, they can be a nuisance to use and can be difficult or expensive to implement for a new service provider.

Systems have also been developed for Internet use. These systems concentrate on authentication of the user and then, once this has been established, provide for un-encrypted connection to the service. When particular transactions are undertaken, the service determines whether encryption is necessary, for example to secure credit card details. Other solutions require entry of credit card details for each transaction. These systems inevitably must provide a balance between security and user convenience as the encryption mechanisms used cause additional work for and complication to the user.

The present invention seeks to provide an improved security system.

According to an aspect of the present invention, there is provided a security system for communicating with a remote service over a public network including a user card or other memory device, a user located card or memory device reader, a user located public network communication device and a processor unit located remotely from the user located public network communication device, wherein the user card includes user details and the user located public network communication device is operable to transmit the user details to the processor unit.

Advantageously, the processor unit is operable to carry out encryption between it and the user and to provide to the user a transparent path to the service. Thus, the user need not be aware of any security steps taken or any encryption system used, this being carried out by the card reader and the processor unit or central gateway.

The card may be any suitable device which can store user information and, preferably, encryption data. The card, can for example be a smart card, a magnetic card such as a credit/debit card or store loyalty card or any other suitable device. In addition to the card, the user may be required to input a secret identification code, such as an identification number.

In the preferred embodiment, the system provides for the user to insert the card into his/her card reader and to initiate the connection to the processor unit or central gateway. Once the connection is made, the processor unit obtains the relevant data from the card and upon verification by the identification code, allows the user access to the authorised service without any intermediate tasks, such as requirements to encrypt or decrypt transmitted data, to provide other user details and, where appropriate account or payment details. Thus, as with the preferred embodiment, all communications between the processor unit and the user can be encrypted, without the user necessarily being aware of or involved in this encryption. The communication between the user and the processor unit can therefore be totally secure yet without user inconvenience.

Advantageously, communications between the service and the processor unit, which are preferably carried out via a secure link, need not be encrypted.

The splitting of the encryption from the service results in being able to provide a dedicated encryption device, the processor unit, which can therefore be designed to maximise encrypted communication efficiency. Typically, encryption of all communications from the service unit is not practicable because the service unit is not designed for such a task and even if it were it would result in a loss of efficiency in providing the service itself.

In the preferred embodiment, the processor unit is also able to determine which of a plurality of services the user is authorised to access and/or the level of access such as spending limit, and to control access to the service or relevant service on this basis. It can also or alternatively undertake transactions against an account identified by the card.

An embodiment of the present invention is described below, by way of example only, with reference to the accompanying drawings, in which:

Figure 1 is a schematic diagram of an embodiment of security system coupled to a processor unit or central gateway and a service; and

Figure 2 is a flow chart of an example of validation routine for use with the system of Figure 1.

Referring to Figure 1, the embodiment of security system shown is designed for communications through the Internet or a similar public network.

The system includes an intelligence client device 10, which may be a personal computer, television, or any other suitable device which can communicate with a remote system. A processor unit, in this example a central gateway 12 is coupled between the client device 10 and one or more service units 14.

Communication between the client device 10 and the central gateway 12 is, in this embodiment, via a public network 18 such as the Internet. Communication between the central gateway 12 and the service units 14, 16 is, on the other hand, via a private network 20 which cannot be accessed by the public.

The client device 10 is provided with an application and user interface 22; which can be the usual computer devices such as monitor, keyboard and software in the case that it is a personal computer; the screen and a suitable keyboard or keypad in the case that the device 10 is a television or any other suitable device. The device 10 could also be a portable telephone with suitable display and keypad.

The device 10 also includes suitable network protocol 24 for allowing communication to the gateway 12 through the chosen network 18 or other public transmission medium.

The device 10 also includes a card reader 26 designed for reading the card-type chosen for the system and a card 28 which is specific to that user. The card 28 could be a smart card or magnetic card of the types well known or any other portable memory device. It is envisaged that the card 28 could have other functions in addition to the security function for this system, for example it could also be a credit/debit card, store loyalty card and the like.

The card 28 has stored thereon one or more user identifiers, one or more encryption keys and the desired service information, that is details of the service to which the user wants access. His/her level of authorisation in the service and so on will be determined by the central gateway 12.

The card reader 26 is designed, in the preferred embodiment, to be able to detect the insertion of the card 28 thereinto and in response to such insertion to commence immediately communication with the gateway 12 via the client device 10.

The central gateway 12 includes an encryption and network protocol stack 30 designed to allow communication via the chosen public network 18 and to provide encryption of all communications between itself and the client device 10. It also includes an authentication and validation unit 32 for authenticating the client data from the client card 28. The authentication and validation unit 32 is coupled to a verification database 34 of the gateway 12 in which is stored the identification data of all the users registered for the services 14,16. The database 34 may be provided either within the gateway 12 or in a remote database 34' accesses through secure network 20.

The authentication and validation unit 32 is also coupled to server access permission unit 36 designed to control the type of access to the service units 14,16 in dependence upon the user's authority.

Also provided in the gateway 12 are a typical HTTP server for management of the gateway 12 and an authentication and attachment unit 38 for communicating with the desired services 14,16 and with any remote verification database 34'.

The central gateway 12 is designed specifically for encrypting all communications over the public network 18 and for carrying out the authentication procedure.

The operation of the this embodiment will now be described with reference to Figure 2.

Insertion 50 of the card 28 into the card reader 26 prompts the card reader 26 to commence automatically the connection to the gateway 12. For this purpose, card reader 26 activates a software component in the device 10 to establish a communication link with the gateway 12 on the basis of information stored on the card 28 about the location on the Internet and access details of the gateway 12.

When a connection with the gateway 12 is established, the gateway 12 requests the user's personal identification code which is then inputted 52 at a suitable prompt on the user interface 22.

Validation 54 of the user's details and identification code is carried out either internally of the gateway 12, by the units 32 and 34, or externally at the verification database 34'.

If the gateway 12 determines 54 that the user's identification code is invalid, the user is rejected 56 and the connection is cut 58. On the other hand, if it is determined 54 the user's identification code is valid, the gateway 12 determines 60 the desired service 14, 16 and level of service to be provided and connects 62 to the desired service unit 14, 16.

During the connection to the desired service 14, 16, all data transfers between the gateway 12 and user device 10 are encrypted on the basis of the encryption keys on the user's card 28 and within verification database 34, while all data transfers between the gateway 12 and the service units 14, 16 through the private network 20 are not encrypted for ease of access and for increased efficiency. In practice, the user will not be aware of the encryption between him/her and the gateway 12 as this will be carried out as a background task. Moreover, the user will not need to re-confirm his/her identity or financial details as these will be provided by the card 28 or gateway 12.

The gateway 12, in some embodiments, records the activities of the client, such as transaction details, either within the gateway 12 or in a remote memory accessed via a private network.

Disconnection from the services 14, 16 is, in this embodiment, effected simply by removing 64 the card 28 from the card reader 26.

Thus, connection is made by a simple two step process of inserting the card 28 into the reader 26 and entering the user identification code and disconnection is effected by removing the card 28 from the card reader 26. The user is not involved in any other authentication or encryption process and need not re-enter personal details.

This system can be used for any remote service, including business to consumer (in which case the card could be designed also to function as a store or credit card), business to business (for example for transactions on account) and for internal networking (where the activity of staff, for example, needs to be secured).

It will be apparent from the above that the system can provide simple but absolutely secure access to a remote service. Moreover, by identifying the user to the desired service, user access can be customised. By removing the need for entry of account details, transactions into the desired service become quicker and less risky for the user's perspective.

Performance of the services can also be enhanced by carrying out the encryption tasks within the gateway rather than in the service units.

In addition, the service company can establish a relationship with the user by providing the user with the card and, possibly, also with the card reader.

It will be apparent that the card 28 and card reader 26 could be configured to communicate with a plurality of separate gateways 12.

CLAIMS

1. A security system for communicating with a remote service over a public network including a user card or other memory device, a user located card or memory device reader, a user located public network communication device and a processor unit located remotely from the user located public network communication device, wherein the user card includes user details and the user located public network communication device is operable to transmit the user details to the processor unit.

2. A security system according to claim 1, wherein the processor unit is operable to carry out encryption between itself and the user.

3. A security system according to claim 1 or 2, wherein the card has stored thereon user information and, preferably, encryption data.

4. A security system according to claim 3, wherein the card is a smart card, a magnetic card or any other suitable device.

5. A security system according to any preceding claim, wherein the card reader is operable to activate communication with the remote processor means upon insertion of a card thereinto.

6. A security system according to any preceding claim, wherein the processor unit is operable to encrypt substantially all communications between the user and itself.

7. A security system according to any preceding claim, wherein the processor unit is operable to determine which of a plurality of services a user is authenticated onto the desired service.

8. A security system substantially as hereinbefore described with reference to and as illustrated in the accompanying drawings.

The Patent Office

INVESTOR IN PEOPLE

10.

Application No: GB 9921227.6
Claims searched: All

Examiner: Michael Logan
Date of search: 20 January 2000

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.R): G4V (VAK); H4P (PDCSA)

Int Cl (Ed.7): G06F 17/60; G07F 7/10

Other: Online: WPI, EPODOC, JAPIO

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
X	EP 0813175 A2	(NCR INTERNATIONAL) whole document relevant	1-6
X	WO 98/32260 A1	(COMMONWEALTH BANK OF AUSTRALIA) see page 2 and fig 1	1-6
X	WO 97/50207 A1	(TELIA AB) see page 9, lines 1-24	1-6
X	WO 97/29416 A2	(INTEGRATED TECHNOLOGIES OF AMERICA) see especially page 7, line 5 - page 8, line 16	1-7
X	US 5809143	(HUGHES) see for example column 10, lines 35-43	1-6

X	Document indicating lack of novelty or inventive step
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.
&	Member of the same patent family
A	Document indicating technological background and/or state of the art.
P	Document published on or after the declared priority date but before the filing date of this invention.
E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

An Executive Agency of the Department of Trade and Industry

8) Family number: 14153892 (JP2000215165 A2) | ✉ | 🖶 | full-text | status | citations | < | > | ^ | ☐ |

Title: METHOD AND DEVICE FOR INFORMATION ACCESS CONTROL AND RECORD MEDIUM RECORDI INFORMATION ACCESS CONTROL PROGRAM

Priority: JP19990017401 19990126
Priority Map

Family: Family Explorer	Publication number	Publication date	Application number	Application date	Lin
	JP2000215165 A2	20000804	JP19990017401	19990126	🖶

Assignee(s) (std): NIPPON TELEGRAPH AND TELEPHONE

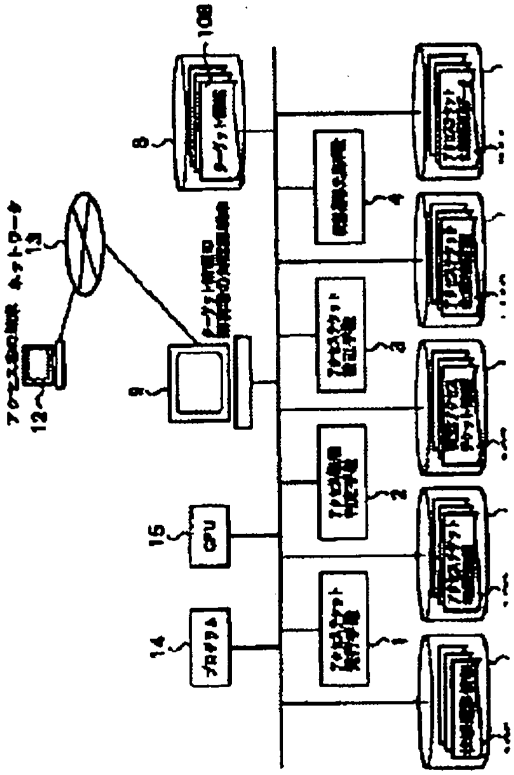
Inventor(s): OHARA YASUHIRO ; OSHIMA YOSHITO

International class (IPC 8): G06F12/14 G06F15/00 G09C1/00 H04L9/32 (Advanced/Invention); G06F12/14 G06F15/00 G09C1/00 H04L9/32 (Core/Invention)

International class (IPC 1-7): G06F12/14 G06F15/00 G06F17/60 G09C1/00 H04L9/32

Abstract:

Source: JP2000215165A2 PROBLEM TO BE SOLVED: To provide the method and device for information access control which can easily change the access authority to be allowed to an accessing person in response to the change of situation of a transaction and also to provide a recording medium which records an information access control program. SOLUTION: An access ticket issuing means 1 issues the access tickets to every accessing person and these tickets prescribe the access authority to the target information for each of plural types and states. Receiving an access request from an accessing person, the means 1 reads the request and the access authority corresponding to the type and state of an inputted access ticket out of an access ticket authority information storing means 6 and decides to permit or not permit the access request based on the access authority. When a state transition request is received from the accessing person, the transition destination state is read out of a state transition information storing means 5 based on the type and state of the access ticket that is inputted together with the state transition request. Based on the transition destination state, the change of the access ticket is updated.



2) Family number: 33529418 (JP2005218143 A2) | ✉ | 🖨 | full-text | status | citations | < | > | ^ | ☐ |

extended family

Title: ENCRYPTION DEVICE USED IN A CONDITIONAL ACCESS SYSTEM

Priority: US19970054575P 19970801

Priority Map

Family: Family Explorer	**Publication number**	**Publication date**	**Application number**	**Application date**	**Lin**
	JP2005218143 A2	20050811	JP20050120426	20050418	
	WO9907150 A1	19990211	WO1998US16145	19980731	

Assignee(s) (std): SCIENTIFIC ATLANTA

Assignee(s): SCIENTIFIC ATLANTA INC

Inventor(s) (std): PALGON MICHAEL S ; PINDER HOWARD G

Designated states: AL AM AT AU AZ BA BB BE BF BG BJ BR BY CA CF CG CH CI CM CN CU CY CZ DE DK EE ES FI F GA GB GE GH GM GN GR GW HR HU ID IE IL IS IT JP KE KG KP KR KZ LC LK LR LS LT LU LV M MD MG MK ML MN MR MW MX NE NL NO NZ PL PT RO RU SD SE SG SI SK SL SN SZ TD TG TJ TR TT UA UG UZ VN YU ZW

International class (IPC 8): G09C1/00 H04L9/08 H04L9/10 H04N7/10 H04N7/16 H04N7/167 (Advanced/Invention); G09C1/00 H04L9/08 H04L9/10 H04N7/10 H04N7/16 H04N7/167 (Core/Invention)

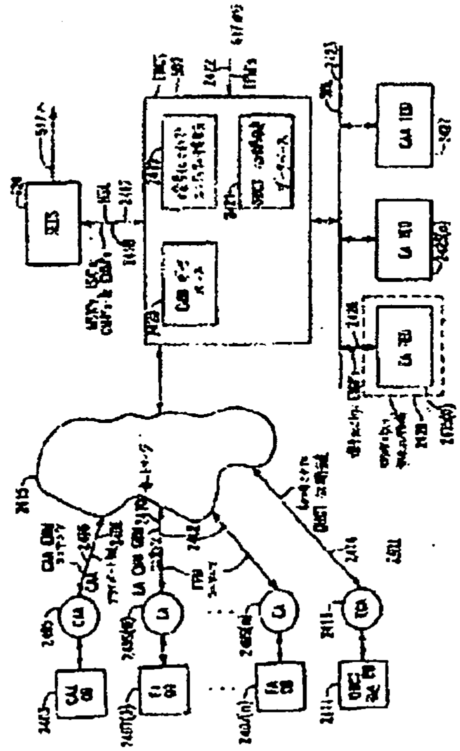
International class (IPC 1-7): H04L9/10 H04N7/16 H04N7/167

European class: H04N7/167D H04N7/16E2

Cited documents: WO9529560, US5787172, US5592552, US5400401, US5341425, EP0752786,

Abstract:

Source: JP2005218143A2 PROBLEM TO BE SOLVED: To provide a cable television system providing conditional access to a service. SOLUTION: The cable television system includes a headend from which service "instances", or programs, are broadcast and a plurality of set top units for receiving the instances and selectively decrypting these instances for display to system subscribers. The service instances are encrypted, by using public and/or private keys provided by service providers or central authorization agents. Keys, used by the set tops for selective decryption may also be public or private in nature, and these keys may be reassigned at different times, to provide a cable television system in which the anxiety for violation actions is minimized. COPYRIGHT: (C)2005,JPO&NCIPI<



(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 840 194 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
06.05.1998 Bulletin 1998/19

(51) Int. Cl.^{6}: G06F 1/00

(21) Application number: 97108754.9

(22) Date of filing: 02.06.1997

(84) Designated Contracting States:
DE FR GB
Designated Extension States:
AL LT LV RO SI

(30) Priority: 29.10.1996 JP 286345/96

(71) Applicant:
MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.
Kadoma-shi Osaka (JP)

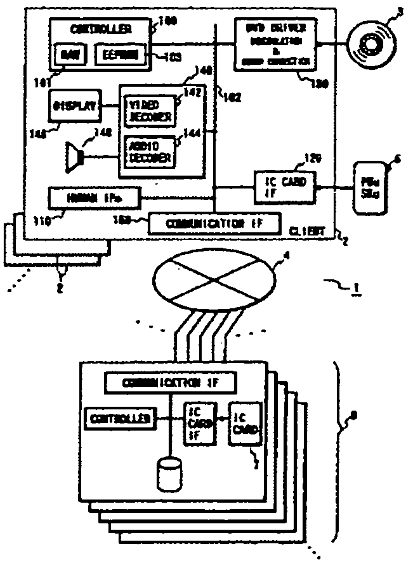
(72) Inventors:
- **Uranaka, Sachiko**
  Tokyo (JP)
- **Kiyono, Masaki**
  Kamakura-shi, Kanagawa-ken (JP)

(74) Representative:
Pellmann, Hans-Bernd, Dipl.-Ing. et al
Patentanwaltsbüro
Tiedtke-Bühling-Kinne & Partner
Bavariaring 4
80336 München (DE)

(54) System and method for controlling the use of a package of distributed application software

(57) A system for permitting only an authentic user to play a desired application contained in a distributed application package in one of predetermined operation, e.g., free play mode, charged mode, limit-attached play mode, etc. The system comprises a client for playing an application under the control of a server connected with the client through a communication network. The application package (the volume) includes a distribution descriptor which contains mode codes assigned to the volume and the applications of the volume. The data of distribution descriptor is decided and stored in the descriptor at the time of distribution of the volume. This feature makes the system flexible. There is also disclosed a system operatable without communicating with a server.

FIG. 1


CONTROLLER
100
RAM
EEPROM
103
101
140
142
102
DISPLAY
VIDEO DECODER
148
146
AUDIO DECODER
144
DVD DRIVER
130
3
120
IC CARD IF
6
HUMAN IFs
110
150
COMMUNICATION IF
CLIENT
2
2
4
1
COMMUNICATION IF
CONTROLLER
IC CARD IF
IC CARD
7
8

EP 0 840 194 A2

Printed by Xerox (UK) Business Services
2.16.1/3.4

Description

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention generally relates to a security system and, more specifically, to a method and system for permitting an authentic user to use charged information which has been distributed via package or transmission media while charging and controlling the use of distributed charged information.

2. Description of the Prior Art

In order to use charged information such as music, movies, games, etc. provided by information providers that provide various programs of such charged information, a user has generally to take two steps. In the first step (or obtaining step), the user obtains a desired program from one of the information providers by purchasing a package media such as an FD (floppy disc), an optical disc (e.g., CD-ROM (compact disc read only memory) and DVD (digital versatile disc or video disc)), etc. on which the desired program is recorded (off-line distribution or obtaining) or by down loading the desired program from the server computer of an information provider through a predetermined procedure (on-line distribution or obtaining). In case of the on-line obtaining, the user may either play the program while obtaining it (i.e., the two steps are executed in parallel) or store the program while obtaining it in the first step and execute the program later as the second step (or using step). In case of the off-line obtaining, in the second step the user loads the obtained recording media into an appropriate device and directly plays (or executes) the program or once stores the program into the memory of the device and then plays the program.

Japanese Patent unexamined publication No. Hei7-295674 (1995) discloses a security system for use in the second or using step for a CD-ROM. In this system, the user can use encrypted information which is recorded together with a public key of a toll center (a center public key) on a CD-ROM by encrypting with the center public key and sending a code of desired program included in the information and a user-generated key to the information provider and by decrypting the information with an encryption key which has been encrypted with the user-generated key and sent by the information provider. However, the identity of the user is not verified, permitting a mala fide user who have obtained other person's CD-ROM to use it. Further, the center public key is pressed together with the encrypted information on the CD-ROM. This makes it difficult to change the center public key. Also, this causes different providers who probably want to use different center public keys to force the CD-ROM manufacturer to use different masters (or stampers) in pressing the CD-ROMs.

Japanese Patent unexamined publication No. Hei7-288519 (1995) discloses a security system for use in both the first and second steps. However, this system is only applicable to a system in which charged information is distributed on line.

Japanese Patent unexamined publication No. Hei8-54951 (1996) discloses a system in which the quantity of used software is monitored, and further software use by the user is impeded if the quantity exceeds a predetermined quantity. Since a dedicated hardware is necessary for impeding of software use, this system is only suitable for the use in a server in a on-line distribution system.

There is also a system for permitting a user to use, only for a trial period, software which has been distributed with data defining the trial period. In this system, a mala fide user may make the software reusable by installing the software again or setting the user system clock for a past time.

There are these and other programs in the art. It is an object of the invention to provide a system for permitting only an authentic user (a user who have legally obtained charged information either on line or off line from an information provider) to use the charged information without any limitation, charging for each time of its use, or within the tolerance of a use-limiting factor (e.g., the quantity used, the days elapsed since the day of its purchase or the current date) according to the type of the charged information.

SUMMARY OF THE INVENTION

According to the principles of the invention, it is assumed that charged information or an application package is distributed, either via package (or recording) media or via transmission media, together with at least control information such as a media title and a media code, etc. However, an illustrative embodiment will be described mainly in conjunction with charged information recorded on and distributed by means of the DVD.

For any type of charged information, charged information has been encrypted with a key and recorded on a DVD when obtained by a user. If distributed charged information to be played is of the limitlessly playable type, the charged information processing is achieved in the following way: the key is first obtained in a user public key-encrypted form from the DVD on which the key has been recorded at the time of selling the DVD; the user public key-encrypted key is decrypted with a user secret key stored in a IC card into a decrypted key; and the encrypted charged information is decrypted with the decrypted key and consumed (that is, played or executed). The user-public key-encrypted key may be obtained on line from the server serving the client (device).

If distributed charged information to be played is of the usage-sensitive charging type, the user is charged for each time of using the information. In this case, prior to processing the charged information, the client double-encrypts and sends a user's credit card number to one of the to 11 servers of the provider of the information; the server adds an amount (e.g., play time or duration) used associated with the information to the value in a total amount (software meter) field in a volume data table, and sends the updated total amount value to the client; and the client displays the updated total amount. Then the client starts the charged information processing.

If distributed charged information to be played is of the limit-attached type, that is, the use of the information is to be limited by the tolerance of a certain limiting factor concerning the Information consumption, then the client is permitted to consume the charged information only if the use-limiting factor is within the preset limit. In case of this type of charged information, prior to processing the charged information, the client sends the identifier (ID) code of a user specified application which is recorded on the DVD to the server; on receiving the ID code the sever tests if the use-limiting factor associated with the user specified application is within the preset limit; if not, then the server informs the client of the test result, and the client displays the test result; if the test was successful, then the server updates the meter (or integrated value) of the use-limiting factor and sends the updated value to the client; and in response to the reception of the updated value the client displays the updated value. Then the client starts the charged information processing.

BRIEF DESCRIPTION OF THE DRAWING

Further objects and advantages of the present invention will be apparent from the following description of the preferred embodiments of the invention as illustrated in the accompanying drawings. In the drawing,

FIG. 1 is a block diagram showing an arrangement of a system for permitting a user to use a distributed application package on the terms of use of the package with a higher security according to a first illustrative embodiment of the invention;

FIG. 2 is a diagram showing an exemplary structure of an application (or a charged information) package recorded on a DVD used in the inventive system;

FIGs. 3 and 4 are diagrams showing, in a detailed form, exemplary data structures of the volume descriptor 22 and the distribution descriptor 23, respectively;

FIG. 5 is a flow chart of a volume control program for playing the application(s) recorded on the DVD according to the principle of the invention;

FIG. 6A is a diagram showing an exemplary structure of a volume data table stored in a server shown in FIG. 1;

FIG. 6B is a diagram showing an exemplary structure of a application data table stored in a server 8;

FIG. 7 is a diagram showing a structure of a server table 75 stored in the EEPROM 103 of the client 2;

FIGs. 8A and 8B are flow charts of initial routines executed interactively by the client 2 and the server 8, respectively, at the beginning of the processes 650, 700 and 800.

FIG. 9 is a flow chart showing a procedure of a free play process shown as step 650 in FIG. 5, wherein connecting adjacent blocks by two flow lines indicates that each block is executed interactively by a client and an associated server;

FIGs. 10A and 10B are flow charts jointly showing a procedure formed of exemplary expected play time informing routines interactively executed;

FIGs. 11A and 11B are flow charts jointly showing a procedure formed of exemplary timed play and metered usage report routines interactively executed for playing an application while timing the duration and displaying a timed play duration after the play;

FIGs. 12A and 12B are flow charts jointly showing a procedure formed of exemplary timed application-play subroutines interactively executed for playing the application while timing the duration;

FIGs. 13A and 13B are flow charts jointly showing a procedure formed of alternative timed application-play subroutines interactively executed in which timing of play time is achieved with a timer in the client;

FIG. 14 is a flow chart of an exemplary application play subroutine called in steps 612 and 622 of FIGs. 12A and 13A, respectively, and executed by the controller 100;

FIG. 15 is a flow chart showing a procedure of a charged play process 700 shown as step 700 in FIG. 5,

FIGs. 16A and 16B are flow charts jointly showing a procedure formed of exemplary expected charge informing routines interactively executed;

FIGs. 17A and 17B are flow charts jointly showing a procedure formed of routines interactively executed in block 650 of FIG. 15;

FIGs. 18A and 18B are flow charts jointly showing a procedure formed of exemplary timed play and metered charge report routines interactively executed for playing an application while timing the duration and displaying a charge and a total amount of charges after the play;

FIG. 19 is a flow chart showing a procedure interactively executed by the client 2 and the server 8 in the operation block 800 of FIG. 5, wherein blocks connected with two flow lines indicates that operation of the blocks is done by the two elements 2 and 8;

FIGs. 20A and 20B are a key-encrypting key table and a user's public key table, respectively, stored in the server; and

FIG. 20C is a flow chart of a process for obtaining the application encrypting key K_v from the server 8;

FIG. 21 is a block diagram of an exemplary decipherer-built-in IC card IF according to the invention;

FIG. 22 is a diagram showing a K_v decoder used in place of the K_v decoder 126 of FIG. 21 in a system 1 using the cryptosystem of FIG. 20C;

FIG. 23 is a diagram for explaining the meanings of the terms-of-use (TOU) codes and the corresponding limit values;

FIG. 24 is a block diagram showing an arrangement of a system for playing a distributed application package on the terms of use of the package without communicating with any server according to a second illustrative embodiment of the invention;

FIG. 25 is a flow chart schematically showing an exemplary control program executed by the controller 100a shown in FIG. 24;

FIGs. 26 and 27 are flow charts showing an operation of a free play mode shown in step 650a of FIG. 25 in a detailed form and a further detailed form, respectively; and

FIG. 28 is a flow chart showing an operation of a limit-attached play mode shown in step 800a of FIG. 25.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

For the sake of better understanding of the following description, it will be useful to define some terms to be used.

Charged information provided by an information provider may be distributed off-line (in off-line distribution) or on-line (in on-line distribution). In off-line distribution, the charged information is recorded on package media or recording media, and distributed through the sales network of the provider, that is, sold at stores in the sales network. The package media include all sorts of portable recording media such as various types of magnetic discs, a variety of optical memory discs (e.g., CD, CD-ROM, DVD), and magnetic tapes and cartridges. In on-line distribution, the charged information is transmitted via transmission media from the servers at the service points of the provider and the distributors aligned with the provider to the client device (e.g., PC (personal computer)) of the user who requested the charged information, and stored in a recording media of the client (device). The transmission media include any telecommunication channels which permit data communication between the servers and the client device. The package media and the transmission media are hereinafter referred to en bloc as "distribution media".

The charged information may be any type of software such as music, movies, games, etc. which are each referred to as an "application" without discrimination. The distribution unit of charged information is referred to as a "charged information package" or an "application package". There may be included one or more applications in an application package.

The present invention relates to a system for permitting a user to use a distributed application package on the terms of use of the package with a higher security.

Embodiment I

For the purpose of simplicity, a first illustrative embodiment will be described in which package media, among other things, DVDs are used as distribution media.

FIG. 1 is a block diagram showing an arrangement of a system for permitting a user to use the application(s) recorded on a DVD on the terms of use of the DVD with a higher security according to the first illustrative embodiment of the invention. In FIG. 1, the system 1 comprises a client or DVD player 2 which plays a DVD 3, a telecommunication network 4, and a server 8 at a toll center of the provider 6 which provides the application package of the DVD 3.

FIG. 2 is a diagram showing an exemplary structure of an application (or a charged information) package 20 recorded on the DVD 3 used in the inventive system 1. In FIG. 2, the application package 20 comprises at least one application 21, a volume (or package) descriptor 22 comprising data concerning the application package 20, and a distribution descriptor 23 comprising data which is determined mainly at the time of, e.g., distribution or sales after the pressing of the DVD 3. (The volume descriptor 22 and the distribution descriptor 23 constitutes the volume control data of the volume 20.) In this embodiment it is assumed that a volume (or package) control program which controls the use of the application package 20 in cooperation with the server 8 is included in and distributed with the application package

20. Thus, the application package 20 further comprises the package control program 24 suited for the terms of use of the package 20. The application(s) 21, the volume descriptor 22 and the package (or volume) control program 24 are recorded in the data area of the DVD 3 at the time of manufacturing the DVD 3, while the distribution descriptor 23 is recorded in the burst cutting area at the time of, e.g., sales of the DVD 3.

FIGs. 3 and 4 are diagrams showing, in a detailed form, exemplary data structures of the volume descriptor 22 and the distribution descriptor 23, respectively. In FIG. 3, the volume descriptor 22 at least contains a volume identifier (VID_v) 25 which the title of the application package 20 is probably used for and which is the same as the application identifier if the package or volume 20 contains only one application; a provider identifier 26; volume creation date and time 27 which may be used for the base point by which volume expiration data and time as described later is determined; and volume effective date and time 28 indicative of date and time until which the volume 20 is available. If the volume 20 contains more than one applications, the volume descriptor 22 further contains application identifiers (AID_a's) 29.

In FIG. 4, the distribution descriptor 23 comprises the fields of: a volume issue number (NO_{v-i}) 30 which contains a serial number given to each of the distributed application packages of an identical volume identifier (volume ID or title) VID_v in the order of distribution; a server public key (PK_s) 31 the data of which is given by the server 6 at a toll center of the provider 6; a PK_u (user-public-key)-encrypted application-encrypting key (K_v) 32; and sales date and time 33. The key PK_s 31 field contains a key which has been used in encrypting each application 21 in the package 20 and which has been encrypted with a user public key (PK_u) of the user who has legally obtained the package 20. Appropriate data are recorded in all of the fields 30 through 34 at the time of distribution of the package 20, i.e., at the time of sales of the DVD 3 in this embodiment.

The distribution descriptor 23 further comprises the field 34 of terms-of-use code (mode code) plus limit value for the volume (the volume limit value field) and, for each of the application IDs 29, the fields 35 of terms-of-use code plus limit value for the application ID 29 (application limit value field). If terms of use are set only to the volume 20, there is no need of the field 35. If terms of use are set to each application, the field is empty.

FIG. 23 is a diagram for explaining the meanings of the terms-of-use (TOU) codes and the corresponding limit values. In FIG. 23, the terms-of-use code may be, e.g., one byte in length. The higher digit (X) of the TOU code indicates the target to which the terms of use is applied as shown in table 36. That is, higher digits of 0, 1, 2,... indicate that the TOU codes beginning with those digits are for the entire volume, application 1, application 2 and so on. The lower digit (Y) of the above mentioned terms-of-use code indicates the terms of use of the package 20 or the application 21 to which the code is set, and is directly followed by a corresponding limit value as shown in table 37 of FIG. 23. Specifically, the terms-of-use code (or TOU code)of 00H means, for example, that the volume 20 is usable freely after distribution. The value '31H' means, for example, that the application 3 to which the TOU code is set can be used by paying per unit of play duration. The lower digit of 2H or more means that the volume 20 or the application to which the TOU code is set can be used freely until the corresponding limit value are reached, which disables further use. As seen from the table, the use-limiting factors determined by the TOU codes whose lower digits are 2H to 5H are the current date and time, the expiration date and time, the amount of used period, and the access count, respectively.

Since the data of the distribution descriptor 23 can be set as described above, this provides both the providers and the users with more flexibility than conventional system can provide.

Again in FIG. 1, the DVD player 2 comprises a controller 100 for controlling the entire DVD player 2; data bus 102 connected with the not-shown CPU (central processing unit), not-shown ROM (read-only memory), RAM (random access memory) 101, and EEPROM (electrically erasable programmable ROM) 103 included in the controller 100; human interfaces (IFs) 110 including input devices such as a keyboard, a voice recognition device, a mouse, a remote controller, etc. ; an IC card interface (IF) 120 for connecting the bus 102 with the ROM (not shown) in a IC card 5; a DVD driver 130 for reading out the data recorded on the DVD 3 and for demodulating and error-correcting the read data; a video and audio output IF 140 for receiving a MPEG 2 bit stream and outputting a video and audio output signals; a display device 146; a loudspeaker 148, and a communication IF 150 for communicating through the public telecommunication network 4. The IC card 5 stores a user's password PW_u and a user's secret key SK_u which corresponds to the user's public key PK_u mentioned in conjunction with the PK_u-encrypted AP-encrypting key (K_v) contained in the field 32 of the distribution descriptor 23 recorded in the burst cutting area of the DVD 3. The video and audio output IF 140 includes a MPEG 2 video decoder 142 and a MPEG 2 audio decoder 144.

As for obtaining the DVD 3, there may be some ways. If one is to buy a DVD 3, e.g., at some book store or through mail order, he or she has to have the PK_u-encrypted version of an application-encrypting key (K_v) recorded in the burst cutting area of the desired DVD 3 by notifying his or her public key PK_u which corresponds to his or her secret key SK_u stored in the IC card 5. If one is a member of a DVD distribution service, he or she can obtain a DVD with a PK_u-encrypted AP-encrypting key recorded without notifying the PK_u each time of obtaining because he or she must have notified the PK_u when he or she applied for the service.

In operation, the user first sets a desired DVD 3 in the DVD driver 130 of the DVD player 2, and issues a start command to the DVD player 2 through an appropriate human IF 110. In response to a receipt of the start command, the controller 100 reads the volume control program 24 from the data area of the DVD 3 through the DVD driver 130 while loading the read program 24 into the RAM 101 of the controller 100, and then executes the volume control program 24.

FIG. 5 is a flow chart of the volume control program 24 for playing the application(s) 21 recorded on the DVD 3 according to the principle of the invention. In FIG. 5, the controller 100 first checks the AID1 field to see if the volume 20 contains a single application in step 500. If not, then the controller 100 displays the application IDs in the field 29 and prompts the user to select a desired one of the applications in step 502, and waits for the selection in step 504. If any application is selected in step 504, the controller 100 registers the application ID of the application as the application to be played in step 506 and proceeds to step 508 to check the field 35 of the terms-of-use (TOU) code plus limit value for the selected application to see if the field is empty. If so, the controller 100 proceeds to step 510 to read the volume limit field 34.

On the other hand, if the test result is YES in step 500, then the controller 100 registers the volume ID as the application to be played in step 512, and reads the volume limit value 34 in step 510.

If the step 510 is completed or the test result of step 508 is NO, then the controller 100 checks the terms-of-use (TOU) code to see if the lower digit of the TOU code is 0 in step 514. If so, then the controller 100 plays an application free of charge in step 650, and otherwise makes another check to see if the lower digit of the TOU code is 1 in step 516. If so, the controller 100 plays an application in a usage-sensitive charging in step 700, and otherwise (if the lower digit of the TOU code is 2 or more) play an application only when the software meter of a use-limiting factor is under a preset value in step 800. On completing any of the steps or processes 650 through 800, the controller 100 ends the program 24. Thus, the DVD player 2 plays a program specified by the user according to the terms of use determined by the TOU code which has been set to either the application package or the specified application.

The processes 650, 700 and 800 are executed interactively with an associated server 8. The servers 8 need various data for executing these processes, and store such data in the form of tables.

FIG. 6A is a diagram showing an exemplary structure of a volume data table stored in a server 8. In FIG. 6A, Each of the records of the volume data table 60 comprises volume ID (VID$_v$) and issue No. (NO$_{v-i}$) fields. The combination of VID$_v$ and NO$_{v-i}$ serves as the user ID of the user of the application package 20 or the DVD 3. For this reason, the table 60 has, for the members or subscribers of DVD distribution service or the like, personal data fields which contains, for example, a member ID, a name, an address, etc. Each record further comprises a volume minute meter field (VM-METER$_{v-i}$) containing a software meter of play duration in minute which is attached to (or associated with) the volume 20; a volume charge meter (VC-METER$_{v-i}$) containing a software charge meter which is attached to the volume 20; a limit value (LV$_{v-i}$) containing a limit value associated with the TOU code (e.g., the effective date and time, the allowable expiration date and time, the allowable access, etc.); a limit value meter (LV-METER$_{v-i}$); an application ID (AID$_{v-i-a}$) field containing the title of the application; an application minute meter (AM-METER$_{v-i-a}$) field containing a software meter of play duration in minute which is attached to the application of AID$_{v-i-a}$; an application charge meter (AC-METER$_{v-i-a}$) field for a software meter of play duration in minute which is attached to the application of AID$_{v-i-a}$; a limit value (LV$_{v-i-a}$) containing a limit value associated with the TOU code; and a limit value meter (LV-METER$_{v-i}$).

FIG. 6B is a diagram showing an exemplary structure of a application data table stored in a server 8. In FIG. 6B, the application data table 70 comprises the fields of, for example, an application code (ACODE$_n$), an application title (AID$_n$), a duration (D), a rate-per-access (RATE/ACCESS), an access count, a minute meter, etc. The duration is a period of time what it takes to play the application. The rate per access is a charge for a play of the whole application, which is used for informing the user of an expected play duration prior to a play. The rate per unit time is a charge for a unit time of play, which is used for the calculation of a charge for an actually timed play duration. The access count and minute meter fields contains the number of accesses to the application and a total amount of play time, which are not necessary for the present invention but will be used in statistical calculations for the analysis of, e.g., the tastes.

FIG. 7 is a diagram showing a structure of a server table 75 stored in the EEPROM 103 of the client 2. In FIG. 7, the fields of the table 75 comprises a server public key (PK$_s$), a server ID (SID$_s$), a server network address (SADD$_s$), etc. this table 75 is used for associating the sever public key (PK$_s$) contained in the distribution descriptor 23 recorded in the burst cutting area of the DVD with the ID and the network address.

Play an Application Free of Charge

The initial routines of the processes 650, 700 and 800 are the same.

FIGs. 8A and 8B are flow charts of initial routines 80a and 80b which are executed interactively by the client 2 and the server 8, respectively, at the beginning of the processes 650, 700 and 800. In FIG. 8, the controller 100 of the client or the DVD 2, in step 82, sends a service request with the network address CADD$_c$ of the client or DVD 2, the TOU code plus limit value, the volume ID (VID$_v$), the issue number (NO$_{v-i}$), the application ID (AID$_{v-i-a}$), and other data to the associated server 8 the ID of which is SID$_s$ (SID$_s$ is obtained from the table 75 in FIG. 7 by using the public key recorded on the DVD 3), and in step 92 waits for a response from the server (SID$_s$) 8. If there is a response from the server (SID$_s$), the client 2 proceeds to the next step through a circle with "A" therein.

On the other hand, in FIG. 8B, the server 8 of SID_s receives the message from the client 2, that is, the service request and the accompanying data and stores date in a predetermined location for subsequent use in step 84. Then, the server 8 searches the table 60 for a record which contains VID_v and NO_{v-i} in the volume ID and issue No. fields thereof, respectively in step 86. If the search is unsuccessful, then the server 8 adds the record for VID_v and NO_{v-i} and fills relevant fields with AID_{v-i-a} and a limit value, if any, in the table 60 in step 88, and proceeds to step 90. Also, if the search in step 86 is successful, the server 9 proceeds to step 90, where the server 8 selects a routine to execute next according to the value of the TOU code and enters the selected routine through a circle with "B" therein. In this case, if the TOU code = x0H (x: an arbitrary HEX number, the letter H in the last position indicates that the preceding number is in hexadecimal), then a routine for playing an application free of charge is selected. If the TOU code = x1H, then a routine for playing an application in usage-sensitive charging is selected. If the TOU code ≥ x2H, then a routine is selected which plays an application only if the software meter of a use-limiting factor is under a preset value.

FIG. 9 is a flow chart showing a procedure of a free play process shown as step 650 in FIG. 5, wherein connecting adjacent blocks by two flow lines indicates that each block is executed interactively by a client of $CADD_c$ and an associated server SID_s as shown in detail later. If the TOU code is 0 in step 514 of FIG. 5, then the server ($CADD_c$) enters the free play process 650 as shown in FIG. 9, and the client and the server (SID_c) execute the initial routine 80 in block 660. In block 670, they executes an expected play time informing routine, that is, displays an expected play time before playing an specified application. In block 680, they execute an application play and metered play time report routine. Since the routine 80 has been detailed in FIG. 8, the expected play time informing routine and the application play and metered play time report routine will be detailed in the following.

FIGs. 10A and 10B are flow charts jointly showing a procedure formed of exemplary expected play time informing routines 97a and 97b interactively executed by the client 2 and the associated server 8, respectively. In FIG. 10B, the server 8 retrieves the duration (D_n) of the application of AID_{v-i-a} from the table 70 in a well known manner in step 91. In the next step 92, the server 8 calculates an expected total amount of play time according to the value of the TOU code. Specifically, if the TOU code is 0xH, then the client adds the duration (D_n) and the value of the $VM\text{-}METER_{v-i}$ field of the record identified by VID_v and NO_{v-i} in the table 60. If the TOU code is axH (a: the application number of the specified application in the volume), then the client adds the duration (D_n) and the value of the $AM\text{-}METER_{v-i-a}$ field of the record identified by VID_v, NO_{v-i}, and AID_{v-i-a} in the table 60. Then the server 8 sends the result to the client whose network address is $CADD_c$ in step 93, and ends the process.

On the other hand in FIG. 10A, the client 2 receives the incoming message or the value of the updated meter in step 94. In the next step 95, the value is displayed as the total amount of usage. Then the client 2 ends the process.

In updating a relevant meter, a predetermined value of duration has been used in the just described routines of FIG. 10 (a preset value metering system). This arrangement is suited mainly for such applications as it takes a constant time to play, and will not cause a problem unless the user discontinues the play. From this point of view, it is preferable to actually measure the playing time in metering (a timed value metering system). However, it is also noted that the preset value metering system is useful in informing the user of expected play time prior to an actual playing.

FIGs. 11A and 11B are flow charts jointly showing a procedure formed of exemplary timed play and metered usage report routines 675a and 675b interactively executed by the client and the server, respectively, for playing an application while timing the duration and displaying a timed play duration after the play. In the routine 675, the client and the server call a timed application-play subroutine for playing the application while timing the duration (play time) in step 200.

Then the server 8 proceeds to step 210, where the client updates a relevant meter according to the TOU code in the same manner as in step 92 of FIG. 10B. Specifically, if the TOU code is 0xH, then the play time is added to the value of the $VM\text{-}METER_{v-i}$ field of the record identified by VID_v and NO_{v-i} in the table 60. If the TOU code is axH (a: the application number of the specified application in the volume), then the play time is added to the value of the $AM\text{-}METER_{v-i-a}$ field of the record identified by VID_v, NO_{v-i}, and AID_{v-i-a} in the table 60. Then the server 8 sends the play time and the value of the updated meter (i.e., the total amount of play time) to the client whose network address is $CADD_c$ in step 212, and ends the process.

On the other hand, the client 2, after step 200, make a test to see if there is a response from the server of SID_s in step 214. This step is repeated until the client 2 receives a call from the server 8, when the client 2 receives the incoming message or the value of the updated meter in step 216. In the next step 218, the client 2 displays the play time and the total amount of play time, and then ends the routine 675.

FIGs. 12A and 12B are flow charts jointly showing a procedure formed of exemplary timed application-play subroutines 205a and 205b executed by the client 2 and the server 8, respectively, for playing the application while timing the duration. The server 8 of SID_s waits for a notice in step 611 to see if the client has started playing the application. On the other hand, the client 2 of $CADD_c$ informs the server of a start of play in step 610 and immediately call an application play subroutine in step 612. This, causes the server 8 to start a timer in step 613, and waits for a notice of a stop of play from the client 2 in step 615. On completing the step 612, the client informs the server 8 of the stop of play in step 614. In response to this notice, the server 8 stops and reads the timer as the play time in step 617. After steps 614 and 617, the client and the server return.

Though the above described arrangement has used a timer of the server, it may be possible to use a timer of the client.

FIGs. 13A and 13B are flow charts jointly showing a procedure formed of alternative timed application-play subroutines 205ac and 205bc interactively executed by the client 2 and the server 8, respectively, in which timing of play time is achieved with a timer in the client. In the alterative subroutine 205a, the client 2 starts a timer in step 620, calls an application play routine in step 622, stops the timer in step 624, sends the play time to the server 8 in step 626, and then returns. On the other hand, the server 8, on entering the subroutine 295b, waits for a call from the client of $CADD_c$ in step 621. If there is a call from the client 2, then the server 8 receives the play time in step 623 and then returns.

However, the arrangement of FIG. 13 has a possibility of permitting a mala fide user to manipulate the timer of the client 2. From this point of view, the arrangement shown in FIG. 12 is preferable to that of FIG. 13.

FIG. 14 is a flow chart of an exemplary application play subroutine called in steps 612 and 622 of FIGs. 12A and 13A, respectively, and executed by the controller 100.

Prior to the description of the flow chart, we define some notation concerning encryption and decryption. If encrypting X with a key EK according to an encrypting algorithm e yields Y, then it is expressed as:

$$e(EK, X) = Y.$$

Similarly, if decrypting Y with a key DK according to a decrypting algorithm d yields Z, then it is expressed as:

$$d(DK, Y) = Z.$$

Assuming that the algorithms e and d and the keys EK and DK correspond each other, that is, $d(DK, Y) = X$, it follows that

$$d(DK, e(EK, X)) = X.$$

Returning now to FIG. 14, the controller 100 read the PK_u-encrypted application-encrypting (AP-encrypting) key (K_v) or $e1(PK_u, K_v)$ from the filed 32 of the distribution descriptor 23 of the DVD in step 602. Here,

$$v = 1, 2, \ldots, V,$$

where V is the number of kinds of the application package. This indicates that different application-encrypting keys K1 through K_v is assigned to respective kinds of applications, that is, volume VID1 through VID_v.

In the next step 604, the user secret key SK_u is read from the IC card 5. In the next step 606, the PK_u-encrypted AP-encrypting key $e1(PK_u, K_v)$ is decrypted with the user secret key SK_u to obtain the application encrypting key K_v. Then in the next step 608, the K_v-encrypted application (AP), i.e., $e(K_v, AP)$ which is recorded on the DVD 3 is decrypted with the obtained AP-encrypting key K_v to obtain $d(K_v, e(K_v, AP)) = AP$, while passing the obtained application data to the video and audio output IF 140. The obtained application data has the form of an MPEG 2 bit stream. The video and audio output IF 140 converts the MPEG 2 bit stream of the application data into video and audio output signals through MPEG 2 video and audio decoding. The video and audio output signals are applied to the display device 146 and the loudspeaker 148, respectively.

Play an Application in Usage-sensitive Charging system

FIG. 15 is a flow chart showing a procedure of a charged play process 700 shown as step 700 in FIG. 5, wherein connecting adjacent blocks by two flow lines indicates that each block is executed interactively by a client of $CADD_c$ and an associated server of SID_s. In FIG. 15, the client 2 enters the process 700 via step 516 of FIG. 5 and proceeds to block 630, where the client 2 and the associated server 8 execute the initial routine 80. In the next block 640, the client 2 displays an expected charge and a total amount of charges received from the server 8, and let the user decide whether to play the desired application.

FIGs. 16A and 16B are flow charts jointly showing a procedure formed of exemplary expected charge informing routines 640a and 640b interactively executed by the client 2 and the associated server 8, respectively. The routines 640a and 640b are very similar to the routine 97 except that in the routine 640, the DURATION (D_n) or "play time" has been replaced with RATE PER ACCESS and "charge"; between steps 92a and 93a, there has been added a step 641 of the server generating and storing a pseudo random number R in a memory location R'; in step 93a, the server sends the pseudo random number R as well; between steps 94 and 95a there has been added a step 643 of the client storing the received pseudo random number R in a memory location R" for subsequent use. The replacement of DURATION (D_n) with RATE PER ACCESS is achieved by accessing a RATE PER ACCESS field 74 instead of a DURATION field

73 in table 70. Further, in the routine 640 there have been added the following steps: in step 644 following the step 96a, the client 2 makes a check to see if the user decides to play the application; if not, the client 2 sends a quit message to the server of $SADD_S$ in step 645, and ends the routine 640; on the other hand, in step 642 following the step 93a, the server 8 of SID_S waits for a call from the client 2 of $CADD_C$; on receiving a call from the client, the server makes another check in step 646 to see if what has been received is a quit message; if so, the client ends the routine 640; and if the user decided to play the application in step 644, which means that what the server has received is not a quit message but an encrypted credit card number as seen from the description below, then the client 2 and the server 8 proceed to the step 650 of FIG. 15.

In the next block 650, the server 8 obtains a user's credit card number (CCNOu) through the client 2 keeping the security of the card number as shown in FIGs. 17A and 17B. In step 647, the client 2 encrypts the credit card number of the user which has been input by the user through a human IF 110 with a key, i.e., the pseudo random number R which has been stored in a memory location R" in step 643 of FIG. 16A to obtain e2(R, CCNOu). In the next step 648, the client 2 further encrypts R + e2(R, CCNOu) with another key or a server public key read from the distribution descriptor 23 recorded in the burst cutting area of the DVD to obtain

$$e1(PK_S, R + e2(R, CCNOu)).$$

In the next step 649, the client 2 sends the encrypted data to the server 8. Through step 646 of FIG. 16B, the server proceeds to step 650, where the server 8 finds that what was received from the client $CADD_C$ is encrypted data. In the next step 651, the server 8 reads a server secret key SK_S from an IC card 7. in the next step, the server 8 decrypts the received encrypted data with the server secret key SK_S as follows;

$$d1(SK_S, \text{encrypted data}) = d1(SK_S, e1(PK_S, R + e2(R, CCNOu)) = R + e2(R, CCNOu).$$

In step 653, the server 8 makes a check to see if the just obtained pseudo random number R coincides with the random number R which has been stored in a memory location R' of the server. if so, the server 8 sends an enable message to the client of $CADD_C$, and in step 655 decrypts e2(R, CCNOu) with the pseudo random number R to obtain the user's credit card number CCNOu. On the other hand, in response to a reception of the enable message in step 657, the client 2 exits from the process. After step 655, the server also exits from the process. If the result is NO in step 653, then the server 8 sends a disable message to the client in step 656, and ends the process. In response to a reception of the disable message in step 657, then the client displays a message to this effect in step 658, and then ends the process.

After operation of block 650, the client 2 waits, in step 663, for a report from the server on whether the credit card for the transmitted card number (CCNOu) is valid or not, while the server 8 refers to the credit company associated with the card number in step 661 to see if the credit card is valid. if not, the server 8 informs the client 2 of the invalidity of the credit card in step 662, and ends the process. If the card is valid in step 661, the server 8 informs the client of the validity in step 667. If the client 2 receives a report from the server in step 663, the client makes another check in step 664 to see if the report indicates the validity of the card. If not, the client display a message to indicate the invalidity in step 665, and ends the process. If the report indicates the validity in step 664, which means the completion of step 667, then the client 2 and the server 8 proceed to the next block 670.

In step 670, the client 2 and the server 8 execute timed play and metered charge report routine. FIGs. 18A and 18B are flow charts jointly showing a procedure formed of routines 675ac and 675bc interactively executed for playing an application while timing the duration and displaying a charge and a total amount of charges after the play. In FIG. 18, the routines 675ac and 675bc are identical to the routine 675a and 675b in FIGs. 11A and 11B except that "time" has been replaced with "charge", and accordingly VM-METER and AM-METER have been replaced with VC-METER and AC-METER.

The operation, in the client 2, of playing an application on usage-sensitive charging is completed by block 675 of FIG. 15 or step 218a of FIG. 18A. After step 212a, the server 8 charges the play to the credit card number CCNOu obtained in step 655 of FIG. 17B in step 680. This completes the whole of the charged application play process of FIG. 15.

In this process, only information on charge is given to the user. It is very easy to provide information on both time and charge by adding steps 91 through 93 and 95 to the routines 640b and 640a, and by adding steps 210 and 218 to the routines 675bc and 675ac.

As described above, expected time and/or charge are (is) displayed before playing a user specified application. This is helpful for the user to decide whether to play the application. Additionally, charging is done based on the actually timed play duration. This makes the charging reasonable.

In the above description, the arrangement is such that the user has to input his or her credit card number CCNOu each time he or she wants to play an application. However, instead of doing this, the credit card number CCNOu may be stored in non-volatile memory or EEPROM 103 in a PW_u-encrypted form. In this case, CCNOu is obtained by decrypting PW_u-encrypted CCNOu (e.g., $e(PW_u, CCNOu)$) with a password entered by the user. That is, $d(\text{entered password}, e(PW_u, CCNOu)) = CCNOu$.

Permit the Play Within a Preset Limit

FIG. 19 is a flow chart showing a procedure interactively executed by the client 2 and the server 8 in the operation block 800 of FIG. 5, wherein blocks connected with two flow lines indicates that operation of the blocks is done by the two elements 2 and 8. In this case, it is assumed that a preset limit is recorded in or on the application package and is transmitted from client 2 to server each time of play. On entering the process 800 via step 516 of FIG. 5, the client 2 proceeds to step 801, where the client 2 and the server 8 executes the initial routines 80. It is noted that in routine 80b, if there is a record for VID_v and NO_{v-i}, then the limit value (LV_{v-i}) field of the table 60 of FIG. 6A contains the limit value transmitted from the client 2, otherwise, the received limit value is stored in the LV_{v-i} field when the record for VID_v and NO_{v-i} is added in step 88.

In step 810, the server 8 makes a check if a meter associated with the TOU code received from the client 2 is under the limit value. This check is made by comparing an LV field and LV-meter field associated with the TOU code in table 60. if the value of the LV-meter is equal to or greater than the LV field value, then the server returns an over limit massage to the client 2 in step 820. If not, the server 8 returns an underlimit message to the client 2 in step 822, and proceeds to step 828. If the client 2 receives the overlimit message in step 824, then the client 2 displays a message to this effect. If not, the client 2 proceeds to the step 828.

Since the expected play time informing routines 97a and 97b and the application play subroutine 600 has been described above, the description of steps 828 and 830 are omitted.

According to this feature of the invention, it is possible to limit the use of charged information. This feature is especially useful in case when a user who have paid in advance for the use of the application package is permitted to use the application package within a limit value.

Though it has been assumed that the limit values are included in the application package, the limit values may be kept in the servers of the provider or distributer from the beginning. In this case, the limit values are fixed. However, if limit values are permitted to be set and recorded in the application package at the time of distribution or sales, the limit values are advantageously set according to an amount paid.

As is apparent from the foregoing, as a limit value, any use-limiting factors will do that can be measured in quantity. Such limit values are, for example, the effective date and time, the allowable expiration date and time, the maximum amount of play time, the allowable access count.

It is also possible to combine this feature with a charged application play feature. That is, an arrangement may be such that the user is permitted to use an application package on usage-sensitive charging only if the value of an LV-meter associated with the TOU is under the value of the corresponding LV or the value recorded in a field 33 or 34 of the distribution descriptor 23.

Modification I

In the above embodiment, applications, if more than one, in one volume are encrypted by an identical application encrypting key K_v. However, the applications APa in one volume may be encrypted with respective AP-encrypting keys K_a, where a lower case "a" following AP and K is a serial number assigned to each application ID. In this case, each of the AP-encrypting keys K_a are encrypted with the user public key PK_u, and stored in the PK_u-encrypted AP-encrypting key (K_a) fields 32a in the distribution descriptor 23.

Modification II

It has been assumed that the user of the DVD 3 is limited to the purchaser thereof who have had the PK_u-encrypted AP-encrypting key (K_v) recorded on the DVD 3. However, the system may be so arranged that predetermined people, e.g., family members FM_1, FM_2,..., FM_N of the purchaser can use the DVD (N is the number of the family members). One of the ways to realize this is to encrypt the AP-encrypting key K_v with a public key PK_{u-n} of each member FMn ($n = 1, 2,...,N$) to obtain $e1(PK_{u-1}, K_v)$, $e1(PK_{u-2}, K_v)$,..., $e1(PK_{u-n}, K_v)$ and to record them in the PK_{u-n}-encrypted AP-encrypting key $e1(PK_{u-n}, K_v)$ fields 32 of the distribution descriptor 23 at the time of purchase of the DVD.

Modification III: K_v Retrieval From Server

In the above description, the AP-encrypting key K_v has been recorded in a PK_u-encrypted form on the DVD 3. However, the AP-encrypting key K_v may be managed by the server 8 and transmitted to the client or the DVD player 2 in response to a request issued from the DVD player 2 each time of use of the DVD 3. In this case, there is no need of providing the distribution descriptor 23 with the PK_u-encrypted AP-encrypting key field 32. Instead each of the servers has to store an AP-encrypting key table (or K_v table) and a PK_u table (shown in FIGs. 20A and 20B) in the hard disc. As shown in FIG. 20A, the K_v table a volume ID (VID_v) field (as the entry of record) and an AP-encrypting key (K_v) field in each record. In FIG. 20B, each record of the PK_u table comprises a volume ID (VID_v) field (as the entry of record), a volume issue number (NO_{v-i}) field and a PK_u field (Successive same values in the first field are shown by showing only the first appearing one). Further, the process (or step) 610 of obtaining the AP-encrypting key K_v, that is, a group of the steps 602, 604 and 606 in the application play routine 600, has to be replaced with a process of FIG. 20C.

FIG. 20C is a flow chart of a process in which the client DVD player 2 obtains the application encrypting key K_v from the server 8. In step 616, the server 8 retrieves a key K_v from the K_v table by using VID_v. In the next step 618, the key K_v is encrypted with an arbitrary number used only in the current process, e.g., a pseudo random number R to obtain $e2(R, K_v)$. In the next step 620, the server 8 retrieves a key PK_u from the PK_u table by reading the PK_u field of the record which contains VID_v and NO_{v-i} in the VID_v and NO_{v-i} fields, respectively. In the next step 622, $R + e2(R, K_v)$ is encrypted with the retrieved key PK_u to obtain a double encrypted AP-encrypting key

$e1(PK_u, R + e2(R, K_v))$,

which is returned to the client with a client network address $CADD_c$ in the next step 624.

On the other hand, the controller 100 of the client 2 waits for a response from the server 8 of SID_s in step 626. If there is any response from the server 8 of SID_s in step 626, then the client DVD 3 receives the data $e1(PK_u, R + e2(R, K_v))$ from the server 8 in step 628. In the next step 630, the received data is decrypted with the user secret key SK_u read from the IC card 5. Specifically, the following calculation is done.

$d1(SK_u, e1(PK_u, R + e2(R, K_v))) \Longrightarrow R + e2(R, K_v)$

In the next step 632, $e2(R, K_v)$ is decrypted with the obtained pseudo random number R. Specifically, the following calculation is done.

$d2(R, e2(R, K_v)) \Longrightarrow K_v$

Thereafter, the controller 100 proceeds to the step 608 of FIG. 14.

In this modification, the applications APa in one volume may be encrypted with respective AP-encrypting keys K_a. In this case, the K_v table has to be replaced with K_a table in which each record comprises an application ID (AID_a) field and an AP-encrypting key (K_a) field. Further in step 612, the controller 100 of the DVD player 2 has to also send the application ID of the application to be played to the server.

Also in this modification, the system may be, again, so arranged that predetermined people, e.g., family members $FM_1, FM_2, \ldots, FM_N$ of the purchaser can use the DVD (N is the number of the family members). In this case, for each member FMn ($n = 1, 2, \ldots, N$), the server 8 has to use the member's own public key PK_{u-n} in encrypting the AP-encrypting key K_v. One way to realize this is to issue a volume issue number NO_{v-i-n} to each member FMn at the time of sales of the DVD, provide the non-volatile memory (not shown) of the DVD player 2 with a table for associating the user's password PWn with the volume issue number NO_{v-i-n}, send the volume issue number (NO_{v-i-n}) associated with the user's password in step 612, and use not the PK_u table but a PK_{u-n} table in which each of the records has the following fields:

$VID_v, NO_{v-i-n}, PK_{u-n}$.

Another way is to issue and record not only a volume issue number NO_{v-i} but also family member numbers FMNn for all members at the time of sales of the DVD, provide the non-volatile memory (not shown) of the DVD player 2 with a table for associating the user's password PWn with the corresponding family member number FMNn, send the volume issue number (NO_{v-i}) and the family member number FMNn associated with the user's password in step 612, and use another PK_{u-n} table in which each of the records has the following fields:

$VID_v, NO_{v-i}, FMNn, PK_{u-n}$.

In the process of FIG. 20C, the server 8 may be authenticated by means of a public-key cryptosystem using a pair of server secret and public keys (SK_s, PK_s). In this case, the server 8 signs the double-encrypted AP-encrypting key

$e1(PK_u, R + e2(R, K_v))$

with a signing key or the server secret key SK_s after step 622. While the client or DVD player 2 tests the signature by the server 8 with a test key or the server public key PK_s contained in the PK_s field 31 of the distribution descriptor 23 recorded in the burst cutting area of the DVD 2 before step 630.

However, even if just described authentication of the server 8 is omitted, an attacker will never go to any greater length than a steal of TOU code plus limit value, a volume ID VID_v, a volume issue number NO_{v-i}, and the client network address $CADD_c$. This is not a serious problem.

In the process of FIG. 20C, a pseudo random number R has been used as a pseudo variable which takes a different value each time of execution of the process. However, as the pseudo variable, any thing will do if the result of encryption with it takes a different value each time of execution of the process.

Modification IV

In the first illustrative embodiment, the decryption of application is achieved by software. For this purpose, the controller 100 has to read the user secret key SK_u from the IC card 5 through the bus 102, which leaves the possibility of permitting a breaker to easily steal the user secret key SK_u through the bus 102. In order to prevent this, the process achieved by the steps 604 through 608 may be realized by hardware as shown in FIG. 21, which is a block diagram of an exemplary decipherer-built-in IC card IF. In FIG. 21, the decipherer-built-in IC card IF 120a comprises an IC card receptacle 121 and a printed wiring board 122 extending from and fixed with the receptacle 121. An IC 123 is mounted on the printed wiring board 122. The IC 123 comprises a memory IF 125 which usually connects the memory of the IC card 5 with the bus 102 and, in response to an instruction from the controller 100, reads and passes the key SK_u to the next stage; a K_v decoder 126 for receiving the key Sk_u and encrypting e1(PK_u, K_v) with the key SK_u to yield K_v; and an AP decoder 127 for receiving the key K_v and encrypting e(K_v, AP) to yield application data (AP). The printed wiring board 122 portion may be preferably molded together with the IC card receptacle 121 portion so as to make the whole a single body. By doing this, leaking of the user secret key SK_u can be prevented.

This modification can be also applied to a system 1 using the cryptosystem of FIG. 20C. In this case, the K_v decoder 126 of FIG. 21 has to be replaced with a K_v decoder 126a as shown in FIG. 22. In FIG. 22, the K_v decoder 126a decrypts the input data, e1(PK_u, R + e2(R, K_v)), from the bus 102 by using the user secret key SK_u passed by the memory IF 125 to obtain R + e2(R, K_v), while decrypting the obtained data e2(R, K_v)) with the obtained random number R and outputting the key K_v.

Embodiment II

FIG. 24 is a block diagram showing an arrangement of a system capable of playing a distributed application package, e.g., a DVD on the terms of use of the DVD without communicating with any server according to a second illustrative embodiment of the invention. In FIG. 24, the system 1a is identical to the client 2 of FIG. 1 except that the communication IF 150 has been eliminated because of no need of communication with a server and the controller 100 has been replaced with a controller 100a. In the controller 100a, a not-shown ROM for storing a control program as described later and the EEPROM 103 have been also replaced with a new ROM (not shown) and an EEPROM 103a. In order to play a role of the server 8, the system 1a has to have table 60 of FIG. 6A in any non-volatile memory, e.g., the EEPROM 103a and an application duration (play time) for each application as defined in table 70 of FIG. 6B has to be included in the control data of each application package.

FIG. 25 schematically shows an exemplary control program executed by the controller 100a shown in FIG. 24. The control program of FIG. 25 is also identical to that of FIG. 5 except that the decision step 516 and the step 700 has been eliminated because the limit-attached play mode is not supported by the system 1a in this embodiment, and the steps 650 and 800 are replaced with steps 650a and 800a. Accordingly, operation after step 514 will be described in the following.

If the lower digit of the terms-of-use (TOU) code is 0 in the decision step 514, then in step 650a the controller 100a plays, in the free play mode, the application stored in the selected application in step 506 or 512 and ends the operation. It should be noted that since the system 1a does not have the charged play mode, the lower digit of the TOU code is defined as follows.

Higher digit of terms-of-use code (Hexadecimal)	Corresponding limit value	Play mode
0	None	Free play mode
2	Effective date and time	Limit-attached play mode
3	Allowable expiration date and time	
4	Maximum amount of used period	
5	Allowable access count	
:	:	
:	:	

Accordingly, if the lower digit of the TOU code is not 0 in the decision step 514, then in step 800a the controller 100a plays, in the limit-attached play mode, the application stored in the selected application in step 506 or 512 and ends the operation.

FIGs. 26 and 27 show an operation of a free play mode shown in step 650a of FIG. 25 in a detailed form and a further detailed form, respectively. In FIG. 26, the controller 100a executes an initial routine 80a in step 660a, in step 670a executes an expected play time informing routine, and in step 680a executes an application play and metered play time report routine.

As shown in FIG. 27, in the initial routine 80c, the controller 100a searches the table 60 for a record which contains VID_v and NO_{v-j} in the volume ID and issue No. fields thereof, respectively in step 86. If the search is unsuccessful, then the controller 100a adds the record for VID_v and NO_{v-j} and fills relevant fields with AID_{v-i-a} and a limit value, if any, in the table 60 in step 88, and proceeds to step 90. Also, if the search in step 86 is successful, the server 9 proceeds to step 90, where the controller 100a selects a routine to execute next according to the value of the TOU code and enters the selected routine. In this case, if the TOU code = x0H (x: an arbitrary HEX number, the letter H in the last position indicates that the preceding number is in hexadecimal), then a routine for playing an application free of charge is selected. If the TOU code ≥ x1H, then a routine is selected which plays an application only if the software meter of a use-limiting factor is under a preset value.

The expected play time informing routine 670a is identical to the routines 97 (FIG. 10) minus communication steps 93 and 94, comprising the above described steps 91, 92 and 95. Similarly, it is seen from FIGs. 11 and 13A that the above described steps 620, 622, 624, 210 and 218 are executed in this order in the timed play and metered usage report routine 680a. In this way, the system 1a permits the user to play the application stored in the selected application (steps 506 and 512 of FIG. 25) free of charge.

FIG. 28 is a flow chart showing an operation of a limit-attached play mode shown in step 800a of FIG. 25. Since this operation is very similar to that of FIG. 19, only the flow is briefly described, omitting the details of each step. In FIG. 28, controller 100a first makes a check if a meter associated with the TOU code has reached the limit value obtained with the TOU code. If so, then the server returns an overlimit massage to controller 100a in step 820. Otherwise, the controller 100a proceeds to the expected play time informing routine 828a (= 670a), where the controller 100a executes the above described steps 91, 92 and 95, and then calls the application play subroutine 600 in step 830, thereby completing the operation. Since the application play subroutine 600 has been detailed above, further description is omitted. In this way, the system 1a permits the user to play the application stored in the selected application (steps 506 and 512 of FIG. 25) only if the limit value associated with the TOU code assigned to the volume or the user-specified application has not been reached.

According to the second embodiment, the system 1a can operate in either of the free play mode and the limit-attached play mode without the need of communication with a server. For this, the system 1a may be made portable.

Modifications

In the above description, the illustrative embodiment has been described in conjunction with the DVD. The same discussion can be applied to such package media as permit write once or more.

Further, the present invention is also applicable to application packages distributed via transmission media. In this case, the distributed application packages are stored in a bulk storage in the user's device. An application package comprises one or more application and application control data, that is, an application descriptor and distribution descriptor. One volume is stored as a file. Since a plurality of application package may be stored in a single storage, each application package does not have to contain a control program. One control program, which may be distributed via either package or transmission media, is enough for one user device. The folder or directory in which the application packages are stored is set for a user specified one in the control program when the control program is installed. The data to be recorded in the distribution descriptor is included in the application package by the provider according to the information given by the user.

As described above, one who is permitted to use an application package is limited to an owner of the IC card which stores a user secret key SK_u corresponding to the user public key PK_u used for encryption of the AP-encrypting key K_v in the application package. For this, even if someone has unjustly obtained an application package, for example, by copying the whole volume from the DVD on which the volume is recorded, he or she can not use it without the IC card of the owner of the DVD. Thus the inventive system can prevent unjust use of an application package (DVD in this case) by any other person than the regular owner of the application package.

Also, the inventive system is so arranged that most part of the application package is recorded by pressing in manufacturing process of the DVDs, whereas at least a part of the volume control data (i.e., the distribution descriptor) can be determined at the time of, e.g., distribution of each of the DVDs after the manufacturing process. This makes the system flexible because control data can be easily changed without changing the stamper.

In the initial routines 80a and 80b in FIG. 8A and 8B, the data transmitted with the service request may be encrypted in the same manner as in case of the transmission of user's credit card number shown in FIG. 17. However, in case of the initial routines, there are a plurality of data. These data may be encrypted in the following way.

If the data to be encrypted are D1, D2,... then they are first encrypted with a key R as follows:

e2(R, D1), e2(R, D2),...

Then further encryption is made with a server public key PK_s as follows:

e1(PK_s, R + e2(R, D1) + e2(R, D2),....).

In the process of FIG. 17, the user may be authenticated by means of a public-key cryptosystem using a pair of user secret and public keys (SK_u, PK_u). In this case, the client 2 signs the double-encrypted credit card number

$$e1(PK_s, R + e2(R, CCNOu))$$

with a signing key or the user secret key SK_u after step 648. While the server tests the signature by the client 2 with a test key or the user public key PK_u before step 650.

Instead of storing a single server public key in the distribution descriptor 23, a plurality of server public keys or all the server public keys may be recorded. By doing this, it is possible, for example, to setting a different charge depending on the server public key which the user have selected by appropriately combining the tables 70 and 75.

Also, application packages with an identical volume ID can have different server public keys recorded. A plurality of toll center may be advantageously provided for application packages of the same title.

In order to prevent any use of IC card by other person than the owner of the IC card, it is possible to add, before the Sku reading step 604, the steps of prompting the user to enter a password through a human IF 110 and proceeding to step 604 only if the entered password coincides with the user password PW_u stored in the IC card.

Though the IC card 5 is used in the above embodiment, the IC card IF 120 may be replaced with a magnetic card reader to permitting the use of the magnetic card. Alternatively, the arrangement may be such that the user enters his or her password each time the user uses the DVD.

Instead of storing the user secret key SK_u in the IC card 5, the key Sku may be stored in non-volatile memory in a PW_u-encrypted form. In this case, the key SK_u is obtained by decrypting PW_u-encrypted SK_u with a password entered by the user.

The discussion of three preceding paragraphs are applied to the IC card used for storing the server secret key in the server. However, in this case the user has to be taken as the administrator of the toll server.

Many widely different embodiments of the present invention may be constructed without departing from the spirit and scope of the present invention. It should be understood that the present invention is not limited to the specific embodiment described in the specification, except as defined in the appended claims.

A system for permitting only an authentic user to play a desired application contained in a distributed application package in one of predetermined operation, e.g., free play mode, charged mode, limit-attached play mode, etc. The system comprises a client for playing an application under the control of a server connected with the client through a communication network. The application package (the volume) includes a distribution descriptor which contains mode codes assigned to the volume and the applications of the volume. The data of distribution descriptor is decided and stored in the descriptor at the time of distribution of the volume. This feature makes the system flexible. There is also disclosed a system operatable without communicating with a server.

Claims

1. An application package for use in a system for playing an application contained in the application package (the volume), the application package comprising:

   application data for at least one application; and
   volume control data for use in controlling said system, wherein said volume control data at least comprises:
   a volume ID for identifying the kind of said application package (said volume);
   an issue number assigned in order of issue to each of the volumes of said kind; and
   application IDs each assigned to one of said at least one application contained in said volume, and wherein:
   at least a part of said volume control data is to be added to said volume after the creation of said volume; and
   said at least a part of said volume control data includes said issue number.

2. An application package as defined in claim 1, wherein:

   said application data has been encrypted with an encrypting key; and
   said at least a part of said volume control data includes a user's public key-encrypted version of said encrypting key used.

3. An application package as defined in claim 1, wherein said at least a part of said volume control data includes mode codes which are assigned to said volume or said at least one application and each indicate a play mode associated with one of said volume or said at least one application to which the mode code is assigned.

4. A package media on which an application package as defined in claim 1 has been recorded.

5. A package media of a write-once type on which an application package as defined in claim 1 has been recorded.

6. A package media on which an application package as defined in claim 1 has been recorded wherein said at least a part of said volume control data is recorded in an area different from data area where said application data is recorded on the package media.

7. A method for sending data with a raised security from a first device to a second device through a public telecommunication network, comprising the steps of:

   in said second device,

      generating a pseudo random number;
      transmitting said pseudo random number to said first device;

   in said first device,

      encrypting said data with said transmitted pseudo random number into encrypted data;
      encrypting concatenated data consisting of said pseudo random number and said encrypted data with a public key of said second device into double-encrypted data;
      sending said double-encrypted data to said second device; in said second device,
      decrypting said double-encrypted data with a secret key of said second device which corresponds to said public key into decrypted data consisting of a decrypted random number portion and another decrypted portion; and
      decrypting said another decrypted portion with said transmitted random number to obtain said data.

8. A method for sending a plurality of pieces of data with a raised security from a first device to a second device through a public telecommunication network, comprising the steps of:

   in said second device,

      generating a pseudo random number;
      transmitting said pseudo random number to said first device;

   in said first device,

      encrypting each of said pieces of data with said transmitted pseudo random number into an encrypted piece of data;
      encrypting concatenated data consisting of said pseudo random number and said encrypted pieces of data with a public key of said second device into double-encrypted data;
      sending said double-encrypted data to said second device; in said second device,
      decrypting said double-encrypted data with a secret key of said second device which corresponds to said public key into decrypted data consisting of a decrypted random number portion and said plurality of decrypted data portions; and
      decrypting each of said decrypted portions with said transmitted random number to obtain said pieces of data.

9. A method as defined in claim 7 or 8, further comprising the steps, executed after said step of decrypting said double-encrypted data, of:

   proceeding to a next step only if said decrypted random number portion coincides with said transmitted pseudo random number; and
   said second device informing said first device of a failure in decryption if said decrypted random number portion does not coincide with said transmitted pseudo random number.

10. In a system provided with means for playing an application contained in an application package, a method for permitting a user to play an encrypting key-encrypted application contained in a distributed application package which further contains, as volume control data, a user's public key-encrypted encrypting key so encrypted as to be able to be decrypted with a secret key of the user into said encrypting key, the method comprising the steps of:

   reading said user's public key-encrypted encrypting key from said distributed application package (said vol-

ume);
obtaining said secret key;
decrypting said user's public key-encrypted encrypting key with said secret key to obtain said encrypting key; and
decrypting said encrypting key-encrypted application with said obtained encrypting key into application data while passing said application data to said means for playing an application.

11. In a system comprising a client provided with means for playing an application contained in an application package and a server connected with the client through a communication network, a method for permitting a user to play one of encrypting key-encrypted applications contained in a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume), an issue number issued to each volume of the kind in an issued order and application IDs, the method comprising the steps of:

    said client reading said volume ID, said issue number and an application ID for said one of encrypting key-encrypted applications (said encrypting key-encrypted application) from said volume and sending to said server;

        in said server,

            retrieving said encrypting key by using said volume ID;
            retrieving a public key of said user by using said volume ID and said issue number;
            generating a pseudo random number;
            double-encrypting said encrypting key with said pseudo random number and said public key into a double encrypted data;
            sending said double-encrypted data to said client; in said client,
            obtaining a secret key of said user which corresponds to said public key;
            obtaining said encrypting key by decrypting said double-encrypted data with said secret key;
            decrypting said encrypting key-encrypted application with said obtained encrypting key into application data while passing said application data to said means for playing an application.

12. A method as defined in claim 10 or 11, wherein said means for obtaining a secret key comprises means for reading said secret key from a portable memory of said user.

13. A method as defined in claim 12, wherein said portable memory is an IC card.

14. In a system comprising a client provided with means for playing an application package and a server connected with the client through a communication network for controlling the client, the application package (the volume) containing, as volume control data, a volume ID and an issue number issued to each of the volumes of said volume ID in an issued order, a method for controlling the amount of play time comprising the steps of:

    said client sending said volume ID and said issue number to said server;
    said server retrieving an expected play time associated with said volume ID and said issue number; and
    said server adding said expected play time to the value of a total play time associated with said volume ID and said issue number.

15. In a system comprising a client provided with means for playing an application contained in an application package and a server connected with the client through a communication network for controlling the client, the application package (the volume) containing, as volume control data, a volume ID, an issue number issued to each of the volumes of said volume ID in an issued order and an application ID for the application, a method for controlling the amount of play time comprising the steps of:

    said client sending said volume ID, said issue number and said application ID to said server;
    said server retrieving an expected play time associated with said volume ID, said issue number and said application ID; and
    said server adding said expected play time to the value of a total play time associated with said volume ID and said issue number.

16. In a system comprising a client provided with means for playing an application contained in an application package and a server connected with the client through a communication network for controlling the client, the application package (the volume) containing, as volume control data, a volume ID and an issue number issued to each of the volumes of said volume ID in an issued order, a method for controlling the amount of play time comprising the steps of:

    said client and said server interactively measuring, as a measured play time, a play time of said application; and
    said server adding said measured play time to the value of a total play time associated with said volume ID and said issue number.

17. A method as defined in claim 16, wherein said step of measuring a play time comprises the step of using a timer of said server.

18. A method as defined in claim 16, wherein said step of measuring a play time comprises the step of using a timer of said client.

19. In a system comprising a client for playing an application package and a server connected with the client through a communication network wherein the application package (the volume) comprises application data and control data and at least a part of the control data has been added to the volume after the creation of said volume, a method for sending desired data from one side of said client and said server to the other side, the method comprising the steps of:

    including a secret key of said other side in said at lest a part of said control data;

    in said other side,

    generating a pseudo random number;
    transmitting said pseudo random number to said one side;

    in said one side,

    encrypting said desired data with said transmitted pseudo random number into encrypted data;
    encrypting concatenated data consisting of said pseudo random number and said encrypted data with said public key of said other side into double-encrypted data;
    sending said double-encrypted data to said other side;

    in said other side,

    decrypting said double-encrypted data with a secret key of said other side which corresponds to said public key into decrypted data consisting of a decrypted random number portion and another decrypted portion; and
    decrypting said another decrypted portion with said transmitted random number to obtain said desired data.

20. A method as defined in claim 19, wherein said generating a pseudo random number includes storing said pseudo random number in memory, and wherein the method further comprises the step, executed prior to said decrypting said another decrypted portion, of:

    in response to a determination that said decrypted random number portion does not coincide with said pseudo random number stored in said means for storing said pseudo random number stored in said memory, informing said one side of a failure in decryption instead of passing the control to next means.

21. In a system comprising a client provided with means for playing an application contained in an application package and a server connected with the client through a communication network, a method for permitting a user to play an application contained in a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume), an issue number issued to each volume of the kind in an issued order, and an application ID for said application, the method comprising the steps of:

proceeding to a next step only if the value of a meter field associated with said volume ID, said issue number and said application ID is under the value of a limit value field associated with said volume ID, said issue number and said application ID in a volume data table; and

displaying a message informing an overlimit on a display device of said client and quit the operation otherwise.

22. In a system comprising a client provided with means for playing an application contained in an application package and a server connected with the client through a communication network, a method for permitting a user to play an application contained in a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume), an issue number issued to each volume of the kind in an issued order, an application ID for said application and a limit value for limiting the play of said application, the method comprising the steps of:

proceeding to a next step only if the value of a meter field associated with said volume ID, said issue number and said application ID in a volume data table is under said limit value; and
displaying a message informing an overlimit on a display device of said client and quit the operation otherwise.

23. A method as defined in claim 21, wherein said limit value is one of effective date and time, allowable expiration date and time, a maximum amount of play time, and an allowable access count.

24. A method as defined in any of claims 11, 15 and 16, wherein said step of said client sending to said server comprises the steps of:

said client encrypting at least one of said volume ID, said issue number and said application ID into encrypted data; and
said server decrypting said encrypted data.

25. A system for sending data with a raised security from a first device to a second device through a public telecommunication network, comprising:

means provided in said second device for generating a pseudo random number;
means provided in said second device for transmitting said pseudo random number to said first device;
means provided in said first device for encrypting said data with said transmitted pseudo random number into an encrypted data;
means provided in said first device for encrypting concatenated data consisting of said pseudo random number and said encrypted data with a public key of said second device into double-encrypted data;
means provided in said first device for sending said double-encrypted data to said second device;
means provided in said second device for decrypting said double-encrypted data with a secret key of said second device which corresponds to said public key into decrypted data consisting of a decrypted random number portion and another decrypted portion; and
means provided in said second device for decrypting said another decrypted portion with said transmitted random number to obtain said data.

26. A system for sending a plurality of pieces of data with a raised security from a first device to a second device through a public telecommunication network, comprising:

means provided in said second device for generating a pseudo random number;
means provided in said second device for transmitting said pseudo random number to said first device;
means provided in said first device for encrypting each of said pieces of data with said transmitted pseudo random number into an encrypted piece of data;
means provided in said first device for encrypting concatenated data consisting of said pseudo random number and said encrypted pieces of data with a public key of said second device into double-encrypted data;
means provided in said first device for sending said double-encrypted data to said second device;
means provided in said second device for decrypting said double-encrypted data with a secret key of said second device which corresponds to said public key into decrypted data consisting of a decrypted random number portion and said plurality of decrypted data portions; and
means provided in said second device for decrypting each of said decrypted portions with said transmitted random number to obtain said pieces of data.

27. A system as defined in claim 25 or 26, further comprising:

means, provided in said second device, activated prior to decrypting each of said decrypted portions and responsive to a determination that said decrypted random number portion does not coincide with said transmitted pseudo random number, for informing said first device of a failure in decryption instead of passing the control to next means.

28. A system for playing an encrypting key-encrypted application contained in a distributed application package which further contains, as volume control data, a user's public key-encrypted encrypting key so encrypted as to be able to be decrypted with a secret key of the user into said encrypting key, the system comprising:

means for reading said user's public key-encrypted encrypting key from said distributed application package (said volume);
means for obtaining said secret key;
means for decrypting said user's public key-encrypted encrypting key with said secret key to obtain said encrypting key;
means for decrypting said encrypting key-encrypted application with said obtained encrypting key to provide application data; and
means for using said application data for playing.

29. A system for permitting a user to play an encrypting key-encrypted application contained in a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume), an issue number issued to each volume of the kind in an issued order and application IDs, the system comprising:

a client for playing an application by using application data; and
a server for controlling said client through a communication network, wherein said client comprises:
means for reading and sending said volume ID, said issue number and an application ID for said one of encrypting key-encrypted applications (said encrypting key-encrypted application) from said volume to said server, said server comprises:

means for retrieving said encrypting key by using said volume ID;
means for retrieving a public key of said user by using said volume ID and said issue number;
means for generating a pseudo random number;
means for double-encrypting said encrypting key with said pseudo random number and said public key into a double encrypted data; and
means for sending said double-encrypted data to said client, and said client comprises:
means for obtaining a secret key of said user which corresponds to said public key;
means for obtaining said encrypting key by decrypting said double-encrypted data with said secret key;
means for decrypting said encrypting key-encrypted application with said obtained encrypting key to provide application data; and
means for using said application data for playing.

30. A system as defined in claim 28 or 29, wherein said means for obtaining a secret key comprises means for reading said secret key from a portable memory of said user.

31. A system as defined in claim 30, wherein said portable memory is an IC card.

32. A system for permitting a user to play a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume) and an issue number issued to each volume of the kind in an issued order, the system comprising:

a client for playing said distributed application package; and
a server for controlling said client through a communication network, wherein:
said client comprises means for sending said volume ID and said issue number to said server; and
said server comprises means for retrieving an expected play time associated with said volume ID and said issue number, and means for adding said expected play time to the value of a total play time associated with said volume ID and said issue number.

33. A system for permitting a user to play an application contained in a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume), an issue number issued to each volume of the kind in an issued order and an application ID for the application, the system comprising:

a client for playing said application; and
a server for controlling said client through a communication network, wherein:
said client comprises means for sending said volume ID, said issue number and said application ID to said server; and
said server comprises means for retrieving an expected play time associated with said volume ID, said issue number and said application ID, and means for adding said expected play time to the value of a total play time associated with said volume ID and said issue number.

34. A system for permitting a user to play an application contained in a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume), an issue number issued to each volume of the kind in an issued order and an application ID for the application, the system comprising:

a client for playing said application; and
a server for controlling said client through a communication network, wherein:
said client and said server comprise means for interactively measuring, as a measured play time, a play time of said application; and
said server further comprises means for adding said measured play time to the value of a total play time associated with said volume ID and said issue number.

35. A system as defined in claim 34, wherein said means for interactively measuring a play time comprises means for using a timer of said server.

36. A system as defined in claim 34, wherein said means for interactively measuring a play time comprises means for using a timer of said client.

37. A system for permitting a user to play an application package (the volume) comprising application data and control data wherein at least a part of the control data has been added to the volume after the creation of said volume, the system comprising:

a client for playing said volume; and
a server for controlling said client through a communication network, wherein said server comprises means for storing a secret key of said server and said at least a part of said control data includes a public key corresponding to said secret key, and wherein the system comprises:
means provided in said server for generating a pseudo random number;
means for storing said pseudo random number;
means provided in said server for transmitting said pseudo random number to said client;
means provided in said client for encrypting desired data with said transmitted pseudo random number into encrypted data;
means provided in said client for encrypting concatenated data consisting of said pseudo random number and said encrypted data with said public key into double-encrypted data;
means provided in said client for sending said double-encrypted data to said server;
means provided in said server for decrypting said double-encrypted data with said secret key into decrypted data consisting of a decrypted random number portion and another decrypted portion; and
means provided in said server for decrypting said another decrypted portion with said transmitted random number to obtain said desired data.

38. A system as defined in claim 37, further comprising:

means, provided in said server, activated prior to said decrypting said another decrypted portion and responsive to a determination that said decrypted random number portion does not coincide with said pseudo random number stored in said means for storing said pseudo random number, for informing said client of a failure in decryption instead of passing the control to next means.

39. A system for permitting a user to play an application contained in a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume), an issue number issued to each volume of the kind in an issued order and application IDs, the system comprising:

a client for playing an application by using application data; and
a server for controlling said client through a communication network, wherein said client comprises:
means for reading and sending said volume ID, said issue number and an application ID for said one of encrypting key-encrypted applications (said encrypting key-encrypted application) from said volume to said server, said server comprises:
means for proceeding to next step only if the value of a meter field associated with said volume ID, said issue number and said application ID is under the value of a limit value field associated with said volume ID, said issue number and said application ID in a volume data table; and
means for causing said client to display a message informing an overlimit on a display device of said client and quit the operation otherwise.

40. A system for permitting a user to play an application contained in a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume), an issue number issued to each volume of the kind in an issued order, application IDs and limit values associated with respective application IDs for limiting the play of respective applications, the system comprising:

a client for playing an application by using application data; and
a server for controlling said client through a communication network, wherein said client comprises:
means for reading and sending said volume ID, said issue number, an application ID for said one of encrypting key-encrypted applications (said encrypting key-encrypted application) and a limit value associated with said application ID from said volume to said server, and wherein said server comprises:
means for proceeding to a next step only if the value of a meter field associated with said volume ID, said issue number and said application ID in a volume data table is under said limit value; and
means for causing said client to display a message informing an overlimit on a display device of said client and quit the operation otherwise.

41. A system as defined in claim 39, wherein said limit value is one of effective date and time, allowable expiration date and time, a maximum amount of play time, and an allowable access count.

42. A system as defined in any of claims 29, 33 and 34, wherein said means for sending to said server comprises means for encrypting at least one of said volume ID, said issue number and said application ID.

43. A method for permitting an authentic user to play a desired one of the applications contained in a distributed application package in a system capable of playing an application, wherein said application package (said volume) contains volume control data including mode codes assigned to said volume and the applications of said volume, the method comprising the steps of:

deciding to use one of predetermined play modes specified by one of said mode codes associated with said desired application; and
playing said desired application in said specified play mode.

44. A method as defined in claim 43, wherein the method further comprises the step of including, in said mode codes, values indicative of a free play mode and at least one limit-attached play mode which correspond(s) to respective limit value(s) used for limiting usage.

45. A method as defined in claim 44, wherein said step of playing said desired application comprises the step of:

in response to a determination that said one of said mode codes associated with said desired application includes a value indicative of said free play mode, simply playing said desired application.

46. A method as defined in claim 44, wherein said step of playing said desired application comprises the step of:

in response to a determination that said one of said mode codes associated with the desired application includes one of values indicative of said at least one limit-attached play mode, displaying a message to the effect that a limit value associated with said one of values has been reached instead of playing said desired application if said limit value has been reached.

47. A method as defined in claim 43, wherein said volume control data further includes a volume ID, an issue number and an application ID for each of said applications, and wherein said step of deciding to use one of predetermined play modes comprises the steps of:

obtaining said one of said mode codes associated with said desired application and corresponding limit value by using said application ID; and
comparing said one of said mode codes with a meter value associated with said volume ID, said issue number and said application ID.

48. A method as defined in claim 45, wherein each of said applications has been each encrypted with an encrypting key and said volume control data includes a user's public key-encrypted version of said encrypting key (a public key-encrypted version encrypting key), and wherein said step of simply playing said desired application comprises the steps of:

reading said user's public key-encrypted encrypting key from said volume;
obtaining a user's secret key which corresponds to said user's public key;
decrypting said user's public key-encrypted encrypting key with said user's secret key to obtain said encrypting key; and
decrypting said desired application with said obtained encrypting key.

49. A system for permitting an authentic user to play a desired one of the applications contained in a distributed application package, wherein said application package (said volume) contains volume control data including mode codes assigned to said volume and the applications of said volume, the system comprising:

means for deciding to use one of predetermined play modes specified by one of said mode codes associated with said desired application; and
means for playing said desired application in said specified play mode.

50. A system as defined in claim 49, wherein the system further comprises means for including, in said mode codes, values indicative of a free play mode and at least one limit-attached play mode which correspond(s) to respective limit value(s) used for limiting usage.

51. A system as defined in claim 50, wherein said means for playing said desired application comprises:

means, responsive to a determination that said one of said mode codes associated with said desired application includes a value indicative of said free play mode, for simply playing said desired application.

52. A system as defined in claim 50, wherein said means for playing said desired application comprises:

means, responsive to a determination that said one of said mode codes associated with the desired application includes one of values indicative of said at least one limit-attached play mode, for displaying a message to the effect that a limit value associated with said one of values has been reached instead of playing said desired application if said limit value has been reached.

53. A system as defined in claim 49, wherein said volume control data further includes a volume ID, an issue number and an application ID for each of said applications, and wherein said means for deciding to use one of predetermined play modes comprises:

means for obtaining said one of said mode codes associated with said desired application and corresponding limit value by using said application ID; and
means for comparing said one of said mode codes with a meter value associated with said volume ID, said issue number and said application ID.

54. A system as defined in claim 51, wherein each of said applications has been encrypted with an encrypting key and said volume control data includes a user's public key-encrypted version of said encrypting key (a public key-encrypted version encrypting key), and wherein said means for simply playing said desired application comprises:

means for reading said user's public key-encrypted encrypting key from said volume;
means for obtaining a user's secret key which corresponds to said user's public key;
means for decrypting said user's public key-encrypted encrypting key with said user's secret key to obtain said encrypting key; and
means for decrypting said desired application with said obtained encrypting key.

55. A method for permitting an authentic user to play a desired one of the applications contained in a distributed application package in a system comprising a client capable of playing an application and a server connected with said client through a communication network, wherein said application package (hereinafter referred to as "said volume") contains volume control data including mode codes assigned to said volume and the applications of said volume, the method comprising the steps of:

said client deciding to use one of predetermined play modes specified by one of said mode codes associated with said desired application; and
playing said desired application in said specified play mode by means of cooperation between said client and said server.

56. A method as defined in claim 55, wherein the method further comprises the step of including, in each of said mode code, a value indicative of one of a free play mode, a charged play mode and at least one limit-attached play mode, wherein said volume control data further comprises a limit value associated with each of said at least one limit-attached play mode.

57. A method as defined in claim 55 or 56, wherein said volume control data further includes a volume ID, an issue number, and an application ID for each of said applications, and wherein said step of playing said desired application in said specified play mode includes an application play step of simply playing said specified application.

58. A method as defined in claim 57, wherein each of said applications contained in a distributed application package has been encrypted with an encrypting key and said volume control data includes a user's public key-encrypted version of said encrypting key (a public key-encrypted version encrypting key), and wherein said application play step comprising the steps of:

reading said user's public key-encrypted encrypting key from said volume;
obtaining a user's secret key which corresponds to said user's public key;
decrypting said user's public key-encrypted encrypting key with said user's secret key to obtain said encrypting key; and
decrypting said desired application with said obtained encrypting key.

59. A method as defined in claim 57, wherein each of said applications contained in a distributed application package has been encrypted with an encrypting key and said volume control data includes a user's public key-encrypted version of said encrypting key (a public key-encrypted version encrypting key), and wherein said application play step comprises the steps of:

in said server,

retrieving an encrypting key by using said volume ID;
retrieving a user's public key associated with said volume ID and said issue number;
double-encrypting said encrypting key with a pseudo random number and said user's public key into a double encrypted data;
sending said double-encrypted data to said client; in said client,
obtaining a user's secret key which corresponds to said user's public key;
obtaining said encrypting key by decrypting said double-encrypted data with said user's secret key;
decrypting said desired application with said obtained encrypting key.

60. A method as defined in claim 57, wherein said step of playing said desired application further comprises the steps, executed prior to said application play step, of:

said server retrieving an expected play time associated with said desired application; and
displaying said expected play time on a display device of said client.

61. A method as defined in claim 57, wherein said step of playing said desired application further comprises the steps of:

measuring, as a measured play time, a duration of said application play step;
adding said measured play time to a play time meter associated with said mode code to obtain a total amount of play time; and
displaying said measured play time and said total amount of play time on a display device of said client after said application play step.

62. A method as defined in claim 61, wherein said step of measuring a duration comprises the step of measuring said play time by using a timer of said server.

63. A method as defined in claim 61, wherein said step of measuring a duration comprises the step of measuring said play time using a timer of said client.

64. A method as defined in claim 57, wherein said step of deciding to use one of predetermined play modes comprises deciding to use said charged play mode if said one of said mode codes associated with said desired application includes a value indicative of said charged play mode, and wherein said step of playing said desired application comprises the steps of:

said client obtaining and sending a credit card number of said user to said server;
proceeding to a next step only if the credit card of said number is found to be valid from a reference to an associated credit company;
displaying, on a display device of said client, a charge for play decided based on a measurement of a duration of said application play step and a total amount of play charges after said application play step; and
said server charging said play to said credit card number.

65. A method as defined in claim 64, wherein said step of playing said desired application further comprises the steps, prior to said application play step, of:

displaying, prior to said application play step, an expected charge and an expected total amount of charges on said display device; and
letting the user decide whether to play said desired application.

66. A method as defined in claim 64, wherein said step of said client obtaining and sending a credit card number of said user to said server comprises the steps of:

in said server,

generating a pseudo random number;
storing said pseudo random number in memory;
transmitting said pseudo random number to said client;

in said client,

prompting said user to input said credit card number;
double-encrypting said credit card number first with said transmitted random number and then with a sever's public key included in said volume control data into a double-encrypted number;
sending said double-encrypted number to said server; in said server,
decrypting said double-encrypted number with a server's secret key into a decrypted random number and another decrypted data; and
decrypting said another decrypted data with said transmitted random number to obtain said credit card number.

67. A method as defined in claim 66, wherein said step of said client obtaining and sending a credit card number of said user to said server further comprises the steps, executed prior to said step of decrypting said another encrypted data, of:

proceeding to a next step only if said decrypted random number coincides with said pseudo random number which has been stored in said memory; and
displaying a message informing a failure in decryption and quitting the operation otherwise.

68. A method as defined in claim 57 wherein said step of deciding to use one of predetermined play modes comprises deciding to use one of said at least one limit-attached play mode if said one of said mode codes associated with said desired application includes a value indicative of said one of said at least one limit-attached play mode, and wherein said step of playing said desired application comprises the step of:

in response to a determination that a meter value associated with said one of said mode codes associated with said desired application in a record identified by said volume ID, said issue number and an application ID of said desired application in a volume data table has reached a limit value associated with said mode code, displaying a message informing an overlimit on a display device of said client instead of executing said application play step.

69. A method as defined in claim 68, wherein said limit value is one of effective date and time, allowable expiration date and time, a maximum amount of play time, and an allowable access count.

70. A system for playing a distributed application package in one of predetermined play modes in concert with a server, wherein the application package contains a data set encrypted with an encrypting key (a K-encrypted data set) for each of at least one application and volume control data for use in controlling operation of the system and the server and the volume control data includes mode codes defining said play modes, the system comprising:

means for permitting a user to select one of said at least one application of said volume;
means for deciding to use one of said predetermined play modes associated with one of said mode codes assigned to said selected application; and
means for playing said selected application in said selected play mode in concert with said server.

71. A system as defined in claim 70, wherein each of said mode codes includes one of values for a free play mode, a charged play mode and at least one limit-attached play mode.

72. A system as defined in claim 70, wherein said volume control data further includes a volume ID, an issue number and an application ID for each of said applications, and wherein said means for playing said selected application in said selected play mode at least comprises:

means for setting said server for said selected play mode by sending to said server said volume ID, said issue number, and the application ID and said mode code associated with said selected application; and
application play means for simply playing said specified application.

73. A system as defined in claim 72, wherein said volume control data further includes a user's public key-encrypted encrypting key, and wherein said application play means comprises:

means for reading said user's public key-encrypted encrypting key from said volume;
means for obtaining a user's secret key which corresponds to said user's public key;
means for decrypting said user's public key-encrypted encrypting key with said user's secret key to obtain said encrypting key; and
means for decrypting the K-encrypted data set of said selected application with said obtained encrypting key.

74. A system as defined in claim 73, wherein means for decrypting said user's public key-encrypted encrypting key and said means for decrypting the K-encrypted data set are realized as an integrated circuit.

75. A system as defined in claim 72, wherein said application play means comprises:

means for receiving double-encrypted data from said server;
means for obtaining a user's secret key which corresponds to said user's public key;

means for obtaining said encrypting key by decrypting said double-encrypted data with said user's secret key; and

means for decrypting the K-encrypted data set of said selected application with said obtained encrypting key.

76. A system as defined in claim 75, wherein means for obtaining said encrypting key and said means for decrypting the K-encrypted data set are realized as an integrated circuit.

77. A system as defined in claim 74 or 76, wherein said integrated circuit is incorporated into said means for obtaining a user's secret key.

78. A system as defined in claim 73, wherein said means for deciding to use one comprises means for deciding to use a free play mode and wherein said means for playing said selected application further comprises: means, prior to said application play means, of:

means for receiving data from said server; and
displaying said data as an expected play time for said selected application.

79. A system as defined in claim 73, wherein said means for deciding to use one of said predetermined play modes comprises means for deciding to use a free play mode, and wherein said means for playing said selected application further comprises:

means for causing said server to obtain, as a measured play time, data of a operation period of said application play means;
means for receiving first and second data from said server; and
means for displaying, just after the completion of operation by said application play means, said first and second data as said measured play time and a total amount of play time. data as said measured play time and a total amount of play time.

80. A system as defined in claim 79, wherein said means for causing said server to obtain data of said operation period comprises means for informing said server of the start and the end of operation by said application play means to utilize a timer of said server.

81. A system as defined in claim 79, wherein said means for causing said server to obtain data of a operation period comprises:

means for measuring said operation period of said application play means; and
means for sending said operation period to said server for use in a calculation of said total amount of play time.

82. A system as defined in claim 72, wherein said means for deciding to use one comprises means for deciding to use a charged play mode and wherein said means for playing said selected application further comprises:

means for obtaining and sending a credit card number of said user to said server;
means responsive to a verification result of said credit card from said server for starting a next process only if said result is positive; and
means for displaying a charge for play decided based on a measured play time of said application play means and a total amount of play charges after operation of said application play means.

83. A system as defined in claim 82, wherein said means for playing said selected application further comprises:

means activated prior to operation of said application play means for displaying an expected charge and an expected total amount of charges and letting the user decide whether to play said selected application.

84. A system as defined in claim 82, wherein said volume control data of said distributed application package further includes a sever's public key, and wherein said means for obtaining and sending a credit card number of said user to said server comprises:

means for prompting said user to input said credit card number;
means for receiving a random number from said server;

means for obtaining said server's public key from said volume;

means for double-encrypting said credit card number first with said random number and then with said server's public key into a double-encrypted data;

sending said double-encrypted number to said server;

85. A system as defined in claim 84, wherein said means for said client obtaining and sending a credit card number of said user to said server further comprises:

means responsive to a positive result of random number check from said server for starting a next process; and

means responsive to a negative result of said random number check from said server for displaying a message indicative of a failure in said random number check and quitting the operation for said selected application.

86. A system as defined in claim 72, wherein:

said means for deciding to use one comprises means for deciding to use a limit-attached play mode; and

said sending to said server includes sending a limit value associated with said mode code, and wherein said means for playing said selected application further comprises:

means operative prior to operation of said application play means for receiving from said server a limit check result indicative of whether a limit value associated with said mode code has been reached; and

means responsive to an over limit case of said result for starting a next operation.

87. A system as defined in claim 86, wherein said limit value is one of effective date and time, allowable expiration date and time, a maximum amount of play time, and an allowable access count.

88. A system for controlling through a communication network a client device to play a distributed application package in one of predetermined play modes, wherein the application package contains a data set encrypted with an encrypting key (a K-encrypted data set) for each of at least one application and volume control data for use in controlling operation of the system and the client and the volume control data includes a volume ID, an issue number, an application ID for each of said applications, and a mode code for said volume or mode codes for said applications, the system comprising:

volume data table for storing, for each volume, said volume ID, said issue number, said mode code for said volume, and said application ID and said mode code for each of said applications;

means for receiving a service request, a volume ID, an issue number, an application ID and a mode code and other data from said client;

means for storing said received application ID, said received mode code and other data in appropriate fields of a record identified by said volume ID and said issue number;

means responsive to a determination that there is no record identified by said volume ID and said issue number in said volume data table for adding said record in said volume data table and storing said received application ID and mode code and said other data in relevant fields of said record; and

means operative on the basis of said received mode code for deciding to subsequently passing the control to means for supporting a play mode associated said received mode code.

89. A system as defined in claim 88, wherein said means for supporting a play mode at least comprises means for supporting application play means, of client, for simply playing an application identified by said received application ID, and wherein said means for supporting said application play means of said client comprises:

first means for associating a given volume ID with a corresponding encrypting key;

second means for associating both a given volume ID and issue number with a corresponding user's public key;

means for retrieving an encrypting key associated with said received volume ID from said first means;

means for retrieving a user's public key associated with said received volume ID and issue number from said second means;

means for double-encrypting said encrypting key with a pseudo random number and said user's public key into a double encrypted data; and

sending said double-encrypted data to said client.

90. A system as defined in claim 89, further comprising an application data table for storing data for each kind of application, wherein said received mode code defines a free play mode, and wherein said means for supporting a play mode associated said received mode code comprises:

means, activated prior to an operation of said means for supporting application play means of said client, for retrieving an expected play time associated with said received application ID from said application data table; and
means for sending said expected play time to said client.

91. A system as defined in claim 89, wherein said received mode code defines a free play mode, and wherein said means for supporting a play mode associated said received mode code comprises:

means for measuring, as a measured play time, a duration of application play;
means for adding said measured play time to a play time meter associated with said received mode code in said volume data table to obtain a total amount of play time; and
means for sending said measured play time and said total amount of play time to said client.

92. A system as defined in claim 91, wherein said means for measuring a duration comprises:

means responsive to a notice of the start of operation by said application play means of said client for starting a timer; and
means responsive to a notice of the end of said operation for stopping said timer.

93. A system as defined in claim 91, wherein said means for measuring a duration comprises:

means for receiving a measured duration from said client.

94. A system as defined in claim 88, wherein said received mode code defines a charged play mode, and wherein said means for supporting a play mode associated said received mode code comprises:

means for receiving a credit card number of said user from said server;
means, responsive to a determination, from a verification of said credit card number, that said credit card number is not valid, for informing said client of invalidity and quitting the operation of said means for supporting a play mode;
means, responsive to a determination, from said verification of said credit card number, that said credit card number is valid, for informing said client of a validity and proceeding to a next operation; and
means for charging said play to said credit card number.

95. A system as defined in claim 94, wherein said means for supporting a play mode associated said received mode code further comprises:

means activated prior to operation of said application play means of said client for retrieving an expected charge from said application data table by using said received application ID;
means for calculating a sum of said expected charge and a value of a charge meter associated with said received volume ID or application ID depending on said received mode code;
means operative prior to operation of said application play means for sending said expected charge and said sum to said client; and
means responsive to a receipt of a message of quitting for quitting said means for supporting a play mode.

96. A system as defined in claim 94, wherein said means for receiving a credit card number of said user from said server comprises:

means for generating a pseudo random number;
means for storing said pseudo random number in memory;
means for transmitting said pseudo random number to said client;
means for waiting for a double-encrypted data from said client;
means for obtaining a server's secret key;
means for decrypting said double-encrypted number with said server's secret key into a decrypted random number and another decrypted data; and

means for decrypting said another encrypted data with said transmitted random number to obtain said credit card number.

97. A system as defined in claim 96, wherein said means for obtaining a user's secret key comprises means for reading said user's secret key from a portable memory of said user.

98. A system as defined in claim 96, wherein said means for receiving a credit card number of said user from said server further comprises:

means responsive to a determination, made prior to said decrypting said another, that said decrypted random number coincides with said pseudo random number which has been stored in said memory for sending an enable message to said client and proceeding to a next operation; and
means responsive to a determination, made prior to said decrypting said another, that said decrypted random number does not coincide with said pseudo random number which has been stored in said memory for sending a disable message to said client and quitting said supporting a play mode.

99. A system as defined in claim 88, wherein:

said received mode code defines a limit-attached play mode; and
means for receiving a service request further receives a limit value associated with said mode code, and wherein said means for supporting a play mode associated said received mode code comprises:
means for proceeding to a next operation only if the value of a software meter associated with said mode code in said volume data table is under said limit value; and
means for sending a message informing an over limit to said client and quitting the operation of said means for supporting a play mode associated said received mode code if the value of a software meter associated with said mode code in said volume data table is not under said limit value.

100. A system as defined in claim 99, wherein said limit value is one of effective date and time, allowable expiration date and time, a maximum amount of play time, and an allowable access count.

101. A system as defined in any of claims 54, 73 and 75, wherein said means for obtaining a user's secret key comprises means for reading said user's secret key from a portable memory of said user.

102. A system as defined in claim 28 or 29, wherein said means for obtaining said secret key comprises means for reading said user's secret key from a portable memory of said user.

103. A method as defined in any of claims 10, 11, 19, 21, 22 and 55, wherein said application package is recorded on a package media.

104. A method as defined in claim 103, wherein said package media is of a write-once type, and said client is a system capable of playing said package media of said write-once type.

105. An application package as defined in claim 1, wherein said package media is distributed to a purchaser thereof or a subscriber thereof via a transmission media.

106. A system as defined in any of claims 28, 29, 37, 39, 40, 70 and 88, wherein said application package is recorded on a package media.

107. A system as defined in claim 106, wherein said application package is recorded on a package media of a write-once type.

108. A system as defined in claim 106, wherein at least a part of said volume control data is recorded, after manufacturing said package media, in an area different from a data area where said at least one application is recorded.

109. A system as defined in claim 108, wherein said client is a system provided with means for playing said package media of said write-once type.

110. A system as defined in any of claims 28, 29, 37, 39, 40, 70 and 88, wherein said application package is recorded

on a DVD and at least a part of said volume control data is recorded, after manufacturing said package media, in a BCA (burst cutting area) of the DVD, and wherein said client is a system provided with means for playing said DVD.

111. A method as defined in any of claims 10, 11, 19, 21, 22, 43 and 55, wherein the application package has been distributed to a purchaser thereof or a subscriber via a transmission media and at least a part of said volume control data has been added to said application package after preparing said application package.

112. A system as defined in any of claims 28, 29, 37, 39, 40, 49, 70 and 88, wherein said application package has been distributed to a purchaser thereof or a subscriber thereof via a transmission media and at least a part of said volume control data has been added to said application package after preparing said application package.

FIG. 1

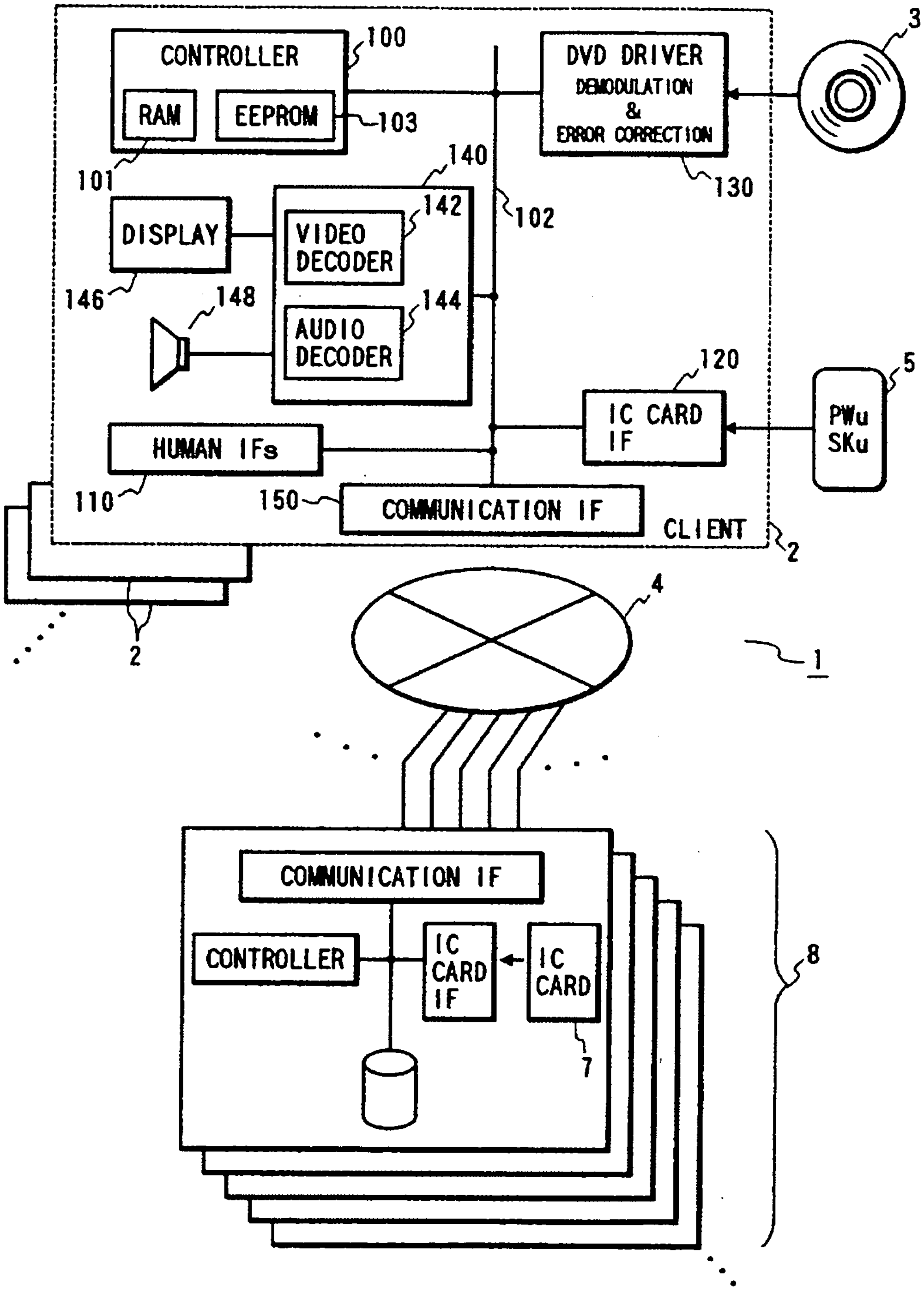

CONTROLLER
RAM
EEPROM
100
103
101
DVD DRIVER
DEMODULATION
&
ERROR CORRECTION
130
3
140
142
102
DISPLAY
VIDEO
DECODER
146
148
144
AUDIO
DECODER
120
5
IC CARD
IF
PWu
SKu
HUMAN IFs
110
150
COMMUNICATION IF
CLIENT
2
2
4
1
COMMUNICATION IF
CONTROLLER
IC
CARD
IF
IC
CARD
7
8

FIG. 2

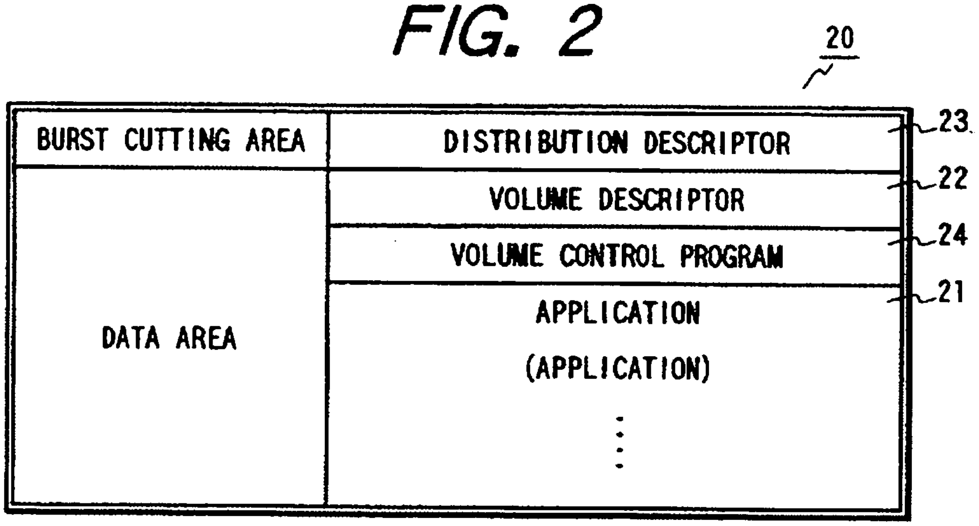

20
BURST CUTTING AREA
DISTRIBUTION DESCRIPTOR
23
VOLUME DESCRIPTOR
22
VOLUME CONTROL PROGRAM
24
DATA AREA
APPLICATION
21
(APPLICATION)
⋮

FIG. 3

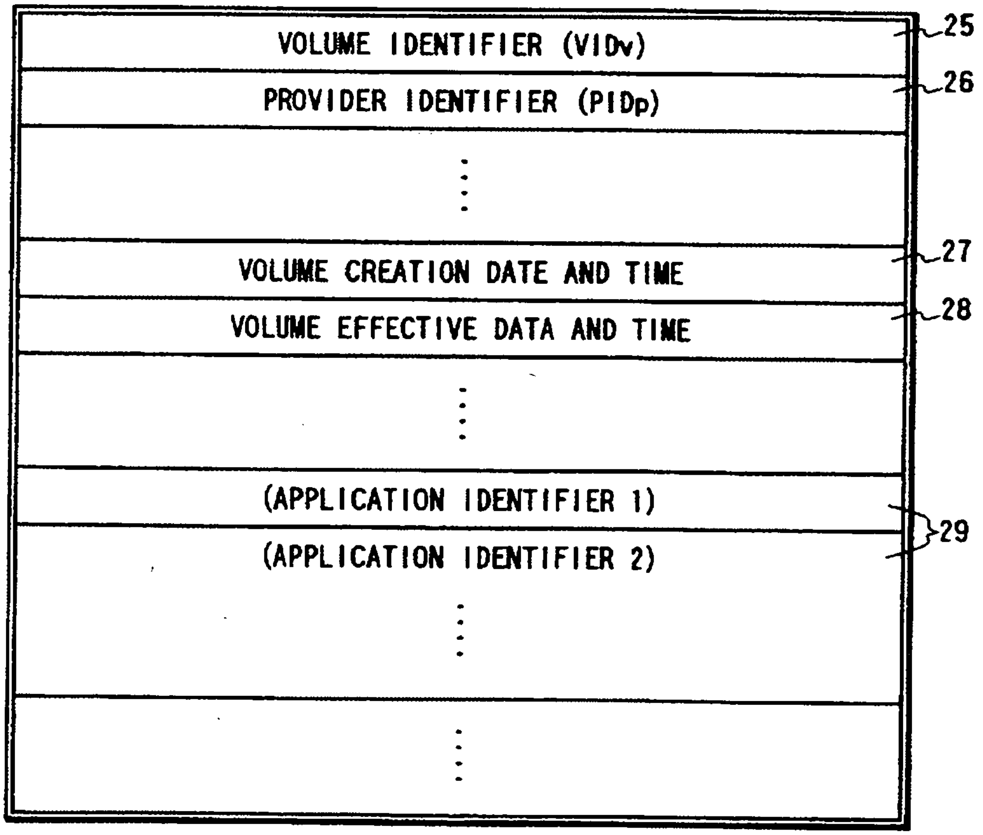

VOLUME IDENTIFIER (VIDv)
25
PROVIDER IDENTIFIER (PIDp)
26
⋮
VOLUME CREATION DATE AND TIME
27
VOLUME EFFECTIVE DATA AND TIME
28
⋮
(APPLICATION IDENTIFIER 1)
(APPLICATION IDENTIFIER 2)
29
⋮
⋮

FIG. 4

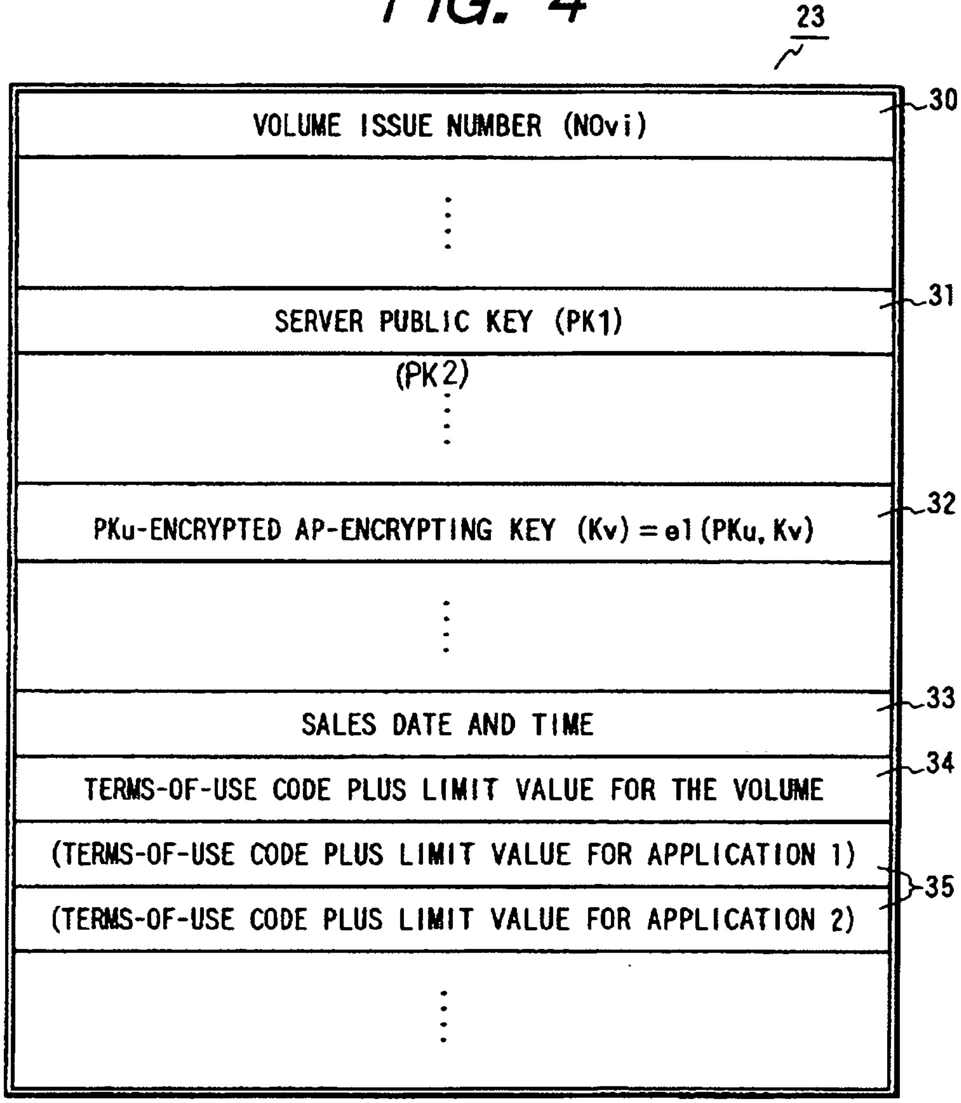

23
VOLUME ISSUE NUMBER (NOvi)
30
SERVER PUBLIC KEY (PK1)
(PK2)
31
PKu-ENCRYPTED AP-ENCRYPTING KEY (Kv) = e1 (PKu, Kv)
32
SALES DATE AND TIME
33
TERMS-OF-USE CODE PLUS LIMIT VALUE FOR THE VOLUME
34
(TERMS-OF-USE CODE PLUS LIMIT VALUE FOR APPLICATION 1)
(TERMS-OF-USE CODE PLUS LIMIT VALUE FOR APPLICATION 2)
35

FIG. 5

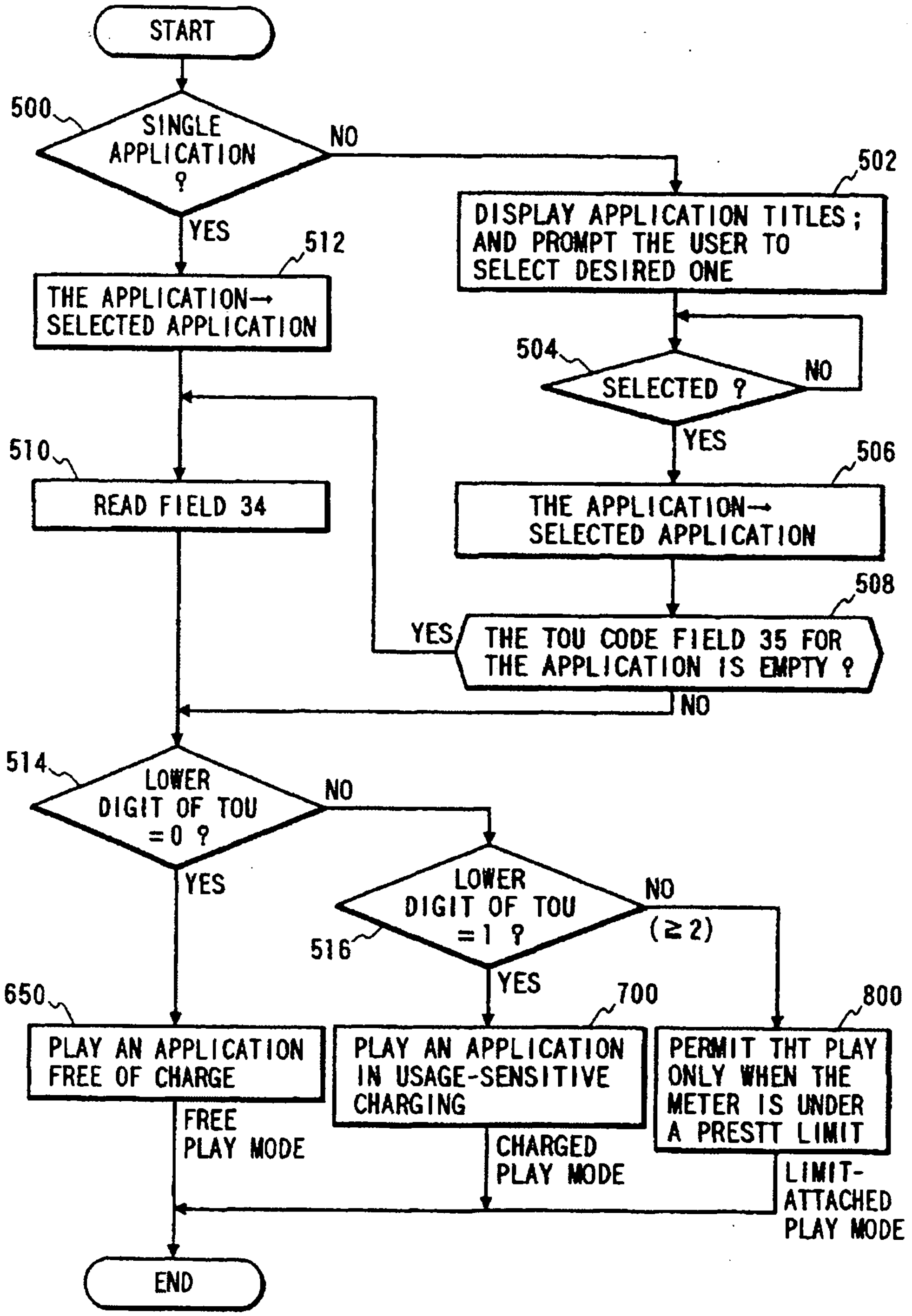

START
500
SINGLE APPLICATION ?
NO
YES
502
DISPLAY APPLICATION TITLES; AND PROMPT THE USER TO SELECT DESIRED ONE
512
THE APPLICATION→ SELECTED APPLICATION
504
SELECTED ?
NO
YES
510
READ FIELD 34
506
THE APPLICATION→ SELECTED APPLICATION
508
YES
THE TOU CODE FIELD 35 FOR THE APPLICATION IS EMPTY ?
NO
514
LOWER DIGIT OF TOU =0 ?
NO
YES
LOWER DIGIT OF TOU =1 ?
NO (≥2)
516
YES
650
700
800
PLAY AN APPLICATION FREE OF CHARGE
PLAY AN APPLICATION IN USAGE-SENSITIVE CHARGING
PERMIT THT PLAY ONLY WHEN THE METER IS UNDER A PRESTT LIMIT
FREE PLAY MODE
CHARGED PLAY MODE
LIMIT-ATTACHED PLAY MODE
END

FIG. 6A

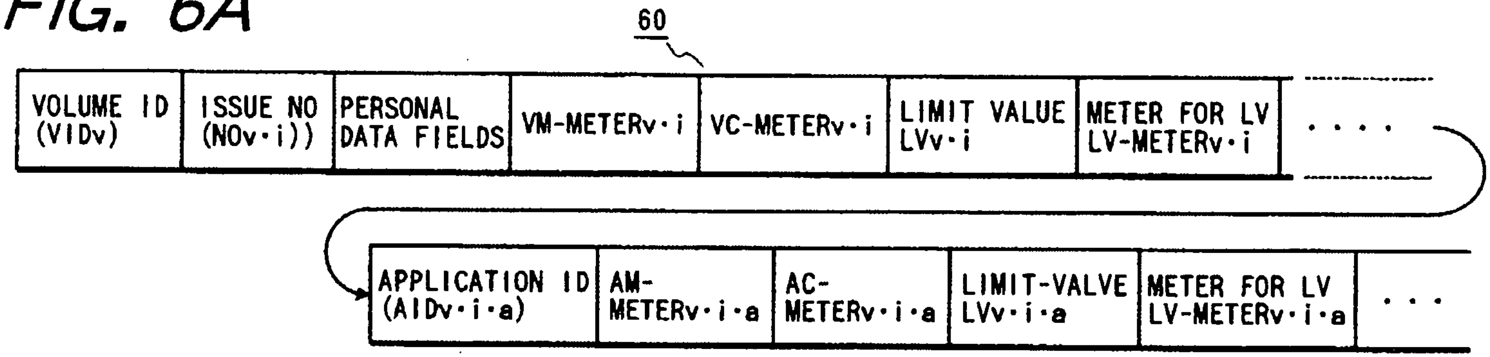

60
VOLUME ID (VIDv)
ISSUE NO (NOv·i))
PERSONAL DATA FIELDS
VM-METERv·i
VC-METERv·i
LIMIT VALUE LVv·i
METER FOR LV LV-METERv·i
. . . .
APPLICATION ID (AIDv·i·a)
AM-METERv·i·a
AC-METERv·i·a
LIMIT-VALVE LVv·i·a
METER FOR LV LV-METERv·i·a
. . .

FIG. 6B

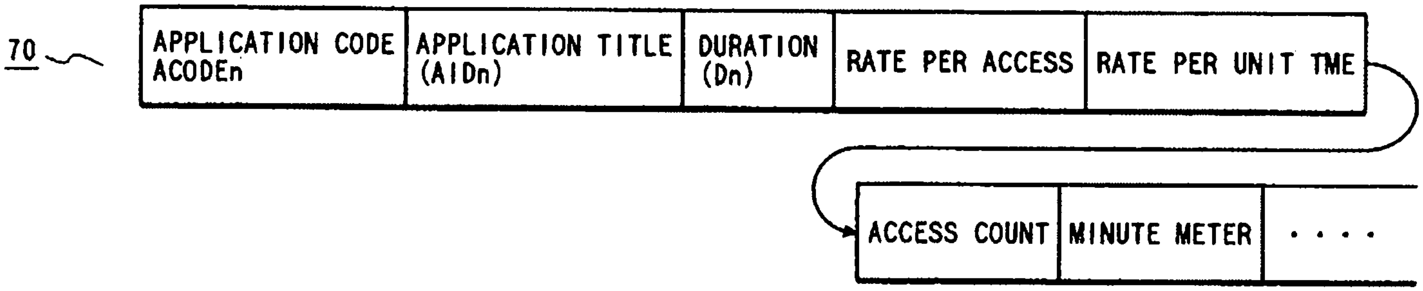

70
APPLICATION CODE ACODEn
APPLICATION TITLE (AIDn)
DURATION (Dn)
RATE PER ACCESS
RATE PER UNIT TME
ACCESS COUNT
MINUTE METER
. . . .

FIG. 7

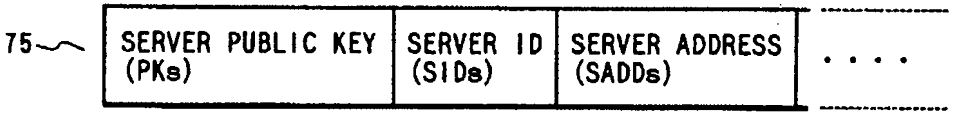

75
SERVER PUBLIC KEY (PKs)
SERVER ID (SIDs)
SERVER ADDRESS (SADDs)
. . . .

FIG. 8A

FIG. 8B

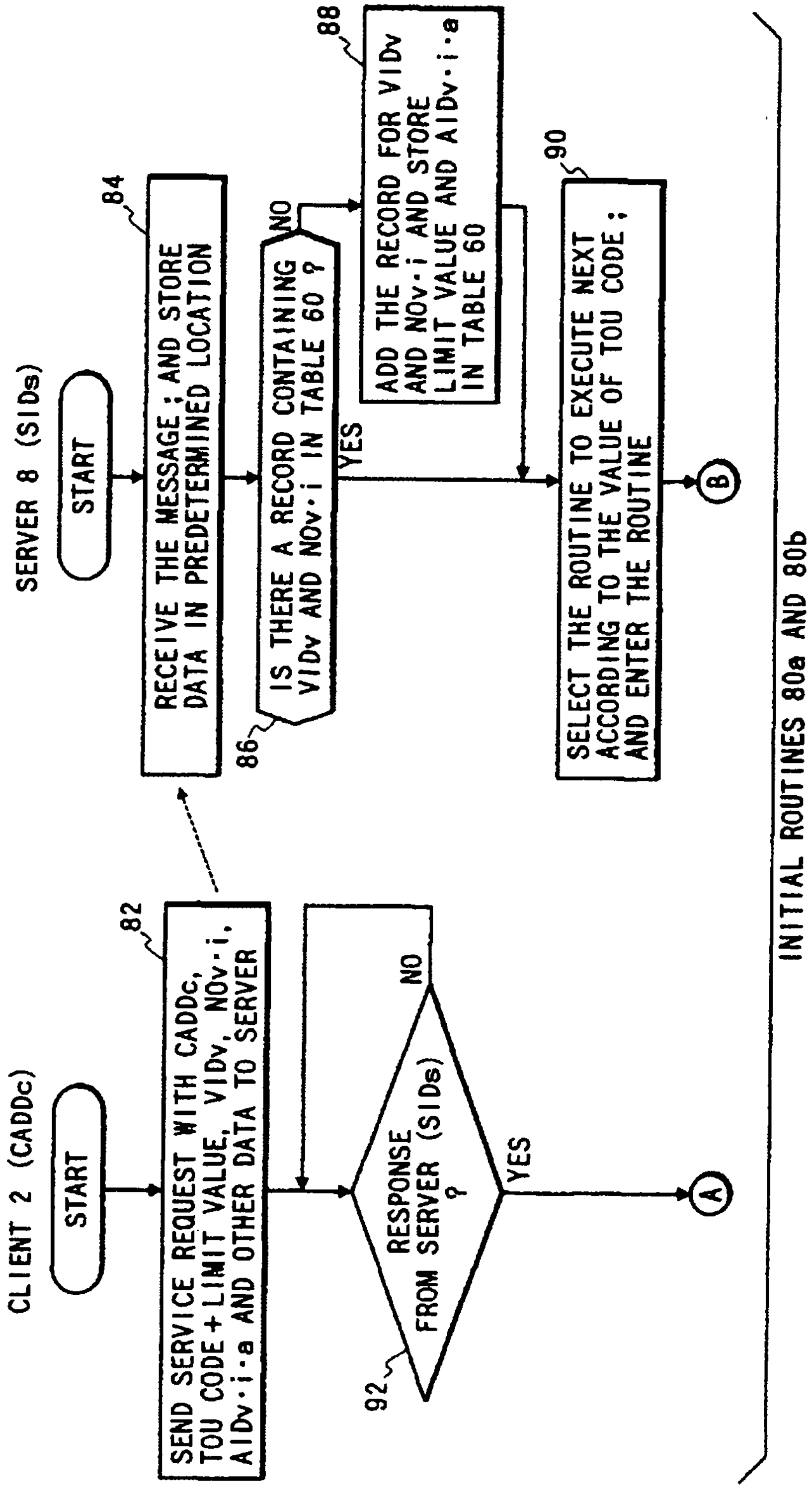

CLIENT 2 (CADDc)
START
SEND SERVICE REQUEST WITH CADDc, TOU CODE + LIMIT VALUE, VIDv, NOv·i, AIDv·i·a AND OTHER DATA TO SERVER
82
RESPONSE FROM SERVER (SIDs) ?
92
NO
YES
A
SERVER 8 (SIDs)
START
RECEIVE THE MESSAGE ; AND STORE DATA IN PREDETERMINED LOCATION
84
IS THERE A RECORD CONTAINING VIDv AND NOv·i IN TABLE 60 ?
86
NO
YES
ADD THE RECORD FOR VIDv AND NOv·i AND STORE LIMIT VALUE AND AIDv·i·a IN TABLE 60
88
SELECT THE ROUTINE TO EXECUTE NEXT ACCORDING TO THE VALUE OF TOU CODE ; AND ENTER THE ROUTINE
90
B
INITIAL ROUTINES 80a AND 80b

FIG. 9

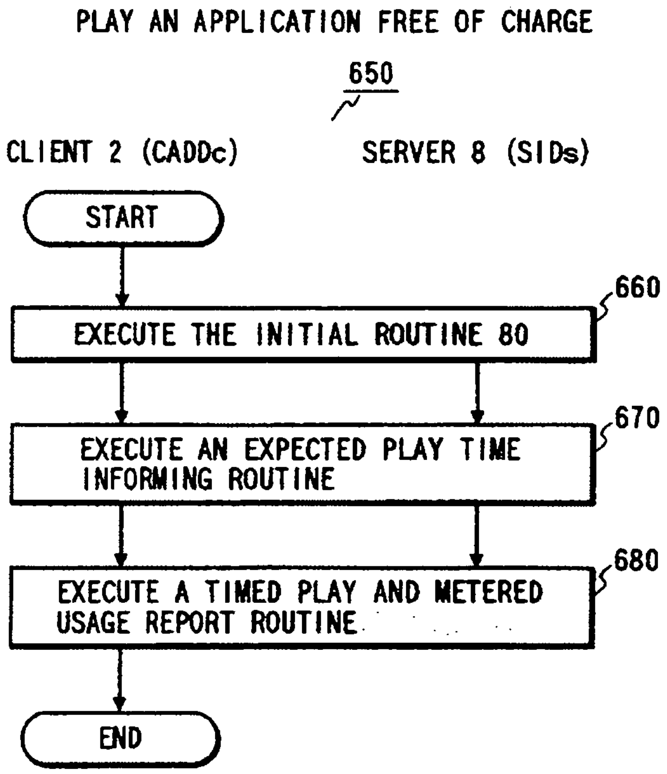
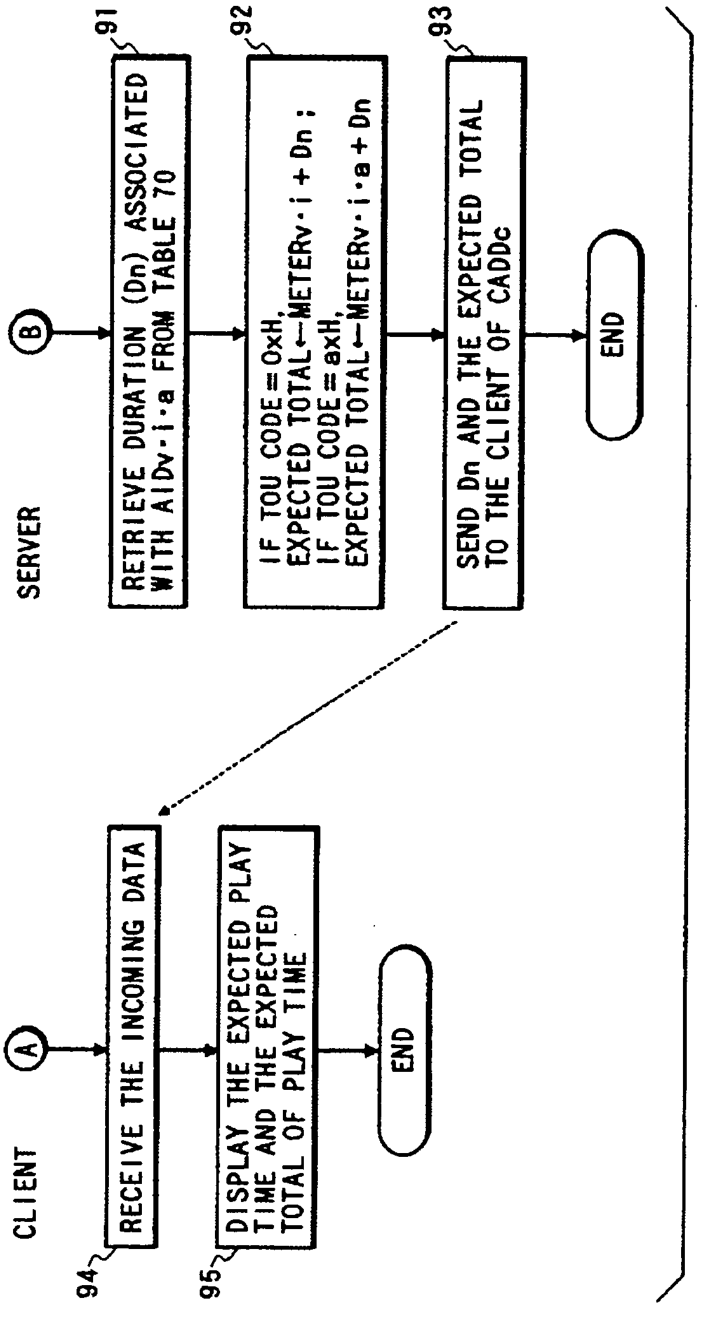

PLAY AN APPLICATION FREE OF CHARGE
650
CLIENT 2 (CADDc)
SERVER 8 (SIDs)
START
660
EXECUTE THE INITIAL ROUTINE 80
670
EXECUTE AN EXPECTED PLAY TIME INFORMING ROUTINE
680
EXECUTE A TIMED PLAY AND METERED USAGE REPORT ROUTINE
END

FIG. 10A

FIG. 10B


CLIENT
A
RECEIVE THE INCOMING DATA
94
DISPLAY THE EXPECTED PLAY TIME AND THE EXPECTED TOTAL OF PLAY TIME
95
END
SERVER
B
RETRIEVE DURATION (Dn) ASSOCIATED WITH AIDv·i·a FROM TABLE 70
91
IF TOU CODE=0xH,
EXPECTED TOTAL←METERv·i + Dn ;
IF TOU CODE=axH,
EXPECTED TOTAL←METERv·i·a + Dn
92
SEND Dn AND THE EXPECTED TOTAL TO THE CLIENT OF CADDc
93
END

EXPECTED PLAY TIME INFORMING ROUTINES 97a AND 97b

FIG. 11A FIG. 11B

TIMED PLAY AND METERED USAGE REPORT ROUTINES 675a AND 675b

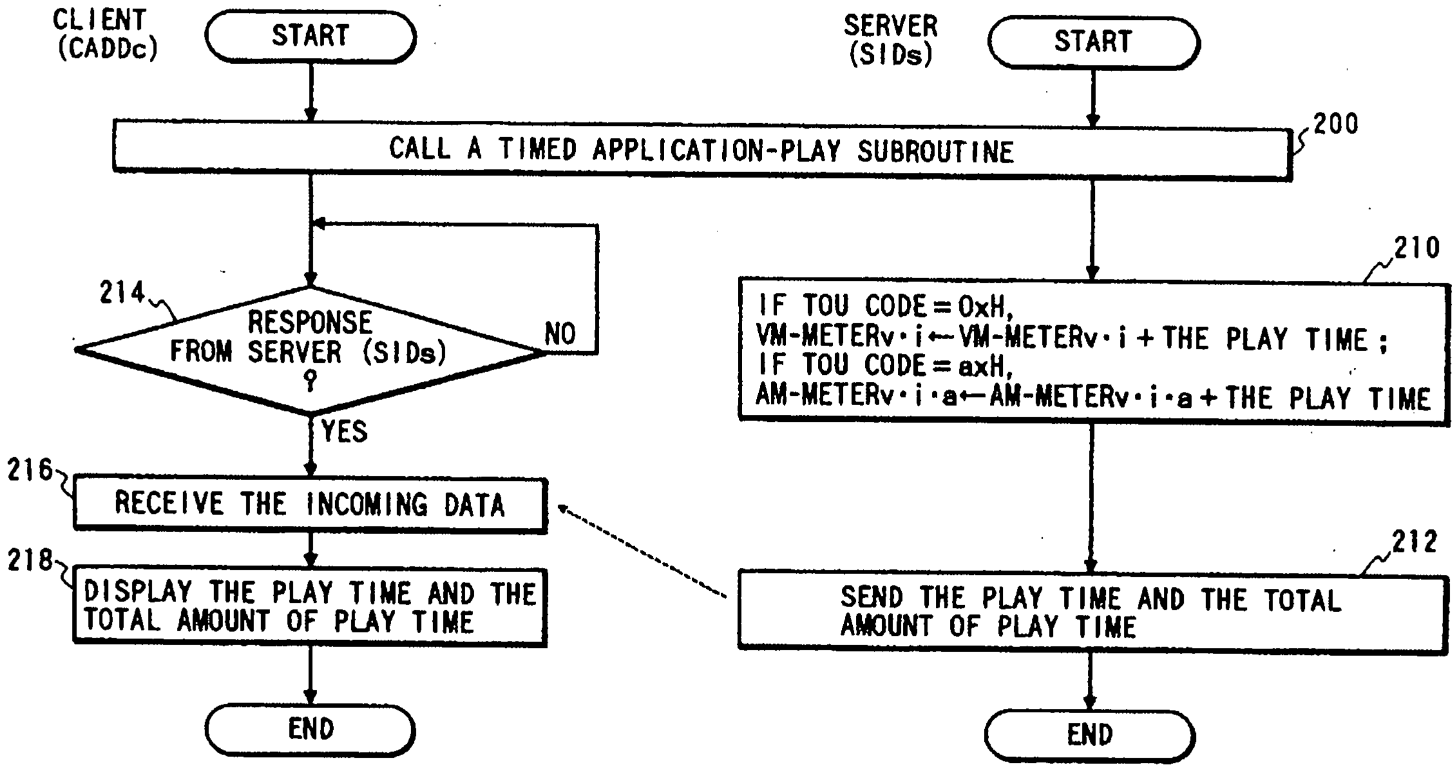

CLIENT (CADDc)
START
SERVER (SIDs)
START
CALL A TIMED APPLICATION-PLAY SUBROUTINE
200
214
RESPONSE FROM SERVER (SIDs) ?
NO
YES
210
IF TOU CODE = 0xH,
VM-METERv·i ← VM-METERv·i + THE PLAY TIME ;
IF TOU CODE = axH,
AM-METERv·i·a ← AM-METERv·i·a + THE PLAY TIME
216
RECEIVE THE INCOMING DATA
218
DISPLAY THE PLAY TIME AND THE TOTAL AMOUNT OF PLAY TIME
212
SEND THE PLAY TIME AND THE TOTAL AMOUNT OF PLAY TIME
END
END

FIG. 12A FIG. 12B

TIMED APPLICATION-PLAY SUBROUTINES

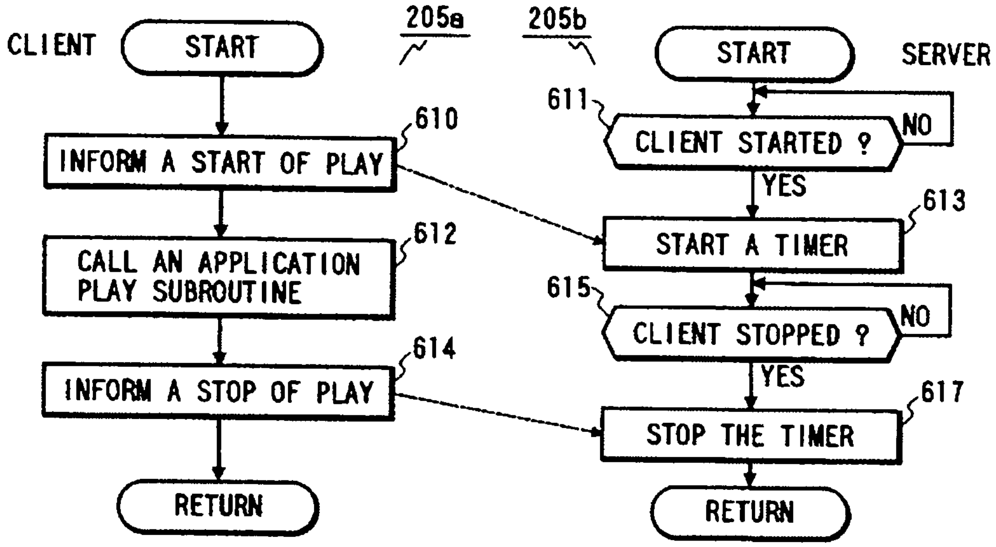

CLIENT
START
205a
205b
START
SERVER
611
610
INFORM A START OF PLAY
CLIENT STARTED ?
NO
YES
613
612
CALL AN APPLICATION PLAY SUBROUTINE
START A TIMER
615
NO
CLIENT STOPPED ?
614
INFORM A STOP OF PLAY
YES
617
STOP THE TIMER
RETURN
RETURN

FIG. 13A FIG. 13B

TIMED APPLICATION-PLAY SUBROUTINES

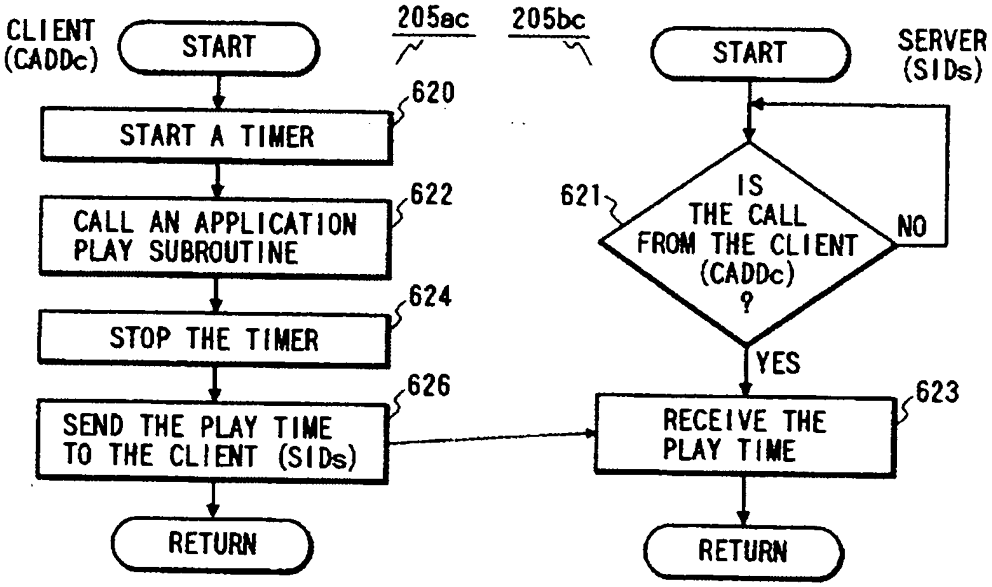

CLIENT (CADDc)
START
205ac
205bc
START
SERVER (SIDs)
620
START A TIMER
622
CALL AN APPLICATION PLAY SUBROUTINE
621
IS THE CALL FROM THE CLIENT (CADDc) ?
NO
624
STOP THE TIMER
YES
626
SEND THE PLAY TIME TO THE CLIENT (SIDs)
623
RECEIVE THE PLAY TIME
RETURN
RETURN

FIG. 14

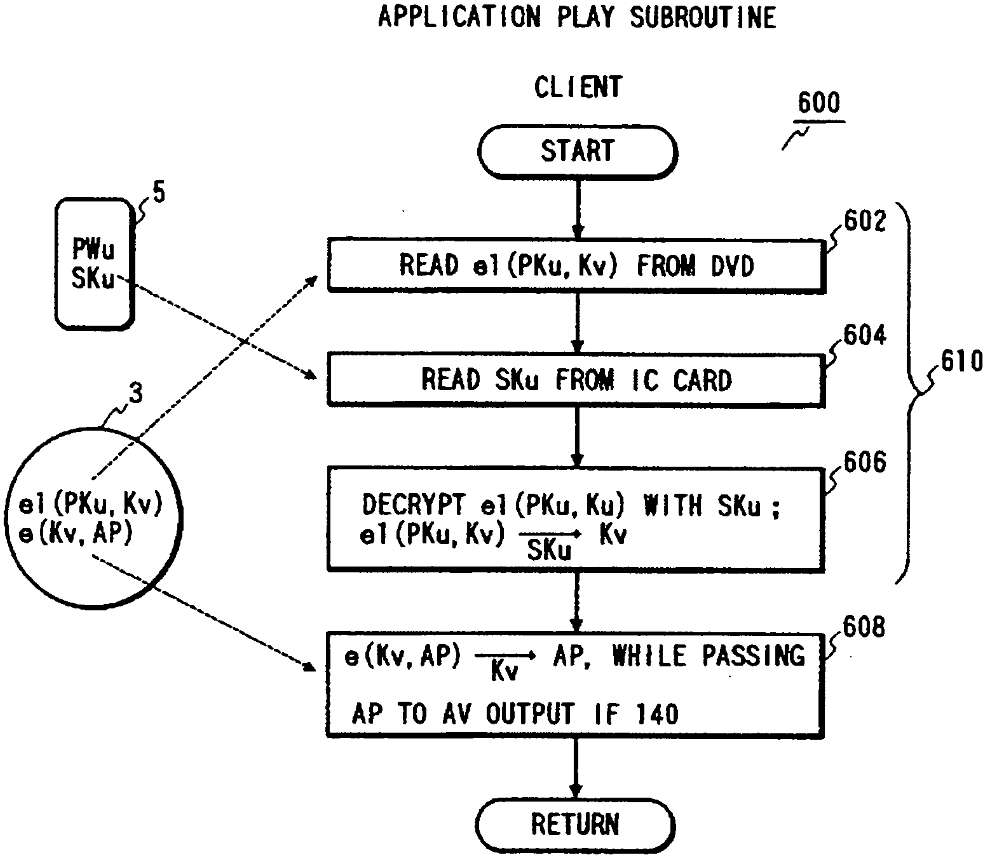

APPLICATION PLAY SUBROUTINE
CLIENT
600
START
5
PWu
SKu
602
READ e1 (PKu, Kv) FROM DVD
604
READ SKu FROM IC CARD
610
3
606
DECRYPT e1 (PKu, Ku) WITH SKu ;
e1 (PKu, Kv) $\xrightarrow[SKu]{}$ Kv
e1 (PKu, Kv)
e (Kv, AP)
608
e (Kv, AP) $\xrightarrow[Kv]{}$ AP, WHILE PASSING
AP TO AV OUTPUT IF 140
RETURN

FIG. 15

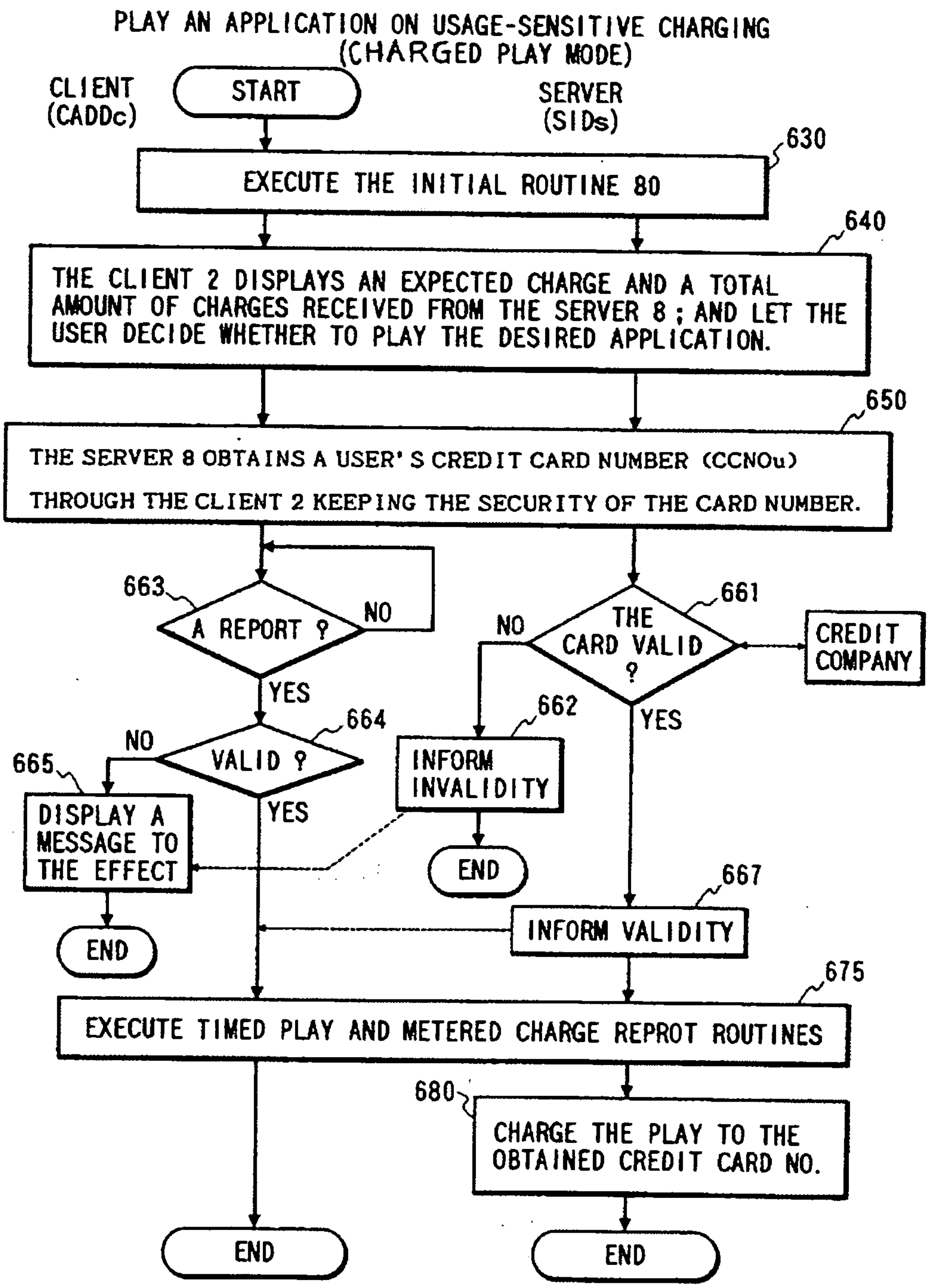

PLAY AN APPLICATION ON USAGE-SENSITIVE CHARGING
(CHARGED PLAY MODE)
CLIENT
(CADDc)
START
SERVER
(SIDs)
630
EXECUTE THE INITIAL ROUTINE 80
640
THE CLIENT 2 DISPLAYS AN EXPECTED CHARGE AND A TOTAL AMOUNT OF CHARGES RECEIVED FROM THE SERVER 8 ; AND LET THE USER DECIDE WHETHER TO PLAY THE DESIRED APPLICATION.
650
THE SERVER 8 OBTAINS A USER'S CREDIT CARD NUMBER (CCNOu) THROUGH THE CLIENT 2 KEEPING THE SECURITY OF THE CARD NUMBER.
663
A REPORT ?
NO
YES
661
THE CARD VALID ?
NO
CREDIT COMPANY
YES
664
VALID ?
NO
YES
665
DISPLAY A MESSAGE TO THE EFFECT
END
662
INFORM INVALIDITY
END
667
INFORM VALIDITY
675
EXECUTE TIMED PLAY AND METERED CHARGE REPROT ROUTINES
680
CHARGE THE PLAY TO THE OBTAINED CREDIT CARD NO.
END
END

FIG. 16A

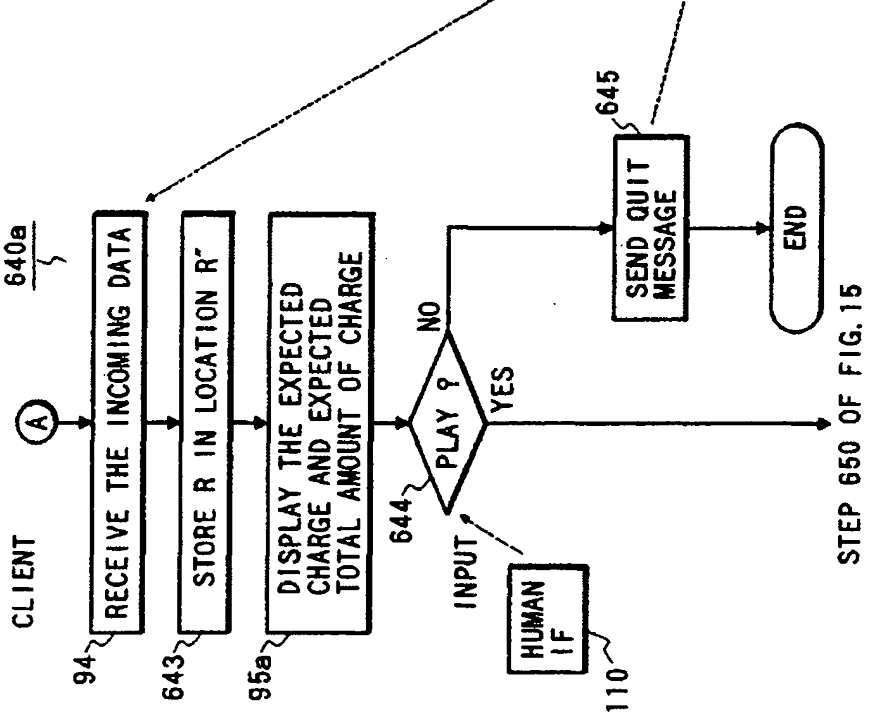

CLIENT
640a
A
RECEIVE THE INCOMING DATA
94
STORE R IN LOCATION R'
643
DISPLAY THE EXPECTED CHARGE AND EXPECTED TOTAL AMOUNT OF CHARGE
95a
PLAY ?
644
YES
NO
INPUT
HUMAN IF
110
SEND QUIT MESSAGE
645
END
STEP 650 OF FIG. 15

FIG. 16B

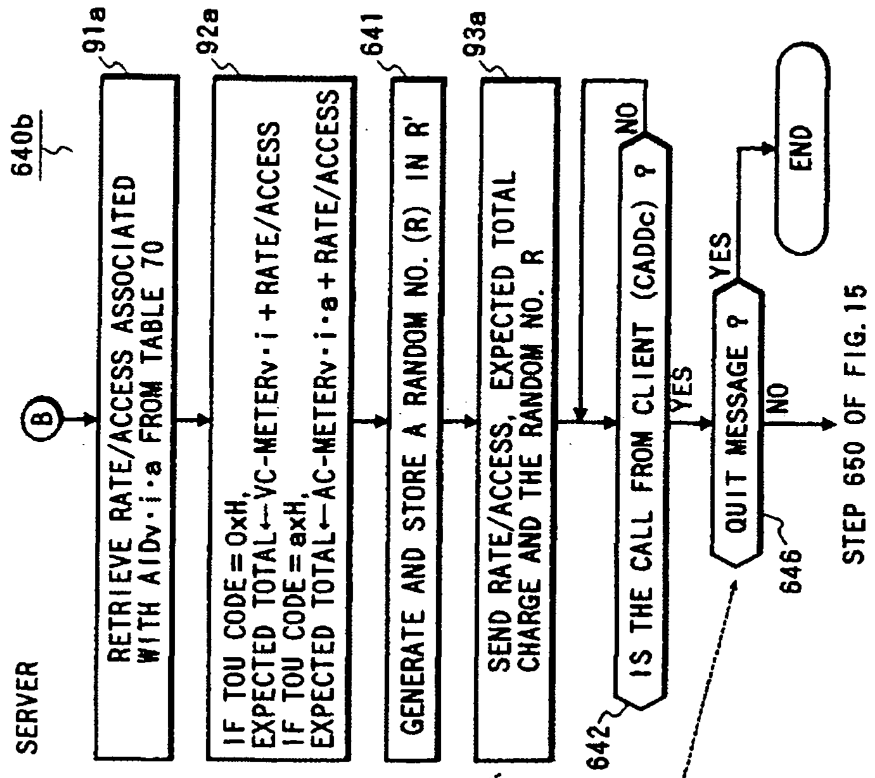

SERVER
640b
B
RETRIEVE RATE/ACCESS ASSOCIATED WITH AIDv·i·a FROM TABLE 70
91a
IF TOU CODE=0xH,
EXPECTED TOTAL←VC-METERv·i+RATE/ACCESS
IF TOU CODE=axH,
EXPECTED TOTAL←AC-METERv·i·a+RATE/ACCESS
92a
GENERATE AND STORE A RANDOM NO. (R) IN R'
641
SEND RATE/ACCESS, EXPECTED TOTAL CHARGE AND THE RANDOM NO. R
93a
IS THE CALL FROM CLIENT (CADDc) ?
642
NO
YES
QUIT MESSAGE ?
646
YES
NO
END
STEP 650 OF FIG. 15

FIG. 17A

FIG. 17B

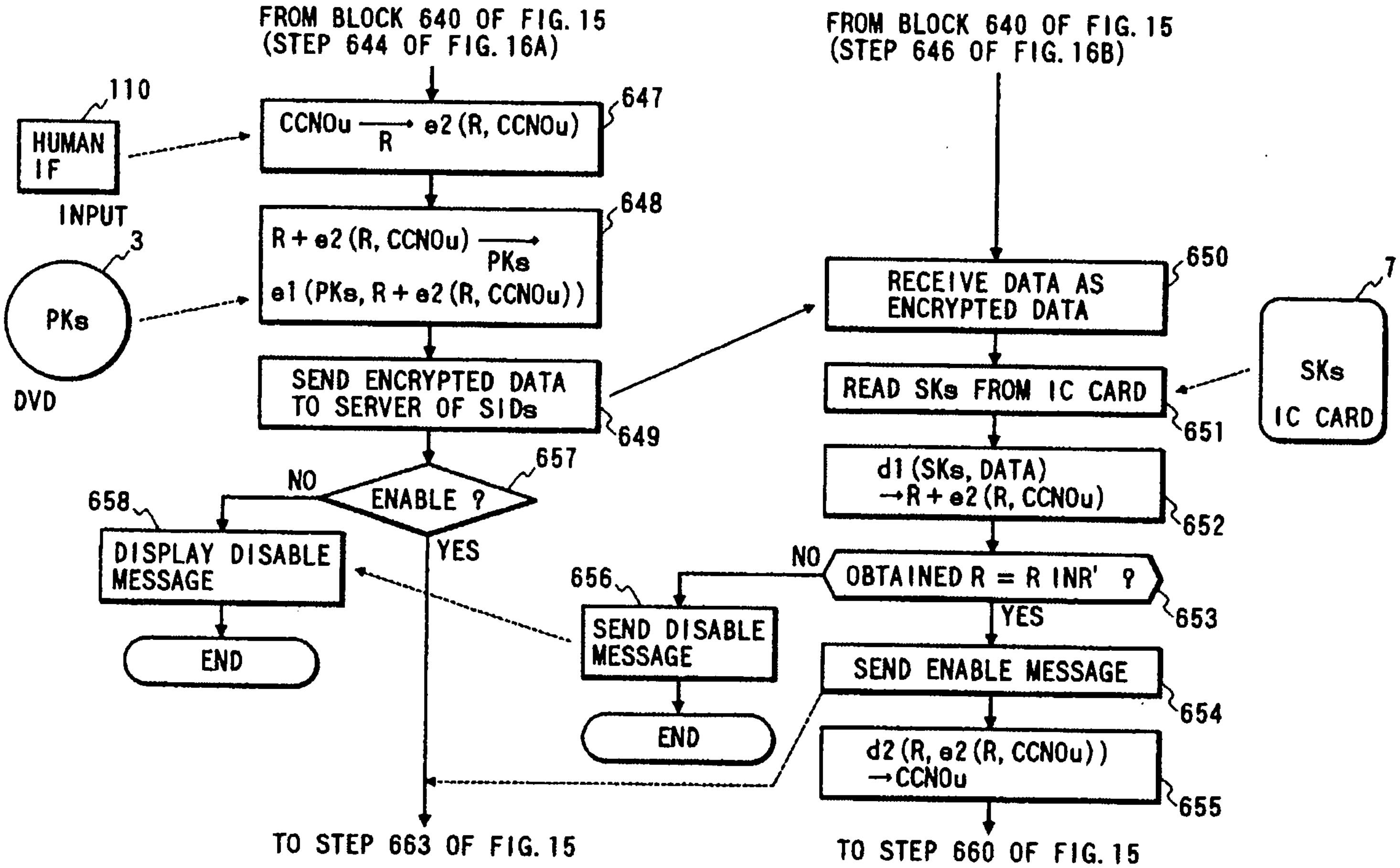

FROM BLOCK 640 OF FIG. 15
(STEP 644 OF FIG. 16A)
110
HUMAN IF
INPUT
3
PKs
DVD
CCNOu $\xrightarrow[R]{}$ e2 (R, CCNOu)
647
R + e2 (R, CCNOu) $\xrightarrow[PKs]{}$
e1 (PKs, R + e2 (R, CCNOu))
648
SEND ENCRYPTED DATA TO SERVER OF SIDs
649
ENABLE ?
657
NO
YES
658
DISPLAY DISABLE MESSAGE
END
TO STEP 663 OF FIG. 15
FROM BLOCK 640 OF FIG. 15
(STEP 646 OF FIG. 16B)
RECEIVE DATA AS ENCRYPTED DATA
650
7
SKs
IC CARD
READ SKs FROM IC CARD
651
d1 (SKs, DATA)
→R + e2 (R, CCNOu)
652
OBTAINED R = R INR' ?
653
NO
YES
656
SEND DISABLE MESSAGE
END
SEND ENABLE MESSAGE
654
d2 (R, e2 (R, CCNOu))
→CCNOu
655
TO STEP 660 OF FIG. 15

FIG. 18A FIG. 18B

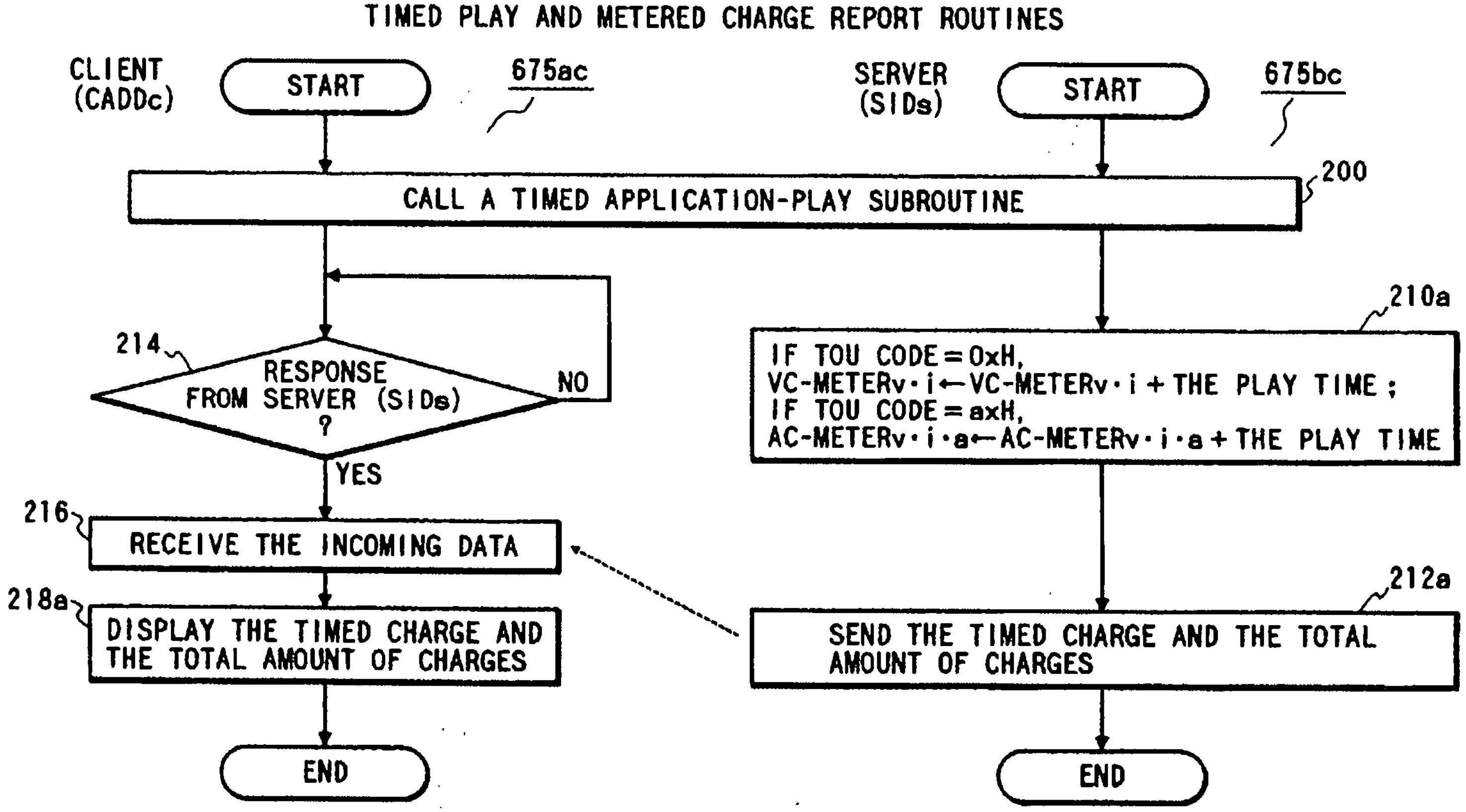

TIMED PLAY AND METERED CHARGE REPORT ROUTINES
CLIENT (CADDc)
START
675ac
SERVER (SIDs)
START
675bc
CALL A TIMED APPLICATION-PLAY SUBROUTINE
200
214
RESPONSE FROM SERVER (SIDs) ?
NO
YES
216
RECEIVE THE INCOMING DATA
218a
DISPLAY THE TIMED CHARGE AND THE TOTAL AMOUNT OF CHARGES
END
210a
IF TOU CODE=0xH,
VC-METERv·i←VC-METERv·i+THE PLAY TIME;
IF TOU CODE=axH,
AC-METERv·i·a←AC-METERv·i·a+THE PLAY TIME
212a
SEND THE TIMED CHARGE AND THE TOTAL AMOUNT OF CHARGES
END

FIG. 19

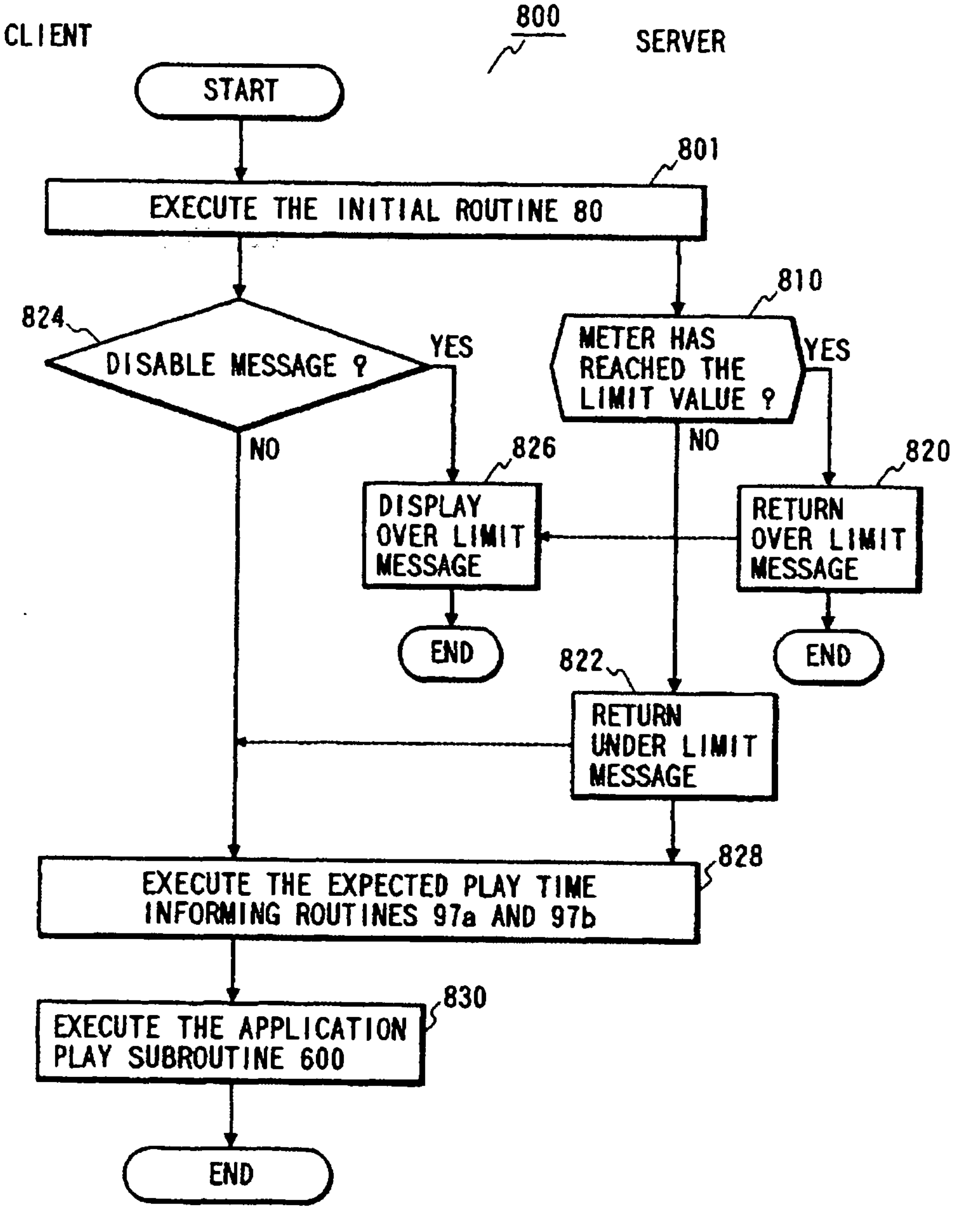

CLIENT
800
SERVER
START
801
EXECUTE THE INITIAL ROUTINE 80
824
DISABLE MESSAGE ?
YES
NO
810
METER HAS REACHED THE LIMIT VALUE ?
YES
NO
826
DISPLAY OVER LIMIT MESSAGE
820
RETURN OVER LIMIT MESSAGE
END
822
END
RETURN UNDER LIMIT MESSAGE
828
EXECUTE THE EXPECTED PLAY TIME INFORMING ROUTINES 97a AND 97b
830
EXECUTE THE APPLICATION PLAY SUBROUTINE 600
END

FIG. 20A

VIDv	Kv
VID1	K1
VID2	K2
⋮	⋮

FIG. 20B

VIDv	NOv·i	PKu
VID1	NO1·1	PK347020
	NO1·2	PK001031
	⋮	⋮
	NO1·365	PK314162
VID2	NO2·1	PK141421
⋮	⋮	⋮
	NO2·77	PK789012
VID3	NO3·1	PK123456
⋮	⋮	⋮

FIG. 20C

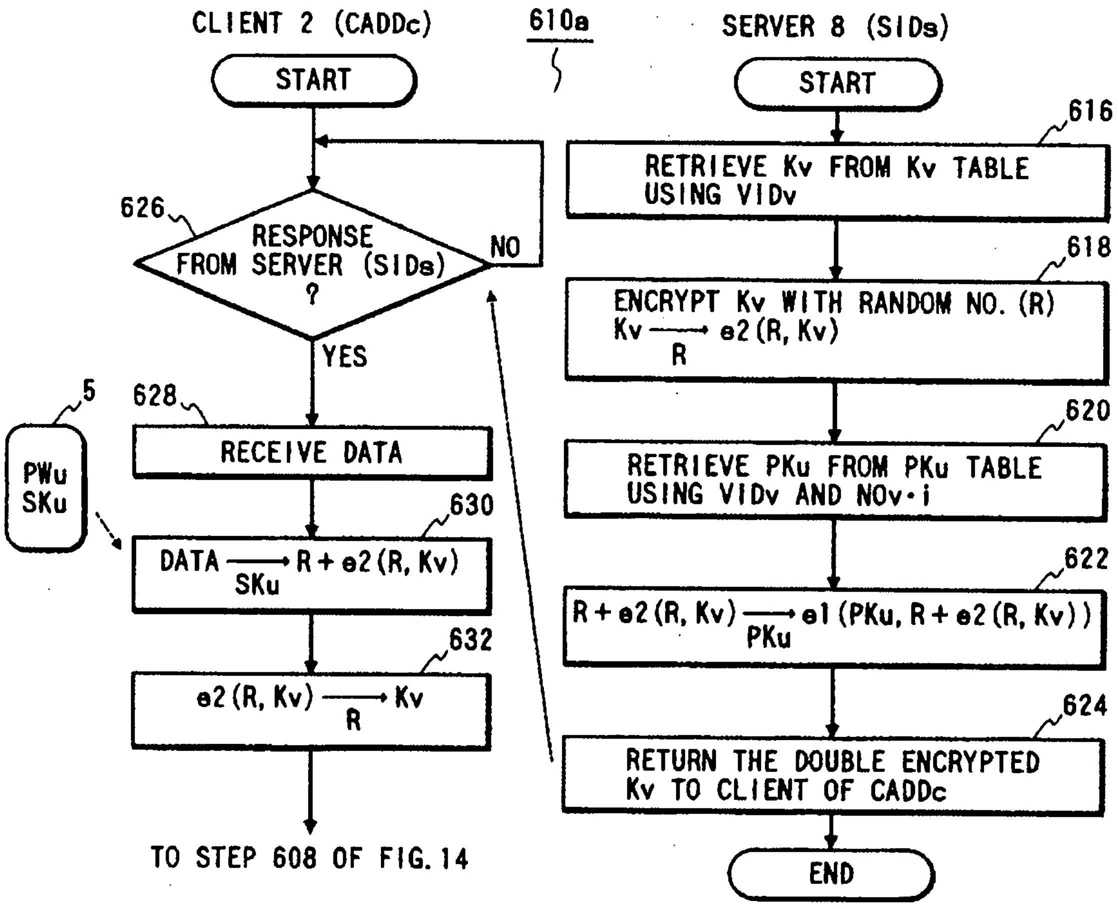

CLIENT 2 (CADDc)
610a
SERVER 8 (SIDs)
START
626
RESPONSE FROM SERVER (SIDs) ?
NO
YES
628
RECEIVE DATA
5
PWu
SKu
630
DATA $\xrightarrow[SKu]{}$ R + e2 (R, Kv)
632
e2 (R, Kv) $\xrightarrow[R]{}$ Kv
TO STEP 608 OF FIG. 14
START
616
RETRIEVE Kv FROM Kv TABLE USING VIDv
618
ENCRYPT Kv WITH RANDOM NO. (R)
Kv $\xrightarrow[R]{}$ e2 (R, Kv)
620
RETRIEVE PKu FROM PKu TABLE USING VIDv AND NOv·i
622
R + e2 (R, Kv) $\xrightarrow[PKu]{}$ e1 (PKu, R + e2 (R, Kv))
624
RETURN THE DOUBLE ENCRYPTED Kv TO CLIENT OF CADDc
END

FIG. 21

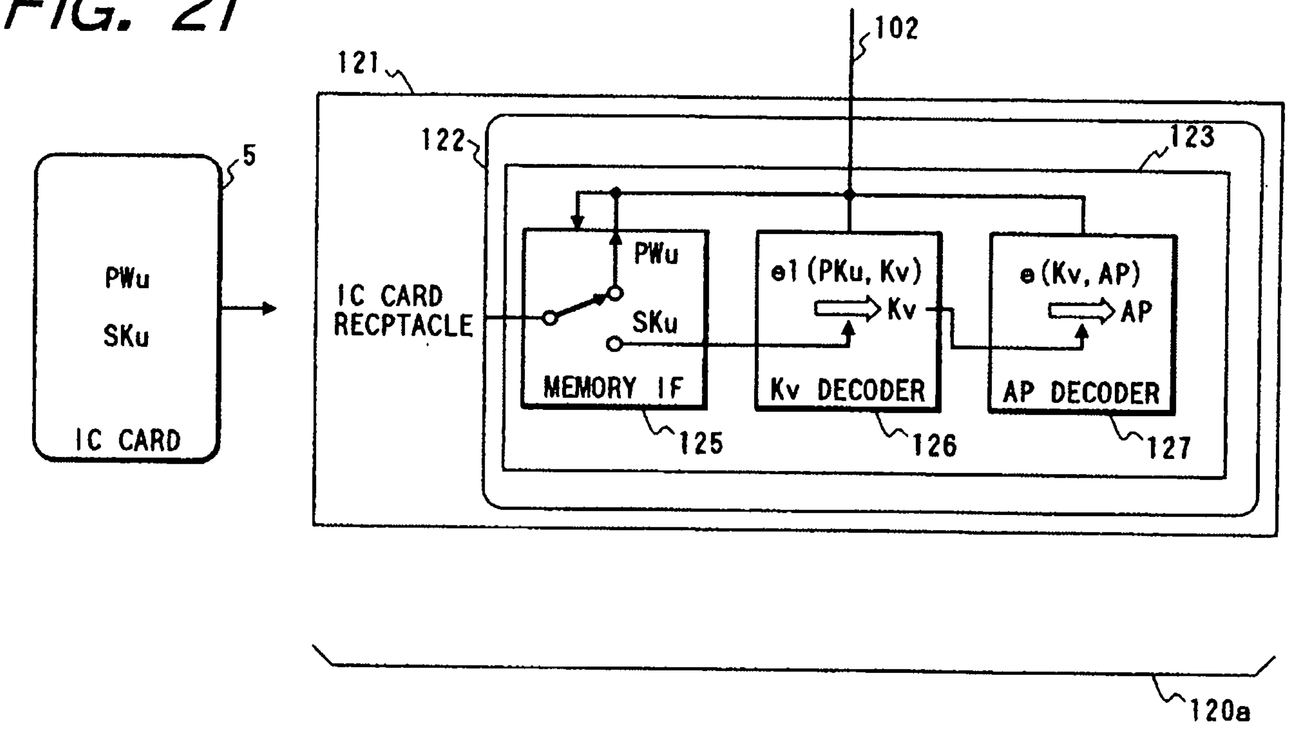

121
102
122
123
5
PWu
SKu
IC CARD
IC CARD RECPTACLE
PWu
SKu
MEMORY IF
e1 (PKu, Kv)
Kv
Kv DECODER
e (Kv, AP)
AP
AP DECODER
125
126
127
120a

FIG. 22

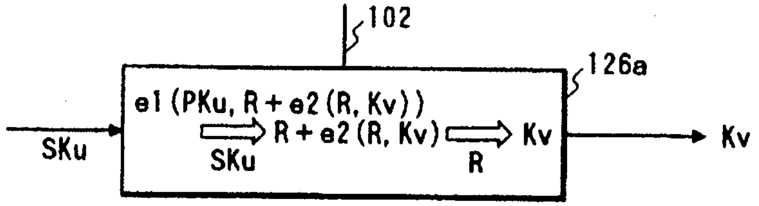
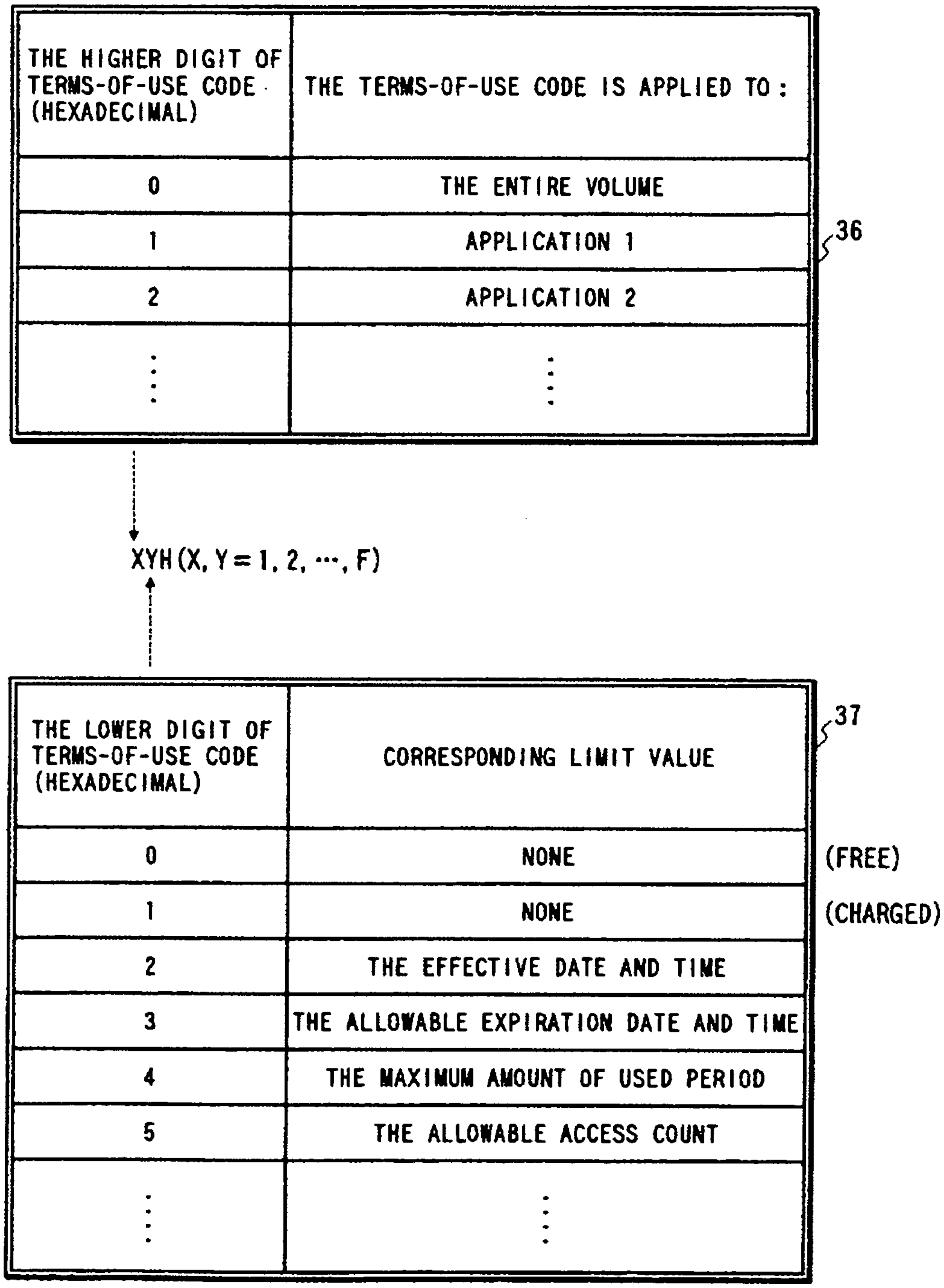

102
126a
e1 (PKu, R + e2 (R, Kv))
SKu
R + e2 (R, Kv)
SKu
Kv
R
SKu
Kv

FIG. 23



THE HIGHER DIGIT OF TERMS-OF-USE CODE (HEXADECIMAL)	THE TERMS-OF-USE CODE IS APPLIED TO :
0	THE ENTIRE VOLUME
1	APPLICATION 1
2	APPLICATION 2
⋮	⋮

THE LOWER DIGIT OF TERMS-OF-USE CODE (HEXADECIMAL)	CORRESPONDING LIMIT VALUE
0	NONE
1	NONE
2	THE EFFECTIVE DATE AND TIME
3	THE ALLOWABLE EXPIRATION DATE AND TIME
4	THE MAXIMUM AMOUNT OF USED PERIOD
5	THE ALLOWABLE ACCESS COUNT
⋮	⋮

FIG. 24

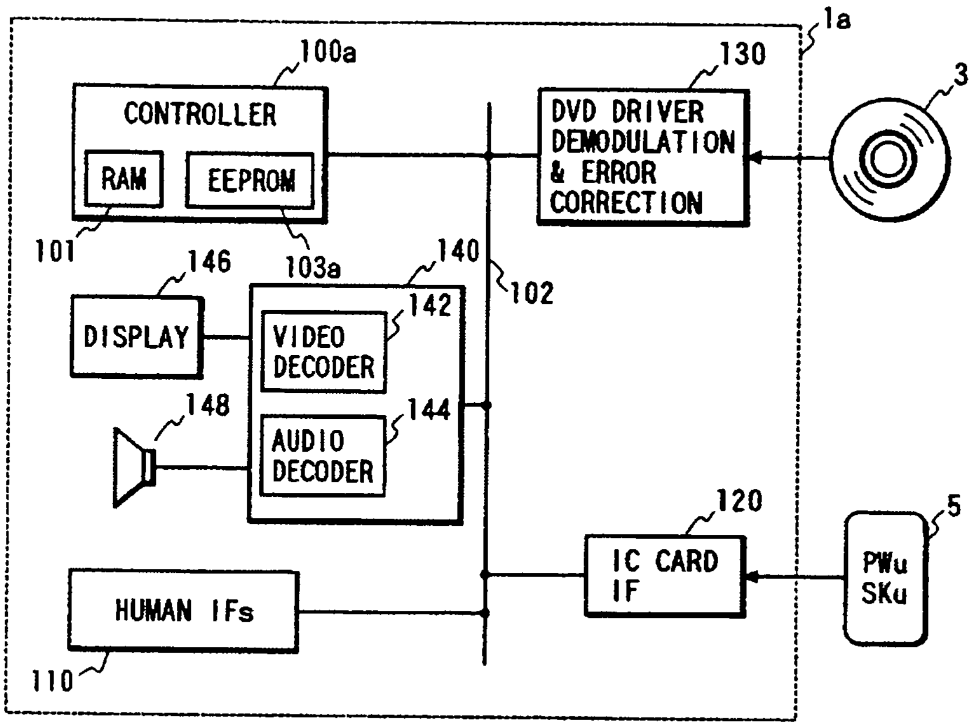

1a
100a
CONTROLLER
RAM
EEPROM
101
103a
130
DVD DRIVER
DEMODULATION
& ERROR
CORRECTION
3
146
DISPLAY
140
142
VIDEO
DECODER
102
148
144
AUDIO
DECODER
120
IC CARD
IF
5
PWu
SKu
HUMAN IFs
110

FIG. 26

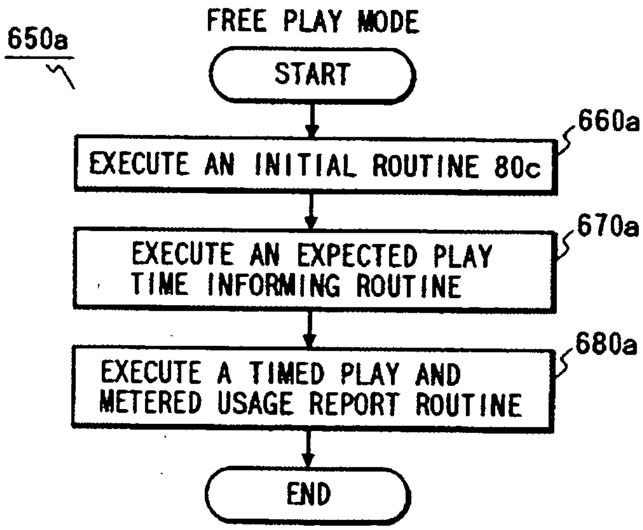

FREE PLAY MODE
650a
START
660a
EXECUTE AN INITIAL ROUTINE 80c
670a
EXECUTE AN EXPECTED PLAY
TIME INFORMING ROUTINE
680a
EXECUTE A TIMED PLAY AND
METERED USAGE REPORT ROUTINE
END

FIG. 25

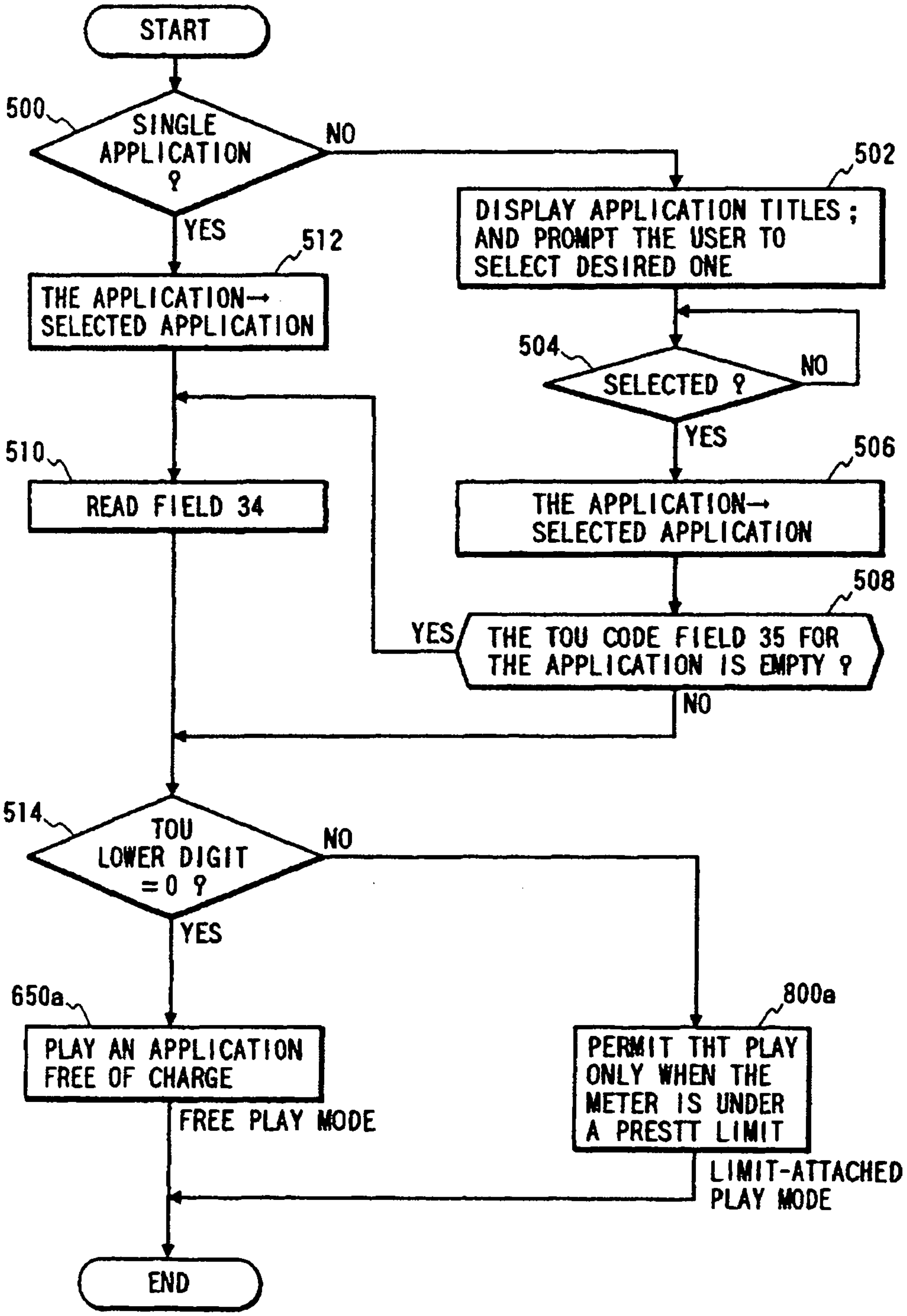

START
500
SINGLE APPLICATION ?
NO
502
DISPLAY APPLICATION TITLES ; AND PROMPT THE USER TO SELECT DESIRED ONE
YES
512
THE APPLICATION→ SELECTED APPLICATION
504
SELECTED ?
NO
YES
506
THE APPLICATION→ SELECTED APPLICATION
510
READ FIELD 34
508
YES
THE TOU CODE FIELD 35 FOR THE APPLICATION IS EMPTY ?
NO
514
TOU LOWER DIGIT =0 ?
NO
YES
650a
800a
PLAY AN APPLICATION FREE OF CHARGE
PERMIT THT PLAY ONLY WHEN THE METER IS UNDER A PRESTT LIMIT
FREE PLAY MODE
LIMIT-ATTACHED PLAY MODE
END

FIG. 27

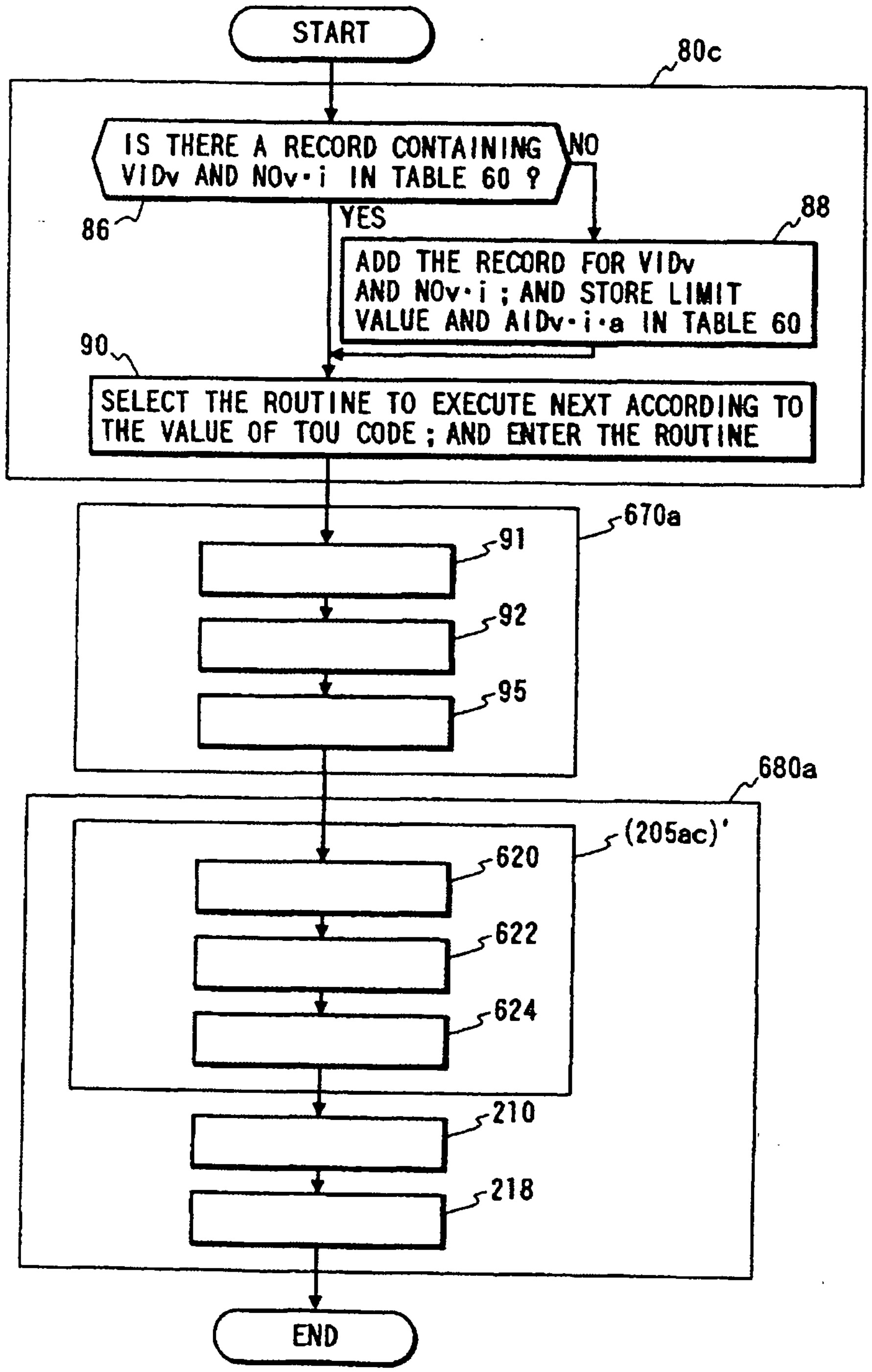
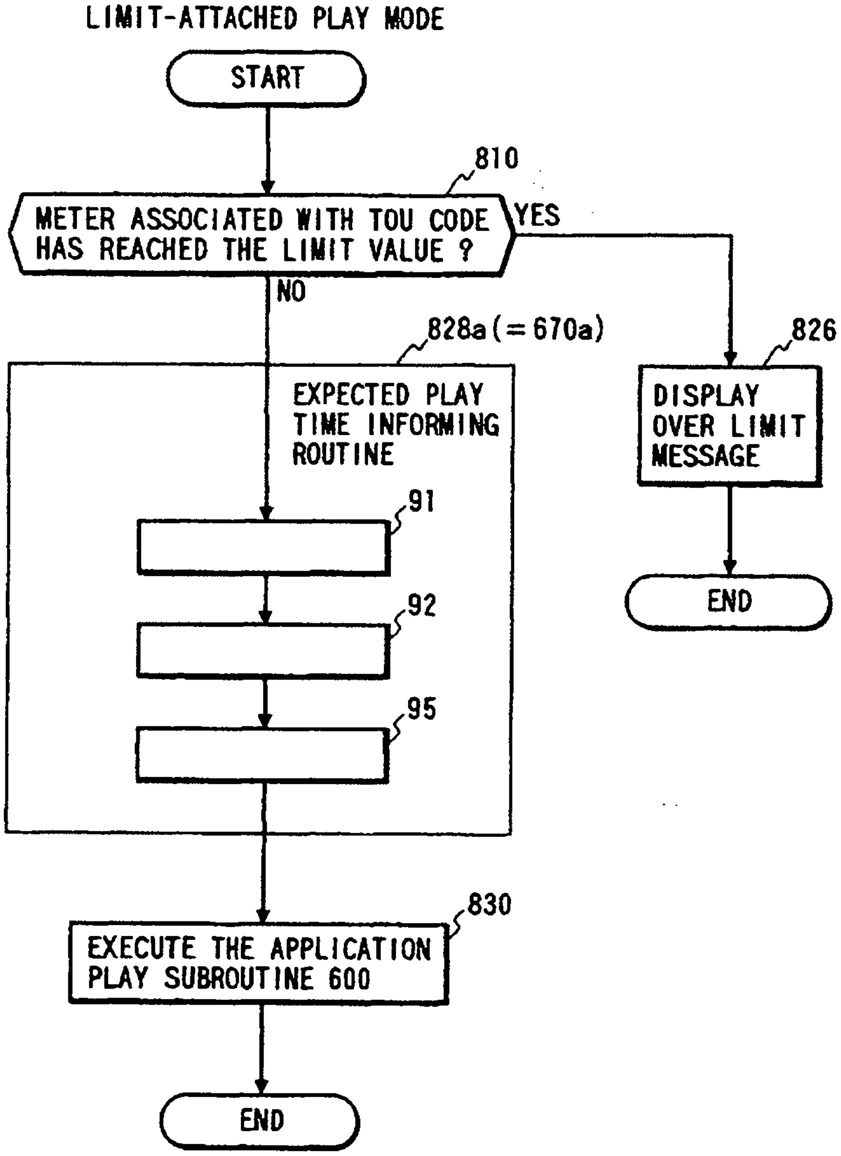

START
IS THERE A RECORD CONTAINING VIDv AND NOv·i IN TABLE 60 ?
NO
YES
ADD THE RECORD FOR VIDv AND NOv·i ; AND STORE LIMIT VALUE AND AIDv·i·a IN TABLE 60
SELECT THE ROUTINE TO EXECUTE NEXT ACCORDING TO THE VALUE OF TOU CODE ; AND ENTER THE ROUTINE
(205ac)'
END

FIG. 28


LIMIT-ATTACHED PLAY MODE
START
810
METER ASSOCIATED WITH TOU CODE HAS REACHED THE LIMIT VALUE ?
YES
NO
828a (= 670a)
EXPECTED PLAY TIME INFORMING ROUTINE
91
92
95
826
DISPLAY OVER LIMIT MESSAGE
END
830
EXECUTE THE APPLICATION PLAY SUBROUTINE 600
END

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 892 521 A2

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
20.01.1999 Bulletin 1999/03

(51) Int Cl.6: **H04L 9/32**

(21) Application number: **98305646.6**

(22) Date of filing: **15.07.1998**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **15.07.1997 US 892792**

(71) Applicant: **Hewlett-Packard Company**
Palo Alto, California 94304 (US)

(72) Inventor: **Zamek, Steven**
Palo Alto, California 94303 (US)

(74) Representative: **Jehan, Robert et al**
Williams, Powell & Associates,
4 St Paul's Churchyard
London EC4M 8AY (GB)

(54) **Method and apparatus for long term verification of digital signatures**

(57) The time over which a digital signature can be verified is extended well beyond the expiration of any or all of the certificates upon which that signature depends. In a "save state" approach, an archive facility is used to store public key infrastructure (PKI) state, e.g. cryptographic information, such as certificates and certificate revocation lists (CRLs), in addition to non-cryptographic information, such as trust policy statements or the document itself. This information comprises all that is necessary to re-create the signature verification process at a later time. When a user wants to verify the signature on a document, possibly years later, a long term signature verification (LTSV) server re-creates the precise state of the PKI at the time the document was originally submitted. The LTSV server restores the state, and the signature verification process executes the exact process it performed (or would have performed) years earlier. Another embodiment of the invention combines the strength of cryptography with the proven resilience of (non-public key) technology and procedures currently associated with secure data stores by saving the PKI state for future verification; and protecting the PKI state information from intrusion by maintaining it in a secure storage facility which is protected by services, such as firewalls, access control mechanisms, audit facilities, intrusion detection facilities, physical isolation, and network isolation.

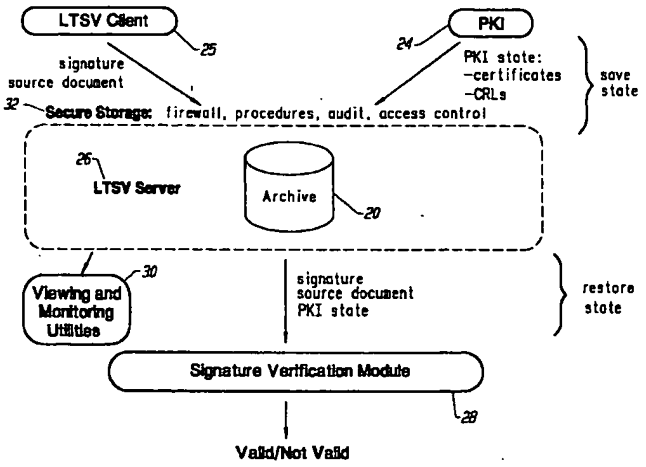

LTSV Client
25
signature
source document
24
PKI
PKI state:
-certificates
-CRLs
save state
32
Secure Storage: firewall, procedures, audit, access control
26
LTSV Server
Archive
20
30
Viewing and Monitoring Utilities
signature
source document
PKI state
restore state
Signature Verification Module
28
Valid/Not Valid

FIG. 3

EP 0 892 521 A2

Printed by Jouve, 75001 PARIS (FR)

Description

This invention relates to a method and apparatus for the long term verification of digital signatures.

The technology of digital signatures opens up the likelihood of increased use of digital networks (including the Internet) for electronic commerce. It is now feasible to send and receive digitally signed documents that represent transactions of some value to one or more parties.

Currently, a digital signature is verifiable only as long as the digital certificates upon which it depends have not expired. Given the expectation that a certificate's life span is in the area of one to two years duration, current technology does not support the emerging needs of the electronic commerce market, where the durability of digital signatures over time is a requirement.

For certain applications, the recipient of digitally signed documents should be able to verify the authenticity of a document years after the document was signed, just as the document's authenticity can be verified at the time of signing. Unfortunately, the current state of the technology does not provide for the verification of these digital signatures after certificate expiration because it is the nature of keys and certificates used for signing and encrypting documents to expire after a specific period of time (typically after a year or two). This is due, at least in part, to the fact that the strength of keys is expected to degrade over time because of such factors as improvements in computing speed and breakthroughs in cryptoanalysis. Moreover, the longer the key is in use, the longer that an adversary has to attempt to crack the key. Therefore, it is standard practice to replace keys periodically. This is why certificates have specific expiration dates.

An examination of the current state of the technology reveals that a digital signature verification module would fail if presented with a request to verify a signed document in which any of the associated certificates had expired. Fig. 1 is a block schematic diagram illustrating certification expiration. This simple example demonstrates that, given a certificate 10 having a two-year life span (*e.g.* from 4/1/96 to 4/1/98), a signature could be successfully verified six months (*e.g.* on 10/1/96) after certificate issuance (100); but this same signature would not be successfully verified three years later (*e.g.* on 4/1/99) (102). This behavior is clearly unacceptable if the duration of a document, for example contract, must extend beyond the duration of the certificates' life.

Further, some current systems use certificate revocation lists (CRLs) to revoke certificates and remove them therefrom, once those certificates expire. This means that a record of those CRLs generally disappears, making long term signature verification impossible using known techniques.

It is known to reconstruct past trust (see A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, pp. 583 (1996)). In this approach, both signature reverification relative to a past point in time and resolution of disputes may require reconstruction of chains of trust from a past point in time. This requires archival of keying material and related information for reconstruction of past chains of trust. Direct reconstruction of such past chains is taught to be unnecessary if a notarizing agent is used. A notarizing agent is defined as a general service capable not only of ascertaining the existence of a document at a certain time, but of vouching for the truth of more general statements at certain points in time. The original verification of the notary is taught to establish the existence of a trust chain at that point in time, and subsequently its record thereof is taught to serve as proof of prior validity. It is taught that details of the original trust chain may be recorded for audit purposes. It is not taught that a document can be verified based upon the existence of expired certificates. Rather, reliance is placed upon the use of the notarizing agent. It is further taught that the archived keying material can be used as evidence at a future time to allow resolution of disputed signatures by non-automated procedures.

It would be advantageous to provide a technique for extending the time over which the authenticity and integrity of digital signatures can be accurately verified beyond the time that any relevant certificates expire.

The present invention seeks to provide improved signature verification.

According to an aspect of the present invention there is provided a method of enabling long term verification of digital signatures as specified in claim 1.

According to another aspect of the present invention there is provided apparatus as specified in claim 11.

The prefered embodiment provides a method and apparatus which effectively extends the time over which a digital signature can be verified, *i.e.* well beyond the expiration of any or all of the certificates upon which that signature depends. The invention can be used for any application domain where users want digital signatures to be applied to long lasting documents (*e.g.* contracts), and be independently verifiable years or decades after signing the document. The preferred embodiment provides two alternative approaches to constructing a solution which delivers long term signature verification (LTSV).

One embodiment of the invention provides an approach for solving the LTSV problem that is referred to herein as the "save state" approach. This embodiment of the invention largely entails the use of cryptographic information and techniques. Thus, an archive facility is used to store the public key infrastructure (PKI) state, *e.g.* cryptographic information, such as certificates and CRLS, in addition to the document itself. This information comprises all that is necessary to re-create the signature verification process at a later time. It may also be desirable to store the source document separately from the cryptographic information (such as the signature, certificates, and CRLs) for reasons of privacy. For example, a user may want to have control over the source document. The PKI state information may contain either or both of cryptographically protected information, such as certificates and CRLs, and information that is not cryptographically protected, such as the public key of a root certification authority or policy information.

When a user wants to reverify the signature on a document, possibly years later, an LTSV server re-creates the precise state of the PKI at the time the document was originally submitted. The LTSV server restores the state, and the signature verification process executes the exact process it performed (or would have performed) years earlier. The time used as the basis for re-creation of the signature verification process does not have to be the time of submittal. Rather, the time could be some other relevant time, such as when a document was signed by the originator or when it was verified by a recipient.

Another embodiment of the invention combines the strength of cryptography with the proven resilience of (non-public key) technology and procedures currently associated with secure data stores. An example of this embodiment provides a mechanism that:

- Saves the PKI state for future reverification; and
- Protects the PKI state information from intrusion by either maintaining it in a secure storage facility which is protected by services, such as firewalls, access control mechanisms, audit facilities, intrusion detection facilities, physical isolation, and network isolation; and/or employing a cryptographic protection mechanism, for example using the LTSV server to sign the PKI state information or using a keyed hash algorithm.

In addition, other non-cryptographic features may be added to such approaches to deliver a highly secure and trusted LTSV solution, including, for example utilities for viewing the PKI state information (cryptographic as well as non-cryptographic) and visually monitoring the security of the system. These utilities can be used to provide visual evidence for purposes of dispute resolution.

One enhancement to the secure storage approach herein disclosed maintains certain evidence, such as certificate chains, in an archive. This information need not be used for actual reverification, but merely as supporting evidence in case of a dispute.

An embodiment of the present invention is described below, by way of example only, with reference to the accompanying drawings, in which:

Fig. 1 is a block schematic diagram illustrating certification expiration;

Fig. 2 is a block schematic diagram illustrating a "save state" embodiment of the invention;

Fig. 3 is a block schematic diagram illustrating a "save state" "secure storage" embodiment of the invention;

Fig. 4 is a flow diagram that provides two alternative scenarios that illustrate the applicability of time stamps to the preferred embodiments;

Figs. 5a-5c provide block schematic diagrams that illustrate three long term signature verification usage scenarios;

Fig. 6 is a block schematic diagram that illustrates trust between two entities ; and

Fig. 7 is a block schematic diagram that illustrates a long term signature verification trust model.

The meanings of some of the terms used herein may differ somewhat from common usage. The following definitions are meant to clarify the meaning of each in the context of its usage herein.

Archive: Any facility for the storage and retrieval of electronic information.

Certificate: An artifact upon which digital signatures are based. A certificate securely binds an entity with that entity's public key.

Cryptographic Refresh: A means of solving the key degradation problem when storing cryptographic information for long periods of time. The process involves re-encoding the old cryptographic artifacts *(e.g.* encrypted data, digital signatures, and message digests) with stronger algorithms and/or longer keys.

Document: A document can be any information which can be represented electronically or optically *(e.g.* an arbitrary bit stream).

Key Degradation/Algorithm Degradation: The process whereby the protection afforded a document by encryption under a key loses effectiveness over time. For example, due to factors such as improvements in computing speed and breakthroughs in cryptoanalysis, it is expected that a document securely encrypted today would be crackable years later. This property could affect any cryptographic information, including digital signatures. This problem can be generalized to keyed and non-keyed cryptographic processes and artifacts, such as one-way hash algorithms. The security provided by these are also expected to diminish over time.

LTSV: Long Term Signature Verification. The herein described method and apparatus for verifying a digital signature after the certificates used for such verification have expired.

LTSV client: The entity which requests/utilizes the services of the LTSV server.

LTSV server: The entity which delivers the LTSV services. This does not imply, however, that this entity must be stand-alone component.

LTSV submission: A request from an LTSV client to an LTSV server to perform the necessary functions required to enable reverification of a digital signature some time in the future *(e.g.* save PKI state).

PKI: Public Key Infrastructure. Refers to all components, protocols, algorithms, and interfaces required to deliver the capabilities to digitally sign and verify documents. For purposes of clarity herein, a PKI does not include a service module for long term signature verification (LTSV server), although in practice a PKI might be designed to encompass such a module.

Signature Reverification: The re-creation of the digital signature verification process after the original verification. This specifically refers to the process associated with the verification process, based upon the restoration of the previously saved PKI state.

Signature Verification: The process by which a digital signature, for a given document, is determined to be authentic or not.

Signature Verification Module: The module which is responsible for performing the verification of digital signatures.

Time stamp: A digital time stamp is an electronic indicator which associates the current date and time with a particular document. Time stamps are useful for proving that a document existed at a particular time. It is desirable that time stamps be secure, durable over time, and trusted by those using them.

The discussion herein assumes an understanding of public key, digital signatures, and PKI infrastructure using X.509 certificates. Practical information concerning application of such techniques is considered to be well known to those skilled in the art. Background information may be found, for example, in B. Schnier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, John Wiley & Sons, Inc. (1996); W. Ford, M. Baum, Secure Electronic Commerce, Prentice Hall PTR (1997); and in the X.509 v.3 specification ([X.509-AM] ISO/IEC JTC1/SC 21, Draft Amendments DAM 4 to ISO/IEC 9594-2, DAM 2 to ISO/IEC 9594-6, DAM 1 to ISO/IEC 9594-7, and DAM 1 to ISO/IEC 9594-8 on Certificate Extensions, 1 December 1996). The system described herein may be built upon the X.509 infrastructure.

The following discussion provides some background on cryptographic techniques. Cryptographic algorithms can generally be divided into two categories: public key *(e.g.* RSA) and secret key *(e.g.* DES). Both types of algorithms transform plain text into cypher text using a key(s) for the encryption and decryption processes.

Both public key and secret key algorithms are considered to be secure. One is not better than another in terms of security. The strength of each algorithm, in terms of it being cracked, is largely a function of the length of the key used. The primary distinguishing characteristic of public key, however, is that it uses two keys (one to encrypt and another to decrypt), while secret key algorithms use only one key (the same key is used for encryption and decryption). For this reason, secret key algorithms are sometime referred to as symmetric algorithms and public key algorithms are called asymmetric.

One problem with secret key algorithms is that a key must be distributed between all participants. This means that some secure channel must be available for the distribution of the keys.

In practice, each entity in a public key-based system has a key pair, *i.e.* one private key and one public key. The private key is known only to its owner, the public key is known to all correspondents. It is computationally infeasible to determine a private key from the public key.

The two primary services provided by public key cryptography are secure exchange of symmetric keys (by using public key techniques to encrypt a symmetric session key), and non-repudiation via digital signatures.

Public key cryptography can be used to solve the key exchange problem associated with secret key algorithms by using this technology to encrypt the secret key under the public key of the recipient. It can then be decrypted by the recipient using his/her private key.

Digital signatures are possible by encrypting data with the private key of the signing entity. Any entity can decrypt it with the signer's publicly available public key and know that no one else could have encrypted it because that private key is only known by that one individual. This particular use of public key provides the non-repudiation service, which is a primary use of public key cryptography. A digital signature is very powerful notion, it generally exhibits the following characteristics:

- Cannot be forged;
- Is independently verifiable;
- Is not reusable or transferable to a different piece of data; and
- Includes data integrity checks, allowing tamper-detection.

The new services provided by public key cryptography do not come for free, however, because these services require the existence of a supporting public key infrastructure. The strength of a public key system depends upon the assurance that all participants know the public key of any entity with whom they wish to correspond. If a secure correspondence between a user and his/her public key cannot be maintained, then it may be possible to impersonate another entity or read encrypted data intended for another.

The standard solution to this problem is the issuance of a digital certificate (X.509 certificate) to each participant. This certificate securely binds its owner's name with his/her public key. It is issued by a trusted third party, called a certification authority (CA), and is signed by that CA, thereby making it tamper proof. Certificates are issued for a limited period of time (start and stop dates), during which the certificate is considered valid. A certificate is considered expired after the ending validity date.

The public keys of entities (which are embedded in the X.509 certificates) must be publicly available. The distribution or access mechanisms available are numerous.

The secure operation of a public key infrastructure rests upon certain points of trust. Certainly each entity must trust its own CA. However, when a given PKI domain is expanded to encompass relationships with multiple CAs, the number of points of trust are also expanded. The trust placed in a particular end entity *(i.e.* that entity's certificate or signature) is directly related to the trust relationships among the CAs which certify those entities.

CAs can create trust relationships with other CAs by certifying each other. This can be a unidirectional trust relationship, whereby one CA can merely issues a certificate to another CA, just as a CA issues a certificate to an end user. Two CAs can also mutually agree to trust each other (bidirectional trust relationship) by issuing a cross-certificate -- a special form of certificate which contain two individual certificates, one for each direction.

If two entities are in the same CA domain, then there is no concem with respect to CA trust because they both trust the same CA. This is not the case, however, when dealing with the scenario where entities which have been certified by different CAs attempt to conduct a secure transaction. The security of this transaction depends upon the trust between the CAs. More generally, the security provided by the PKI depends upon the trust models embodied in the trust relationships among the various CAs which choose to trust one another. In concrete terms, any change in these trust relationships can cause a signature verification to either succeed or fail.

The preferred method and apparatus effectively extend the time over which a digital signature can be verified, *i.e.* well beyond the expiration of any or all of the certificates upon which that signature depends. They can be used for any application domain where users want digital signatures to be used on long lasting documents *(e.g.* contracts), and be independently verifiable years or decades after signing the document. The preferred embodiment of the invention provides two alternative approaches to constructing a solution which delivers long term signature verification (LTSV).

Fig. 2 is a block schematic diagram illustrating a "save state" embodiment of the invention. This embodiment, largely entails the use of cryptographic information and techniques. Thus, an archive facility 20 is used to store a public key infrastructure (PKI) state 24, *e.g.* cryptographic information, such as certificates and CRLS, in addition to the source document itself. For example, the LTSV client 25 requests the services of an LTSV server 26 to accomplish storage of such information. This step is shown as the "save state" step in Fig. 2. The PKI state information may contain either or both of cryptographically protected information, such as certificates and CRLs, and information that is not cryptographically protected, such as the public key of a root certification authority or policy information.

This information comprises all that is necessary to re-create the signature verification process at a later time, *i.e.* during the "restore state" step, for example, as requested by the LTSV client. It may also be desirable to store the source document separately from the cryptographic information (such as the signature, certificates, and CRLs) for reasons of privacy. For example, a user may want to have control over the source document.

When a user wants to reverify the signature on a document, possibly years later, the LTSV server 26 recreates the precise state of the PKI at the time the document was originally submitted. The LTSV server restores the state, and the signature verification process 28 executes the exact process it performed (or would have performed) years earlier. The time used as the basis for re-creation of the signature verification process does not have to be the time of submittal. Rather, the time could be some other relevant time, such as when a document was signed by the originator or when it was verified by a recipient.

Fig. 3 is a block schematic diagram illustrating a "save state" "secure storage" embodiment of the invention. This embodiment of the invention combines the strength of cryptography with the proven resilience of (non-public key) technology and procedures currently associated with secure data stores. An example of this embodiment:

- Saves the PKI state for future reverification (as described above in connection with Fig. 2); and
- Protects the PKI state information from intrusion by maintaining it in a secure storage facility which is protected by services, such as firewalls, access control mechanisms, audit facilities, intrusion detection facilities, physical isolation, and network isolation; and/or employing a cryptographic protection mechanism, for example using the LTSV server to sign the PKI state information or using a keyed hash algorithm.

In addition, other non-cryptographic features may be added to such approach to deliver a highly secure and trusted LTSV solution, including, for example utilities 30 for viewing the PKI state information (cryptographic as well as non-cryptographic) and visually monitoring the security of the system. These utilities can be used to provide visual evidence for purposes of dispute resolution.

One enhancement to the secure storage approach herein disclosed maintains certain evidence, such as certificate chains, in an archive. This information need not be used for actual reverification, but merely as supporting evidence in case of a dispute. See A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, pp. 583 (1996), for one manner in which this enhancement may be implemented in the context of a notary service (discussed above).

There are other embodiments of the invention in which a hybrid LTSV solution could be constructed by combining cryptographic and non-cryptographic techniques. The best combination for a particular application domain depends upon the security requirements of the application(s), in combination with cost constraints.

It is presently preferred to employ the second embodiment of the invention (discussed above) due to the cryptographic strength associated with its ability to recreate the complete digital signature verification process, combined with the trust instilled by more conventional techniques used for providing secure storage, and in conjunction with audit and viewing facilities with which to view evidence and monitor the secure storage controls. In practice, the most useful embodiment of the invention for a particular application may be that which is the least expensive and which still meets the user or application requirements.

Several issues related to the design of a system which implements LTSV are described below. Alternatives for the resolution of the issues are presented, as well as a discussion of the advantages and disadvantages associated with each alternative. The best approach to any given solution depends upon the security requirements of the application(s) using the LTSV services, as well as the cost constraints. There is no best solution for all applications.

When to Save the PKI State

Signature reverification is preferably associated with a particular time because the outcome of this process could change, depending upon the state of the PKI *(e.g.* because of certificate revocations or the creation/removal of cross certificates). There are numerous possibilities with regard to when the PKI state should be saved, including:

- At signature creation time. This approach is used when an individual wants to document the validity of his/her signature at the time it was created. This is the most accurate time to store the PKI state because it reflects the state at the time of signing, which is presumably the critical time in evaluating the authenticity of that signature. Changes to the PKI state occur after that time, some of which could impact the outcome of a signature reverification. Therefore, saving of the PKI state at any time after signing introduces the possibility of inconsistencies between the signer's and recipient's perspectives on a signature's validity.
- At signature verification time. This approach is useful when a recipient wants to document the validity of a signed document received from another individual.
- At archival time. When a user decides that a document should be archived for long term storage is also an appropriate time to save the PKI state.
- When explicitly requested. There may occur certain application specific document life cycle milestones, at which time the user may desire the PKI state to be saved for future reverification.
- Just before changes in PKI state *(e.g.* certificate revocation). This approach requires a tight integration with the PKI because changes in the PKI must be monitored.

The correct time at which to save the PKI state is preferably determined by the constraints and needs of the application using the LTSV services. A robust LTSV solution is able to accommodate the needs of all (or most) applications in this respect.

Contents of the PKI State.

The exact composition of the PKI state varies somewhat from one PKI vendor's product to another's, depending upon the implementation chosen by each vendor. Moreover, certain information is stored in a different format from one vendor to another. In addition, the contents of a PKI state may change over time as well, as new capabilities (and supporting data) are added to the system. Finally, the required contents of the PKI state may change from one application to another, depending upon the needs *(e.g.* level of security and legal requirements) of each application.

Notwithstanding these uncertainties, there are classes of PKI state information which are candidates for saving. These classes include:

- Certificate chain (list of certificates from one entity to another, including certification authorities (CAs) and the end entities).
- CRLs (one for each CA in certificate chain).

  CA policy statements or identifiers.
- Attribute certificates.
- Date and time.
- Trust information (e.g., public key(s) or certificate(s) of trusted root CA(s), policy constraints).

  Policy constraints are, for example, non-cryptographic information stored within the LTSV archive. The public key of the trusted root CA may or may not be cryptographically protected. If it is embedded in a certificate, then it is signed by the CA. However, it could just as well be an isolated public key, in which case it is unprotected by cryptography.

It is possible that the items in the above list may not be supported or available from certain PKI implementations. Further, the PKI state from another implementation might include some additional data. Therefore, the list above is only an example of what might be considered important pieces of PKI state information, given the current state of the technology. An implementation of an LTSV service is preferably tied to the implementation of a specific PKI until such time as the technology evolves and comprehensive standards emerge.

How to Store the PKI State.

Storage of the PKI state is preferably accomplished in either of two general ways:

- Store all of the PKI state relevant to each document separately; and
- Store the PKI state centrally, and only store references to the PKI state information with each document. This approach enables storage efficiencies by eliminating the redundant storage of PKI state information over multiple documents. For example, given two documents submitted to the LTSV server at about the same time, it is possible that the CRLs contained in the PKI state are exactly the same for both submissions. Central storage of this information allows the LTSV server to store this information only once.

The storage requirements for the save state solution for LTSV may be quite large, depending upon the size of the certificates, the length of the certificate chains and -- more importantly -- the size of the CRLs. The choice of storage technique may have a great impact on the total data storage requirements. It is clearly undesirable to store massive CRLs with every document that is stored for long term archival and possible future reverification. For this reason, the second alternative listed above is presently considered to be the preferred approach.

However, this second approach may present certain difficulties in applications where the LTSV server is an entirely separate component from the PKI, and where support of multiple PKIs is a primary design goal of the LTSV server. In this case, it would be advantageous for the PKI state to remain opaque to the LTSV server, thereby providing ease of support of multiple PKI vendors. Given that what constitutes the PKI state for one vendor may be different for another vendor, it is desirable to maintain an opaque interface between the LTSV server and the PKI. On the other hand, storage efficiencies can be derived only if the LTSV server is informed about the contents and format of the PKI state information. These conflicting requirements -- acceptable storage size and opaqueness -- pose a challenge for the design of an LTSV service.

Some of the possible alternatives are listed below:

- Keep the interface opaque and store the PKI state as it currently exists (full certificate chains and CRLs). This option focuses entirely on the opaqueness requirement, and sacrifices the data size requirement. The primary advantage of this solution is simplicity and quick deployment.
- Remove the opaqueness requirement by making the PKI state visible to the LTSV server. This allows the LTSV server to manage the certificates and CRLs manually -- thereby avoiding duplication of these objects in the data store. This solution potentially sacrifices the ease of multi-vendor support at the expense of achieving efficient storage.
- Compromise by making the CRLs visible to the LTSV server, where other PKI state information is opaque. This solution is interesting because it is probable that the CRLs are the largest piece of data comprising the PKI state. Because CRLs are standard across nearly all PKIs, the visibility should not pose a problem in terms of multi-vendor support. This solution address both of the requirements, but does put the burden of management of the CRLs onto the LTSV server.
- An alternative embodiment of the invention provides a variation on the solution above that breaks up the PKI state into multiple pieces, each of which is opaque. The PKI indicates which of these objects is common across multiple signed documents *(e.g.* CRLs and certificates). The PKI labels these objects with unique handles (identifiers), thereby allowing the LTSV server to store these objects and retrieve them efficiently when needed for signature reverification -- all the while maintaining the opaqueness of these objects.
- Encourage PKI vendors to make concise cryptographically protected assertions about the state of revocation, as an alternative to using CRLs. (For example, CRLs indicate who has been revoked. It would be more efficient if the PKI could make a statement that a certificate has not been revoked at a given point in time. This could eliminate the need for storing CRLs.) This approach is non-standard, but acceptable because these PKI-generated assertions are not seen by any application outside the PKI. A major benefit of this approach is that the opaqueness of the state is preserved while some of

the storage inefficiencies of the state information are removed.

For cases where the LTSV server is dedicated to a particular PKI, it is preferred to create a close integration between the two components, thereby allowing the LTSV server to know about the content and format of the PKI state information, and store it in the most efficient manner possible. For cases where the LTSV server must be insulated from the PKI *(e.g.* for portability across multiple PKIs), one of the options listed above (with the possible exception of the first two) may be used.

Location of Source Data.

The source data associated with an LTSV submission can be stored either by the client or by the LTSV server itself. Some LTSV clients do not choose to submit clear text to the LTSV server for storage because of concerns over privacy. (Privacy of the channel between the LTSV client and the LTSV server can be achieved by having the client encrypt the submission under the public key of the LTSV server.) A submission to the LTSV may be encrypted, such that the LTSV is not able to decrypt it. That is acceptable with the LTSV server. However, the client must determine how to decrypt the submission.

Given that the LTSV server views the source data as a bit stream, it is possible that the message could be encrypted by the LTSV client before submission. (The fact that a general purpose LTSV server treats the source document as a bit stream does not preclude the possibility of implementing an application specific LTSV server that is aware of the contents of the submitted data.) The LTSV server treats the encrypted data as the source. Such prior encoding may be sufficient for some applications' needs for privacy. In this case, however, either the client must maintain the decryption key, or the key must be divulged and stored by the LTSV server (which is probably not acceptable).

Alternatively, the LTSV client may submit a message digest (resulting from a one-way hash function) as the source document. The client, in this case, is responsible for maintaining the real source document. If the source document is stored by the client, then only the PKI state information is stored in the LTSV server's archive (along with some reference to the source document or the submitter).

Whether the source data is stored by the client or the LTSV server, it must be produced if and when a reverification of that document is required. It is a required component of any signature verification process.

Key and Algorithm Degradation.

If cryptographically encoded information *(e.g.* digital signatures or encrypted data) is stored for a significant period of time, the issue of key and algorithm degradation must be addressed, *i.e.* the probable loss in effectiveness of a cryptographic key or algorithm over time. Although signed documents are expected to be sealed securely with strong cryptographic algorithms, the strength of an algorithm and associated key length decreases over time with the advent of faster computers and new developments in cryptoanalysis. It is expected that cryptographic algorithms and key lengths have limited life spans. It is generally acknowledged that they should be examined, modified, and/or replaced at periodic intervals. This legitimate security concern increases with the length of time for which a document is valid, and it becomes a very serious threat as the time span approaches multiple decades.

For example, a digital signature performed today, using RSA and a 512-bit key, is considered very strong *(i.e.* it would take years to forge it). But, it is also expected that this same signature may be easily forgeable within ten years or so. This is because of the increased ability to search the key space faster (and thereby find the key used to sign the message) with newer computers or computing techniques. Similarly, there may continue to be developments in techniques for factoring large prime numbers (the difficulty of which is the basis for the strength of the RSA algorithm). It is reasonable for both of these abilities to improve over time (although the pace of these changes is less certain).

It is, therefore, prudent to protect cryptographically encoded documents from this threat when those documents must live beyond a few years. This is the case with the documents expected to be submitted to the LTSV server, and especially so when using the save state approach herein disclosed. Hence, the LTSV facility should address this problem. Not only must the signed documents stored in the archive be protected from this threat, but all other cryptographic data or metadata stored in the archive should be protected. (The cryptographic data primarily include keyed information. That is, any information that is signed or encrypted with a private key. Such information may also include non-keyed cryptographic data, such as the output from a hash algorithm, such as MD5.) This data could also include such items as certificates and CRLs, which are, themselves, digitally signed by the issuing CA.

There are any number of ways that the LTSV facility addresses this problem. For example:

- Periodically countersign all data in need of cryptographic refresh through the use of nested signatures. Under this approach, the LTSV server effectively refreshes the cryptographic strength of the data by signing it with successively longer keys (or stronger algorithms) every few years. Each counter signature has the effect of locking in the cryptographic strength of the enclosed signature(s), thereby extending the cryptographic life of the enclosed document. This countersignature is preferably performed by the LTSV server using a key owned by that server. Performance shortcuts may be required to avoid the costly unraveling of signatures at reverification time, or the potentially time consuming task of countersigning every document in the archive. Such shortcuts include, for example, removing a previous countersignature before applying a new one, or countersigning the entire archive or portions thereof instead of each individual document.

- A modification of the cryptographic approach suggested above provides for countersigning the information in the archive once, but with an extremely long key, *i.e.* a key which is expected to be unbreakable for decades or more. This eliminates all need for countersigning. This may be merely a theoretical solution because finding an algorithm and key length which is secure for that long is impossible to predict. Therefore, there is still a need to provide some backup mechanism, just in case the original algorithm were cracked, for example.

- Protect the cryptographic information in the archive by insulating the archive itself, rather than the individual documents contained in the archive, thereby eliminating the need for a cryptographic solution. In this approach, the archive is protected via access controls and other procedural controls. If the archive can be effectively insulated from intrusion and modification, then key degradation is not an issue and cryptographic refresh is not necessary.

- Use a time stamp facility to seal the cryptographic information in time. Such a facility is expected to provide all of the necessary characteristics required for solving the key degradation problem. This time stamp facility could use one of the techniques listed above, or it could even be an independent service (see below for a discussion of time stamping).

Relationship to Time stamping.

A secure and comprehensive LTSV solution preferably includes an association with a time stamping mechanism. For long term verification of digital signatures, it is often necessary to know the time at which particular events occurred *(e.g.* time of signing or verifying a signature) to determine if a document was valid at that specific time. If there were uncertainty concerning when a document was signed, then the later reverification of that document could be compromised because of the uncertainty of when it was signed.

Fig. 4 is a flow diagram that provides two alternative scenarios that illustrate the applicability of time stamps.

In scenario 1:

- Alice signs a document at time T1, and sends it to Bob (140).

- Alice's certificate is revoked at time T2 (142).

- Bob verifies Alice's signature at time T3 (144).

In scenario 2:

- Alice's certificate is revoked at time T1 (150).

- Alice signs a document at time T2, and sends it to Bob (152).

- Bob verifies Alice's signature at time T3 (154).

When Bob performs the verification (at time T3), he does not know when Alice signed the document. This is critical, because if Alice's key (certificate) were revoked before signing the message, then the signature verification by Bob should fail, and Bob should not trust the contents of the message. If, on the other hand, the revocation occurred after the act of signing, then the signature can be presumed to be valid and trustworthy. For simplicity, this example does not consider the complicating issue of CRL latency, *i.e.* the time between the initiation of certificate revocation and the time when this information becomes available on a CRL.

This example demonstrates the need for a secure and trusted time stamp mechanism in the domain of digital signatures. The mere recording of the current date and time when creating a digital signature is not sufficient for most application because the source of that time may not be trusted by the recipient. The impact, however, also applies not only to the short term signature verification process, but also to the long term verification of digital signatures. Given the example above, the LTSV server could save the PKI state (at time T1) associated with scenario 1 or scenario 2 (or both). The outcome of a signature verification on this message years later is greatly affected by the PKI state used for this verification process, as well as the target time for the verification.

The problem highlighted above demonstrates the preference that the LTSV service to be cognizant of time. It should:

- Be able to determine in a secure fashion the time at which a document was originally signed;

- Be able to re-create accurately the PKI state which was active at a target time in the past;

- Be able to determine the current date and time accurately; and

- At a minimum, save the PKI state associated with a particular target time.

These requirements establish the preference for the integration of a time stamp facility with the signing and verification (and reverification) process. When a document is signed, it is also preferably time stamped to document in a secure fashion the precise moment at which that event occurred. The LTSV service should know the time for which the PKI state is to be saved, be sure to save the appropriate state (the state active at the target time), and execute its signature reverification process in the context of the correct time.

Usage Scenarios.

Figs. 5a-5c provide block schematic diagrams that illustrate three long term signature verification usage scenarios.

In scenario 1, a client (EntityA) 50 submits a document to a LTSV facility 52 for long term signature verification. This is a simple case where EntityA is interested in documenting that it possessed some piece of information.

In scenario 2, EntityB 56 receives a document from EntityA 54 and submits that document to the LTSV facility 58. In this case, EntityB wants to document that it received some information from EntityB.

In scenario 3, EntityA 60 sends the same document to EntityB 64 and to the LTSV facility 62. This case represents a carbon copy feature, whereby EntityA can document the information it sent to EntityB by additionally filing it with the LTSV facility.

Each of the scenarios described above raises issues with respect to encryption, private key access, and trust models.

Encryption and Private Key Access.

A document can be encrypted and/or signed. Ideally, the LTSV facility accepts any such document. This raises a problem, however, with respect to how the LTSV module works with respect to the encryption. When encrypting under a public key system, the document is effectively encrypted under the public key of the recipient, thereby guaranteeing that the recipient (the possessor of the corresponding private key) is the only entity which can decrypt the information. (For purposes of this discussion, interaction with symmetric keys and algorithms is ignored.)

When applying this principle to scenario 1, it is clear that if the signed message is also encrypted, then it could be encrypted under the public key of the LTSV module. This allows the LTSV component to unwrap the signed document and preserve it for long term verification. This is a useful feature because it provides confidentiality between EntityA and the LTSV service. This scenario does not preclude the possibility that the source document sent signed and encrypted to the LTSV module could itself be encrypted under a key known only to EntityA. That is, it is not necessary that the LTSV have access to the plain text version of the source document. The LTSV module treats that encrypted document as the source. If EntityA does decide to encrypt the document first under a secret key before submitting the document to the LTSV service, then it is the responsibility of EntityA to maintain possession of that key if and when decryption of that document is required.

In Scenario 2, if the message from EntityA to EntityB is encrypted (under the public key of EntityB) and then forwarded -- unchanged -- to the LTSV service by EntityB, then it is unreadable by the LTSV component because it does not possess the private key required to decipher and unwrap the enclosed signed document. This unwrapping (decipherment) is essential for the LTSV module to do its job.

There exist several alternatives for addressing this problem:

- Allow the LTSV facility to have access to EntityB's private key;
- Do not allow EntityA to send encrypted documents to EntityB; or
- Have EntityB strip off the privacy aspect of the signed and encrypted document received from EntityA. Because EntityB wants to preserve EntityA's signature on the document, and be able to verify it at a later time, this stripping process can not alter the validity of the signature. EntityA can then either send the stripped (*i.e.* plain text) document to the LTSV service, or it can re-encrypt it (still preserving the original signature by EntityA) under the public key of the LTSV module.

The latter approach above is presently the preferred approach. The first approach above raises significant security concerns because it requires distribution of an entity's private key. The second approach above is unacceptably restrictive on the usage of the system.

Trust.

Digital signature verification is always performed between two (and only two) entities. The verification process is based upon (among other things) the trust relationship(s) in place between those two entities -- the originator (signer) and the recipient (verifier).

Fig. 6 is a block schematic diagram that illustrates trust between two entities according to the invention. In this situation, EntityA 70 has been issued a certificate by CA1 72, EntityB 74 has been issues a certificate by CA2 76, and CA's 1 and 2 have been cross certified. (A cross-certificate is a special type of certificate which indicates mutual trust between two CAs.) The resulting trust model sets up a path of trust between EntityA and EntityB, enabling them to verify digitally signed documents from one another successfully. (A trust model is comprised of the trust relationships among PKI entities (CAs and end users), embodied by the certificates and cross-certificates issues among these entities, as well as the underlying policies which enable this trust.) Note that if any of the three paths in this model were not in place, then sufficient trust would be lacking for the successful exchange of digitally signed messages between the two end parties. Signature verification would fail if any entity in this path is not trusted.

This trust path is commonly referred to as the certificate chain because it is composed of the certificates between the two entities. When considering the save state approach to long term signature verification, it is this entire trust path (among other things) which must be archived as part of the PKI state for later signature reverification. Moreover, the trust path stored by the LTSV facility must contain the relevant trust information existing at the time of the request, not at some other time (before or after) where the trust relationships may be different between the entities. For example, a cross certificate between to CAs could either be created or removed at some point in time. This could effect the trust between two entities and affect the outcome of a signature verification.

As discussed above, the time associated with the existing trust model between two entities is extremely important to the LTSV facility, but there are also ramifications with respect to how the LTSV module works -- specifically, what trust information is needed and stored by the LTSV component for later signature verification. This gets complicated when the LTSV component is included, which may or may not be trusted (via some trust path) by some entities.

Consider the three scenarios illustrated in Figs. 5a-5c:

Scenario 1 is fairly straightforward. There are only two entities involved. The trust path stored by the LTSV facility is the path between those two parties (EntityA and LTSV). It is assumed that trust exists between these entities, otherwise EntityA would not submit a request to that service.

Scenario 2, however, raises certain issues. When EntityB sends a request to the LTSV service, what signature does EntityB want to later verify? Most likely, EntityB wants to reverify EntityA's signature at a later time -- it wants the LTSV service to document that the signed document received from EntityA was valid (contained a valid signature) at the time it was received. This raises two general questions:

- Whether the LTSV service is trusted by EntityA. It can be assumed that the communicating parties (EntityA with EntityB, and EntityB with the LTSV) have developed some trust between themselves. But in this case, it is possible that there exists no trust path between EntityA and the LTSV component.
- The trust path that is to be stored by the LTSV facility. There exist three possible trust paths which can be stored by the LTSV, *i.e.* the path between Entities A and B; the path between EntityB and the LTSV component itself; and the path between EntityA and the LTSV component, if it exists.

Fig. 7 is a block schematic diagram that illustrates a long term signature verification trust model. Given scenario 2, where EntityB 84 submits a signed document, received from EntityA 80, to the LTSV component 88, the LTSV can save the trust model embodied in the original signed document (EntityA 80 → CA1 82 → CA2 86 → EntityB 84). Later verification of this signature recreates the verification process originally performed by EntityB when it received this document from EntityA. If, however, the PKI state stored by the LTSV service were to contain only the trust path between the submitter and the service (EntityB 84 -> CA2 86 -> CA3 90 -> LTSV 88), then reverification of the original document, as originally performed, is impossible. In fact, this is exactly the paradigm described in scenario 1, where the trust path between the submitter and the LTSV are of interest.

The above discussion reveals that there are good reasons for the LTSV component to be able to store either trust path, depending upon the requirements of the client.

In scenario 2, the LTSV would most likely store the trust path corresponding to the message from EntityA to EntityB (to reverify the signed document from EntityA to EntityB). In scenario 1, the LTSV would store the trust path corresponding to the submission itself -- from EntityA to the LTSV.

Similarly, scenario 3 represents a case where flexibility in which trust path(s) to store is required. In this case, EntityA's submission to the LTSV facility may be with the intent to either reverify its correspondence with EntityB, or to reverify the submission itself (between EntityA and the LTSV). In fact, both trust paths may be of use to the client. The requirements on the LTSV are determined by the business of the particular application being deployed. For this reason, the interface to the LTSV preferably supports the ability of the client to indicate the needs in terms of trust paths as it impacts the requirements for later reverification.

The disclosures in United States patent application no 08/892,792, from which this application claims priority, and in the abstract accompanying this application are incorporated herein by reference.

Claims

1. A method of enabling long term verification of digital signatures, comprising the steps of:

   submitting a source document or digest thereof to a signature verification entity; and using an archive facility to store a public key infrastructure (PKI) state relative to said document at a selected archival time.

2. A method as in claim 1, comprising the steps of:

   using said archived PKI state to re-create said PKI state relative to said document at a selected time after a certificate associated with said signature has expired;
   wherein the time over which a digital signature associated with said document can be verified is extended beyond expiration of any or all of any certificates upon which that signature depends.

3. A method as in claim 1 or 2 comprising the step of:
   storing said source document separately from any associated cryptographic information.

4. A method as in claim 1, 2 or 3 wherein the selected archival time used as the basis for subsequent re-creation of a signature verification process is the time of said source document submittal;
   is the time when said source document was signed by its originator; or in the time when said source document was verified by a recipient.

5. A method as in any preceding claim, comprising the step of;
   protecting said PKI state information from intrusion by maintaining it in a secure storage facility preferably comprising of at least one of a firewall, access control mechanism, audit facility, intrusion detection facility, physical isolation and network isolation; or protecting non-cryptographic PKI state information from intrusion by protecting it in an archive via any of a signature and keyed hash algorithm.

6. A method as in any preceding claim comprising the step of:
   providing utilities for viewing said PKI state information and for visually monitoring system security.

7. A method as in any preceding claim, wherein classes of PKI state information may include one or more of certificate chain from one entity to another, including certification authorities (CAs) and the end entities; certificate revocation lists (CRLs), one for each CA in certificate chain; certificate practice statements; attribute certificates; policy constraints; trust information; and date and time.

8. A method as in any preceding claim, comprising the step of:
   periodically countersigning all data in need of cryptographic refresh through the use of nested signatures and/or countersigning information in said archive facility once with an extremely long key.

9. A method as in any preceding claim, comprising at least one of the steps of:

   protecting said archive facility itself, rather than individual documents contained in said archive; and
   employing a cryptographic protection mechanism at said signature verification entity.

10. A method as in any preceding claim, comprising the step of:
    using a time stamp facility to seal cryptographic information in time.

11. Apparatus for long term verification of digital signature, comprising:

    a source document; and
    an archive facility for storing a public key infrastructure (PKI) state relative to said document at a selected archival time.

12. Apparatus as in claim 11, comprising:
    either of a signature and a keyed hash system for protecting non-cryptographic PKI state information from undetected modification, wherein said noncryptographic PKI state information is maintained in an archive.

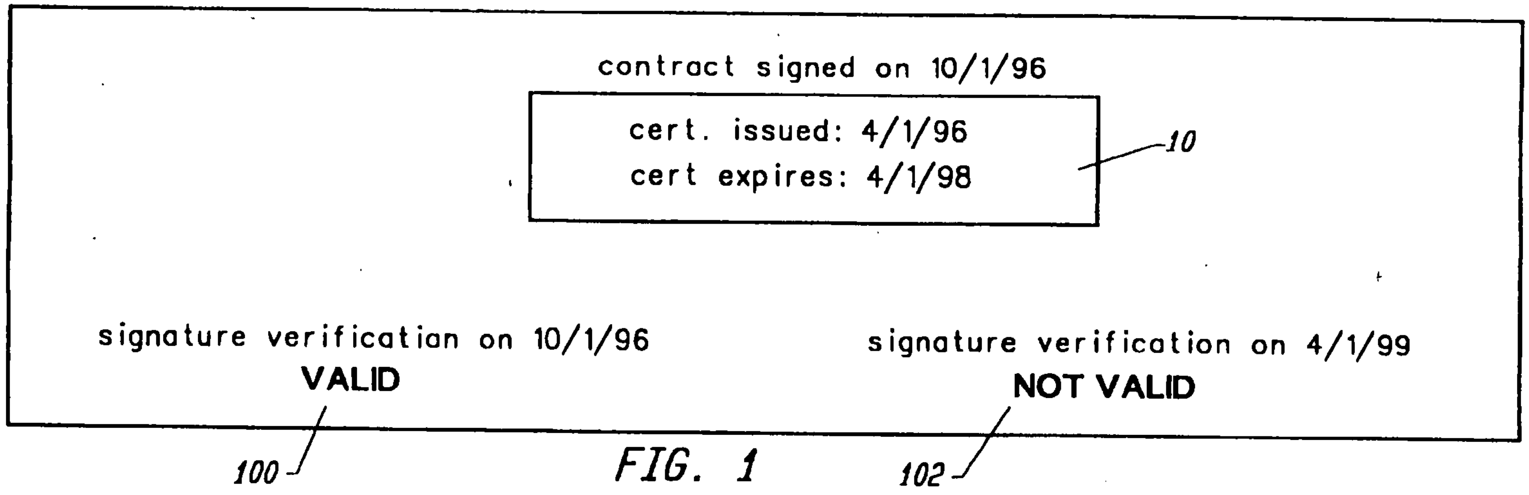

contract signed on 10/1/96
cert. issued: 4/1/96
cert expires: 4/1/98
10
signature verification on 10/1/96
VALID
100
signature verification on 4/1/99
NOT VALID
102

FIG. 1
(PRIOR ART)

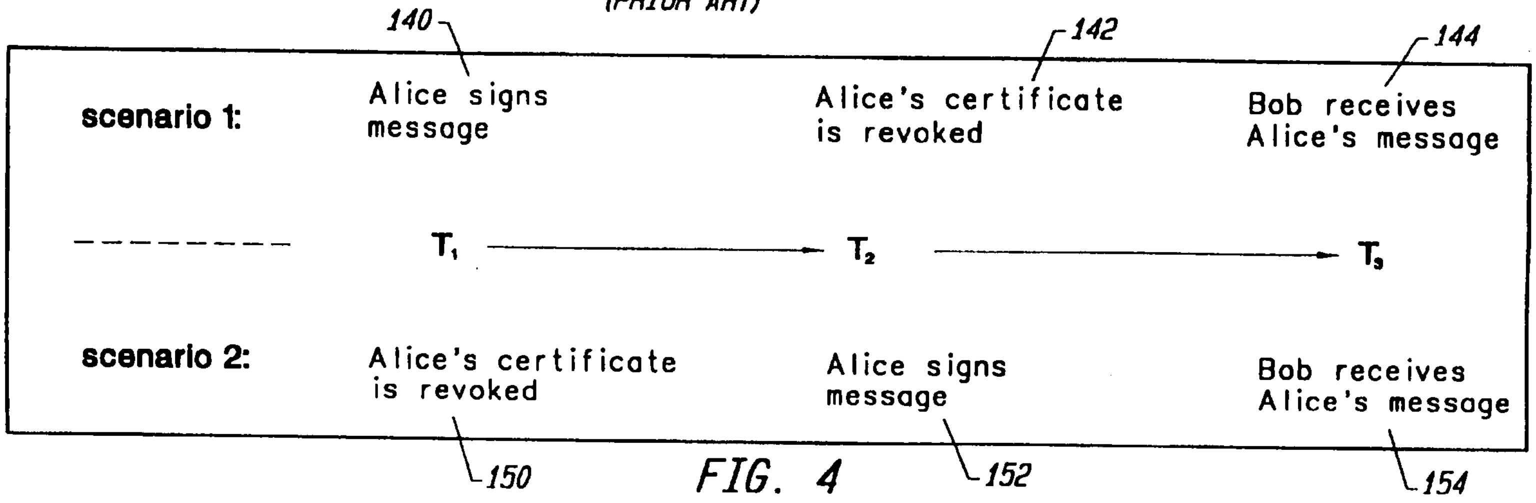

140
142
144
scenario 1:
Alice signs message
Alice's certificate is revoked
Bob receives Alice's message
T_1
T_2
T_3
scenario 2:
Alice's certificate is revoked
Alice signs message
Bob receives Alice's message
150
152
154

FIG. 4

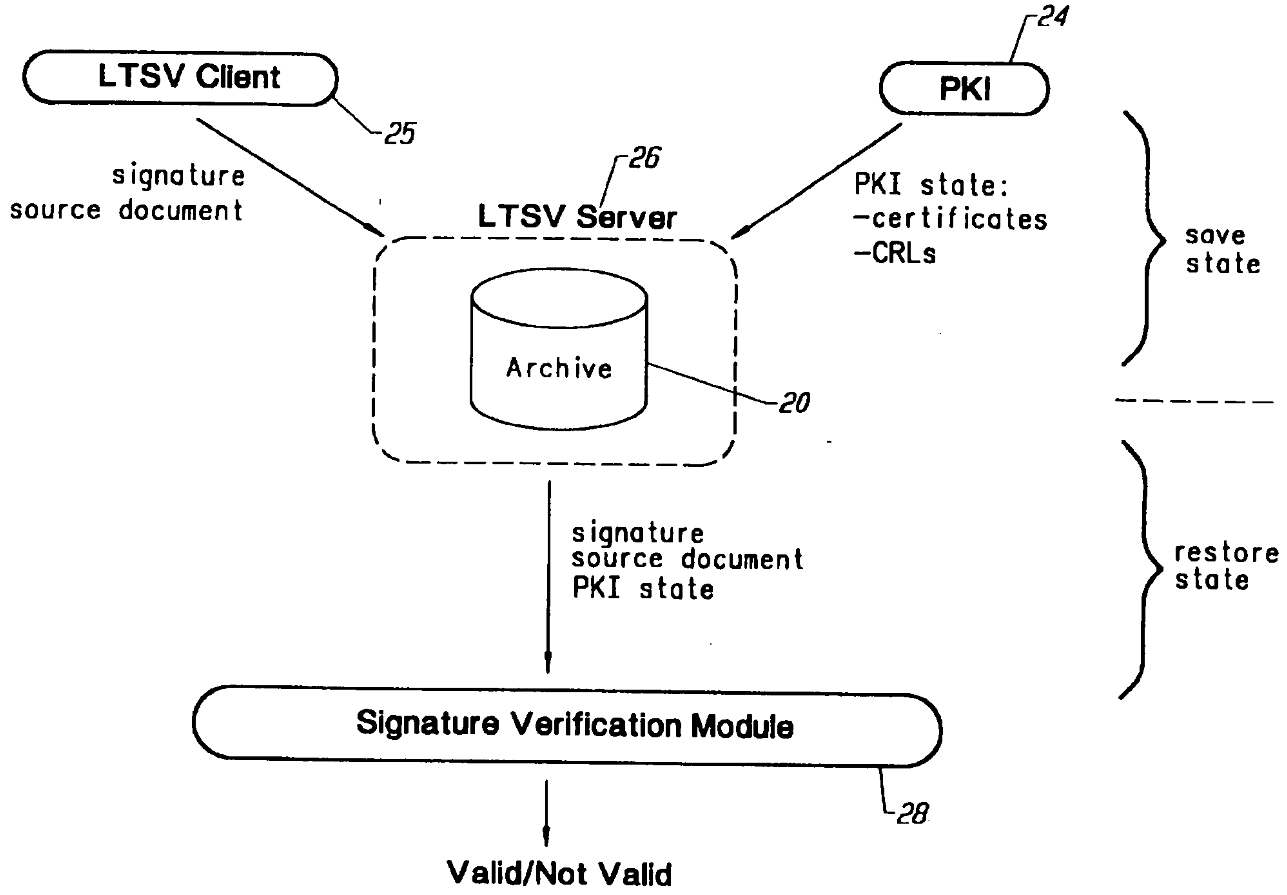

LTSV Client
25
signature
source document
PKI
24
PKI state:
-certificates
-CRLs
26
LTSV Server
Archive
20
save
state
restore
state
signature
source document
PKI state
Signature Verification Module
28
Valid/Not Valid

FIG. 2

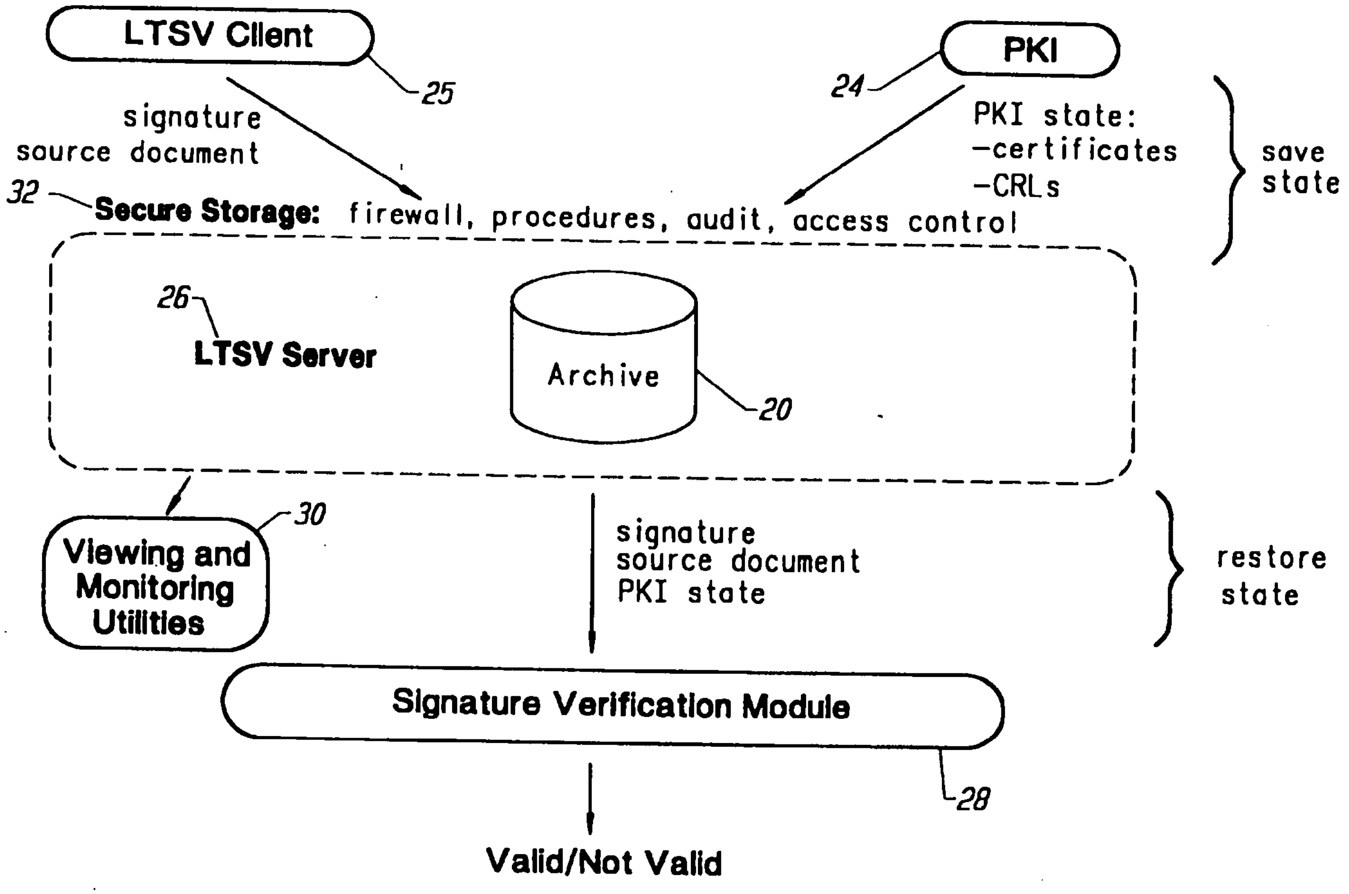

LTSV Client
25
signature
source document
PKI
24
PKI state:
-certificates
-CRLs
save
state
32
Secure Storage: firewall, procedures, audit, access control
26
LTSV Server
Archive
20
Viewing and
Monitoring
Utilities
30
signature
source document
PKI state
restore
state
Signature Verification Module
28
Valid/Not Valid

FIG. 3

SCENARIO 1

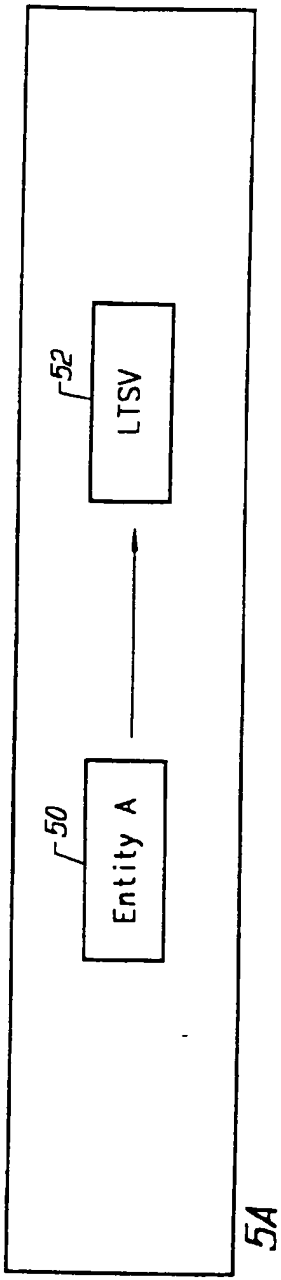

50
Entity A
52
LTSV

FIG. 5A

SCENARIO 2

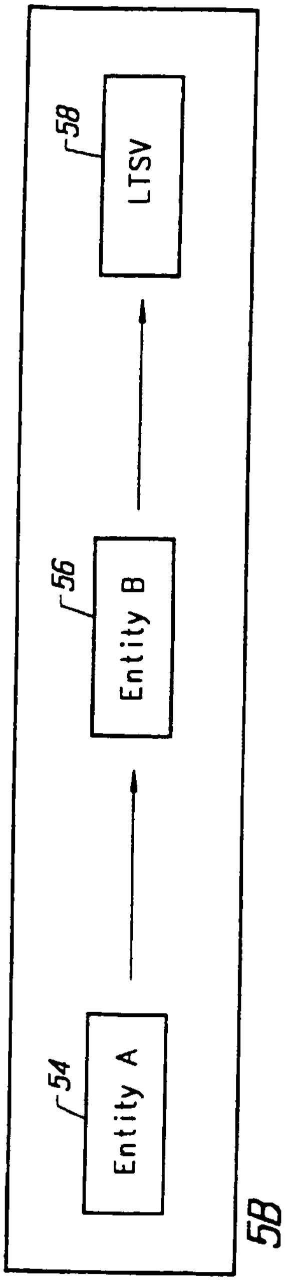

54
Entity A
56
Entity B
58
LTSV

FIG. 5B

SCENARIO 3

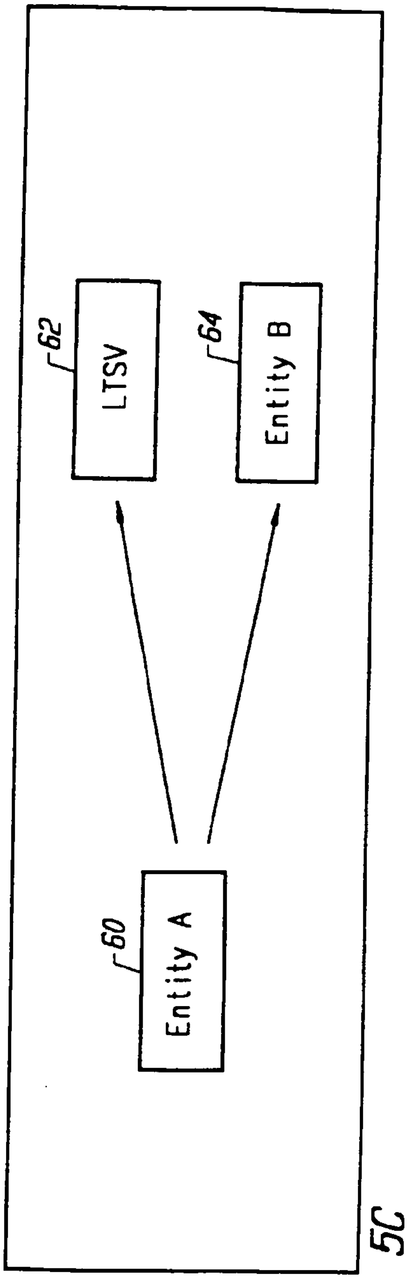

60
Entity A
62
LTSV
64
Entity B

FIG. 5C

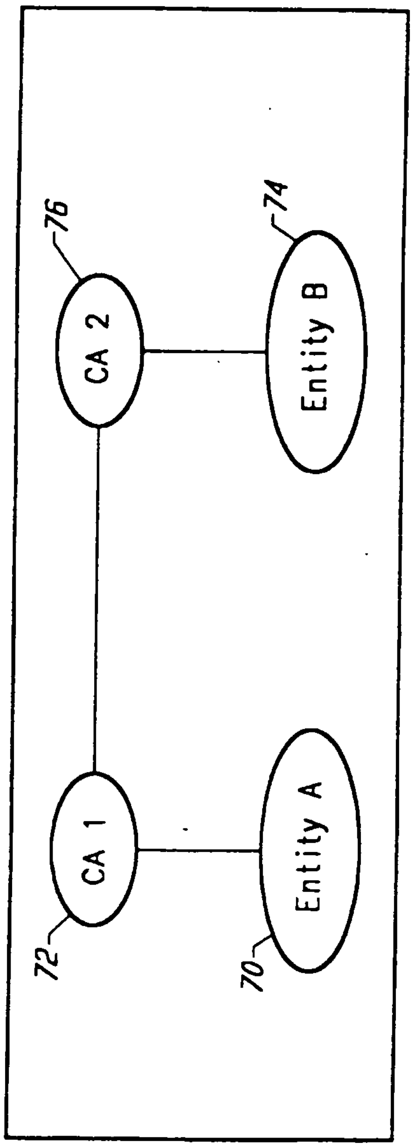

CA 1
CA 2
Entity A
Entity B
70
72
74
76

FIG. 6

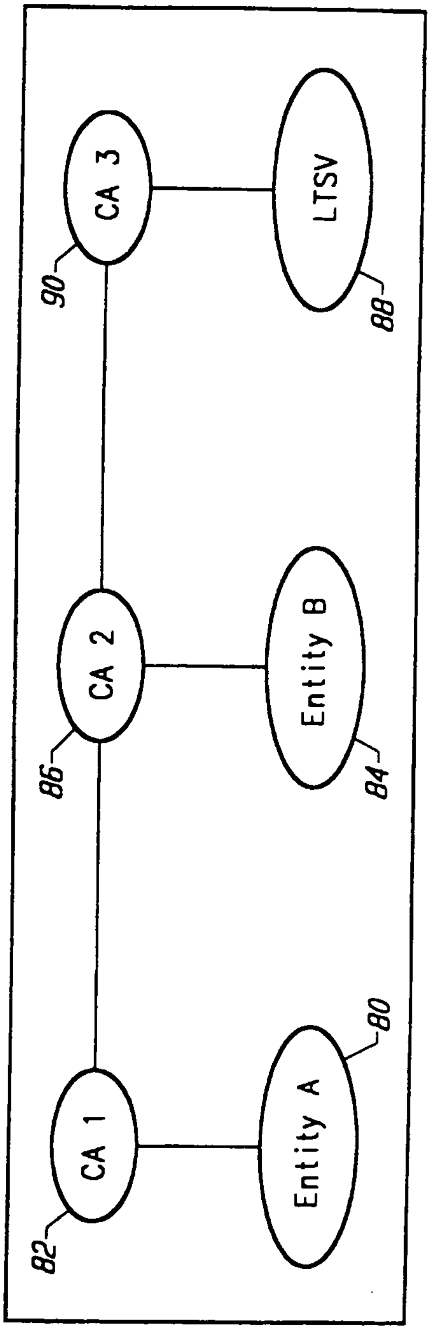

CA 1
CA 2
CA 3
Entity A
Entity B
LTSV
80
82
84
86
88
90

FIG. 7

PATENT SPECIFICATION (11) 1 483 282

1 483 282

(21) Application No. 52131/74 (22) Filed 2 Dec. 1974
(31) Convention Application No. 7342706
(32) Filed 30 Nov. 1973 in
(33) France (FR)
(44) Complete Specification published 17 Aug. 1977
(51) INT CL² G06F 13/00
(52) Index at acceptance
G4A 10EX 13E 13M 17B4 17P 6G 6H 6X AP ND NR

(19)

(54) APPARATUS FOR PROTECTING THE INFORMATION IN AN VIRTUAL MEMORY SYSTEM IN PROGRAMMED DATA PROCESSING APPARATUS

(71) We, COMPAGNIE INTERNATIONALE POUR L'INFORMATIQUE CII-HONEYWELL-BULL (formerly Compagnie Honeywell-Bull), a French Body Corporate, of 94 Avenue Gambetta, Paris 75020, France, do hereby declare the invention, for which we pray that a patent may be granted to us, and the method by which it is to be performed, to be particularly described in and by the following statement:—

The present invention concerns apparatus for protecting the information in a virtual memory system in programmed data processing apparatus.

Several schemes have been utilized in the past in order to protect information. Some of them are detailed by Robert M. Graham in a paper entitled "Protection in an Information Processing Utility", published in CACM (May 1968).

This type of memory protection is inadequate for present day multiprogramming systems because there is no provision for gradations of privilege or gradations of accessability, and severely limits the control over access to information. There should be provisions for different access rights to the different types of information. A partial answer to this problems is found in the concept of a memory having a segment as the unit of information to which access is controlled (see Patent Application No. 21630/74, (Serial No. 1,465,344), filed on 15 May 1974). Varying degrees of access to each segment is possible by providing for different types of privileges attached to each segment such as master/slave, write/no-write and execute/non-execute. However, this method of protecting the privacy and integrity of information does not take into account the user of the information. Under this type of protection, privilege is not accorded the user but the information being protected. Hence a user if he has access at all to a segment has access similar to all other users who have access to the segment. David C. Evans and Jean Yves LeClerc in a paper entitled "Address Mapping and the Control of Access in an Interactive Computer," SJCC 1967, recognized the problem and attempted a solution. Evans and LeClerc said in that article p. 23, "The user of a computing system should be able to interact arbitrarily with the system, his own computing processes, and other users in a controlled manner. He should have access to a large information storage and retrieval system called the file system. The file system should allow access by all users to information in a way which permits selectively controlled privacy and security of information. A user should be able to partition his computation into semi-independent tasks having controlled communication and interaction among tasks. Such capability should reduce the human effort required to construct, debug, and modify programs and should make possible increased reliability of programs. The system should not arbitrarily limit the use of input/output equipment or limit input/output programming by the user". Evans and LeClerc proposed conditioning access rights on the procedure-in-execution. The segment, under their proposal, is still the unit of information to which access is controlled; however, a segment's access control attributes are recorded substantially in a user-name versus procedure tables whose entries are the access modes. Such a solution, however, has serious drawbacks. For one, the construction and updating of each segment's table of access control attributes presents a formidable task. For another, too many uses of the segment and event occurrences must be foreseen. To overcome this problem access control by procedure-set was suggested. Under this suggestion, related procedures are grouped into "sets of procedures" and access rights to segments is based on the identity of the set to which the procedure seeking access belongs. This method alleviated the problem of constructing and updating each segment's voluminous tables of access control attributes, but introduced the problem of determining to which set a given procedure belonged, particularly when a procedure was or could be a number of many sets. This ambiguity in defining sets, and the possible transitions between sets makes the implementation of access control based on "sets of procedures" extremely difficult.

To overcome the difficulties encountered with the "set" technique a ring concept was developed. The ring concept groups the sets of procedures into rings that can unambiguously be ordered by increasing power or level of privilege. By assigning a collection of sets to a collection of concentric rings, and assigning numbers to each ring with the smallest ring having the smallest number and each succeeding larger ring having a progressively greater number, different levels of privilege can then be unambiguously assigned to the user of a segment. Under this concept the innermost ring having the smallest number assigned to it has the greatest privilege. Hence it can be postulated that users in the lowest ring number can access information having higher ring numbers, but users in a higher ring number cannot access information having lower ring numbers or can access information in a lower ring number only in a specified manner. This palpable change of power or level of privilege with a change in rings is a concept which overcomes the objections associated to a change of sets.

Multics (*Multi*plexed *I*nformation and *C*omputing *S*ervice) is an operating system developed primarily by Massachusetts Institute of Technology, in cooperation with General Electric Co. and others which first utilized the ring theory of protection in software on a converted Honeywell 635 (Registered Trade Mark) computer and later on a Honeywell 645 (Registered Trade Mark) computer. The Multics philosophy utilizes 64 rings of protection numbered as rings 0—63 and is set forth generally in a paper entitled "Access Control to the Multics Virtual Memory" published by Honeywell Information Systems Inc. in the Multics Technical Papers, Order No. AG95, Rev. O. A more detailed description of Multics ring protection is to be found on chapter 4 of a book entitled "The Multics System: An Examination of its Structure", by Elliott I. Organick, published by MIT Press, and also in the Multics System Programmers Manual 1969, MIT Project MAC. Briefly, the Multics system does not utilize a "pure ring protection strategy" but rather employs the "ring bracket protection strategy" wherein a user's access rights with respect to a given segment are encoded in an access-mode and a triple of ring number (r1, r2, r3) called the user's "ring brackets" for a given segment. A quotation from pages 137—139 from the Multics Technical Paper entitled, "Access Control to the Multics Virtual Memory" sets out the rules and conditions for using and changing rings.

This "ring protection concept" was first implemented with software techniques utilizing 64 separate rings. Subsequently an attempt was made to define a suitable hardware base for ring protection. The Honeywell 645 (Registered Trade Mark) computer represents a first such attempt. The Honeywell 645 (Registered Trade Mark) system differs from the "ringed hardware" concepts described supra in several respects which when taken together, add up to the fact that the Honeywell 645 (Registered Trade Mark) is a 2-ring rather than a 64-ring machine, and has in lieu of a "ring register", a master mode and a slave mode, which imparts greater power to the processor when in master mode than when in slave mode. "The access control field of the 645's SDW (segment descriptor word) contains no information about rings; in particular its does not contain ring brackets. It does, however, contain either:

a) access-mode information possibly including either of the two descriptors;
accessible in master mode only,
master mode procedure;
b) the specification of one of eight special 'directed' faults (traps) which is to occur whenever the segment descriptor word (SDW) is accessed.

"The procedure is only 'in master mode' when executing a procedure whose SDW indicates a 'master mode procedure'. The processor may enter master mode while executing a slave mode procedure by:
faulting,
taking an interrupt".

"The 645 processor's access control machinery interprets the SDW during the addressing cycle and causes the appropriate action to occur depending on the SDW and (usually) on the attempted access, as follows:

a. If the SDW implies a particular "directed fault", then that fault occurs.

b. Otherwise, if the SDW does not permit the attempted access, the appropriate access violation fault occurs.

c. Otherwise, the SDW permits the attempted access and the access is performed.

"When a fault occurs, the 645 enters master mode and transfers control to the appropriate master mode fault handling procedure". (Access Control to the Multics Virtual Memory, supra pps. 157—158).

Another paper by Michael D. Schroeder and Jerome H. Saltzer entitled "A Hardware Architecture for Implementing Protection Rings" published in Communications of the ACM, March 1972 Vol. 15, No. 3, sets forth background and theory of ring protection and describes a hardware implementation of "ring protection".

Because the Multics and Honeywell 645 version of ring protection was implemented mainly in software, considerable operating system supervisor overhead was entailed particularly when calls to greater or lesser power were made by trapping to a supervisor procedure. What was required was an access control mechanism which had the functional capability to perform effectively its information protection function, was relatively simple in operation, was economic to build, operate and maintain, and did not restrict programming generality. The Honeywell 6000 (Registered Trade Mark) computer system met these requirements by implementing most of the ring protection mechanism in hardware. Hence special access checking logic, integrated with the segmented addressing hardware was provided to validate each virtual memory reference, and also some special instructions for changing the ring of execution. However certain portions of the ring system particularly outward calls and returns or calls to a lesser power and returns therefrom presented problems which required the ring protection function to be performed by transferring control to a supervisor. What is now needed are further improvements in hardware and techniques that will permit a full implementation of ring protection in hardware/firmware and will meet the criteria of functional capability, economy, simplicity and programming generality.

Accordingly the present invention has for an object to provide an improved computer ring protection mechanism.

Accordingly the present invention consists in an internally programmed data processing apparatus CPU having a virtual memory system, and being responsive to internally stored instruction words for processing information and having stored in said virtual memory system a plurality of different types of groups of information each information group-type associated with an address space bounded by a segment having adjustable bounds, and comprising means for protecting the information in said-virtual memory system from unauthorized users by restricting accessability to the information in accordance to levels of privilege, said means comprising in combination with an access checking mechanism:

(a) first means arranged in operation to store in said virtual memory system at least one segment table comprising a plurality of segment descriptors with each segment descriptor being associated with a predetermined one of said segments and each segment descriptor having a predetermined format containing an access information element and a base address element in predetermined positions of said format, said base address element being used for locating in said virtual memory system the starting location of a selected one of said segments, and said access information element for specifying the minimum level of privilege required for a predetermined type of access that is permitted in a selected one of said segments;

(b) a plurality of second means having a predetermined format, communicating with said first means, arranged to store in a predetermined portion of said second means, a segment number SEG for identifying a segment table and the location of a segment descriptor within said segment table, said second means also being arranged to store in a predetermined other portion of said second means, an offset address within the segment identified by said segment descriptor said offset address locating from said segment base the first byte of a word within said segment;

(c) third means responsive to an address syllable element of an instruction being executed for addressing one of said plurality of second means;

(d) fourth means arranged to store a displacement from said address syllable;

(e) fifth means, communicating with said first, second, third and fourth means, arranged to add the displacement D and said base address to said offset; and,

(f) sixth means responsive to said access information element in a selected one of said segment descriptors, restricting the accessability to the segment associated with said selected one of said segment descriptors in accordance to the level of privilege and the type of access specified in said access information element, wherein each group-type of information is associated with a predetermined ring number indicative of a level of privilege said legel of privilege decreasing as the associated ring number increases comprising means for determining the maximum effective address ring number EAR (i.e. minimum level of privilege) of a selected process to access a selected group of information, said means comprising:

(a) first means to store first information indicating the maximum ring number RD (i.e. minimum level of privilege) required to read information from said selected group;

(b) second means to store second information indicating the maximum ring number WR (i.e. minimum level of privilege) required to write information into said selected group;

(c) third means to store third information indicating the maximum ring number MAXR (i.e. minimum level of privilege) required to process information from said selected group; and,

(d) fourth means communicating with said first, second and third means, to determine the maximum of the contents of said first, second and third means whereby the effective address ring number EAR is generated.

The present invention, however, both as to organization and operation thereof may best be understood by reference to the following description which is given by way of example in conjunction with the accompanying drawings in which:

Figure 1 is a block diagram of a computer system utilizing the invention.

Figure 2 is a schematic diagram illustrating the levels of privilege of the invention.

Figure 3 is a flow diagram of the segmented address scheme utilized by the invention.

Figures 4A—4J are schematic diagrams of various novel hardware structures utilized in the invention.

Figure 5 is a schematic diagram of the computer ring protection hardware.

Figure 6 is a schematic diagram of the computer segmented addressing hardware.

Figures 7a—7h and Figures 8a—8c are detailed logic block diagrams of the ring protection hardware.

Figures 9a—9k is a legend of the symbols utilized in the diagrams of the invention.

Figure 10 is a schematic diagram of three stack segments, one each for ring 0, 1 and 3 respectively.

Figure 11A shows the format of the Enter Procedure instruction.

Figure 11B shows the format of a procedure descriptor.

Figure 11C shows the format of a gating procedure descriptor GPD the first word of the segment containing the procedure descriptors.

Figure 11D shows the format of the Exit Procedure instruction.

Figure 12 is a flow diagram of a portion of the Enter Instruction pertaining to ring crossing and ring checking.

Figure 13 schematically shows a segment descriptor and the segment containing procedure descriptors.

Figures 14—16 are flow diagrams showing various operations that are performed when the Enter Procedure instruction is executed.

Figure 17 is a flow chart of the Exit Instruction.

As previously discussed the ring concept of information protection was originated on MULTICS and implemented on various Honeywell (Registered Trade Mark) Computer Systems. The original MULTICS concept required 64 rings or level of privilege and later implementation had the equivalent of two rings on the Honeywell 645 and 8 rings on the Honeywell 6000 (Registered Trade Mark). The embodiment described herein groups data and procedure segments in the system into a hierarchy of 4 rings or classes. (Refer to Figure 2). The 4 rings or privilege levels are identified by integers 0—3; each ring represents a level of privilege in the system with level 0 having the most privilege and level 3 the least. Level 0 is known as the inner ring and level 3 as the outer ring. The basic notion as previously discussed is that a procedure belonging to an inner ring has free access to data in an outer ring. Conversely a procedure in an outer ring cannot access data in an inner ring without incurring a protection violation exception. Transfer of control among procedures is monitored by a protection mechanism such that a procedure execution in an outer ring cannot directly branch to a procedure in an inner ring. This type of control transfer is possible only by execution of a special "procedure-call" instruction. This instruction is protected against misuse in a number of ways. First, a gating mechanism is avilable to ensure that procedures are entered only at planned entry points called gates when crossing rings. The segment descriptor of such a procedure contains a gate bit indicating that procedures in this segment can be entered only via gates; information regarding these gates is contained at the beginning of the segment and is used by the hardware to cause entry at a legal entry-point. The procedure itself must then verify (in a way which, of necessity depends on the function of the procedure) that it is being legitimately called. A further hardward protection mechanism is available in the case that the calling procedure supplies an address as a parameter; it is then possible that the more privileged procedure would invalidly modify information at this address which the less privileged caller could not have done, since the ring mechanism would have denied him access; an address validation instruction is available to avoid this possibility.

An important convention is required

here in order to protect the procedure call mechanism. This states that it is not in general permissible to use this mechanism to call a procedure in a less privileged ring and return to the more privileged one. This restriction is necessary since there is no assurance that the procedure in the higher ring will, in fact, return; that it will not, accidentally or maliciously, destroy information that the more privileged procedure is relying upon; or that it will not, accidentally or maliciously, violate the security of the stack (see GLOSSARY for definition). Any of these could lead to unpredictable results and crash the system.

The level of privilege are quite independent of the process control mechanism and there is no notion here of privileged and non-privileged processes as in the IBM system 360 (Registered Trade Mark). Instead the same process can execute procedures at different levels of privilege (rings) subject to the restrictions imposed by the ring mechanism. In this sense the ring mechanism can be viewed as a method for subdividing the total address space assigned to a process according to level of privilege.

The ring mechanism defined herein permits the same segment to belong to up to 3 different rings at the same time i.e. there are 3 ring numbers in each segment descriptor, one for each type of possible access. Thus the same segment can be in ring one with respect to "write" access, ring two with respect to "execute" access and ring three with respect to "read" access. One obvious use for this is in the case of a procedure segment which can be written only by ring zero (perhaps the loader) but can be executed in ring three.

Of the four available rings, two are allocated to the operating system and two to users. Ring zero, the most privileged ring, is restricted to those operating system segments which are critical to the operation of the whole system. These segments form the hard core whose correctness at all times is vital to avoid disaster. Included would be the system information base, those procedures dealing with the organisation of physical memory or the initiation of physical data transfer operations, and the mechanisms which make the system function, like the "exception supervisor, the scheduler, and the resource management".

Ring one contains a much greater volume of operating system segments whose failure would not lead to catastrophe but would allow recovery. Included herein are the language translators, data and message management, and job and process management. Through the availability of two rings for the operating system, the problem of maintaining system integrity is made more tractable, since the smaller hard core which is critical is isolated and can be most carefully protected.

Rings two and three are available to the user to assign according to his requirement. Two important possibilities are debugging and proprietary packages. Programs being debugged may be assigned to ring two while checked out programs and data with which they work may be in ring two; in this way the effect of errors may be localized. Proprietary programs may be protected from their users by being placed in ring two while the latter occupy ring three. In these and other ways, these two rings may be flexibly used in applications.

The General Rules of the Ring System

1. A procedure in an inner ring such as ring 2 on Figure 2 has free access to data in an outer ring such as ring 3 and a legal access (arrow 201) results. Conversely a procedure in an outer ring such as ring 3 cannot access data in an inner ring such as ring 2 and an attempt to do so results in an illegal access (arrow 202).

2. A procedure in an outer ring such as ring 3 can branch to an inner ring such as ring 1 via gate 204 which results in a legal branch 203, but a procedure operating in an inner ring such as ring 2 may not branch to an outer ring such as ring 3.

3. Each segment containing data is assigned 2 ring values, one for read (RD) and one for write (WR). These ring values specify the maximum ring value in which a procedure may execute when accessing the data in either the read or write mode.

Each time a procedure instruction is executed, the procedure's ring number (effective address ring, EAR) is checked against the ring numbers assigned to the segment containing the referenced data. The EAR is the maximum number of process ring numbers in the processor instruction counter (see later description) and all ring numbers in base registers and data descriptors found in the addressing path. Access to the data is granted or denied based on a comparison of the ring numbers. For example, if a system table exists in a segment having a maximum read/ring value of 3 and a maximum write/ring value of 1, then a user procedure executing in ring 3 may read the table but may not update the table by writing therein.

Procedure Calls and the Stack Mechanism:

The procedure call and stack mechanism is an apparatus being described herein Procedure calls are used to pass from one procedure to another; to allow user procedures to employ operating system services; and to achieve a modular structure within the operating system. A procedure call is effected by instructions and a hardware recognized entity called a stack.

A stack is a mechanism that accepts, stores and allows retrieval of data on a last-in-first-out basis. Stacks reside in special segments called stack segments. A stack segment consists of a number of contiguous parts called stack frames which are dynamically allocated to each procedure. The first stack frame is loaded into the low end of the segment and succeeding frames are loaded after it. The last frame loaded is considered the top of the stack. A T-register 114 (see Figure 1) locates the top of the stack for the currently active process. A virtual T-register exists in the process control block (PCB) of all other processes in the system.

A stack frame consists of three areas: a work area in which to store variables, a save area in which to save the contents of registers, and a communications area in which to pass parameters between procedures. Prior to a procedure call, the user must specify those registers he wishes saved and he must load into the communications area the parameters to be passed to the called procedure. When the call is made, the hardware saves the contents of the instruction counter and specified base registers to facilitate a return from the called procedure.

Each procedure call creates a stack frame within a stack segment and subsequent calls create additional frames. Each exit from one of these called procedures causes a stack frame to be deleted from the stack. Thus, a history of calls is maintained which facilitates orderly returns.

To ensure protection between procedures executing in different rings, different stack segments are used. There is one stack segment corresponding to each protection ring per process. A process control block (PCB) contains three stack base words (SBW) which point to the start of the stack segment for rings 0, 1 and 2 associated with the process. The ring 3 stack segment can never be entered by an inward call; therefore, its stack starting address is not required in the PCB.

The procedure call is used by users who have written their programs in a modular way to pass from one program module to another. It is used by user programs to avail themselves of operating system services. It is used by the operating system itself to achieve a responsive modular structure. The procedure call as is described in the above referenced patent application is effected by hardware instructions and the hardware recognizable stack mechanism.

The main requirements on a procedure call mechanism are:

1. Check the caller's right to call the caller;

2. Save the status of the caller which includes saving registers, instruction counter (for return), and other status bits;

3. Allow for the passing of parameters;

4. Determine valid entry point for the called procedure;

5. Make any necessary adjustments in the addressing mechanism;

6. Enter the new procedure.

When the called procedure terminates or exits, whatever was done in the call must be undone so that the status of the calling procedure is restored to what it was before the call.

As a preliminary to making a procedure call, the instruction PREPARE STACK is executed. This instruction causes those registers specified by the programmer in the instruction to be saved in the stack. It causes the status register (see Figure 1) to be saved, and provides the programmer with a pointer to parameter space which he may now load with information to be passed to the called procedure.

Another instruction ENTER PROCEDURE permits the procedure call via the following steps corresponding to the requirement specified above:

1. Ring checking—the caller's ring is checked to make sure that this ring may call the new procedure; the call must be to a smaller or equal ring number; and if ring crossing does occur the new procedure must be gated through a gate 204 of Figure

2. The new ring number will then be that of the called procedure.

2. The instruction counter is saved;

3. Base register 0 (see Figure 1) is made to point effectively to the parameters being passed;

4. The entry-point of the called procedure is obtained from a procedure descriptor whose address is contained in the ENTER PROCEDURE INSTRUCTION;

5. A point to linkage information is loaded in base register number 7.

6. The new procedure is entered by loading the new ring number and the address of the entry-point in the instruction counter.

The remainder of the current stack-frame is also available to the called procedure for storage of local variables.

When the called procedure wishes to return, it executes the instruction EXIT PROCEDURE. The registers and the instruction counter are then restored from their saving areas in the stack.

Referring to Figure 1 there is shown a block diagram and a computer hardware system utilizing the invention. A main memory 101 is comprised of four modules of metal-oxide semi-conductor (MOS) memory. The four memory modules 1—4 are interfaced to the central processor unit 100 via the main store sequencer 102. The four main memory modules 1—4 are also interfaced to the peripheral subsystem such as magnetic tape units and disk drive units (not shown) via the main store sequencer 102 and the IOC (not shown). The main store sequencer gives the capability of providing access to and control of all four memory modules.

Operations of the CPU are controlled by a read only memory ROM, herein called the control store unit 110.

The control store interface adapter 109 communicates with the control store unit 110, the data mangagement unit 106, the address control unit 107 and the arithmetic logic unit 112 for directing the operation of the control store memory. The control store interface adapter 109 includes logic for control store address modification, testing, error checking, and hardware address generation. Hardware address generation is utilized generally for developing the starting address of error sequencers or for the initialization sequence.

The buffer store memory 104 is utilized to store the most frequently used or most recently used information that is being processed by the CPU.

The data management unit 106 provides the interface between the CPU 100 and main memory 101 and/or buffer store memory 104. During a memory read operation, information may be retrieved from main memory or buffer store memory. It is the responsibility of the data management unit to recognize which unit contains the information and strobe the information into the CPU registers at the proper time. The data management unit also performs the masking during partial write operations.

The instruction fetch unit 108 which interfaces with the data management unit 106, the address control unit 107, the arithmetic and logic unit 112 and the control store unit 110 is responsible for keeping the CPU 100 supplied with instructions.

The address control unit 107 communicates with the instruction fetch unit 108, the buffer store directory 105, the main store sequencer 102, the arithmetic logic unit 112, the data management unit 106, and the control store unit 110 via the control store interface adapter 109. The address control unit 107 is responsible for all address development in the CPU.

Interfacing with the address control unit 107, the instruction fetch unit 108 and the control store unit 110 is the arithmetic logic unit 112 which is the primary work area of the CPU 100. Its primary function is to perform the arithmetic operations and data manipulations required of the CPU.

Associated with the arithmetic logic unit 112 and the control store unit 110 is the local store unit 111 which typically is comprised of a 256-location (32 bits per location) solid state memory and the selection and read/write logic for the memory. The local store memory 111 is used to store CPU control information and maintain ability information. In addition, the local store memory 111 contains working locations which are primarily used for temporary storage of operands and partial results during data manipulation.

The central processing unit 100 typically contains 8 base registers (BR) 116 which are used in the process of address computation to define a segment number, an offset, and a ring number. The offset is a pointer within the segment and the ring number is used in the address validity calculation to determine access rights for a particular reference to a segment.

The instruction counter 118 communicates with the main memory local register (MLR) 103 and with the instruction fetch unit 108, and is a 32-bit register which contains the address of the next instruction, and the current ring number of the process (PRN). Also contained in the central processing unit is a T register 114 which also interfaces with the instruction fetch unit 108 and is typically a 32-bit register containing a segment number and a 16-bit or 22-bit positive integer defining the relative address of the top of the procedure stack. The status register 115 is an 8-bit register in the CPU which among other things contains the last ring number—i.e. the previous value of the process ring number (PRN).

The main memory 101 is addressed by the memory address register (MAR) 119, and the information addressed by (MAR) 119 is fetched and temporarily stored in the memory local register (MLR) 103.

Referring now to Figure 3 there is shown a flow diagram of the general rules for segmented address development shown in detail in the above mentioned copending patent application No. 21630/74, Serial No. 1,465,344. Figure 3 when read in conjunction with the above referenced patent application is self-explanatory. There is however one major difference between the address development as shown on Figure 3 to that of the above mentioned application and that is that in the address development of Figure 3 of the instant application as many as 16 levels of indirection may be utilized in the address development whereas in the above referenced application the levels of indirection were limited to a maximum of two. This of course is a matter of choice with the designer and in no way alters the high level inventive concept.

Referring now to Figures 4A—4J, Figures 4A and 4B show the format of the instruction counter designated by reference numeral 118 on Figure 1. The instruction counter (IC) 118 is a 32-bit register which contains the address of the next instruction, and the current ring number of the process (PRN). Referring specifically to Figures 4A and 4B the TAG is a 2-bit field which corresponds to the TAG field of data descriptors shown and described in the above reference application entitled "Segmented Address Development". PRN is a 2-bit field which defines the current ring number of the process to be used in determination of access rights to main storage. SEG is typically either a 12-bit or a 6-bit field which defines the segment number where instructions are being executed. The OFFSET is typically either a 16-bit or a 22-bit field which defines the address of the instruction within the segment SEG.

Figures 4C—4F show the format of segment descriptors with Figures 4C and 4D showing the first and second word of a direct segment descriptor whereas Figures 4E and 4F show the first and second word of an indirect segment descriptor. Segment descriptors are two words long each word comprised of 32 bits. Referring to Figures 4C—4D which show the first and second word respectively of a direct segment descriptor, P is a presence bit. If P equals one, the segment defined by the segment descriptor is present in main storage. If P equals zero, the segment is not present and a reference to the segment descriptor causes a missing segment exception. All other fields in a segment descriptor have meaning only if P equals one. A is the availability bit. If A equals zero, the segment is unavailable (or locked) and a reference to the segment causes an unavailable segment exception. If A equals one, the segment is available (or unlocked, and can be accessed). I is the indirection bit. If I equals zero, the segment descriptor is direct. If I equals one, the segment descriptor is indirect. U is the used bit. If U equals zero, the segment has not been accessed. If U equals one, the segment has been accessed. U is set equal to one by any segment access. W is the written bit. If W equals zero, no write operation has been performed on the segment. If W equals one, a WRITE operation has been performed on the segment. W is set to one by any WRITE operation. GS is the gating-semaphore bits. When the procedure call mechanism referred to above requires that the segment be a gating segment or when the process communication mechanism (not shown) requires that the segment be a segment descriptor segment (SD) the GS bits are examined. To be a valid gating segment, the GS bits must have the value 10. To be a valid SD segment, the GS bits must have the value 01. If a gating or SD segment is not required, these bits are ignored. The BASE is a 24-bit field which defines the absolute address in quadruple words of the first byte of the segment. This field is multiplied by 16 to compute the byte address of the segment base. The SIZE is a field which is used to compute the segment size. If the segment table number, subsequently referred to as STN, is greater or equal to zero but less than or equal to six, the SIZE field is 18 bits long. The STN is a field indicating the segment table entry STE for selecting a segment descriptor. If the STN is greater than or equal to 8 but less than or equal to 15, the SIZE field is 12 bits long. The number of bytes in the segment is equal to 16 times (SIZE+1). If SIZE equals zero, the segment size is 16 bytes. RD is the read access field. This is a 2-bit field which specifies the maximum EAR (effective address ring number) for which a read operation is permitted on the segment. (A procedure is always permitted to read its own segment if EAR equals PRN). WR is the write access field. This is a 2-bit field which specifies the maximum EAR for which a write operation is permitted on the segment and the minimum PRN at which the segment may be executed. MAXR is the maximum ring number. This is a 2-bit field which specifies the maximum PRN at which the segment may be executed. WP is the write permission bit. This bit indicates whether a WRITE operation may be performed on the segment. If WP equals zero, no WRITE operation may be performed. If WP equals one, a WRITE operation may be performed if EAR is greater than or equal to zero but less than or equal to WR. EP is the execute permission bit. This bit specifies whether the segment may be executed. If EP equals zero, the segment may not be executed. If EP equals one, the segment may be executed at any PRN for which PRN is greater than or equal to WR but less than or equal to MAXR. MBZ is a special field which must be set to zero by software when the field is created, before its initial use by hardware.

Referring to Figures 4E—4F the definitions of the various fields are similar as above however word 0 includes a LOCATION field and word 1 includes a

RSU field. The LOCATION field is a 28-bit field which defines the absolute address of a direct segment descriptor. The value in the LOCATION field must be a muliple of 8. The RSU field is a special field which is reserved for software use.

Figures 4G—4H show the format of the base registers (BR) which are used in the process of address computation to define a segment table number, a segment table entry number, an offset, and a ring number. There are typically 8 base registers as shown by reference numeral 116 on Figure 1. A base register is specified or identified as base register 0 through 7. The size of a base register is 32 bits long. The base register format of Figure 4G is utilized for small segment i.e. where STN is greater or equal to 8 but less than or equal to 15, whereas the format of base register of Figure 4H is utilized for large segments i.e. STN is greater or equal to zero but less than or equal to six. Referring to Figures 4G—4H, TAG is a 2-bit field which corresponds to the TAG of a data descriptor referenced previously. RING is a 2-bit field which contains the ring number associated with the segmented address for protection purposes. SEG is a field previously referred to, which identifies a segment described in a segment table. STN is the segment table number, and STE is the segment table entry number. OFFSET is a 16-bit field or a 22-bit field depending on segment table number, which defines a positive integer. The OFFSET is used in the process of address development as a pointer within a segment.

Referring to Figures 4I—4J there is shown the format of the T-register. The T-register is a 32-bit register containing a segment number and a 16-bit or 22-bit positive integer defining the relative address of the top of the procedure stack previously mentioned. The T-register is shown by reference numeral 114 on Figure 1. The various fields of the T-register have the same definition as described above.

Referring now to Figures 3 and 4A—4J a more defined description of absolute address calculation and access checking is made. In general absolute address calculation consists of fetching a segment descriptor specified by STN and STE and using the segment descriptors in four ways: access checking, computation of the absolute address, bound checking, and updating (U and W flags). As described in copending patent application No. 21630/74, (Serial No. 1,465,344) the absolute address may be direct or indirect and is derived by first deriving an effective address from STN, STE, and SRA (segment relative address). STN is extracted from bits 4 through 8 of the base register BR specified in the address syllable of an instruction. If STN is 7, an out of segment table word array exception is generated. STE is extracted from the base register specified in the address syllable. If STN 4:4 (i.e., beginning at bit 4 and including the next 4 bits) is greater than or equal to zero or less than or equal to six, STE is in a base register bits 8 and 9. If STN 4:4 (i.e. 4 bits beginning at bit 4) is greater than or equal to 8 but less than or equal to 15, STE is in a base register BR bits 8 through 15. The segment relative address SRA for direct addressing is computed by adding the displacement in the address syllable; the offset of the base register BR; and the 32-bit contents of an index register, if specified in the address syllable. The sum of these three quantities is a 32-bit unsigned binary integer which must be less than the segment size appropriate to the segment STN, STE.

Indirect addressing is developed by fetching a data descriptor and developing an address from that descriptor. The effective address of the data descriptor is computed as in the direct addressing case with the exception that the index register contents are not used. In developing the address from the data descriptor the effective address may be computed by an indirection to segment ITS descriptor and an indirection to base ITBB descriptor. If the descriptor is ITS the STN and STE are extracted from the descriptor in the same manner as from a base register. SRA is computed by adding the displacement in the descriptor and the contents of an index register as specified in the syllable. If the descriptor is an ITBB descriptor then STN and STE are extracted from the base register specified in the BBR field (i.e. the base register implied by ITBB descriptor) of the descriptor as in direct addressing. SRA is computed by adding the displacement in the descriptor, the offset of the base register, and the contents of an index register is specified in the address syllable.

As shown on Figure 3 the indirection process may be extended up to 16 levels.

Every effective address contains protection information which is computed in address development and checks for access rights by the ring protection hardware of the absolute address calculation mechanism. The effective address contains protection information in the form of an effective address ring number EAR (see Figures 2J and 2K of above application No. 21630/74, (Serial No. 1,465,344). The EAR is computed from the base register ring number BRN and from the current process ring number PRN by taking the maximum ring number. In developing the EAR for indirect addressing a somewhat more tedious but essentially similar procedure as indirect addressing is used. In indirect addressing the EAR for extraction of the first descriptor (EAR 1) is once again the maximum of the ring number from the base register specified in the address syllable and the current process ring number PRN in the instruction counter 115 of Figure 1 and stored in 00 register 512 of Figure 5. The EAR for extraction of the second descriptor (EAR 2), of multiple level indirection is the maximum of:

a. EAR 1;

b. The ring number in the first descriptor if indirection is indirection to segment;

c. The ring number from a base register 116 utilized as a data base register BBR if the first descriptor is an indirection to segment descriptor ITBB.

The EAR for extraction of the data of multiple level indirection is the maximum of:

a. EAR 2;

b. The ring number in the second descriptor if it is an indirection segment descriptor ITS;

c. The ring number in one of the base registers utilized as a data base register BBR if the second descriptor is an indirection to base descriptor ITBB.

Referring now to Figures 5 and 6, the transfers and manipulation of the various type ring numbers will be described at the system level. Detailed logic block diagrams for effecting the transfers and operations of Figure 5 will be later described. Referring first to Figure 6 an associative memory 600 is utilized in segmented address development. The associative memory 600 comprises essentially a UAS associator 609 which has circuitry which includes associative memory cells, bit sense amplifiers and drivers, and word sense amplifiers and drivers (not shown). A word or any part of a word contained in UAS associator 609 may be read, compared to another word with a match or no match signal generated thereby, or be written either in whole or in a selected part of the associator 609. For example, US register 607 may contain a segment number which may also be in the associative memory 600. A comparison is made with UAS associator 609 and if a match is found a "hit" results. The match or "hit" signal is provided to encoder 610. The function of encoder 610 is to transform the "hit" signal on one of the match lines to a 4 bit address. Encoder 610 provides this 4 bit address to UAB associator buffer 611 so that the information contained in that particular location of UAB associator buffer 611 is selected. Information in UAB associator buffer 611 may be transferred to UV register 613 for temporary storage or for transfer to QA or QB bus 614 and 615 respectively. By thus locating a prestored segment number of the associative memory 600 (which may have been placed there after a generation of an absolute address) regeneration of the same address is not necessary. In the drawing of Figure 6, UAB associator buffer 611 is shown as storing a first and second word of a segment descriptor; however other types of information may just as well be stored therein. This buffer 611 provides a function similar to that of buffer 104 in the more generalised diagram of Figure 1.

As mentioned supra the development of an absolute address of an operand from an effective address is disclosed in patent application No. 21630/74, (Serial No. 1,465,344). Briefly and with reference to Figure 6 any of 8 base registers 602 are addressed via UG and UH registers 603 and 604 respectively which contain base register addresses from an instruction address syllable or base register specified by the instruction formats. The base register 602 contain such information as TAG, base register ring number BRN, segment table number STN, segment table entry STE and OFFSET as shown or contained by base registers 1 and 2 of the group of base registers 602. Writing into the base registers is performed under micro-op control by UWB logic 601. For example it is shown that information from the UM register 502 of Figure 5 may be written into bit positions (2, 3) of a selected base register; also information from the QA bus may be written into the base registers and provisions are made to clear a selected base register i.e. write all zeroes. Reading out of any of the base registers is performed by UBR logic 605. In general the UBR logic 605 permits the appropriate base register to be strobed out onto bus QA or QB, or into UN register 608. Note that UN register 608 holds bits 8 through 31 of the base registers which is the OFFSET part of the segmented address. Moreover UBR logic 605 when addressed by an address contained in instruction buffer IB (not shown) reads out the segment number SEG (which is comprised of STN and STE) into US register 607 via UBS transfer logic 606. The comparison of the segment number SEG in US register 607 with the associative memory 600 may then be performed as previously described. It will be noted that bits (4—15) of QA bus 614 may also be read into or from US register 607. Similarly bits (8—31) from QA bus 614 may read into UN register 608. Also bits (9—11) of the US register 607 may be read into QA bus 614 as denoted by US (9—11) arrow (the arrows into various register and/or logic circuitry denote the source of data and that followed by a number denote the bit numbers of that data).

Referring now to Figures 5 and 6, a 2-bit UP register 501 stores the current process ring number PRN. The current process ring numbers PRN is obtained from bits 2 and 3 of the instruction counter (118 or Figure 1) via bits IC (2—3) of the QA bus 614 of Figure 6. Bits IC (2—3) of QA bus 614 are transferred to 2-bit UV register 503 under control of a micro-operation UV9QA0. The micro-operations are obtained from micro-instructions in the control store unit 110. (On Figure 5 the dot surrounded by a circle indicates a micro-operation and the first two letters of the name of the micro-operation indicate the destination of the data to be transferred; the fourth and fifth letters indicate the source of the data transferred; the third character indicates whether a full or partial transfer is made with F indicating a full transfer while the sixth character indicates whether the signal doing the transferring is high or low with even numbers indicating a low signul and odd numbers indicating a high signal. As an example of the use of this convention bits 2 and 3 on QA bus indicating the tail of the arrow QA (2, 3) indicate PRN is the PRN process ring number that is being transferred under control of the micro-op UV9QA0 which says the transfer is made to register UV, is a partial transfer of the bus QA, and the source of the data is the bus QA and is an unconditional transfer as indicated by the sixth character being 0. Transfer to UV register from QA bus source is unconditional. This 0 will be the corresponding seventh character in the logic file name of the subcommand UV9QA10. Once the process ring number PRN is transferred from the QA bus 614 to the UV register 503 another transfer takes place under control of the micro-operation UM9UV0 from UV register 503 to UM register 502. Finally another transfer takes place from UM register 502 to UP register 501 under control of a micro-operation UP9UM0.

Two bit register UM 502 is utilized to generate the effective address ring number EAR during ITS and ITBB (i.e. indirection to segment and indirection to base), (EAR=MAX (BRN, PRN, DRN,/BBR (BRN) etc.) address formation for address syllable 1 and address syllable 2 type instruction format. The EAR is generated according to the rules previously enunciated by utilizing one or more tests shown in block 510 and the maximum of the ring number is obtained and stored in UM register 502 which stores the effective address ring number EAR (detailed logic or making the comparisons of block 510 are later shown and described in detail). The UO register is used to save address syllable 1 effective address ring number EAR in the event the address syllable 2 is being utilized to extract EAR 2.

Two-bit UV register 503, and 2-bit UW register 504 is utilized mainly as storage for various ring numbers that are obtained from the outside of the ring checking hardware of Figure 5 and transferred or processed to other parts of the ring checking hardware. For example the base register ring number BRN is transferred from bit positions 2 and 3 of UBS transfer logic 606 to UV register 503 under control of the micro-operation UVFBS0; the maximum ring number MAXR of word 2 of the segment descriptor (also shown stored in bits 36 and 37 of UAB associator buffer 611) is transferred from UAB buffer 611 to UV register 503 under control of the micro-operation UVFAB1; also bits 34 and 35 of UAB buffer 611 which is the write ring number WR is transferred to UV register 503 under control of micro-operation UVFAB0. UW register 504 has similar transfers of other ring numbers from various parts of the system. For example bits 34 and 35 which are the write ring number WR of UAB buffer 611 may also be transferred to UW register 504 under control of micro-operation UWFAB1; bits 32 and 33, the read RD ring number of UAB buffer 611 may also be transferred to UW register 504 under control of micro-op UWFAB0; also bits 0 and 1 of QA bus 614 may be transferred to UW register 504 under control of micro-operation UW9QA0. Note also several transfer paths of UW register 504 into UV register 503 under control of the micro-operation UV9UW0; the transfer path of UV register 503 into UM register 502 under control of micro-operation UM9UV0; the transfer path of UM register 502 into UP register 501 under control of the micro-operation UP9UM0; the transfer path of UP register 501 into UM register 502 under control of micro-operation UM9UP0; the transfer path of UM register 502 into UO register 512 under control of micro-operation UO9UM0; and finally the transfer path of UO register 512 into UM register 502 under control of the micro-operation UM9UO0.

Briefly therefore UP register 501 holds the current process ring number PRN; UM register 502 and UO register 512 are utilized for transfer operations and also to generate the EAR; UV register 503 may shore for various purposes and at different times the current process ring number PRN, the base register ring number BRN, the maximum ring number MAXR, the write ring number WR, or the read ring number RD. UW register 504 may at various times hold the read ring number RD, the write ring number WR, and bits 0 and 1 of bus QA. UMR 505 is logic, the details of which are shown on Figure 8d, which compares the contents of registers UM and UV and produces the greater of the two values in the registers and this value is stored in UM register 502 under micro-operation control UMFMR0. This is one way of generating the effective address ring number EAR. UMR logic 505 may also produce the greater value of the contents of register UP or of bits 2 and 3 of UBS logic 606. This is another method and/or additional step in generating the effective address ring number EAR. UMR logic 505 is also utilized to determine whether or not a write violation has occurred by transferring a write ring number WR into UV register 503 and then comparing the contents of the UM register 502 (holding EAR) with the contents of UV register 503 in order to determine which one has the greater contents. Since UM register 502 stores the effective address ring number EAR a comparison of the UM register and the UV register will indicate whether EAR is greater than WR or vice versa. If WP (i.e. write permission bit in the segment descriptor) is equal to 1 and if EAR lies in the range of $0 \leq EAR \leq WR$ then a write operation may be performed into the segment. Note that UMR logic 505 may have inputs directly or indirectly from all registers 501—504, from other logic 506, 507 and also from UBS logic 606.

UWV logic 506 corresponds to the detail logic of Figure 8a. UWV logic 506 has inputs directly or indirectly from registers 501—504 and from logic 505, 507 respectively and generates an execute violation signal when a comparison of UW, UM and UV registers 504, 502, and 503 respectively indicates that the statements that the maximum ring number MAXR is greater or equal to the effective address ring number EAR, and that EAR is greater or equal to the write ring number WR are not true i.e. in order for a procedure to be able to execute in a given segment indicated by the effective address the maximum ring number MAXR must be greater or equal to the effective address ring number and the effective address ring number EAR must be equal or greater than the write ring number WR. UWV logic 506 also performs tests shown in block 510. Indications may be given that the contents of UW register is less than or equal to the contents of the UV register; the contents of the UM register is greater than or equal to the contents of the UV register; the contents of the UV register is equal to the contents of the UM register; the contents of the UV register is greater or equal to the contents of the UM register; and the contents of the UM register is greater than the contents of the UW register. Of course when performing these tests different values of ring numbers may occupy the registers.

UEP logic 507 corresponds to the detail logic of Figure 8b. UEP logic 507 in combination with UWV logic 506 generates the read violation exception. However the read violation exception may be overridden if the effective address ring number EAR equals the current process ring number PRN, since a procedure is always permitted to read its own segment, and if the segment number of the procedure segment descriptor (not shown herein) and the segment number of the address syllable utilized in generation of the effective address are the same.

To illustrate the overriding of the read violation signal assume that the effective address read number EAR is greater than the read number RD which would generate a read violation high signal which would be applied as one input of AND gate 522. However the read violation exception signal may not be generated even though there is a read violation signal if the following two conditions exists:

1. The effective address ring number EAR is equal to the process ring number PRN; i.e. the contents of register UM is equal to the contents of the register UP; and,

2. The segment number contained in the address syllable of the segment in which a procedure desires to read is equal to the segment number of the procedure segment descriptor (not shown) of the current procedure in execution and this is indicated by setting a bit called a P bit and located as the thirteenth bit of UE register 650. (UE register 650 is a store for the contents of UAS associator 609 when a "hit" has resulted by a comparison of the contents of US register 607). Since this example assumes that EAR equals PRN, UEP logic 507 will apply a high signal to AND gate 520 as one input, and since it is also assumed that the segment number SEG of the address syllable of the segment being addressed is equal to the segment number SEG of the procedure segment descriptor (not shown) of the currently executing procedure, then the P bit of the procedure segment descriptor will be set and hence the other input applied to AND gate 520 will be high thus enabling AND gate 520; a high signal is therefore applied to the input of inverter 521 resulting in a low signal at the output of inverter 521 which low signal is then applied as another input of AND gate 522. Since there is a low signal to AND gate 522 no read violation exception signal can be generated by amplifier 523 even if

the third input signal applied to AND gate 522 is high.

To illustrate how a read violation signal is generated and not overridden, assume that the output of UEP logic 507 indicates that the contents of UM register is not equal to the contents of UP register. Then that input to AND gate 520 would be low and hence AND gate 520 would not be enabled and its output would be low and would be applied to the input of inverter 521. Since the input of inverter 521 is low its output would be high which would be applied as one input of AND gate 522. If also the effective address ring number EAR is greater than the read ring number RD (i.e. contents of UM register is greater than contents of UW register) that signal would be high and would be also applied to another input of AND gate 522. AND gate 522 has still a third input which must also be high in order to enable AND gate 522. This third input is high when AND gate 526 is enabled. Since AND gate 526 has one input terminal which is high when the 00 terminal of URV1F flop 524 is low, AND gate 526 is enabled by applying the micro-operation read violation interrogate signal AJERVA to one input terminal of AND gate 526 while the 00 terminal of URV1F flop 524 is low. Thus AND gate 522 will have all input terminals high, generating the read violation exception signal.

The execute violation exception is generated in two ways. It was seen earlier that an execute violation signal results when UWV logic 506 indicates that the inequalities WR is less than or equal to EAR, and EAR is less than or equal to MAXR are not true. This high execute violation signal is applied to a one-legged AND gate 550 which in turn is applied to the input terminal of two-legged AND gate 553 via amplifier 552. When an execute violation interrogate micro-operation signal AJEEVA is applied as another input of two-legged AND gate 553, this gate is enabled which in turn generates the execute violation exception via amplifier 554. The other method by which the execute violation exception is generated by the execute violation hardware 511 is when the execute permission bit EP is not set. When this condition is true it is indicated by the seventh bit of UY register 613 being high; this bit is then applied to the input terminal of one-legged AND gate 551 which is applied as a high signal to one input terminal of AND gate 553 via amplifier 552. When the execute violation interrogate micro-operation signal AJEEVA goes high, AND gate 553 is enabled and generates an execute violation exception via amplifier 554.

The write violation exception is also generated in two ways. It was seen previously how the UMR logic 505 generates a write violation signal when EAR is greater than WR. This write violation signal is applied to one input terminal of AND gate 545. AND gate 545 is enabled when its second input terminal goes high thus generating a write violation exception through amplifier 547. The second input terminal of AND gate 545 goes high when AND gate 542 is enabled. AND gate 542 is enabled when the input signals applied to its input terminals are high. One input signal is high when UWV1F flop 541 is low which in turn applies a low signal to the input terminal of inverter 543 which in turn applies a high signal to one input terminal of AND gate 542; the other input signal is high when the write violation interrogate micro-op signal AJEWVA is high and this happens when it is desired to interrogate a procedure for the write violation exception. (Flip-flops URV1F, URN1F, and UWV1F are set low when any interrupts or softward occurs). (UWV2F, URV2F, and URN2F flip-flops are utilized to store back-up excess checking information for ring checking). The other method for generating a write violation exception is when the write permission bit WP is not set. This condition is indicated by bit 6 of UV register 613 being high. When this condition exists and the high signal (i.e. the sixth bit of UV register) is applied as one input of AND gate 546 and the interrogate signal AJEWVA is high and applied as another input of AND gate 546, then AND gate 546 is enabled and a write violation exception occurs via amplifier 547.

Logic circuitry 591 comprised of flip-flops 532 and 533 in conjunction with amplifier 530 and AND gate 531 and inverter 530A permit the formation in register UM 502 of the maximum value of ring number (i.e. EAR) under control of a splatter instruction subcommand (not described herein) from the instruction fetch unit IFU. Assuming URN1F flip-flop 532 is set to logical 0 whereas URN2F flip-flop 533 is set to logical 1, then during the execution of the splatter subcommand, input terminal 531A of AND gate 531 will be high; therefore if flip-flop 532 is low (logical 0) then the signal will be inverted by inverter 530A and AND gate 531 will be enabled. Hence the maximum value of the contents of UP register 501 or bits 2 and 3 of logic vector UBS 606 will be strobed into UM register 502. Conversely if flip-flop 532 is a logical 1, then the contents of UM register 502 is not changed via the above mentioned sources and the EAR derived in UM register 502 via the addressing process of indirection is the one utilized. Flip-flop

533 is the back-up store for the EAR of address-syllable 2 when utilized.

Referring now to Figures 7 and 8 and Figure 5 there is a correspondence wherein the detailed logic for hardware in Figure 5 is shown in Figures 7 and 8 as follows: Figure 7a and UW register 504; Figure 7b and UV register 503; Figure 7c and block 590; Figure 7d and block 591; Figure 7e and block 592; Figure 7f and UP register 501; Figure 7g and UO register 512; Figure 7h and UM register 502; Figure 8a and UWV logic 506; Figure 8b and UEP logic 507; and Figure 8d and UMR logic 505.

Referring to Figure 7a, the UW register 504 is comprised of two flip-flops 715a and 720a respectively, each flip-flop capable of holding one bit of information of the UW register. Coupled to flip-flop 715a are 4 AND gates 711a—714a which are OR'ed together, with each gate (except gate 713a) having two input terminals, and with at least one signal applied to each input terminal. AND gate 714a has one of its input terminals coupled to the set terminal OW00010 of the flip-flop 715a. Flip-flop 715a is also coupled to the terminal H27 for receiving from a clock a timing signal called a PDA signal. Flip-flop 720a coupled to AND gates 716a—719a which are OR'ed together. One input terminal of AND gate 716a is coupled to an input terminal of AND gate 711a; one input terminal of AND gate 717a is coupled to one input terminal of AND gate 712a and one input terminal of AND gate 719a is coupled to an input terminal of AND gate 714a, whereas the other input terminal of AND gate 719a is coupled to the set terminal UW00110 of the flip-flop 720a. Flip-flop 720a is also coupled to the H27 terminal for receiving PDA pulses.

AND gates 701a—704a are OR'ed together each having their output terminal coupled to the input terminal of inverter 705a. AND gate 706a is coupled to amplifier 708a; whereas AND gate 707a is coupled to amplifier 709a; one input terminal of AND gate 706a is coupled to one input terminal of AND gate 707a. The output terminal of inverter 705a is coupled to one input terminal of AND gate 714a and 719a; the output terminal of amplifier 708a is coupled to the input terminal of AND gate 713a and the output terminal of amplifier 709a is coupled to the input terminal of AND gate 718a.

The signals applied to the inputs of AND gates and the signals derived as outputs from amplifier, inverters, or flip-flops are designated by letters forming a special code. Since both data signals and control signals are either applied or derived there are two codes, one code for the control signals and one code for the data signals. The code for the control signals are previously described in detail and is summarized here. Briefly the first two characters of a control signal indicate the destination of data to be transferred; the third character indicates whether a full or partial transfer is to be effected with the letter F indicating full transfer and any other character indicating a partial transfer; the fourth and fifth character indicates the source of the data, and if the source is identified by more than two letters only the last two letters need be used; the sixth and seventh characters are usually numerals and indicate whether the signal is high or low i.e. an odd numeral in the sixth position indicates assertion and an even numeral in the sixth position indicates negation; the seventh position indicates whether this is the first, second, third, etc. level of occurrence of the signal. Data, on the other hand, is indicated differently. The first three characters of data indicates the source of the data, the fourth and fifth characters which may be numerals indicate the bit positions where the data is located in the source, and the sixth and seventh position are similar to the control signals in that they indicate whether the signal is high or low and the level of occurrence of the signal. Generally the format itself indicates whether the signal is a control signal or a data signal and by reference to Figures 5 and 6 the source and destination may be determined. There are exceptions to this general rule and they will be spelled out in the specification, and addenum.

As an example of this convention it will be noted on Figure 7a that the following signals are control signals: UWFAB11, UWFAB10, UW9QA10. The following signals are data signals UAB3410, UAB3210, UAB3510, UAB3310, QA00110, and QA00010. The following signals are exception PDARG10 is a timing signal whose source is the PDA clock; UWHOL10 is a hold signal for holding the information in the flip-flops 715a and 720a UW0BK10 and UW1BK10 are back-up logic whose main function is to extend the input capability of flip-flops 715a and 720a by connecting the UW register which is in fact formed by flip-flops 715a and 720a, to bit zero and bit 1 represented by flip-flops 715a and 720a respectively; and finally USCLR10 is the clear signal for clearing and setting the flip-flops to zero.

As an illustration of the above mentioned convention herein adopted the signal UWFAB11 applied to the input of one-legged AND gate 702a is a control signal which transfers data (bits 34 and 35) contained in UAB associator buffer 611 (the U in the signal has been omitted) to UW register 504 and is a full transfer to the

UW register 1; the odd number indicates the signal is assertion. Signal UWFAB10 applied to the input of one-legged AND gate 703a is a control signal with the same source and destination as the signal applied to AND gate 702a except that bits 32 and 33 of UAB are transferred to UW register. The signal UW9QA10 applied to one-legged AND gate 704a is also a control signal wherein data is transferred from QA bus 614 to the UW register and may be a partial transfer. The signal QA00010 applied to AND gate 706a is a data signal where data is on QA bus 614 (the third position is not herein utilized since the first two positions adequately describe where the data is) and this data signal represents the bit identified as 00 on QA bus 614. The signal QA00110 is similar to the previous signal except the data identified by this signal is the data on position 01 of the QA bus 614. Thus by utilizing this convention and Figures 5 through 9 the ring protection hardware is fully defined and may be easily built by a person of ordinary skill in the computer art.

Referring to Figure 7b there is shown the detailed logic block diagram for UV register 503. Signal UVHOL10 is a hold signal for UV register 503 which is generated via inverter 703b when none of the one-legged AND gates 701b—708b has a high signal applied to it. UVHOL10 signal is applied to AND gate 723b and causes information stored in the UV register 503 to be held therein. Signal UVHOL1E coupled to the input of AND gate 704b and to the outputs of AND gates 705b—708b extends the number of control signals that may generate the hold signal UVHOL10. Signal UV0BK10 coupled to the outputs of AND gates 710b—713b and to the input of AND gate 722b is also utilized to extend the number of inputs signals that may be applied to flip-flop 724b. Signal UV1BK10 coupled to the outputs of AND gates 716b—718b and to the input of AND gate 727b similarly extends the number of input signals that may be applied to flip-flop 729b.

Referring now to Figure 7g there is shown the detailed logic block diagram of UO register 512. AND gates 701g—704g are OR'ed together and their output is applied as an input to inverter 705g. AND gates 706g—709g are also OR'ed together and their outputs are coupled to flip-flop 710g. Also one input of AND gate 709g is coupled to the U000010 terminal of flip-flop 710g. AND gates 711g—714g are also OR'ed together and are similarly coupled to flip-flop 715g. It will be noted also that an input of AND gate 706g is coupled to an input of AND gate 711g; an input of AND gate 707g is coupled to an input of AND gate 712g and an input of AND gate 709g is coupled to an input of AND gate 714g. The UOHOL10 signal generated by inverter 705g is also coupled to an input of AND gate 709g and 714g and is utilized to hold information in the UO register 512. XOO represents a ground, whereas XNU means unused input.

Figure 7f is a detailed logic block diagram of UP register 501. It is similar to Figure 7g described supra except that different signals from different destinations and different sources are applied.

Referring now to Figure 7h there is shown the detailed logic block diagram of UM register 502. AND gate 701h—704h are OR'ed together to produce the UMHOL10 hold signal via inverter 705h. AND gates 706h—709h are OR'ed together and are coupled to the input of AND gate 704h in order to extend the range of signals that may be applied to produce the UMHOL10 hold signal. Similarly AND gates 711h—714h are OR'ed together and coupled to the input of AND gate 723h in order to extend the range of signals that may be applied to flip-flop 730h; and also AND gates 716h—719h are OR'ed together and are coupled to the input of AND gate 727h in order to extend the range of signals applied to flip-flop 731h. A line 740h for applying the PDA signals to flip-flop 730h and 731h is coupled at point 734h and 735h respectively. The input of AND gate 703h is also expanded to provide two further inputs URN1F00 and IRNUM10 by coupling the output of amplifier 733h to the input of AND gate 703h.

Referring now to Figures 7c—7e there is shown detailed logic block diagrams of write exception control logic 590, IFU subcommand control logic 591, and read violation exception control logic 592 respectively. Referring first to Figure 7c there is shown flip-flops 705c and 710c which correspond to flip-flops 541 and 540 respectively. Under a micro-operation URW2F10 subcommand the information in flip-flop 710c is transferred to flip-flop 705c. The UWV1H10 hold signal is utilized to hold the information transferred to flip-flop 710c, whereas the UWV2H10 signal is utilized to hold the information transferred to flip-flop 705c. Similarly in Figure 7d information is transferred from flip-flop 710d to flip-flop 705d under micro-operation signal URNSW10, and in Figure 7e information from flip-flop 710e is transferred to flip-flop 709e under control of micro-operation signal URW2F10.

Referring now to Figures 8a, 8b and 8d there is shown detailed logic block diagrams of UWV logic 506, UWEP logic 507, and UMR logic 505 respectively. Referring first to Figure 8a there is shown logic for generating a high signal when one of the test conditions 510 is true and also for generating the execute violation signal when the contents of UW register is less than or equal to the contents of UM register is less than or equal to the contents of UV register is not true. When the signal UWLEV10 is generated it indicates that the contents of UW register 504 is less than or equal to the contents of UV register 503. The logic for generating this signal was derived pursuant to the following Boolean expression:

$$X_1=\overline{(B\overline{CD})+(AB\overline{D})\times(A\overline{C})}$$

Where X_1 represents the output of amplifier 805a and the various letters of the expression represent different input terminals of AND gates 801a—804a.

An indication that the contents of UV register 503 is greater than or equal to the contents of UM register 502 is had when UVGEM10 signal is generated. This signal is generated via inverter 820a in response to various inputs on AND gates 816a—819a which are OR'ed together and coupled to the input of inverter 820a. The logic for generating the UVGEM10 signal is made pursuant to the following Boolean expression:

$$X_2=\overline{(B\overline{CD})+(AB\overline{D})+(A\overline{C})}$$

An indication that the contents of UM register 502 is greater than or equal to the contents of UV register 503 is indicated by generating signal UMGEV10 via inverter 810a in response to the various inputs of AND gates 806a—809a which are OR'ed together. The logic for generating this signal is derived from the following Boolean expression:

$$X_3=\overline{(B\overline{CD})+(AB\overline{D})+(A\overline{C})}$$

(Wherein X_3 is the generated output signal).

Similarly the UVEQM10 signal is generated pursuant to the following Boolean expression:

$$X_4=\overline{(A\overline{C})+(\overline{A}C)+(B\overline{D})+(\overline{B}D)}$$

Generation of the UVEQUM10 signal indicates that the contents of the UV register 503 is equal to the contents of the UM register 502.

The generation of the UMGEW10 signal indicates that the contents of the UM register 502 is greater or equal to the contents of the UW register 504 and is generated pursuant to logic having the following Boolean expression:

$$X_5=\overline{(B\overline{CD})+(AB\overline{D})+(A\overline{C})}$$

Generation of the UMGTW10 signal indicates that the contents of UM register 502 is greater than the contents of UW register 504 and this signal is generated by logic defined by the following Boolean expression:

$$X_6=(AB\overline{D})+\overline{C}(B\overline{D}+A)$$

The generation of the UWGMV00 signal indicates that the contents of UW register less than or equal to the contents of UM register less than or equal to the contents of UV register is not true. It is obtained when the UVGEM10 signal indicating that the contents of UV register is greater than or equal to the contents of the UM register, and the UMGEW10 signal indicating that the contents of the UM register is greater than or equal to the contents of the UW register are both high.

Referring now to Figure 8b a UMEQP10 signal is generated by logic derived from the following Boolean expression:

$$X_7=\overline{(A\overline{C})+(\overline{A}C)+(B\overline{D})+(\overline{B}D)}$$

When this signal is high it indicates that the contents of UM register 502 is greater than the contents of UP register 501.

Referring to Figure 8d there is shown the detailed logic block diagram for performing the operations of UMR logic 505 shown on Figure 5. One of the operations of this logic is to determine the maximum value of the contents of UP register 501 and of bits 2 and 3 of UBS logic 606. In order to do this there must be an indication whether contents of UP is less than the contents of UBS or the contents of UP is greater than the contents of UBS. The generation of UPBEB10 signal indicates that the contents of UP register 501 is less than or equal to bits 2 and 3 of UBS logic 606; whereas the generation signal UPGTB10 indicates that the contents of UP register 501 is greater than bits 2 and 3 of UBS logic 606. These signals are generated by logic which has been defined by the following Boolean expression:

$$X_8=\overline{(B\overline{CD})+(AB\overline{D})+(A\overline{C})}$$

Where X_8 is the output of inverter 805d and the letters of the expression are various inputs of the AND gates 801d—803d.

To illustrate how the maximum value of the contents of UP register and UBS logic may be determined by the output signals UMPB010 and UMPB110 of amplifier 814d and 817d respectively, assume first that the contents of register UP are less than or equal to bits 2 and 3 of UBS logic because bit 2 is 1 and bit 3 is 1 whereas UB register contains 01. This is indicated by the signal UPLEB10 being high and the signal UPGTB10 being low since it is the inverse of signals UPLEB10. This high UPLEB10 signal is applied to one input of AND gate 813d and also one input of AND gate 806d. If bit 2 of UBS logic is a 1 as indicated by signal UBS0210 then AND gate 813d is enabled and signal UMPB010 goes high and indicates that bit 2 on UBS logic is a 1. Moreover if bit 3 of UBS logic is a 1 indicated by input signal UBS0310 being applied as another input of AND gate 816d then AND gate 816d is enabled and signal UMPB110 is high or a 1. Therefore under the assumed conditions where bits (2, 3) UBS logic is greater or equal to the contents of UP register the maximum value of the two quantities is in UBS, and its number is binary 11 or decimal 4. Hence it is seen how a comparison is first made to determine which hardware contains the maximum, and then a determination is made as to the value of that maximum. By similar analysis one may see how the value of the UP register may be determined by signals UMPB010 and signals UMPB110 when the contents of UP register is greater than the second and third bit of UBS logic. Similarly the maximum value of UM register 502 or UV register 503 may be determined by signals UVGEM10 and UMGTV10 respectively, when UV register 503 is greater than or equal to UM register 502, and conversely when UM register 502 is greater than UV register 503.

Referring now to Figures 9a—9i a legend of symbols utilized in Figures 7 and 8 is shown. Figure 9a shows the symbol when there is a connection internally within the logic board. Figure 9b illustrates an output pin connection. Figure 9c indicates an input pin connection and is generally a source outside of the logic board illustrated. Figure 9d is the symbol utilized for an AND gate. Figure 9e is the symbol utilized for an amplifier; whereas Figure 9f is the symbol utilized for an inverter. Figure 9g illustrates three AND gates 901g—903g that are OR'ed together thus causing output 904g to go high when any one of AND gates 901g—903g is high. Figure 9h shows the symbol of a flip-flop having a 00 reset terminal and a 10 set terminal. A PDA line supplies the clock pulse for causing the flip-flop to switch states when other conditions are present on the flip-flop. Figure 9i represents a micro-operation control signal.

In order to enforce the ring protection scheme between procedures executing in different rings, the invention employs push-down stacks for its procedure linkage mechanism wherein a portion of each stack called a stack frame is dynamically allocated to each procedure. Different stack segments are used for each ring with one stack segment corresponding to one ring. Thus when a procedure is executed in ring RN its stack frame is located in the RN stack segment. Referring to Figure 10 there is shown three stack segments 1001—1003, with each stack segment having stack frames S1—S3 respectively. Ring 3 is assigned to stack segment 1001, ring 1 assigned to stack segment 1002 and ring 0 is assigned to stack segment 1003. Within each stack segment there is a procedure 11 associated with stack frame S1 of segment 1001, a procedure P2 associated with stack frame S2 of stack segment 1002 and a procedure P3 associated with stack frame S3 of stack segment 1003. The segmented addresses (i.e. segment number and segment relative address SEG, SRA) of the first bytes of the stack segments for rings 0, 1 and 2 respectively are located in stack base words SBW0—SBW2 respectively which are in turn located in process control block 104. Since the ring 3 stack segment can never be entered by an inward call (i.e. from a ring higher than ring 3) its stack starting address is not needed. Each stack frame S1, S2, S3 is divided into a working area 1005, 1006, 1007 respectively; an unused portion 1008, 1009, 1010, which is utilized for alignment purposes; a register saving area 1011, 1012, and 1013; and a communication area 1014, 1015, and 1016 respectively. The working area is utilized by its procedure as needed and may contain material required by the process such as local variables, etc. The saving area of the stack frame is utilized to save the contents of various registers such as the status register, the T-register and the instruction counter contents ICC. The communications area stores information which is needed to pass parameters between procedures. Prior to a call to a given procedure the user saves those registers he wishes saved and moreover loads into the communication area the parameters to be passed to the called procedure. When the call is made, the hardware saves the contents of the instruction counter and other specified registers to facilitate a return from the called procedure. Each procedure call creates a stack frame within a stack segment and subsequent procedure calls create additional frames. Hence a stack is created and consists of a number of contiguous parts called stack frames which are dynamically allocated to each procedure. These stacks reside in stack segments. Generally the first stack frame is loaded into the beginning of the segment and succeeding frames are loaded after it. The last frame loaded is considered the top of the stack. A T-register 114 on Figure 1, locates the top of the stack for the currently active process. A procedure such as for example P1 which is executing in ring 3 may call a procedure P2 executing in ring 1 which in turn calls a procedure P3 which is now executing in ring 0. As each procedure is called it creates within its ring stack segment a stack frame (i.e. defining the environment for the procedure execution) and the T-register 114 is loaded which gives the address of the top of the stack for the current active process. The procedure P1 (as previously assumed) may call procedure P2 which in turn may call procedure P3 and since these calls are from a higher ring number to a lower ring number a ring crossing entailing an inward call is required and is accomplished in a manner to be described infra. During each change of procedure the necessary registers and parameters are saved in order to facilitate a return from the called procedure.

A procedure is always accessed through a procedure descriptor 1110 by means of the ENTER PROCEDURE INSTRUCTIONS. The format of the ENTER PROCEDURE INSTRUCTION 1100 is shown on Figure 11a. The operation code (OP) 1101 occupies bit positions 0 through 7. The complementary code 1102 is a one bit code and occupies bit position 8 to 9; if the complementary code is set to logical 1 the instruction is ENT, whereas if the complementary code is logical 0 the instruction is ENTSR and the base register must be base register 0 (BRO). The address syllable AS 1104 occupies bit positions 12 thru 31 and provides the address syllable AS of the procedure descriptor 1110. When an ENTER PROCEDURE INSTRUCTION requires a ring crossing a gating procedure descriptor 1120 is obligatorily accessed. This is indicated by the GS field 1302 of segment descriptor 1301 being set to logical 10. Generally the GS field is set to 10 when one of the ENTER PROCEDURE INSTRUCTIONS is utilized. As described in the application No. 21630/76, Serial No. 1,465,344, the segment descriptor is utilized to point to the base of the segment desired, in this instance the segment 1300 containing gate procedure descriptors GPD 1120. The first word of the segment 1300 containing the gating procedure descriptors (GPD's) is formatted as shown in Figure 11c. The TAG 1121 occupies bit positions 0 and 1 and must indicate a fault descriptor i.e. the TAG field must be set to logical 11. The Caller's Maximum Ring Number CMRN 1122 occupies bit positions 2 and 3, and indicates the maximum ring from which a calling procedure through the gated procedure descriptor GPD is legal. A call violation exception is generated if the caller's ring number is greater than CMRN 1122. The gated procedure descriptor address boundary GPDAB 1124 occupies bit positions 10 through 31 and it must be greater than the segment relative address SRA (i.e. the GPD's displacement in the segment of procedure descriptors 1300), otherwise an illegal GPD access exception occurs. Thus a gating procedure descriptor GPD is utilized as the first word of the segment containing procedure descriptors and is utilized to determine whether the caller has a right to access the segment via the caller's maximum ring number CMRN and whether or not the procedure descriptor called is within the gating procedure descriptor's address boundary. Once it is determined that there is a legal call to the segment and the caller has a right to enter the segment the address is obtained from the address syllable AS 1104 of enter instruction 1100 and the required procedure descriptor 1110 (see also Figure 13) is accessed. The format of procedure descriptor 1110 is shown on Figure 11b and is comprised of two 32 bit words—word 0 and 1 respectively. Word 0 contains the segmented address 1113 of the entry point EP of the procedure desired. The segmented address, as is the case with the segmented address of any operand, is comprised of the segment number SEG and the segment relative address SRA. Word 0 of the procedure descriptor includes an entry point ring number EPRN 1112 and a TAG field 1111. The value of the TAG is interpreted as follows:

a. if the TAG contains logical 00 the procedure descriptor is direct;

b. if the TAG is logical 01 the procedure descriptor is an extended descriptor and includes word 1 making a total of two words;

c. if the TAG is logical 10 the procedure descriptor is indirect and an illegal procedure descriptor exception occurs; and

d. if the TAG is logical 11 it is a fault procedure descriptor and an exception occurs.

Word 1 of the procedure descriptor is 32 bits long and is utilized when the TAG indicates an extended descriptor and contains the segmented address of a linkage section whose contents are loaded in base register BR 7 at procedure entry time.

Referring to Figure 12 a portion of the ENT instruction is shown and more specifically that portion which pertains to the ring crossing and ring checking requirements. The ENT instruction is called, 1201 and a comparison is made 1202 wherein the segmented part of the base register BRn is compared to the segmented part of the address of the T register, and if they are not equal an illegal stack base register 1208 is indicated. If on the other hand they are equal another comparison 1203 is made wherein the 30th bit including the next two bits (i.e. bits 30 and 31) of base register, BRn is compared to 0 and if it is not equal to 0, then once again an illegal stack base register 1208 is indicated. If it is equal to 0 it indicates that the contents of BRn is aligned with respect to the word boundary and another comparison 1204 is performed to determine that the TAG of BRn (i.e. the two bits starting from bit 0) is equal to 0. A TAG having a logical 0 indicates information is accessed via a direct descriptor which is one of the requirements of the ENT instruction. If the TAG (i.e. bits 0 and 1 of BRn) is equal to 0 then the functions stated in flow charts of Figures 14 through 16 are performed (see flow chart Figure 12 block 1205). If these meet the necessary requirements a further check 1206 is made to determine whether the segment relative address of the entry point which was given (SRA_{EP}) is even, because instructions start on a half-word boundary. If it is not even then an illegal branch address exception is generated 1209 however if it is legal the ENT instruction is executed 1207 via further steps not shown.

Referring now to the flow charts of the access checking mechanism Figures 14—16, generally the following operations are performed each time the instruction ENTER PROCEDURE is issued:

a. the caller's right to call the callee is checked by first determining from the second word of the segment descriptor the call bracket in which the caller is executing. (The call bracket is determined by taking the minimum ring number from the write ring number field WR and the maximum ring number from the maximum ring number field MAXR).

b. a decision is made about the next process ring number by determining whether the caller is in the same call bracket as the callee, which implies don't do anything; whether the caller is in a call bracket requiring that he make an outward call in which case an exception condition is generated which is handled by a mechanism not described herein; or finally whether the caller is in a call bracket which requires an inward call (i.e. going to a call bracket which requires ring crossing from a larger ring number to a smaller ring number in which case the ring crossing must be at a valid entry point EP and the entry point must be validated).

c. a stack frame is created for the callee (i.e. space in the aforementioned format of the appropriate segment is allocated), and the stack frame and the stack frame registers are updated;

d. a branch to the entry point of the procedure pointed to by the procedure descriptor is performed.

Referring now to Figure 14 the access checking is started 1401 by obtaining the address syllable AS containing the effective address ring number EAR, the segment number of the procedure descriptor SEG_{PD}, and the segment relative address of the procedure descriptor SRA_{PD}. Having developed this information the procedure descriptor 1110 is fetched 1403 from (SEG_{PD}, SRA_{PD}) ignoring access rights to scratch pad memory. The procedure descriptor 1110 will yield the TAG which determines whether the descriptor is direct, extended, indirect, or a fault descriptor; the entry point ring number EPRN; the segment (SRA_{EP}) which contains the entry point and the segment relative address (SRA_{EP}) of the entry point. The TAG is tested 1404 to determine whether the descriptor 1110 is direct, extended, indirect or a fault descriptor by checking its field in accordance to the code hereinbefore described. Only a direct or extended procedure descriptor is legal. An indirect or fault descriptor is illegal and upon access invokes an exception mechanism not herein described. Once it is determined that a legal procedure descriptor has been accessed the actual call right checking begins at point A 1405.

Referring now to Figure 15 and continuing from point A 1405 the maximum ring number MAXR, the write ring number WR, and the execute permission bit EP of the segment containing the entry points SEG_{EP} are fetched; this information is contained in the segment descriptor for the segment containing the entry points (SEG_{EP}). The write ring number WR is compared to the maximum ring number MAXR 1503 and if the write ring number WR is greater than the maximum ring number MAXR the segment is nonexecutable and an execute violation exception 1513 occurs. If the write ring number WR is less than or equal to the maximum ring number MAXR then the execute permission bit EP is compared to logical 1 and if the EP bit is not logical 1 then once again an execute violation exception 1513 occurs; however if the EP bit is equal to one the effective address ring number EAR of the calling procedure is maximized with EPRN to give a new EAR_2,—[MAX (EAR_1 EPRN)] where EAR, is the maximum of PRN as found in the instruction counter IC, and all ring numbers in base registers and data descriptors, if any, found in the path which leads to the procedure descriptor. The effective address ring number EAR_2 is then compared 1506 to the maximum ring number MAXR of the MAXR segment descriptor of SEG_{EP} which is the maximum ring number at which a procedure may execute. If EAR_2 is greater than MAXR the procedure call is an inward call which requires that the procedure be entered by a valid entry point and the access checking operation branch to point B 1507. The following checking operations are then performed:

a. the SEG_{EP} is checked to determine if it is a legal gate segment; and,

b. the caller's maximum ring number CMRN is checked to determine if it is greater than or equal to the effective address ring number EAR of the caller.

If these conditions are not true then an illegal gate segment exception 1603 or call violation exception 1615 occurs.

Referring now to branch point B 1507 of Figure 16 the first check 1602 that is made is to determine whether or not the segment which contains the procedure descriptors is a gate segment. This is done by examining the Gating/Semaphore field GS of the segment descriptor pointing to the segment of procedure descriptors, to determine if it is set to logical 10. If the GS field of the segment descriptor of the segment containing procedure descriptors is set to 10 it is then a gate segment and the first word of the segment containing procedure descriptors is a gated procedure descriptor GPD 1120 of Figure 11C and Figure 13. The first word 1120 of the segment containing procedure descriptors is then fetched from address SEG_{PD}, 0 ignoring access rights to scratch pad memory. It will be noted that the TAG field of the first word 1120 of the segment containing procedure descriptor SEG_{PD} 1300 must be a logical 11 (Figure 13) which indicates it is a fault descriptor. Moreover the MBZ field must be set to zero. These conditions are checked by hardware/firmware (arithmetic logic unit) stop 1605 and if these conditions do not hold an illegal gate segment exception 1603 results. However if these conditions do hold a check 1606 is further made to determine that the segment relative address of the procedure descriptor SRA_{PD} 1110 is a multiple of 8. If the condition of step 1606 does not hold an illegal system object address exception 1613 results otherwise the next step 1607 is performed. Step 1607 checks to determine whether or not the segment relative address of the procedure descriptor SRA_{PD} is within the address boundary GPDAB 1124 of the gated procedure descriptor 1120; if it is not within that address boundary it is an illegal procedure descriptor and an illegal GPD gated procedure descriptor access exception 1614 occurs. However if it is within the address boundary of the gated procedure descriptor (i.e. SRA_{PD} is less than GPDAB) then the caller's right to call the callee is checked 1608. This is performed by comparing the effective address ring number EAR_2 to the caller's maximum ring number CMRN 1122 as found in the first word 1120 of the segment of procedure descriptors 1300. If EAR_2 is greater than the caller's CMRN a call violation exception 1615 occurs which indicates that the caller in this particular instance has no right to legally call inward i.e. from a higher ring number to a lower ring number. On the other hand if EAR_2 is equal or less than CMRN, then the inward call is legal and a check is made 1609 to determine that the process ring number PRN which is the current process ring number found in the instruction counter IC just before the call was made is less than the maximum ring number MAXR of SEG_{EP}; and if it is the accessing mechanism branches to point C 1508, otherwise a new process ring number NPRN is calculated and set to a maximum ring number MAXR 1611. Generally the effective address ring number EAR_2 is the same as the process ring number PRN of the caller. Sometimes however, in cases where it is necessary to give maximum assurance that the caller will not be denied access to a given segment the EAR_2 is greater than the PRN. In those cases 1 RN is forced to take the value of EAR_2 in order to make sure that the call is returned to the maximum ring number upon an exit. To this point it will be noted that this checking mechanism was invoked because the EAR_2 was greater than the MAXR hence greater than the top of the call bracket of the procedure and hence an inward call was necessary which necessitated going through a valid gate, and the mechanism included these gating checks. By branching back to C 1508 (Figure 15) a further check 1509 is made to determine then that the process ring number PRN is greater than the write ring number WR of SEG_{EP} which in this context is the minimum ring number at which a procedure may execute. If the write ring number WR is greater than the process ring number PRN an outward call exception 1514 occurs. However if WR is less than or equal to PRN the call is legal and NPRN is set to PRN 1510.

Having made the above checks the inward call is made, and after performance of the desired operation a return back to the original point of the program in execution is made by the EXIT INSTRUCTION. During the ENTER INSTRUCTION the instruction counter IC was saved in the saving area of the caller's stack frame before making the call. Moreover the caller's ring number was also saved during the ENTER INSTRUCTION and this was saved in base register 0 BRO.

The format of the EXIT INSTRUCTION 1130 is shown on Figure 11D. The operation code OP 1131 is found in bit positions 0—7 and the complementary code C 1133 is found in bit positions 12—15. The complementary code allows other instructions to use the same 8 bit op code. The MBZ field 1132 in bit positions 8—11 must be 0 otherwise an illegal format field exception occurs. (BRO is generally a pointer to the communications area of the caller's stack frame).

In performing the EXIT INSTRUCTION it is necessary to perform predetermined checks in order to ascertain that the caller didn't change his image which would permit him to operate a a different privilege than was intended. Referring to Figure 17 the first check performed 1701 is to determine if the TAG of the instruction counter content (ICC) indicates a direct descriptor. A logical 00 in the TAG field indicates that it is direct if it is not an illegal stack data exception 1702 occurs, whereas if it is equal to 0 the ring field in the instruction counter content ICC is set to the new process ring number NPRN 1703. This sets the new process ring number NPRN to what it used to be when the call was first made. However further checks are made in order to ascertain that there was no further cheating. Hence the base register 0 ring number located at bit position 2 and extending for 2 bit positions from and including bit position 2 must be equal to the new process ring number NPRN 1704. (It will be recalled that when the ENTER INSTRUCTION was called the ring number of the caller before the call was made was stored in bits 2 and 3 of base register 0 (BRO). If check 1704 indicates that the new process ring number NPRN is not equal to the ring number in bit positions 2 and 3 of the base register 0 (BRO) an illegal stack data exception 1702 occurs. The next check 1705 determines whether an inward or an outward return must be performed. Since an inward call was previously performed an outward return is implied in order to reach the original point from which the procedure was called. Moreover since the invention does not permit an outward call there is never a necessity to return inward. Hence the new process ring number NPRN is compared to the process ring number PRN 1705, and if NPRN is less than PRN an inward return is implied and an inward return exception 1706 is generated. However if check 1705 is passed successfully (i.e. NPRN is greater or equal to PRN) then a check is made to determine that a return is made to the segmented address SEGr that called the procedure and a return to the call bracket of the calling procedure is made and moreover that the execute bit EP is set. This is performed by fetching the segment descriptor SEGr of the calling procedure 1707 and making checks 1709, 1711, 1712. In performing checks 1709, 1711, 1712, check 1709 and 1711 determine that the new process ring number NPRN is greater than the minimum ring number WR but less than the maximum ring number MAXR (i.e. that the ring number is in the call bracket of the calling procedure where it should be). Finally check 1712 makes sure that the execute permission bit EP is set to 1. Thus a full cycle is concluded a call was performed via an ENTER INSTRUCTION; the required operation or processing was performed via the called procedure; then a return via an EXIT INSTRUCTION to the calling procedure was performed.

Having shown and described the preferred embodiment of the invention, those skilled in the art will realize that many variations of modifications can be made to produce the described invention and still be within the scope of the claimed invention.

Glossary of Terms

JOB—The job is the major unit of work for the batch user. It is the vehicle for describing, scheduling, and accounting for work he wants done.

JOB STEP—A smaller unit of batch work. It is generally one step in the execution of a job consisting of processing that logically belongs together.

TASK—The smallest unit of user-defined work. No user-visible concurrency of operation is permitted within a task.

PROGRAM—A set of algorithms written by a programmer to furnish the procedural information necessary to do a job or a part of a job.

PROCESS GROUP PLEX—The system's internal representation of a specific execution of a job.

PROCESS GROUP—A related set of processes, usually those necessary for performance of a single job step.

PROCESS—The controlled execution of instructions without concurrency. Its physical representation and control are determined by internal system design or convention.

Glossary of Terms (cont.)

PROCEDURE—A named software function or algorithm which is executable by a computational processor without concurrency. Its physical representation (code plus associated information, invocation, and use are determined by internal system or designed convention).

LOGICAL PROCESS—The collection of hardware resources and control information necessary for the execution of a process.

ADDRESS SPACE (SEGMENTATION)—The set of logical addresses that the CPU is permitted to transform into absolute addresses during a particular process. Although a processor has the technical ability of addressing every single cell of timing memory, it is desirable to restrict access only to those cells that are used during the process associated with the processor.

LOGICAL ADDRESS—An element of the process address space such as for example segment number SEG and Displacement D.

BASIC ADDRESS DEVELOPMENT—A hardware procedure which operates on a number of address elements to compute an absolute address which is used to refer to a byte location in core.

PROCESS CONTROL BLOCK—A process control block PCB, is associated with each process and contains pertinent information about its associated process, including the absolute address of tables defining the segment tables the process may access.

J. P. TABLES—A collection of logical addresses for locating a process control block associated with a process.

SEG$_{PD}$—The segment which contains the procedure descriptor.

SEG$_{EP}$—The segment which contains the entry point, as found in the procedure descriptor.

PRN—The process ring number, found in the instruction counter IC just before the call, or calculated by the ENTSR instruction.

EAR—The effective address ring number which is the maximum of:
(a) the process ring number PRN as found in the IC: or
(b) all ring numbers in the base register and data descriptors (if any) found in the path which leads to the procedure descriptor from the call instruction, including the entry point ring number EPRN located in the procedure descriptor itself.

MAXR—The maximum ring number at which a procedure may execute; MAXR is found in the segment descriptor of SEG$_{EP}$.

WR—The minimum ring number at which a procedure may execute; WR is found in the segment descriptor of SEG$_{EP}$.

EP—Execution permit bit found in the segment descriptor of SEG$_{EP}$.

CMRN—The caller's maximum ring number, as found in the first word of the segment SEG$_{PD}$ if this segment is identified as a gate segment (i.e. with the code "gate" set).

NPRN—New process ring number.

EPRN—Entry point ring number (found in the process procedure descriptor).

Addendum

Signal Name	Type	Function
(1) WSCLR	Control	Clears register to which it is connected.
(2) PDARG	Control	Clock Signal PDA.
(3) PDURGIT	Connecting	Pin connected to PDA at one end and resistor at the other.
(4) UWOBK	Connecting	Expands inputs to UW register.
(5) UWHOL	Control	Holds information in register to which it is connected.
(6) UW1BK	Control	Same as UWOBK but is connected to different input terminal of UW register.
(7) UW00000		Reset terminal of one flip-flop of register UW.
(8) UW00010		Set terminal of flip-flop of register UW.
(9) UW00100 UW00110		Same as 7+8 but different flip-flop.
(10) UVSPS	Control	Spare Control Input.

Addendum (cont.)

Signal Name	Type	Function
(11) UVSPD	Data	Spare Data Input.
(12) UVOBK	Expander	Same as UW0BK and UW1BK, but it connects different registers and gates.
(13) UV00000 UV00010 UV00100 UV00110		Same as UW00000, UW00010, UW00100, UW00110, but applies to flip-flop UV.
(14) UWV1S	Control	Control input for UWV1F.
(15) UWV1D	Data	Data input for UWV1F.
(16) UWV2F	F/F	Write control flip-flop.
(17) UWV1S UWV2S	Control	Control unit for UWV1F, UWV2F.
(18) UWV1D	Data	Data input for UWV1F.
(19) UWV1H	Control	Hold UWV1F flip-flop.
(20) UWV1C	Control	Clear UWV1F.
(21) UWV2C	Control	Clear UWV2F.
(22) URN1S URN2S	Control	Control inputs for URN1F, URN2F.
(23) URN1D	Data	Data Input for URN1F.
(24) URNSW	Control	Transfer URN1F to URN2F and URN2F to URN1F.
(25) URN2F	F/F	Control loading max (UP, UBS2, 3 to UM).
(26) URN1H	Control	Hold URN1F flip-flop.
(27) URN2C	Control	Clear URN2F.
(28) URW1S URW2S	Control	Control inputs for URV1F, URV2F.
(29) URW1D	Data	Data Input for URV1F.
(30) URV2F	F/F	Read control flop.
(31) XNU		Indicates terminal not used herein.
(32) XOO		Grounded Input.

WHAT WE CLAIM IS:—

1. An internally programmed data processing apparatus CPU having a virtual memory system, and being responsive to internally stored instruction words for processing information and having stored in said virtual memory system a plurality of different types of groups of information each information group-type associated with an address space bounded by a segment having adjustable bounds, and comprising means for protecting the information in said-virtual memory system from unauthorized users by restricting accessability to the information in accordance to levels of privilege, said means comprising in combination with an access checking mechanism;

(a) first means arranged in operation to store in said virtual memory system at least one segment table comprising a plurality of segment descriptors with each segment descriptor being associated with a predetermined one of said segments and each segment descriptor having a predetermined format containing an access information element and a base address element in predetermined positions of said format, said base address element being used for locating in said virtual memory system the starting location of a selected one of said segments, and said access information element for specifying the minimum level of privilege required for a predetermined type of access that is permitted in a selected one of said segments;

(b) a plurality of second means having a predetermined format, communicating with said first means, arranged to store in a predetermined portion of said second means, a segment number SEG for identifying a segment table and the location of a segment descriptor within said segment table, said second means also being arranged to store in a predetermined other portion of said second means, an offset address within the segment identified by said segment descriptor said offset address locating from said segment base the first byte of a word within said segment;

(c) third means responsive to an address syllable element of an instruction being executed for addressing one of said plurality of second means;

(d) fourth means arranged to store a displacement from said address syllable,

(e) fifth means, communicating with said first, second, third and fourth means, arranged to add the displacement D and said base address to said offset; and,

(f) sixth means responsive to said access information element in a selected one of said segment descriptors, restricting the accessability to the segment associated with said selected one of said segment descriptors in accordance to the level of privilege and the type of access specified in said access information element, wherein each group-type of information is associated with a predetermined ring number indicative of a level of privilege said level of privilege decreasing as the associated ring number increases comprising means for determining the maximum effective address ring number EAR (i.e. minimum level of privilege) of a selected process to access a selected group of information, said means comprising;

(a) first means to store first information indicating the maximum ring number RD (i.e. minimum level of privilege) required to read information from said selected group;

(b) second means to store second information indicating the maximum ring number WR (i.e. minimum level of privilege) required to write information into said selected group;

(c) third means to store third information indicating the maximum ring number MAXR (i.e. minimum level of privilege) required to process information from said selected group; and,

(d) fourth means communicating with said first, second and third means, to determine the maximum of the contents of said first, second and third means whereby the effective address ring number EAR is generated.

2. Apparatus according to claim 1, wherein said second means for storing the maximum ring number WR additionally indicates the minimum ring number WR (i.e. maximum level of privilege) required to process information from said selected group.

3. Apparatus according to claim 1 or claim 2, wherein said fourth means to generate the effective address ring number comprises a comparator for comparing binary numbers.

4. Apparatus according to any one of claims 1 to 3 wherein the sixth means restricting the accessibility to the segment includes comparator means, communicating with said second means, to compare the effective address ring number EAR with the write ring number WR, and further including means communicating with said comparator means to generate a write-violation-exception signal when EAR is greater than WR.

5. Apparatus according to claim 4, wherein the sixth means restricting the accessibility to the segment includes seventh means, communicating with said second and third means thereof to compare the maximum ring number MAXR and the write ring number WR with the effective address ring number EAR, and further including eighth means, communicating with said seventh means for generating an execute-violation-exception signal when the MAXR is not equal or greater than EAR which in turn is not equal or greater than WR.

6. Apparatus according to claim 5, wherein in that the sixth means restricting the accessibility to the segment includes ninth means, communicating with said first means, for comparing the effective address ring number EAR with the read ring number RD, and further including tenth means, communicating with said ninth means, to generate a read-violation-exeption signal when EAR is greater than RD.

7. Apparatus according to claim 6, wherein in that the sixth means restricting the accessibility to the segment includes eleventh means to store a process ring number PRN of a currently executing process, and also including twelfth means to communicate with said eleventh means, and further including thirteenth means communicating said said twelfth means for overriding said read-violation-exception signal when the effective address ring number EAR is equal to the process ring number PRN of the currently executing process.

8. Apparatus according to any one of the preceding claims wherein the access checking mechanism supervises transfer of control of said CPU from a first selected procedure (i.e. caller) having a first ring number indicative of a minimum level of privilege associated with said caller, to a second selected procedure (i.e. the callee) having a second ring number associated with said callee indicative of a minimum level of privilege associated with said callee, said access checking mechanism comprising

(a) first means for checking the caller's right to call the callee;

(b) second means, communicating with said first means, to compare the caller's ring number to the callee's ring number;

(c) third means responsive to said second means to permit a transfer of control of said CPU from said caller to said callee when the ring number of the caller is greater than the ring number of callee (i.e. inward call); and,

(d) fourth means also responsive to said second means to deny a transfer of control of said CPU from said caller to said callee when the ring number of said caller is less than the ring number of the callee (i.e. outward call).

9. Apparatus according to claim 8, wherein the access checking mechanism includes a plurality of ring stack-segment means each of said ring stack-segment means having associated with it a ring stack-segment number, indicative of the minimum level of privilege required by a selected one of said procedures to access a selected one of said ring stack segments.

10. Apparatus according to claim 9 wherein there are four ring stack segment means having ring numbers 0 to 3 respectively.

11. Apparatus according to claim 9 or claim 10 wherein the access checking mechanism includes stack-frame-element means associated with selected ones of said procedures, said stack-frame-element means being grouped within said ring stack-segment means in accordance with the ring number of the associated procedure of said stack-frame-element means, said stack frame element means to save said register of said caller prior to passing control to said callee.

12. Apparatus according to claim 11, wherein the access checking mechanism includes first sub-element means, responsive to said first, second, third and fourth means, for communicating between a selected one of said stack-frame-means in a first ring stack-segment being associated with one ring number, and a selected other of said stack-frame-means in a second ring stack-segment associated with another ring number.

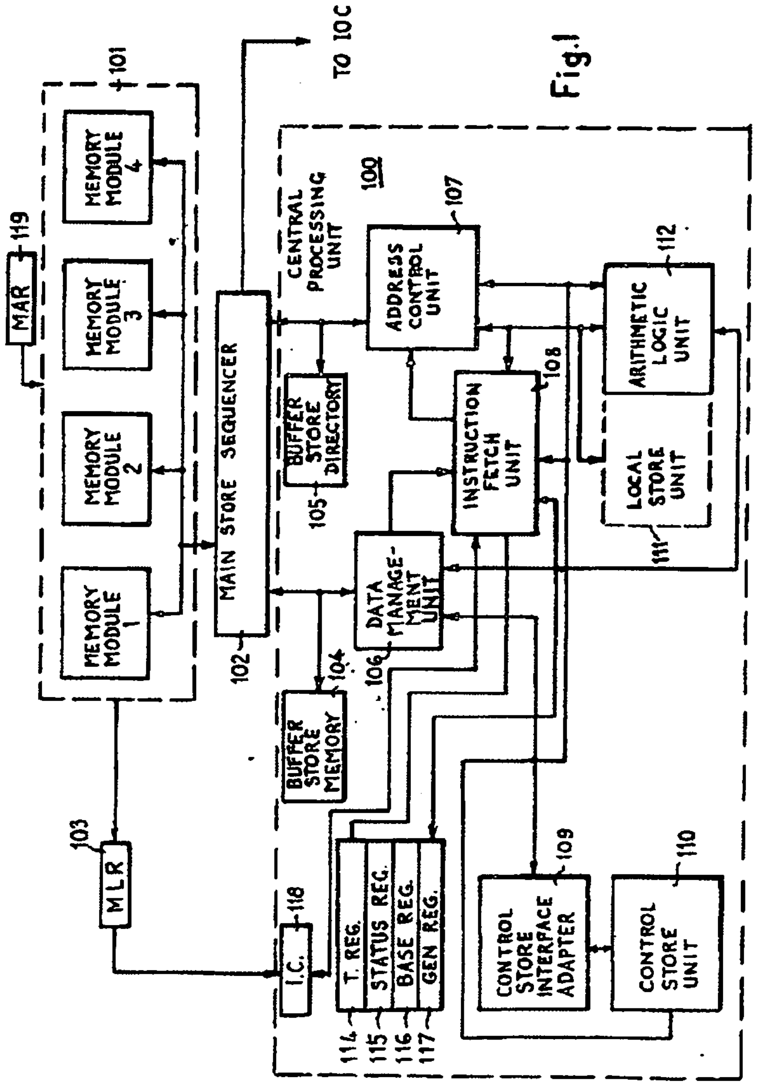
BARON & WARREN,
16, Kensington Square,
London, W8 5HL.
Chartered Patent Agents.

Printed for Her Majesty's Stationery Office, by the Courier Press, Leamington Spa, 1977
Published by The Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.

1483282 COMPLETE SPECIFICATION

29 SHEETS *This drawing is a reproduction of the Original on a reduced scale*

Sheet 1



MAR
119
101
MEMORY MODULE 4
MEMORY MODULE 3
MEMORY MODULE 2
MEMORY MODULE 1
MAIN STORE SEQUENCER
102
TO IOC
103
MLR
CENTRAL PROCESSING UNIT
100
ADDRESS CONTROL UNIT
107
BUFFER STORE DIRECTORY
105
INSTRUCTION FETCH UNIT
108
ARITHMETIC LOGIC UNIT
112
LOCAL STORE UNIT
111
DATA MANAGE-MENT UNIT
106
BUFFER STORE MEMORY
104
I.C.
118
T. REG.
STATUS REG.
BASE REG.
GEN REG.
114
115
116
117
CONTROL STORE INTERFACE ADAPTER
109
CONTROL STORE UNIT
110

Fig.1

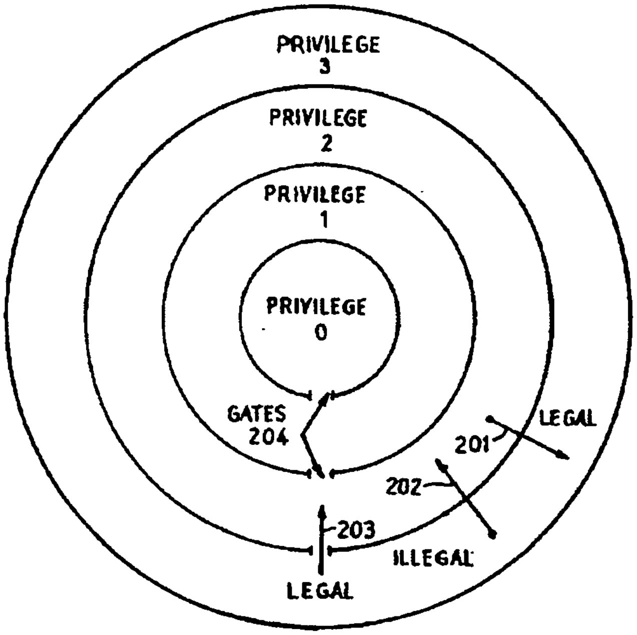

PRIVILEGE 3
PRIVILEGE 2
PRIVILEGE 1
PRIVILEGE 0
GATES 204
LEGAL
201
202
203
ILLEGAL
LEGAL

Fig.2

1483282 COMPLETE SPECIFICATION

29 SHEETS *This drawing is a reproduction of the Original on a reduced scale*

Sheet 3

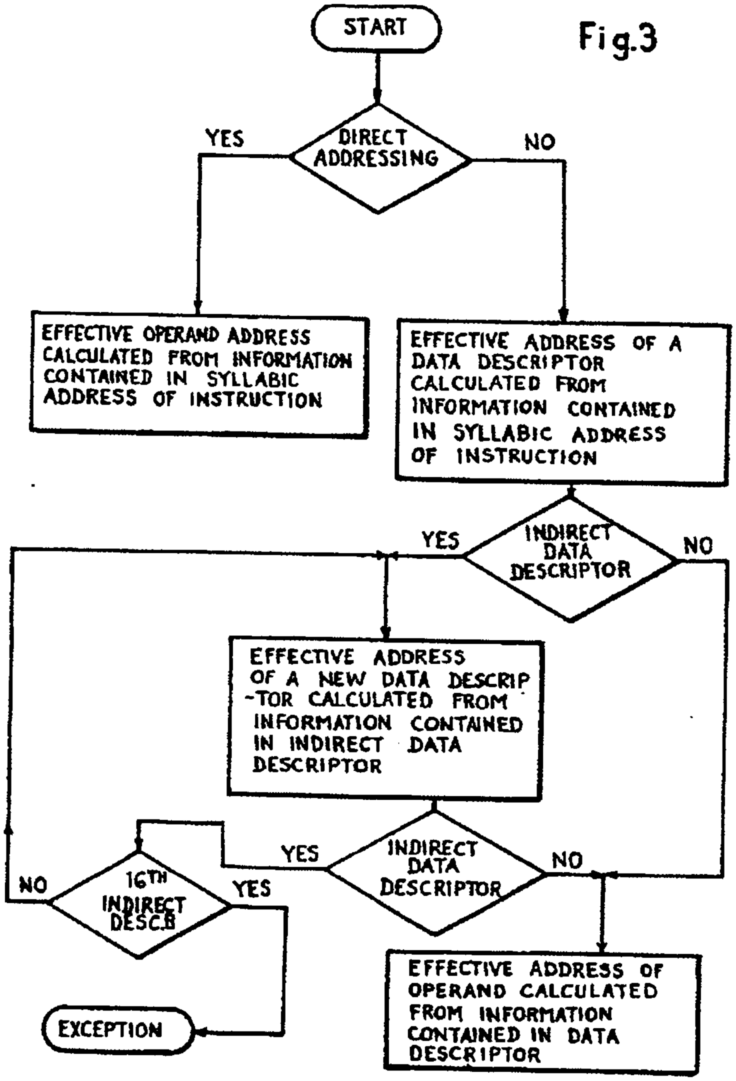

START
DIRECT ADDRESSING
YES
NO
EFFECTIVE OPERAND ADDRESS CALCULATED FROM INFORMATION CONTAINED IN SYLLABIC ADDRESS OF INSTRUCTION
EFFECTIVE ADDRESS OF A DATA DESCRIPTOR CALCULATED FROM INFORMATION CONTAINED IN SYLLABIC ADDRESS OF INSTRUCTION
YES
INDIRECT DATA DESCRIPTOR
NO
EFFECTIVE ADDRESS OF A NEW DATA DESCRIPTOR CALCULATED FROM INFORMATION CONTAINED IN INDIRECT DATA DESCRIPTOR
NO
16TH INDIRECT DESC.R
YES
YES
INDIRECT DATA DESCRIPTOR
NO
EXCEPTION
EFFECTIVE ADDRESS OF OPERAND CALCULATED FROM INFORMATION CONTAINED IN DATA DESCRIPTOR

Fig.3

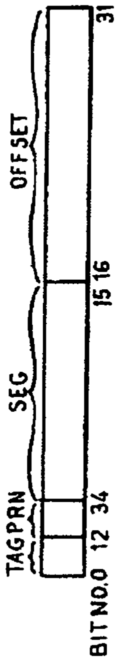

TAGPRN
SEG
OFFSET
BIT NO. 0 1 2 3 4
15 16
31

Fig.4 A

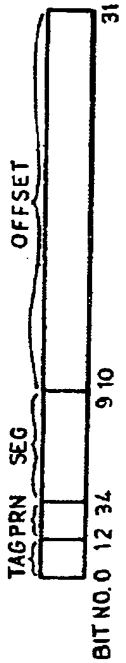

TAGPRN
SEG
OFFSET
BIT NO. 0 1 2 3 4
9 10
31

Fig.4 B

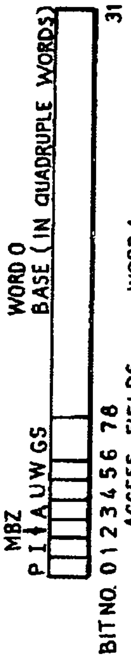

WORD 0
BASE (IN QUADRUPLE WORDS)
MBZ
P I A U W G S
BIT NO. 0 1 2 3 4 5 6 7 8
31

Fig.4C

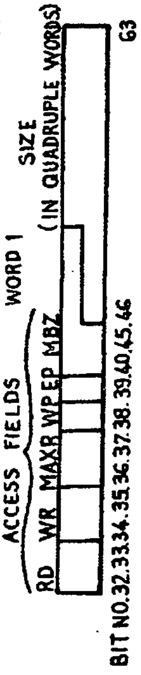

WORD 1
ACCESS FIELDS
RD WR MAXR WP EP MBZ
SIZE
(IN QUADRUPLE WORDS)
BIT NO. 32 33 34 35 36 37 38 39 40 45 46
63

Fig.4 D

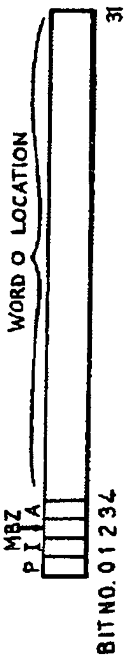

WORD 0 LOCATION
MBZ
P I A
BIT NO. 0 1 2 3 4
31

Fig.4 E

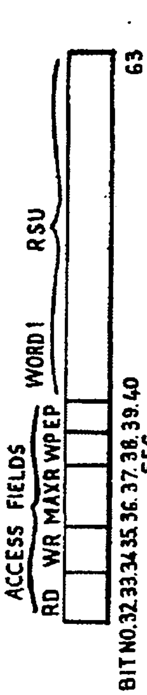

ACCESS FIELDS
RD
WR
MAXR
WP
EP
WORD1
RSU
BIT NO.32 33 34 35 36 37 38 39 40
63

Fig.4F

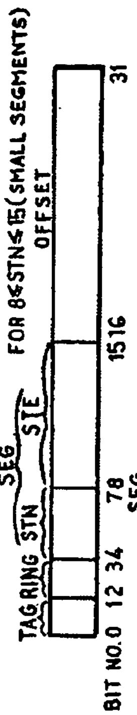

FOR 8≤STN≤15 (SMALL SEGMENTS)
OFFSET
TAG
RING
STN
SEG
STE
BIT NO.0 12 34 78 1516 31

Fig.4G

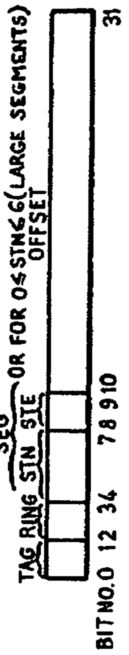

OR FOR 0≤STN≤6 (LARGE SEGMENTS)
OFFSET
TAG
RING
STN
SEG
STE
BIT NO.0 12 34 78 910 31

Fig.4H

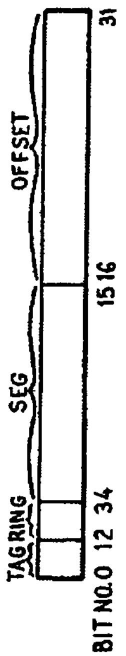

TAG
RING
SEG
OFFSET
BIT NO.0 12 34 1516 31

Fig.4I

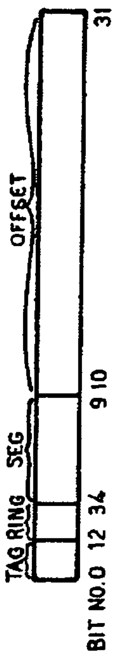

TAG
RING
SEG
OFFSET
BIT NO.0 12 34 910 31

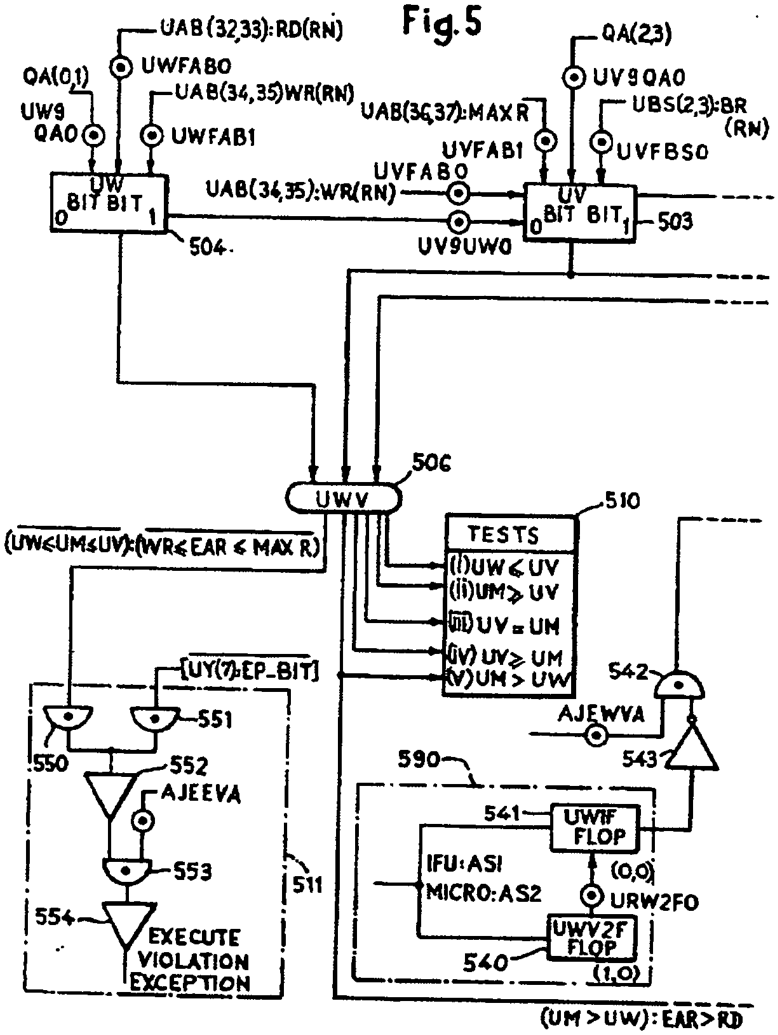
Fig.4J

1483282 COMPLETE SPECIFICATION

29 SHEETS *This drawing is a reproduction of the Original on a reduced scale*

Sheet 6

Fig.5



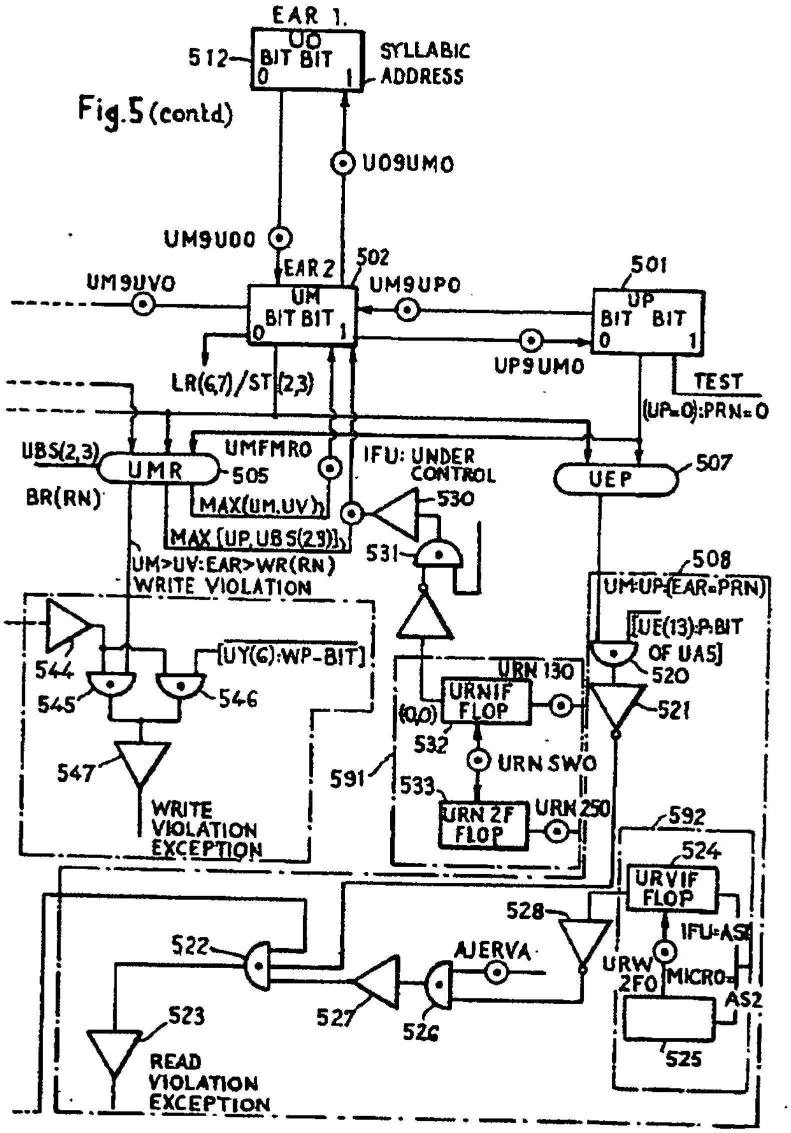
UAB(32,33):RD(RN)
UWFAB0
QA(0,1)
UW9
QA0
UAB(34,35)WR(RN)
UWFAB1
UW
0 BIT BIT 1
504
UAB(34,35):WR(RN)
UVFAB0
UV9UW0
QA(2,3)
UV9QA0
UAB(36,37):MAX R
UVFAB1
UBS(2,3):BR (RN)
UVFBS0
UV
0 BIT BIT 1
503
506
UWV
510
TESTS
(i) UW ≤ UV
(ii) UM ≥ UV
(iii) UV = UM
(iv) UV ≥ UM
(v) UM > UW
(UW≤UM≤UV):(WR≤EAR ≤ MAX R)
[UY(7):EP_BIT]
550
551
552
AJEEVA
553
554
EXECUTE VIOLATION EXCEPTION
511
542
AJEWVA
543
590
541
UWIF FLOP
IFU:AS1
MICRO:AS2
(0,0)
URW2F0
UWV2F FLOP
540
(1,0)
(UM > UW): EAR>RD

1483282 COMPLETE SPECIFICATION

29 SHEETS *This drawing is a reproduction of the Original on a reduced scale*

Sheet 7

Fig.5 (contd)



EAR 1.
UO
BIT BIT
0
1
512
SYLLABIC ADDRESS
UO9UMO
UM9UOO
EAR 2
502
UM9UVO
UM9UPO
UM
BIT BIT
0
1
501
UP
BIT BIT
0
1
UP9UMO
LR(6,7)/ST(2,3)
TEST
(UP=0):PRN=0
UBS(2,3)
UMR
505
UMFMRO
BR(RN)
MAX(UM,UV)
MAX(UP,UBS(2,3))
IFU: UNDER CONTROL
530
UEP
507
531
UM>UV:EAR>WR(RN) WRITE VIOLATION
508
UM=UP:(EAR=PRN)
(UE(13)=P-BIT OF UAS)
520
521
544
UY(6):WP-BIT
545
546
547
WRITE VIOLATION EXCEPTION
URN130
URNIF FLOP
(0,0)
532
URN SWO
591
533
URN 2F FLOP
URN250
592
524
URVIF FLOP
528
IFU=AS2
URW 2FO
MICRO=AS2
525
522
AJERVA
527
526
523
READ VIOLATION EXCEPTION

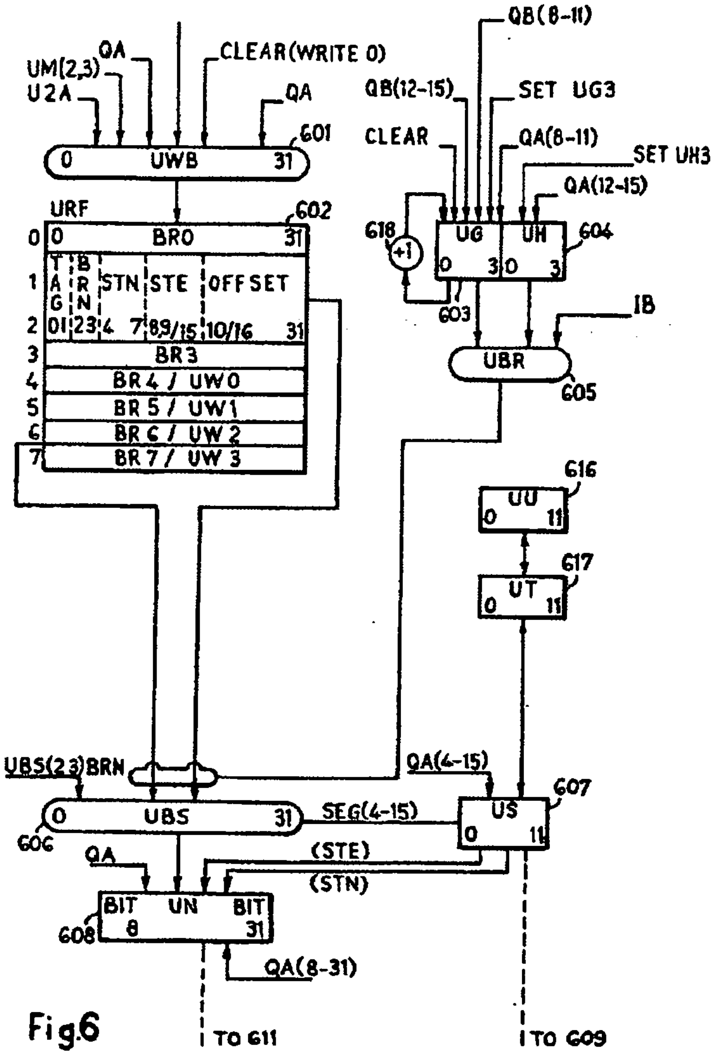

QB(8-11)
QA
UM(2,3)
U2A
CLEAR(WRITE 0)
QA
601
0 UWB 31
QB(12-15)
SET UG3
CLEAR
QA(8-11)
SET UH3
QA(12-15)
URF
602
0 BR0 31
TAG 01
BRN 23
STN 4 7
STE 89/15
OFFSET 10/16 31
BR3
BR4 / UW0
BR5 / UW1
BR6 / UW2
BR7 / UW3
0
1
2
3
4
5
6
7
618
+1
UG 0 3
UH 0 3
604
603
IB
UBR
605
616
UU 0 11
617
UT 0 11
UBS(2,3)BRN
QA(4-15)
607
0 UBS 31
SEG(4-15)
US 0 11
606
QA
(STE)
(STN)
BIT 8 UN BIT 31
608
QA(8-31)
TO 611
TO 609

Fig.6

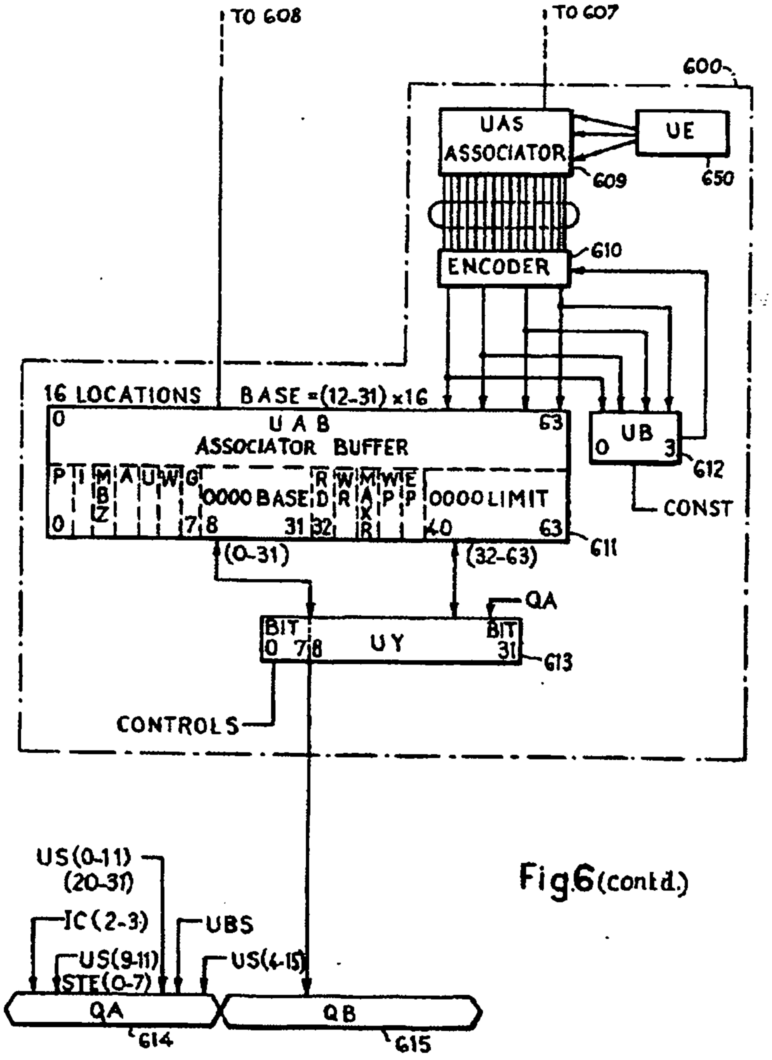

TO 608
TO 607
600
UAS ASSOCIATOR
UE
609
650
610
ENCODER
16 LOCATIONS
BASE =(12-31)×16
0
U A B
ASSOCIATOR BUFFER
63
UB
0
3
612
CONST
P
I
M B Z
A
U
W
G
0000 BASE
R D
W R
M A X R
W P
E P
0000 LIMIT
0
7
8
31
32
40
63
611
(0-31)
(32-63)
QA
BIT 0 7
8
U Y
BIT 31
613
CONTROLS
US(0-11) (20-31)
IC(2-3)
UBS
US(9-11)
STE(0-7)
US(4-15)
QA
QB
614
615

Fig.6(contd.)

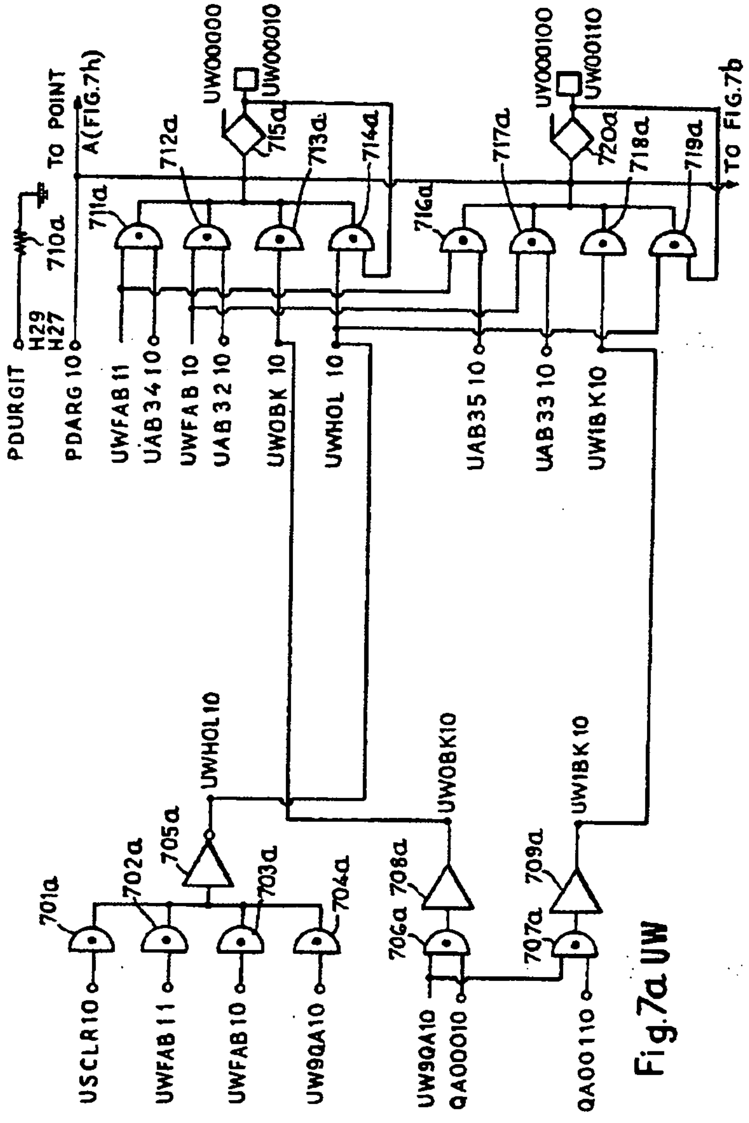

PDURGIT
H29
H27
PDARG 10
710a
TO POINT A(FIG.7h)
UWFAB 11
UAB 34 10
UWFAB 10
UAB 32 10
UWOBK 10
UWHOL 10
UAB35 10
UAB33 10
UW1BK10
711a
712a
713a
714a
715a
716a
717a
718a
719a
720a
UW00000
UW00010
UY000100
UW00110
TO FIG.7b
701a
702a
703a
704a
705a
706a
707a
708a
709a
USCLR10
UWFAB11
UWFAB10
UW9QA10
UW9QA10
QA00010
QA00110
UWHOL10
UWOBK10
UW1BK10

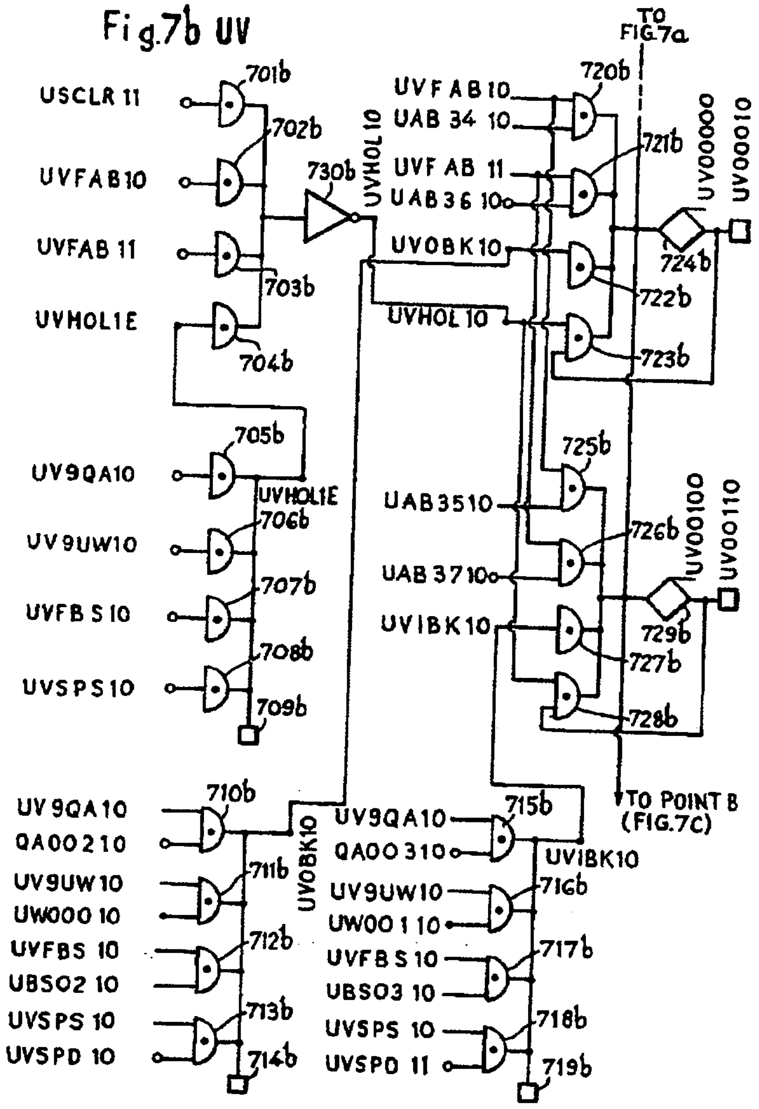
Fig.7a UW

1483282 COMPLETE SPECIFICATION

29 SHEETS *This drawing is a reproduction of the Original on a reduced scale*

Sheet 11

Fig.7b UV



TO FIG.7a
701b
USCLR 11
702b
UVFAB10
730b
UVHOL10
UVFAB 11
703b
UVHOL1E
704b
UVFAB10
UAB 34 10
UVF AB 11
UAB36 10
UV0BK10
UVHOL10
720b
721b
UV00000
UV00010
724b
722b
723b
705b
UV9QA10
UVHOL1E
706b
UV9UW10
707b
UVFBS10
708b
UVSPS10
709b
725b
UAB3510
726b
UAB3710
UV1BK10
UV00100
UV00110
729b
727b
728b
TO POINT B (FIG.7C)
710b
UV9QA10
QA00210
711b
UV9UW10
UW000 10
712b
UVFBS 10
UBSO2 10
713b
UVSPS 10
UVSPD 10
714b
UV0BK10
715b
UV9QA10
QA00310
UV1BK10
716b
UV9UW10
UW001 10
717b
UVFBS10
UBSO3 10
718b
UVSPS 10
UVSPD 11
719b

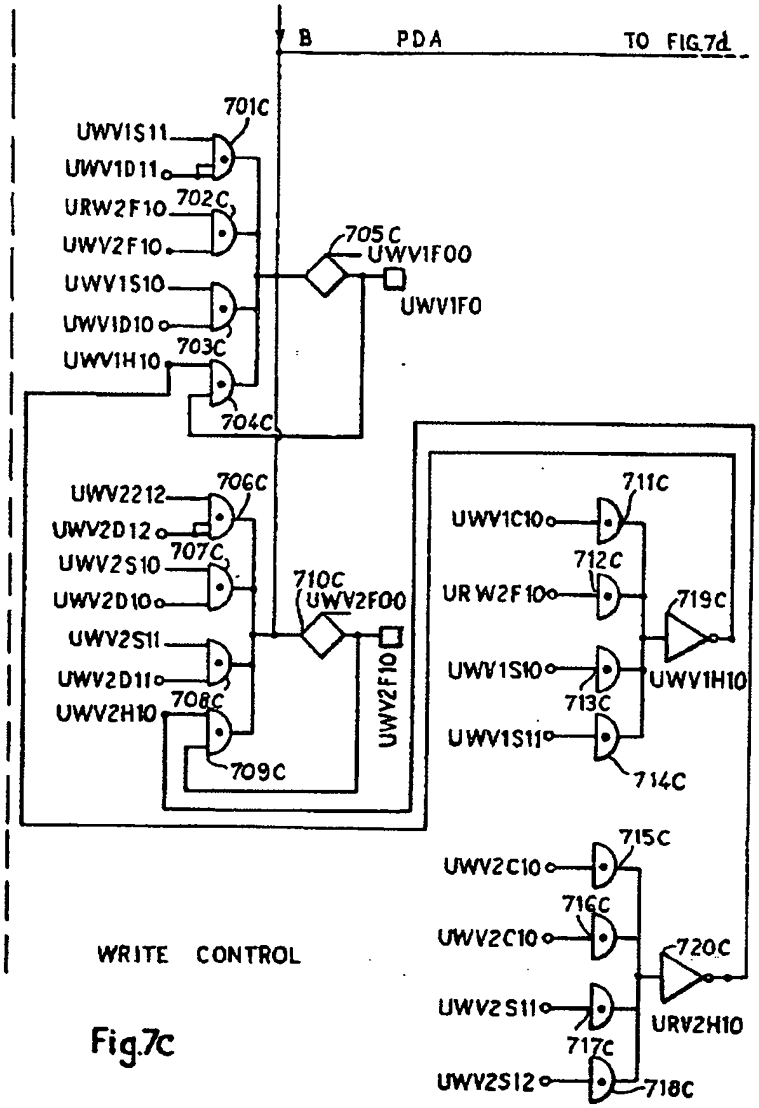

B
PDA
TO FIG.7d
701C
UWV1S11
UWV1D11
URW2F10
702C
UWV2F10
705C
UWV1F00
UWV1F0
UWV1S10
UWV1D10
703C
UWV1H10
704C
706C
UWV2212
UWV2D12
707C
UWV2S10
UWV2D10
710C
UWV2F00
UWV2S11
UWV2D11
UWV2F10
708C
UWV2H10
709C
711C
UWV1C10
712C
URW2F10
719C
UWV1S10
UWV1H10
713C
UWV1S11
714C
715C
UWV2C10
716C
UWV2C10
720C
UWV2S11
URV2H10
717C
UWV2S12
718C
WRITE CONTROL

Fig.7c

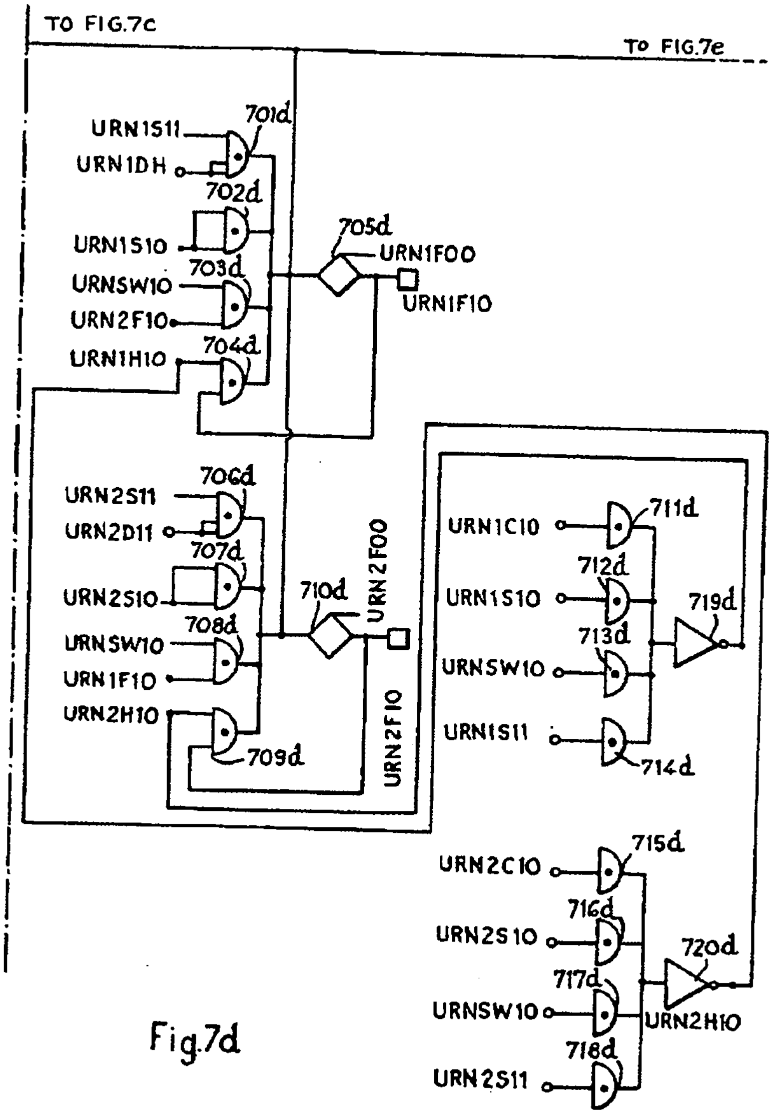

TO FIG.7c
TO FIG.7e
URN1S11
URN1DH
701d
702d
URN1S10
703d
URNSW10
URN2F10
URN1H10
704d
705d
URN1F00
URN1F10
706d
URN2S11
URN2D11
707d
URN2S10
708d
URNSW10
URN1F10
URN2H10
709d
710d
URN2F00
URN2F10
URN1C10
711d
712d
URN1S10
713d
URNSW10
URN1S11
714d
719d
URN2C10
715d
716d
URN2S10
717d
URNSW10
718d
URN2S11
720d
URN2H10

Fig.7d

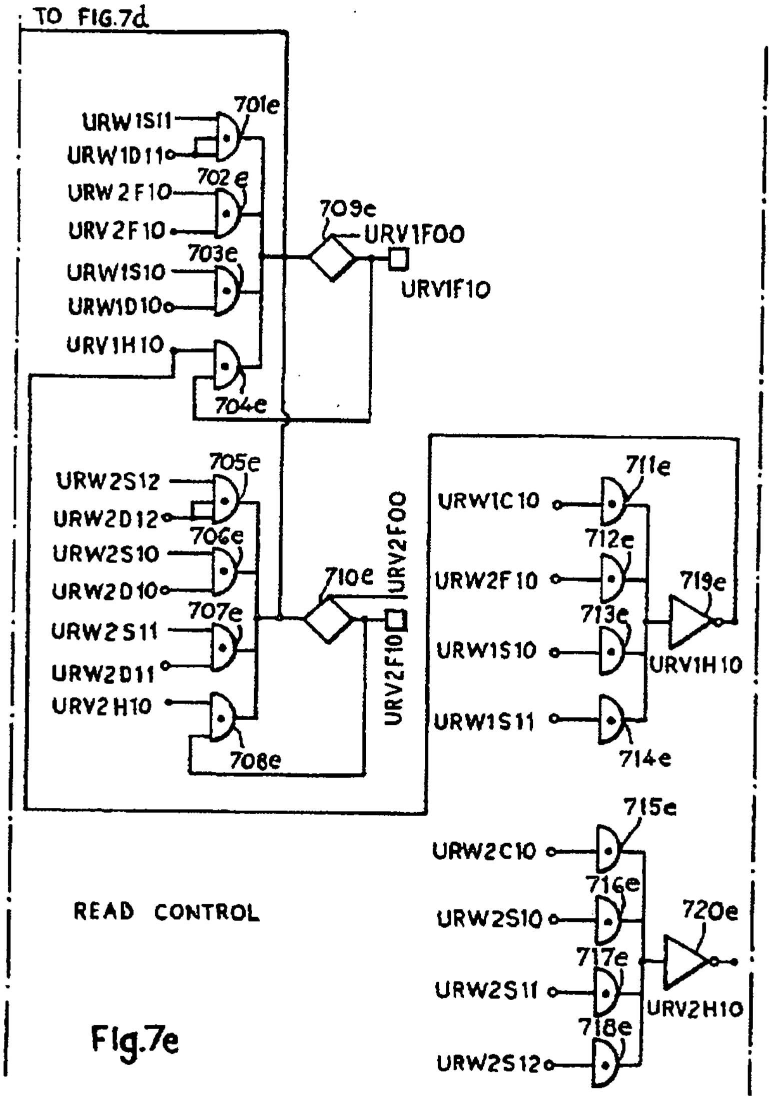

TO FIG.7d
URW1S11
URW1D11
701e
URW2F10
702e
URV2F10
709e
URV1F00
703e
URW1S10
URV1F10
URW1D10
URV1H10
704e
705e
URW2S12
URW2D12
706e
URW2S10
URV2F00
URW2D10
710e
707e
URW2S11
URV2F10
URW2D11
URV2H10
708e
711e
URW1C10
712e
URW2F10
719e
713e
URW1S10
URV1H10
URW1S11
714e
715e
URW2C10
716e
URW2S10
720e
717e
URW2S11
URV2H10
718e
URW2S12
READ CONTROL

Fig.7e

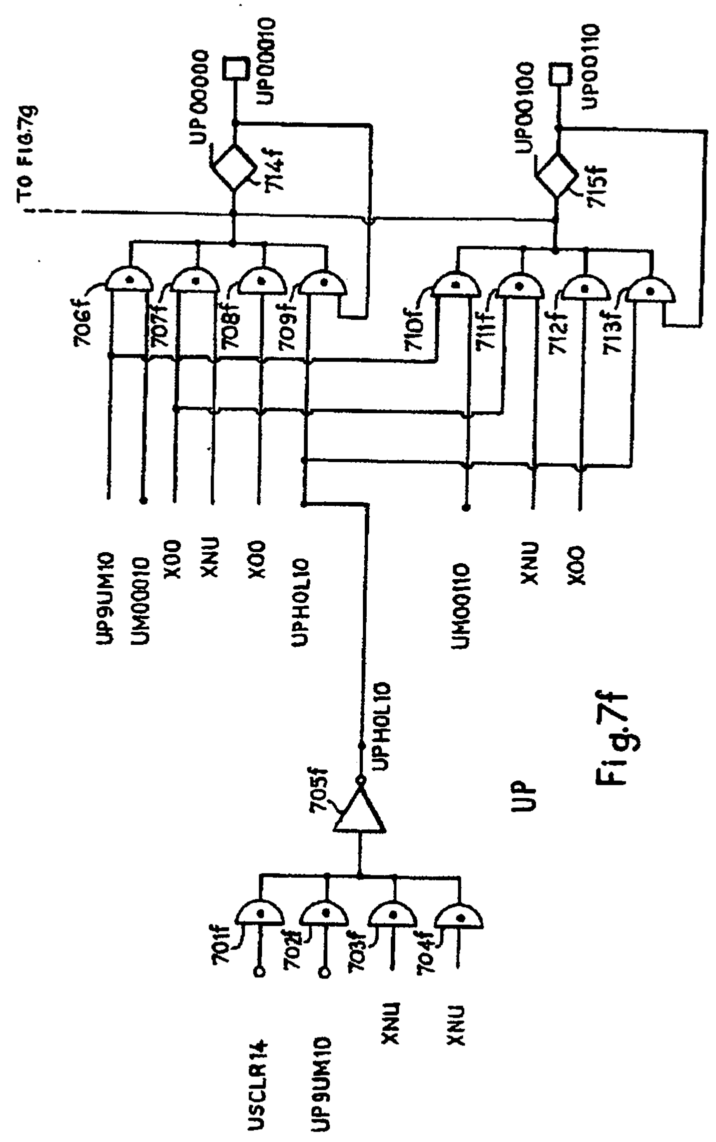

TO FIG.7g
UP00000
UP00010
714f
UP00100
UP00110
715f
706f
707f
708f
709f
710f
711f
712f
713f
UP9UM10
UM00010
X00
XNU
X00
UPHOL10
UM00110
XNU
X00
UPHOL10
705f
701f
702f
703f
704f
USCLR14
UP9UM10
XNU
XNU
UP

Fig.7f

1483282 COMPLETE SPECIFICATION

29 SHEETS *This drawing is a reproduction of the Original on a reduced scale*

Sheet 16

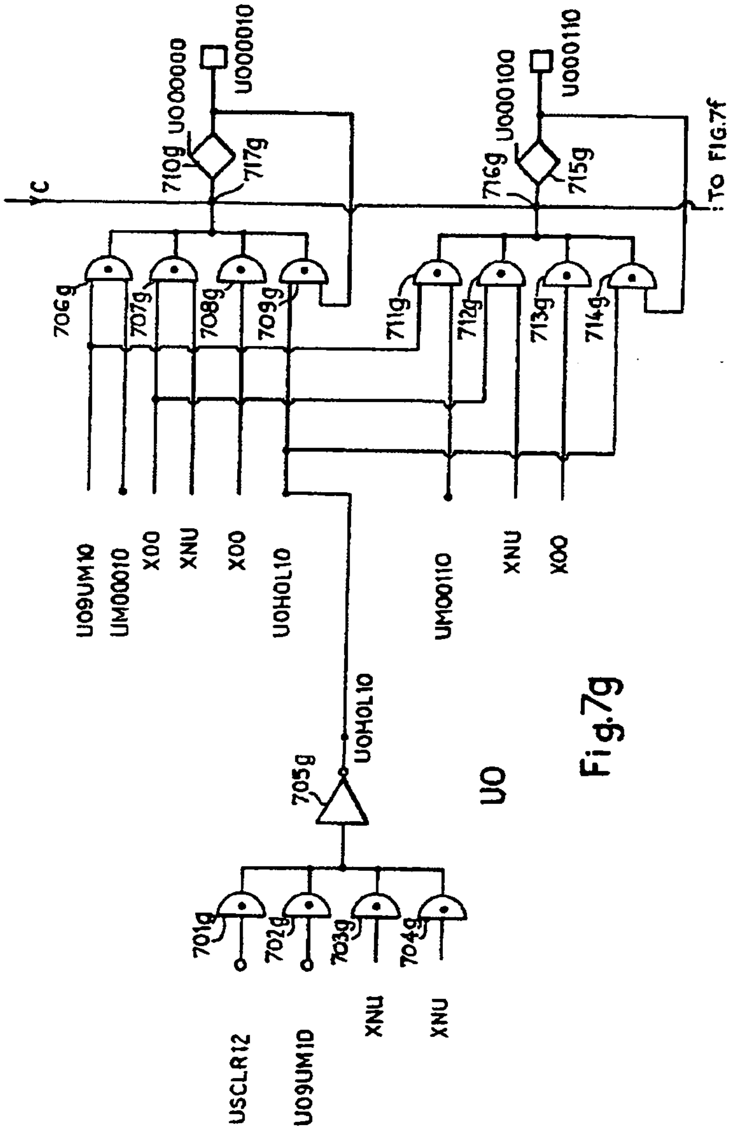
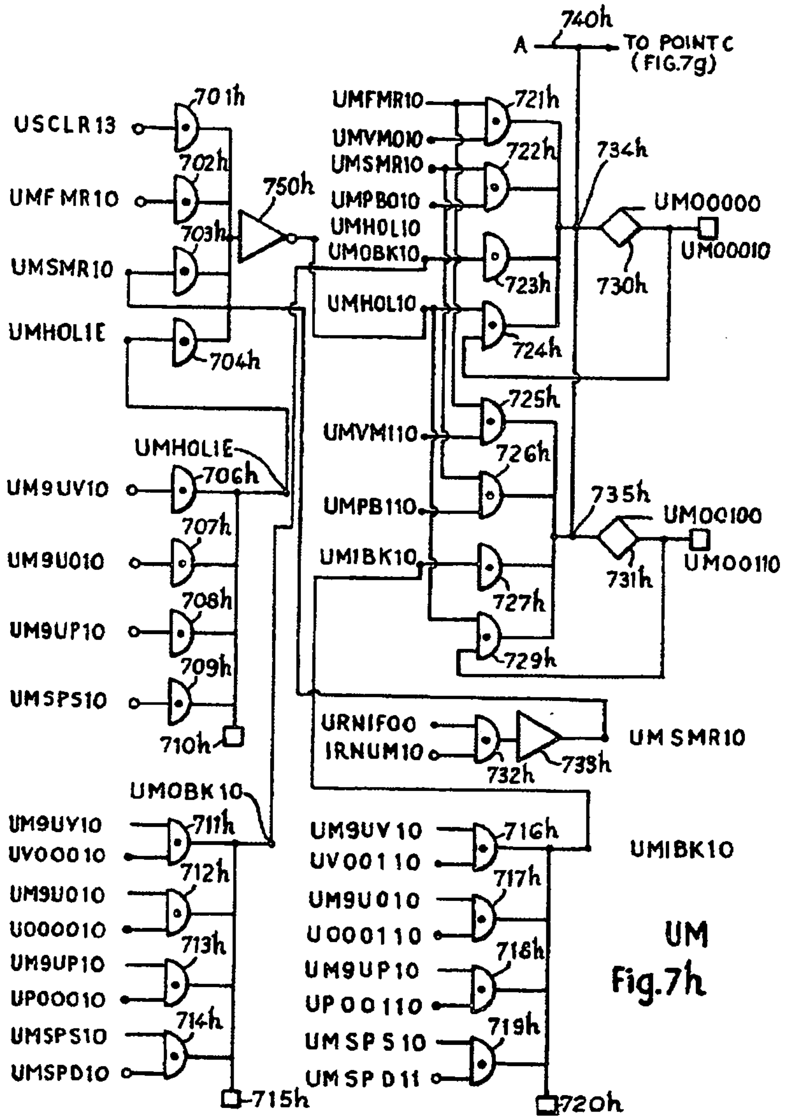

U000000
U000010
710g
717g
C
U000100
U000110
716g
715g
TO FIG.7f
706g
707g
708g
709g
711g
712g
713g
714g
U09UM10
UM00010
X00
XNU
X00
U0HOL10
UM00110
XNU
X00
U0HOL10
705g
U0
701g
702g
703g
704g
USCLR12
U09UM1D
XNU
XNU

Fig.7g

1483282 COMPLETE SPECIFICATION

29 SHEETS *This drawing is a reproduction of the Original on a reduced scale*

Sheet 17



740h
A
TO POINT C (FIG.7g)
701h
USCLR13
702h
UMFMR10
750h
703h
UMSMR10
UMHOL1E
704h
UMHOL1E
706h
UM9UV10
707h
UM9U010
708h
UM9UP10
709h
UMSPS10
710h
UMOBK10
711h
UM9UV10
UV00010
712h
UM9U010
U00010
713h
UM9UP10
UP00010
714h
UMSPS10
UMSPD10
715h
UMFMR10
721h
UMVM010
722h
UMSMR10
UMPB010
UMHOL10
UMOBK10
723h
UMHOL10
724h
734h
UM00000
UM00010
730h
725h
UMVM110
726h
UMPB110
735h
UM00100
UM1BK10
731h
UM00110
727h
729h
URNIF00
IRNUM10
732h
733h
UMSMR10
UM9UV10
716h
UV00110
UM1BK10
717h
UM9U010
U000110
718h
UM9UP10
UP00110
719h
UMSPS10
UMSPD11
720h

UM

Fig.7h

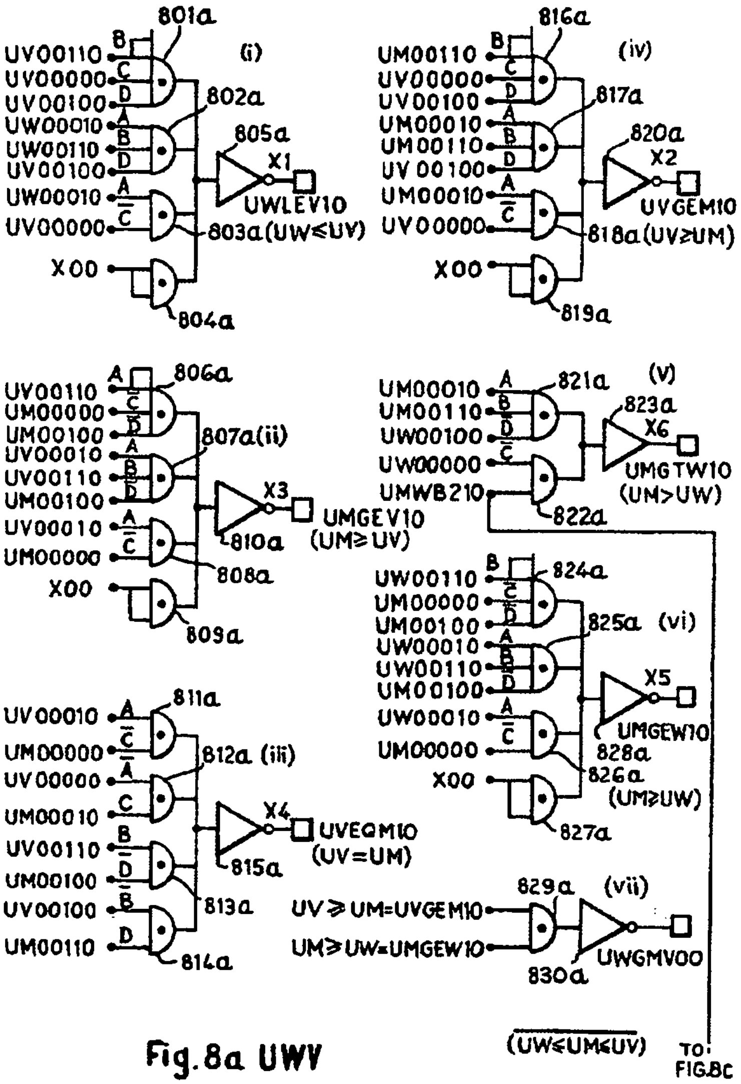

801a
(i)
UV00110
UV00000
UV00100
UW00010
UW00110
UV00100
UW00010
UV00000
X00
B
C
D
A
B
D
A
C
802a
805a
X1
UWLEV10
803a(UW≤UV)
804a
816a
(iv)
UM00110
UV00000
UV00100
UM00010
UM00110
UV 00100
UM00010
UV00000
X00
817a
820a
X2
UVGEM10
818a(UV≥UM)
819a
806a
UV00110
UM00000
UM00100
UV00010
UV00110
UM00100
UV00010
UM00000
X00
807a(ii)
X3
UMGEV10
810a
(UM≥UV)
808a
809a
UM00010
UM00110
UW00100
UW00000
UMWB210
821a
(V)
823a
X6
UMGTW10
(UM>UW)
822a
UW00110
UM00000
UM00100
UW00010
UW00110
UM00100
UW00010
UM00000
X00
824a
825a
(vi)
X5
UMGEW10
828a
826a
(UM≥UW)
827a
UV00010
UM00000
UV00000
UM00010
UV00110
UM00100
UV00100
UM00110
811a
812a
(iii)
X4
UVEQM10
(UV=UM)
815a
813a
814a
829a
(vii)
UV≥UM=UVGEM10
UM≥UW=UMGEW10
UWGMV00
830a
$\overline{(UW\leq UM\leq UV)}$
TO FIG.8C

Fig.8a UWV

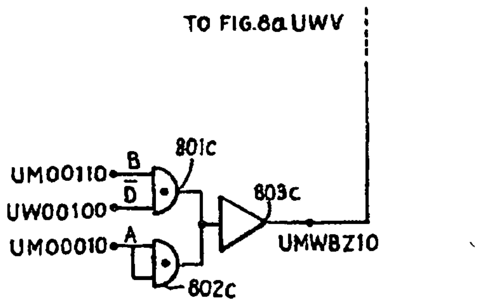

TO FIG.8a UWV
801c
UM00110
B
D̄
UW00100
UM00010
A
803c
UMWBZ10
802c

Fig.8c

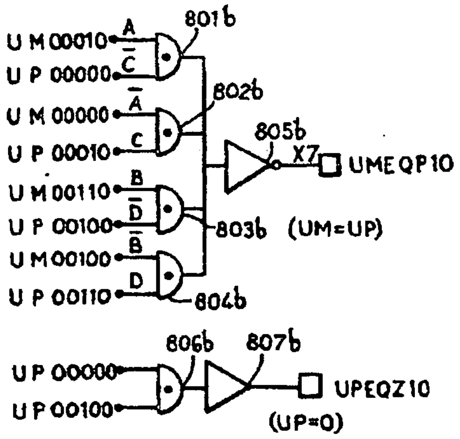

801b
U M 00010
A
C̄
U P 00000
802b
U M 00000
Ā
C
U P 00010
805b
X7
UMEQP10
U M 00110
B
D̄
U P 00100
803b
(UM=UP)
U M 00100
B̄
D
U P 00110
804b
806b
807b
U P 00000
U P 00100
UPEQZ10
(UP=0)

Fig.8b UEP

1483282 COMPLETE SPECIFICATION

29 SHEETS *This drawing is a reproduction of the Original on a reduced scale*

Sheet 20

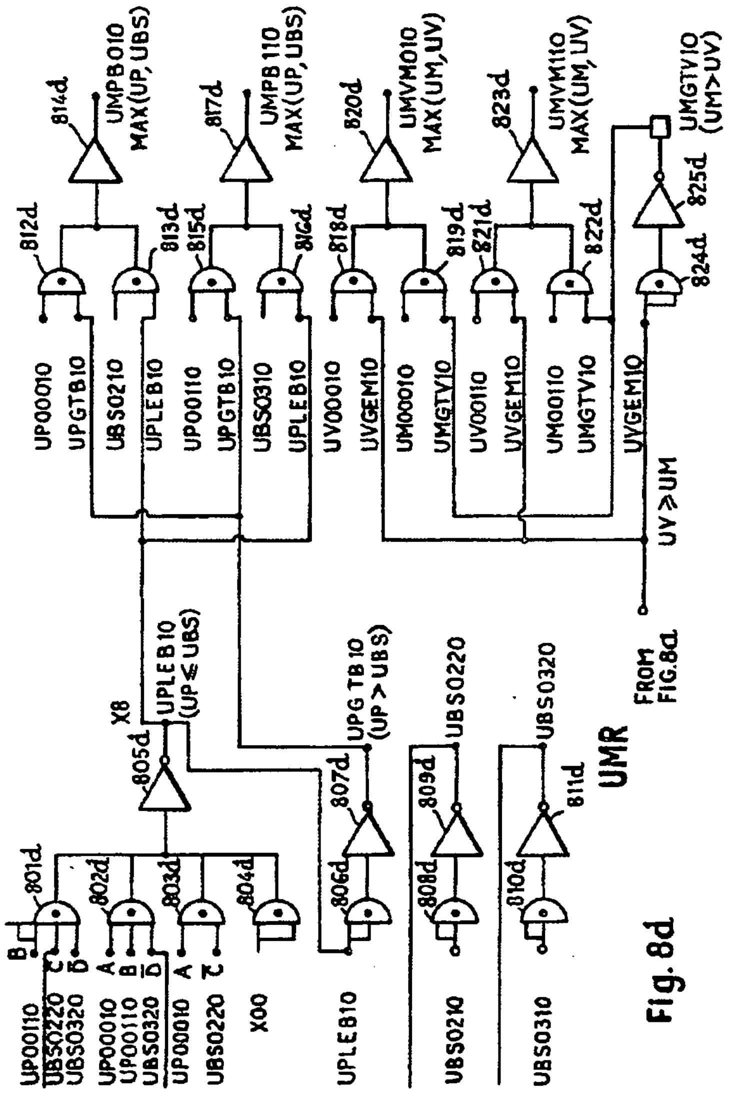
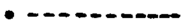

UP00110
UBS0220
UBS0320
B
C
D
801d
UP00010
UP00110
UBS0320
A
B
D
802d
UP00010
UBS0220
A
C
803d
X00
804d
805d
X8
UPLEB10
(UP ≤ UBS)
UPLEB10
806d
807d
UPGTB10
(UP > UBS)
UBS0210
808d
809d
UBS0220
UBS0310
810d
811d
UBS0320
UMR
FROM FIG.8d
UV≥UM
UP00010
UPGTB10
UBS0210
UPLEB10
UP00110
UPGTB10
UBS0310
UPLEB10
UV00010
UVGEM10
UM00010
UMGTV10
UV00110
UVGEM10
UM00110
UMGTV10
UVGEM10
812d
813d
814d
UMPB010
MAX(UP,UBS)
815d
816d
817d
UMPB110
MAX(UP,UBS)
818d
819d
820d
UMVM010
MAX(UM,UV)
821d
822d
823d
UMVM110
MAX(UM,UV)
824d
825d
UMGTV10
(UM>UV)

Fig. 8d

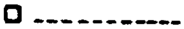
KEY TO SYMBOLS

Fig. 9a



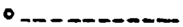
INTERNAL SIGNAL SOURCE

Fig. 9b



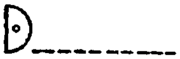
OUTPUT PIN

Fig. 9c



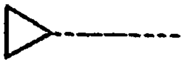
INPUT PIN

Fig. 9d



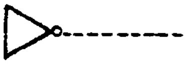
AND GATE

Fig. 9c



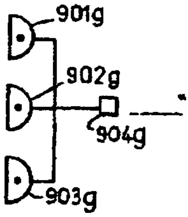
AMPLIFIER

Fig. 9f



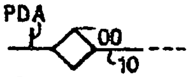
INVERTER

Fig. 9g



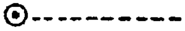
"AND-OR" GATES

Fig. 9h



FLIP-FLOP

Fig. 9i



MICRO-OPERATION

Fig. 9j X : : Y .... X : : Y

Fig. 9k a : : β .... START OF BIT α WHERE THERE ARE β BIT POSITIONS INCLUDING BIT α

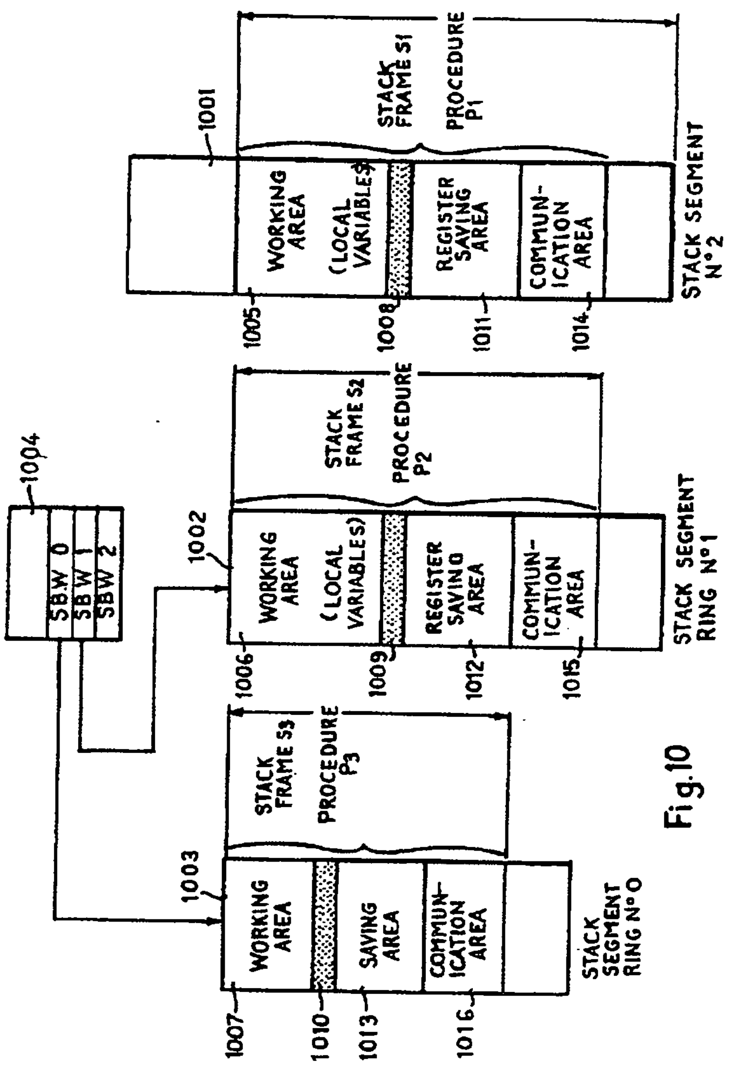

1001
STACK FRAME S1
PROCEDURE P1
WORKING AREA
(LOCAL VARIABLES)
REGISTER SAVING AREA
COMMUNICATION AREA
STACK SEGMENT N°2
1005
1008
1011
1014
STACK FRAME S2
PROCEDURE P2
1004
SBW 0
SBW 1
SBW 2
1002
WORKING AREA
(LOCAL VARIABLES)
REGISTER SAVING AREA
COMMUNICATION AREA
STACK SEGMENT RING N°1
1006
1009
1012
1015
STACK FRAME S3
PROCEDURE P3
1003
WORKING AREA
SAVING AREA
COMMUNICATION AREA
STACK SEGMENT RING N°0
1007
1010
1013
1016

Fig.10

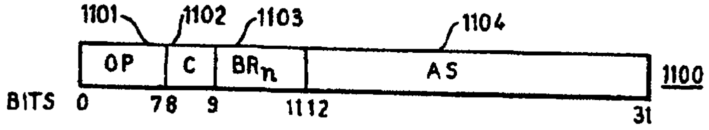

1101
1102
1103
1104
OP
C
BRn
AS
1100
BITS
0
7 8
9
11 12
31

Fig. 11A

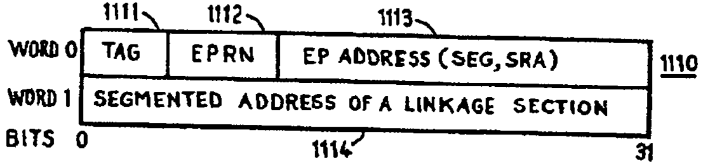

1111
1112
1113
WORD 0
TAG
EPRN
EP ADDRESS (SEG, SRA)
1110
WORD 1
SEGMENTED ADDRESS OF A LINKAGE SECTION
BITS
0
1114
31

Fig. 11B

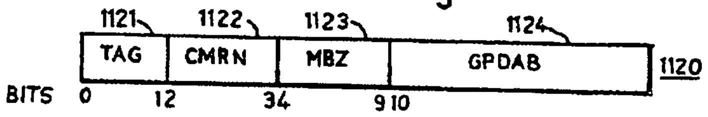

1121
1122
1123
1124
TAG
CMRN
MBZ
GPDAB
1120
BITS
0
1 2
3 4
9 10

Fig. 11C

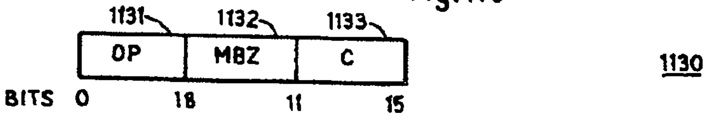

1131
1132
1133
OP
MBZ
C
1130
BITS
0
1 8
11
15

Fig. 11D

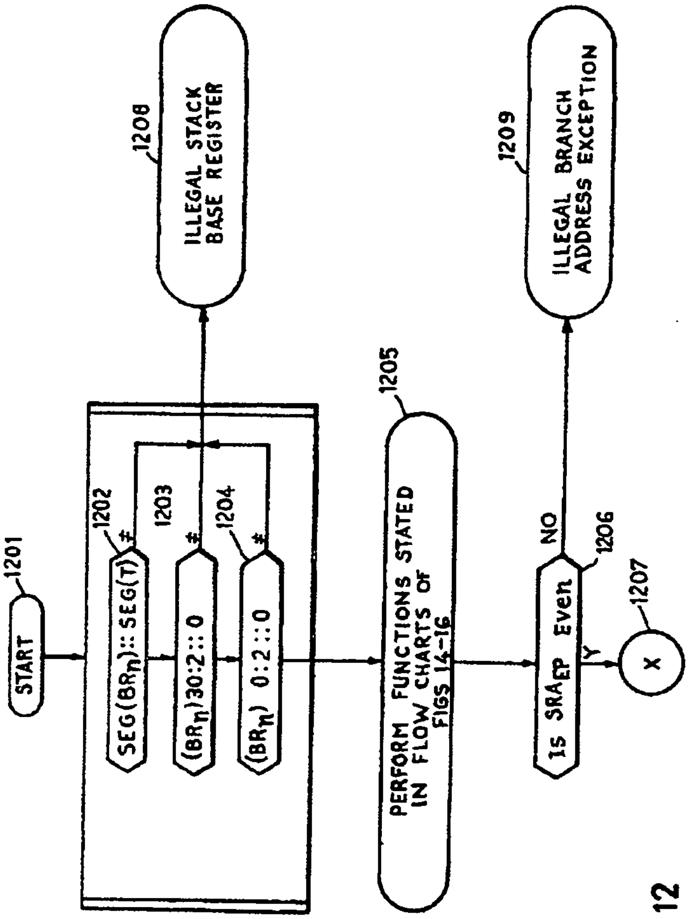

START
1201
SEG(BRn):: SEG(T)
1202
≠
(BRn)30:2::0
1203
≠
(BRn) 0:2::0
1204
≠
1208
ILLEGAL STACK BASE REGISTER
1205
PERFORM FUNCTIONS STATED IN FLOW CHARTS OF FIGS 14-16
IS SRA EP Even
1206
NO
Y
1207
X
1209
ILLEGAL BRANCH ADDRESS EXCEPTION

Fig. 12

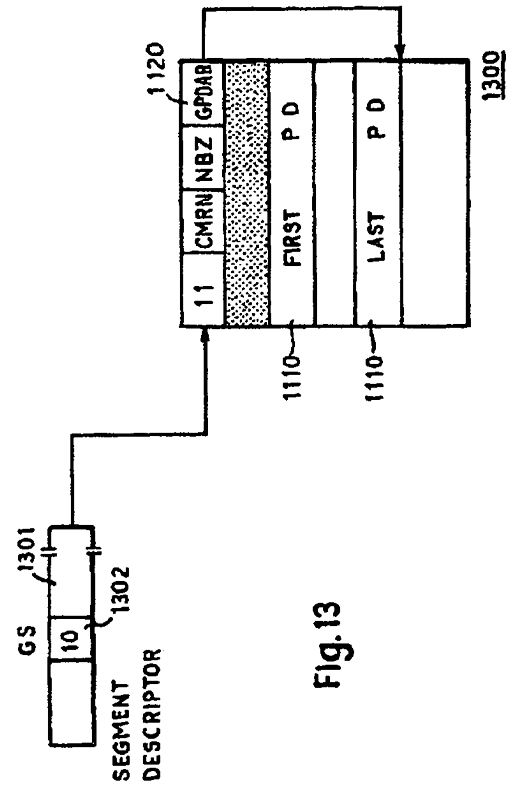
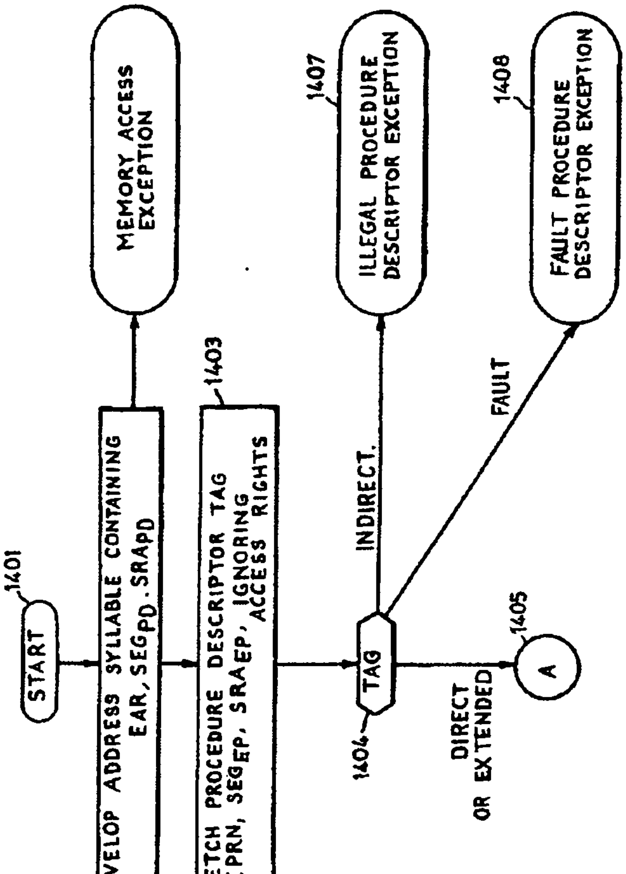

1120
GPDAB
NBZ
CMRN
11
FIRST P D
LAST P D
1300
1110
1110
1301
1302
GS
10
SEGMENT DESCRIPTOR

Fig.13

1483282 COMPLETE SPECIFICATION

29 SHEETS *This drawing is a reproduction of the Original on a reduced scale*

Sheet 26



START
1401
DEVELOP ADDRESS SYLLABLE CONTAINING EAR, SEG$_{PD}$. SRA$_{PD}$
MEMORY ACCESS EXCEPTION
FETCH PROCEDURE DESCRIPTOR TAG EPRN, SEG$_{EP}$, SRA$_{EP}$, IGNORING ACCESS RIGHTS
1403
TAG
1404
INDIRECT.
FAULT
DIRECT OR EXTENDED
A
1405
1407
ILLEGAL PROCEDURE DESCRIPTOR EXCEPTION
1408
FAULT PROCEDURE DESCRIPTOR EXCEPTION

Fig.14

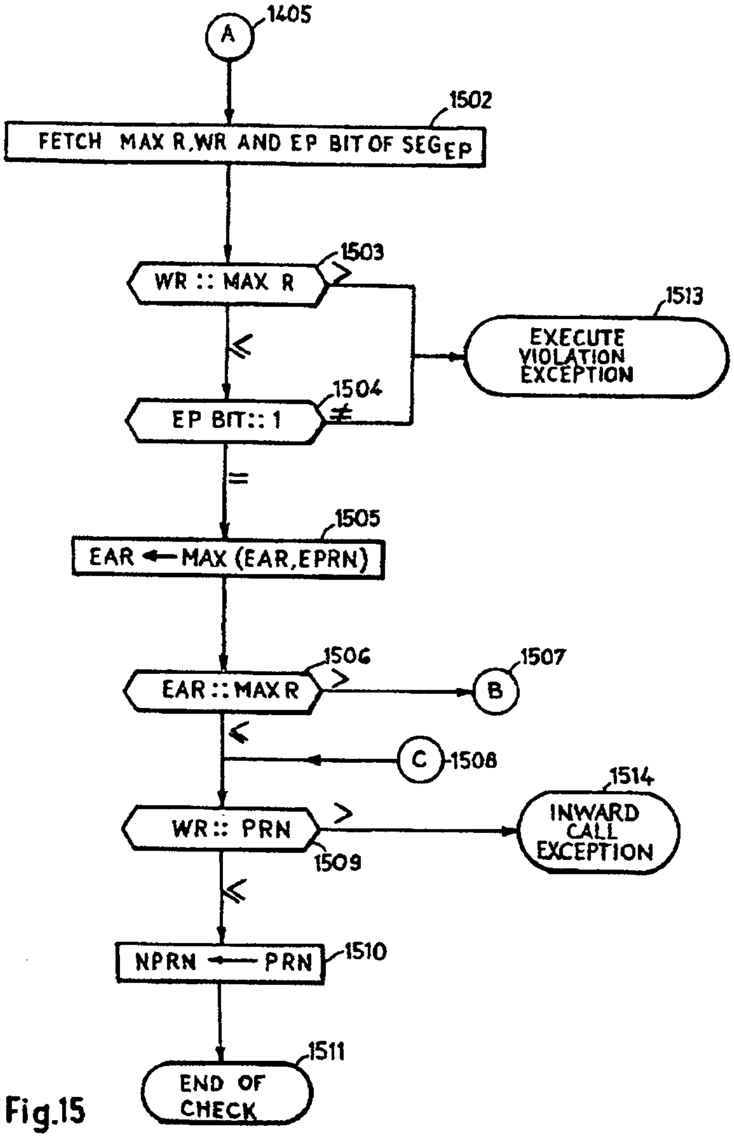

A
1405
FETCH MAX R,WR AND EP BIT OF SEG_{EP}
1502
WR :: MAX R
1503
>
≤
EXECUTE VIOLATION EXCEPTION
1513
EP BIT:: 1
1504
≠
=
EAR ← MAX (EAR,EPRN)
1505
EAR :: MAX R
1506
>
B
1507
≤
C
1508
WR:: PRN
1509
>
INWARD CALL EXCEPTION
1514
≤
NPRN ← PRN
1510
END OF CHECK
1511

Fig.15

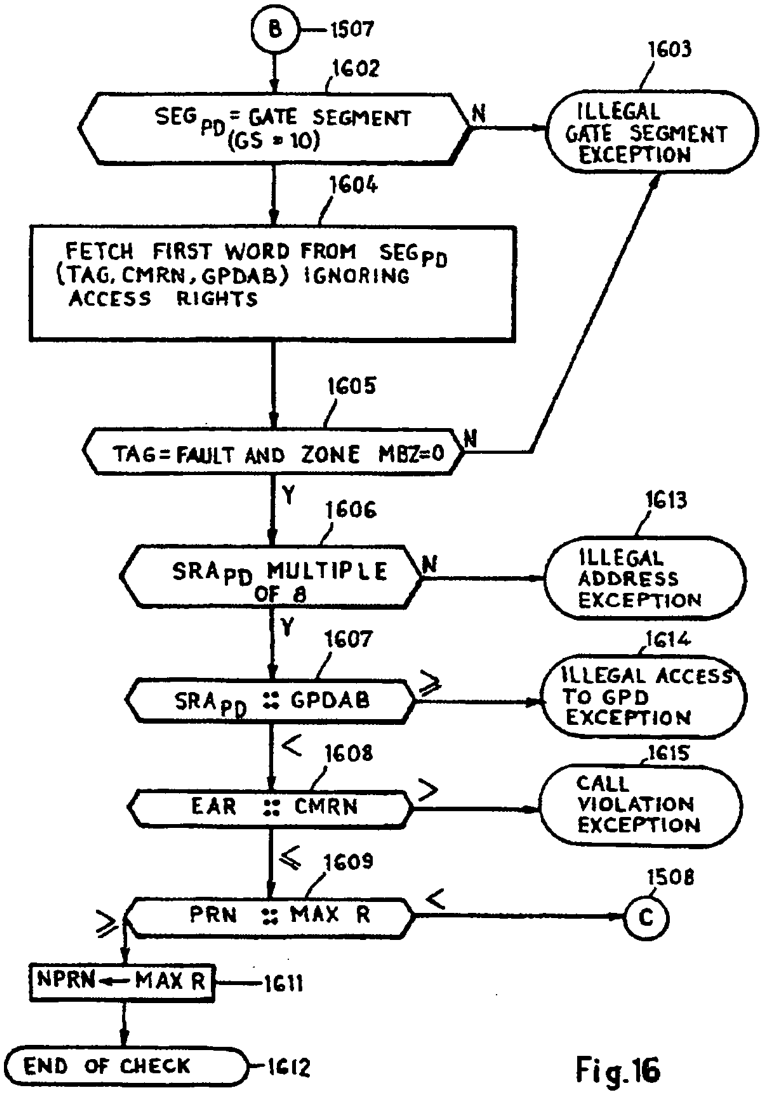
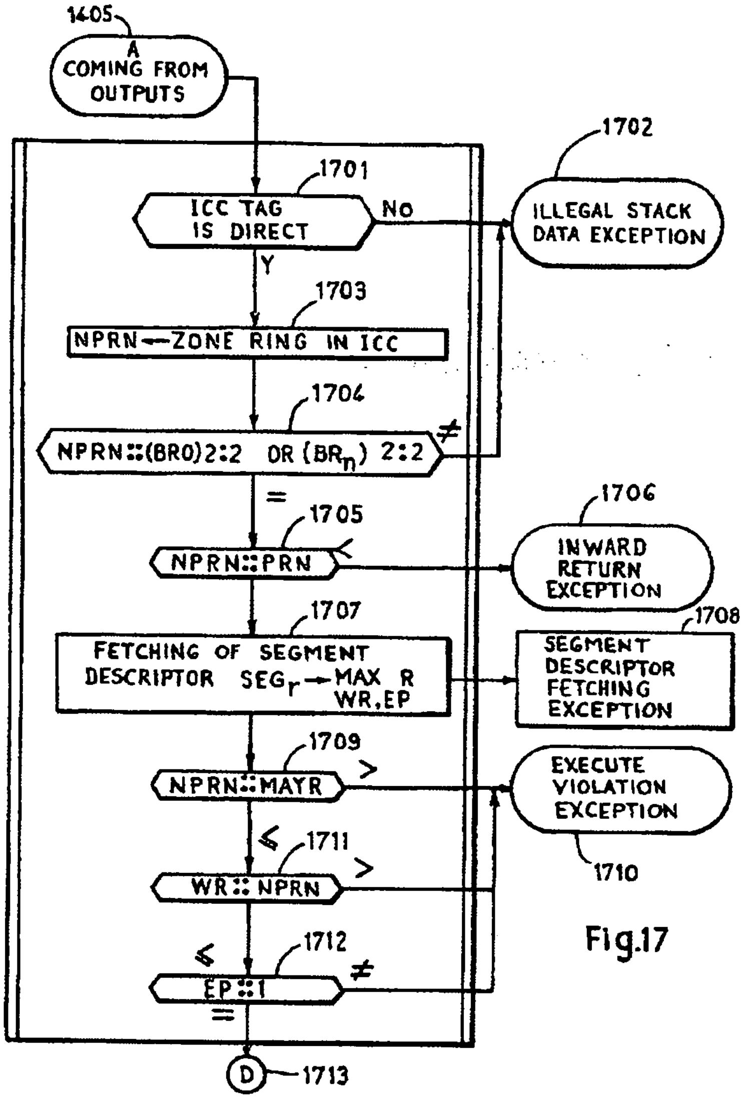

B
1507
1602
SEG_{PD} = GATE SEGMENT (GS = 10)
N
1603
ILLEGAL GATE SEGMENT EXCEPTION
1604
FETCH FIRST WORD FROM SEG_{PD} (TAG, CMRN, GPDAB) IGNORING ACCESS RIGHTS
1605
TAG = FAULT AND ZONE MBZ=0
N
Y
1606
1613
SRA_{PD} MULTIPLE OF 8
N
ILLEGAL ADDRESS EXCEPTION
Y
1607
1614
SRA_{PD} :: GPDAB
≥
ILLEGAL ACCESS TO GPD EXCEPTION
<
1608
1615
EAR :: CMRN
>
CALL VIOLATION EXCEPTION
≤
1609
1508
PRN :: MAX R
<
C
≥
NPRN ← MAX R
1611
END OF CHECK
1612

Fig.16

1483282 COMPLETE SPECIFICATION

29 SHEETS *This drawing is a reproduction of the Original on a reduced scale*

Sheet 29



1405
A
COMING FROM OUTPUTS
1701
ICC TAG IS DIRECT
NO
Y
1702
ILLEGAL STACK DATA EXCEPTION
1703
NPRN←ZONE RING IN ICC
1704
NPRN::(BRO)2:2 OR (BRn) 2:2
≠
=
1705
NPRN::PRN
<
1706
INWARD RETURN EXCEPTION
1707
FETCHING OF SEGMENT DESCRIPTOR SEGr→MAX R WR,EP
1708
SEGMENT DESCRIPTOR FETCHING EXCEPTION
1709
NPRN::MAYR
>
EXECUTE VIOLATION EXCEPTION
1710
≤
1711
WR::NPRN
>
1712
≤
EP::1
≠
=
D
1713

Fig.17

(12) UK Patent Application (19) GB (11) 2 236 604 (13) A

(43) Date of A publication 10.04.1991

(21) Application No 9009655.3

(22) Date of filing 30.04.1990

(30) Priority data
(31) 415984 (32) 02.10.1989 (33) US

(71) Applicant
Sun Microsystems Inc

(Incorporated in the USA - Delaware)

2550 Garcia Avenue, Mountain View, California 94043,
United States of America

(72) Inventor
John Richard Corbin

(74) Agent and/or Address for Service
Potts Kerr and Co
15 Hamilton Square, Birkenhead, Merseyside, L41 6BR,
United Kingdom

(51) INT CL^5
G06F 1/00

(52) UK CL (Edition K)
G4A AAP

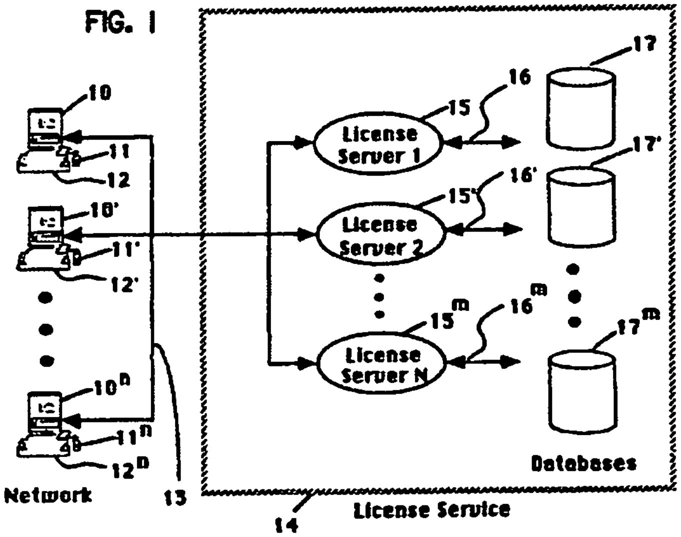
(56) Documents cited
EP 0002390 A1 WO 88/02202 A1

(58) Field of search
UK CL (Edition K) G4A AAP
INT CL^5 G06F 1/00 12/14
Online database: WPI

(54) **Protecting against the unauthorised use of software in a computer network**

(57) The present invention provides to a software application the verification and licence check out functions which are normally performed by a licence server. The encrypted licence information is contained in a licence token, and is stored in a database 17 controlled by the licence server 15. In contrast to the prior art where the server either grants or denies the request after verifying the user's credentials, the server in the preferred embodiment of the present invention finds the correct licence token for the software application and transmits the token to a licencing library. A licence access module attached to the application decodes the token. Routines in the licencing library coupled to the software application verify the licence information before issuing the licence and updating the token. The access module then encodes the updated token before returning it to the server. Because the verification and issuing function of a token are performed by a software application, the application rather than the server becomes the point of attack by unauthorised users. Reverse engineering the access module is less rewarding than attacking the server because the module reveals the contents of a small fraction of a database of licences.

FIG. 1



At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

GB 2 236 604 A

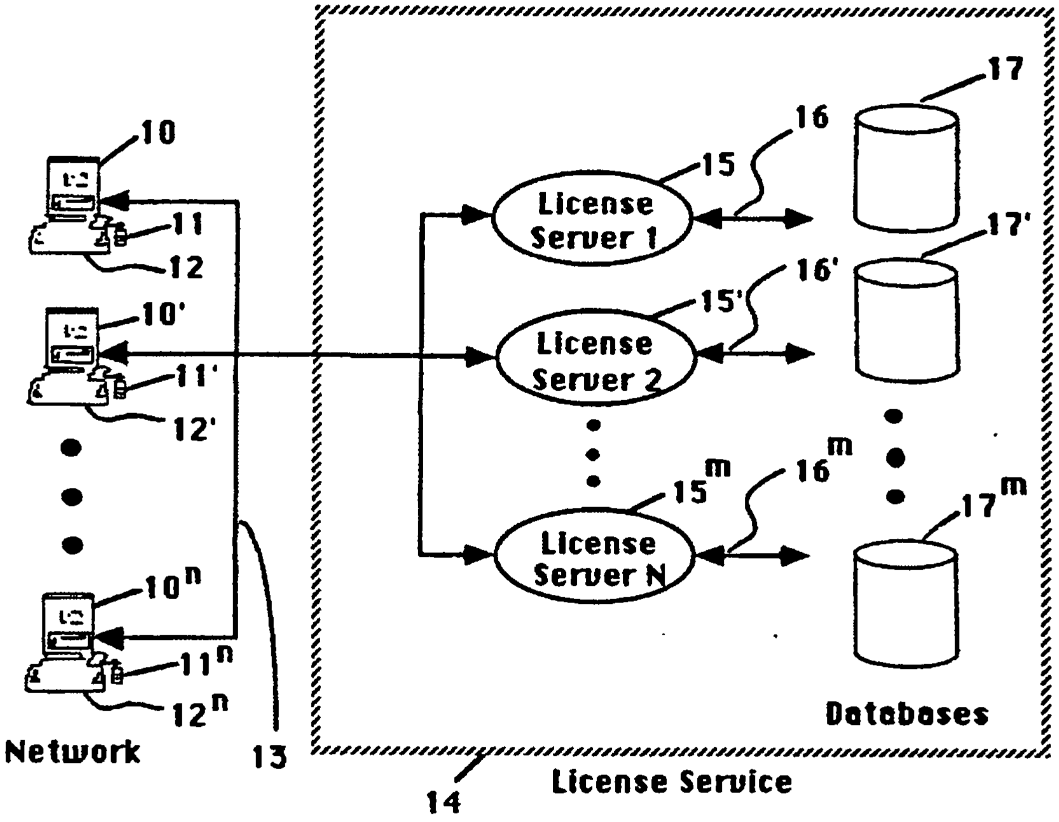

10
11
12
10'
11'
12'
10n
11n
12n
Network
13
License Server 1
License Server 2
License Server N
15
15'
15m
16
16'
16m
17
17'
17m
Databases
License Service
14

FIG. 1

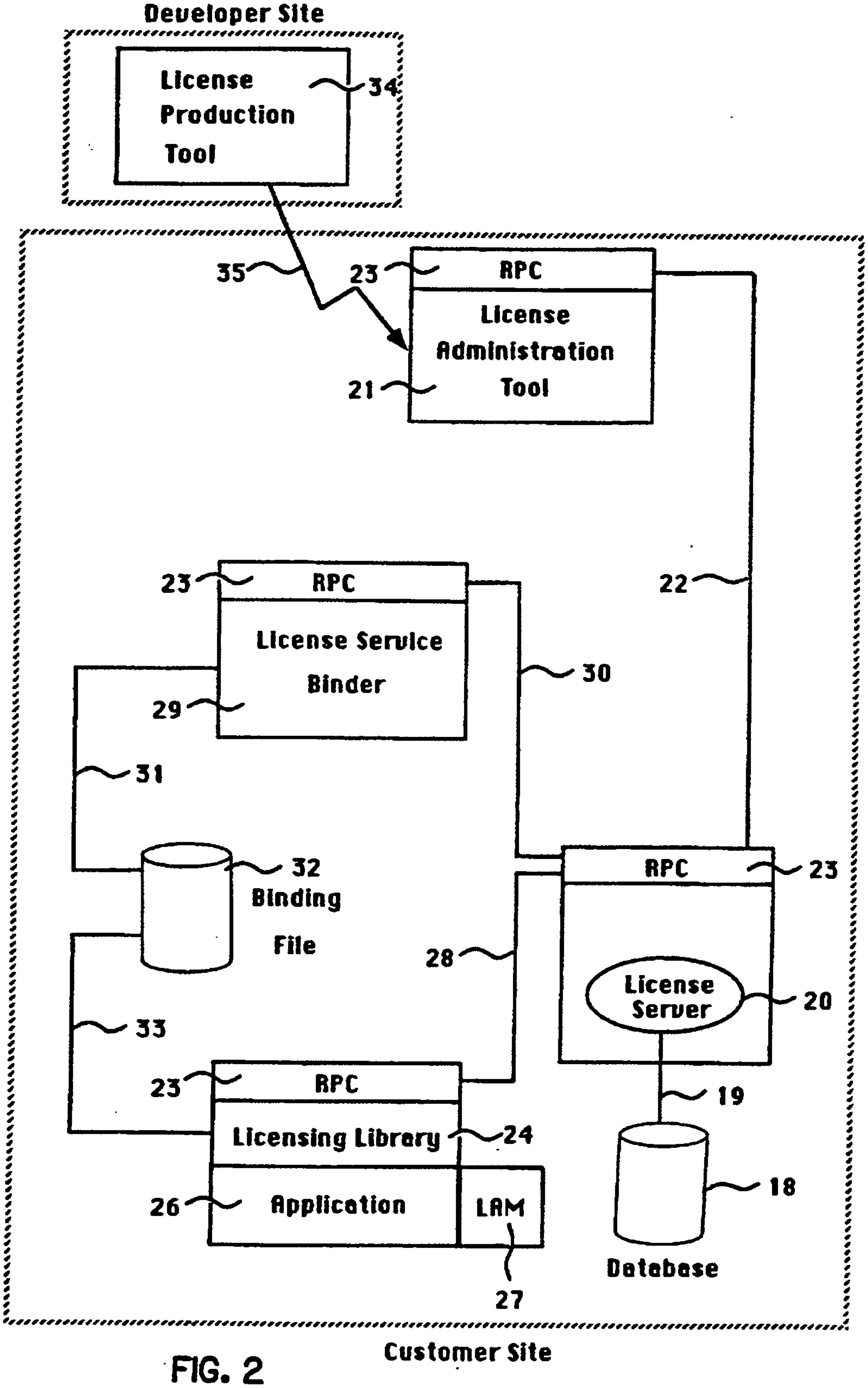

Developer Site
License Production Tool
34
35
23
RPC
License Administration Tool
21
22
23
RPC
License Service Binder
29
30
31
32
Binding File
28
RPC
23
License Server
20
33
23
RPC
Licensing Library
24
26
Application
LAM
27
19
18
Database
Customer Site

FIG. 2

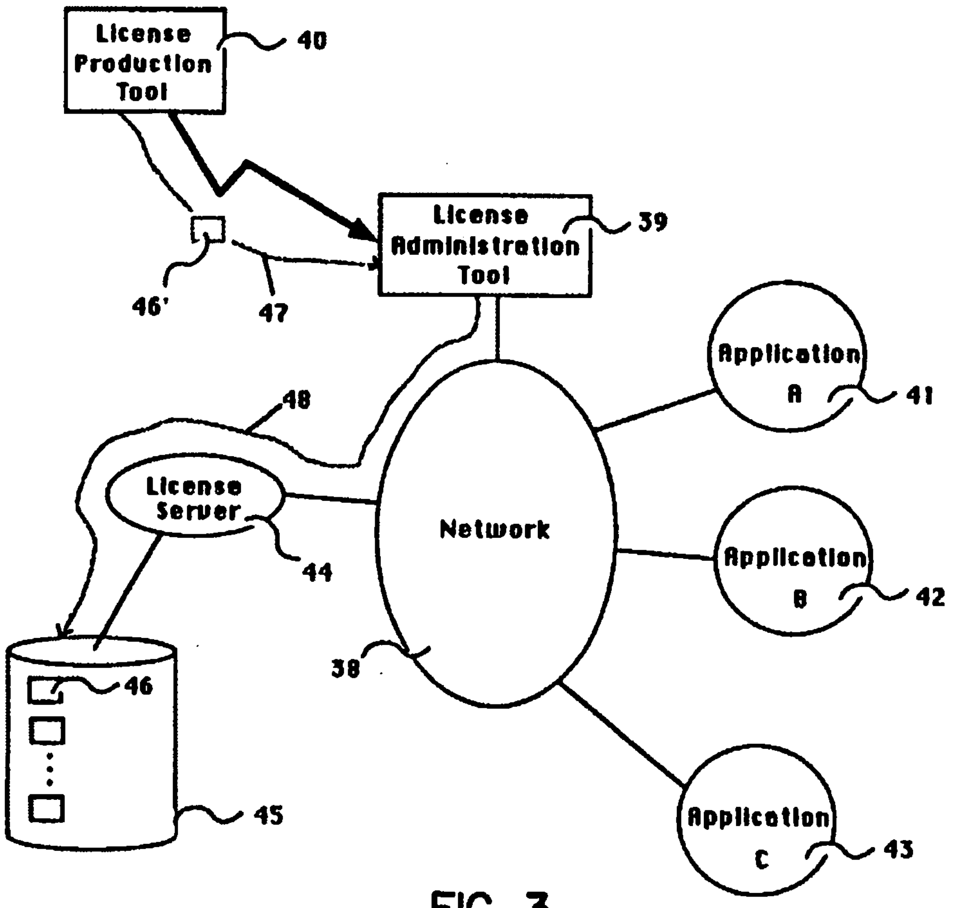

License Production Tool
License Administration Tool
Application A
Application B
Application C
Network
License Server

FIG. 3

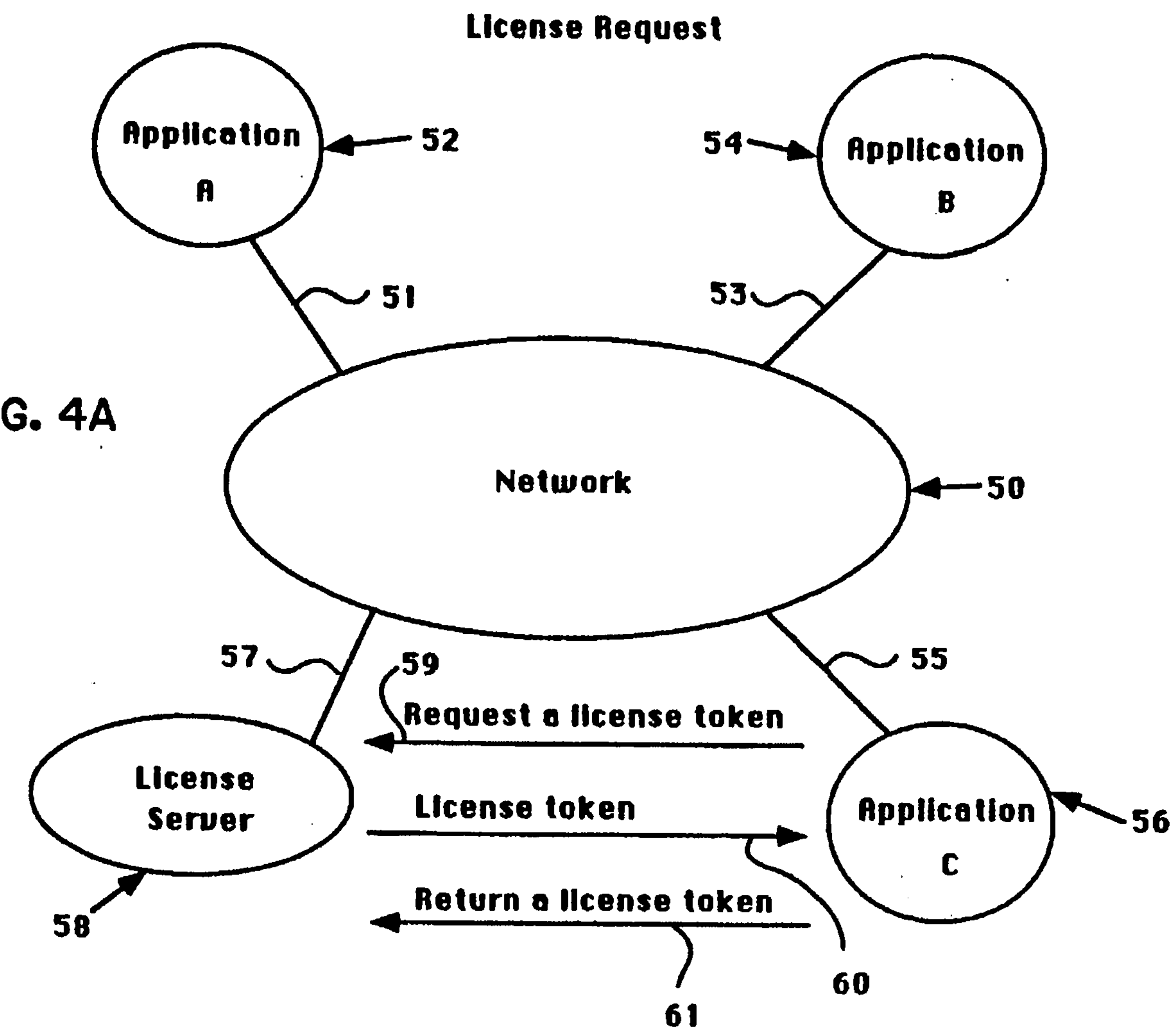

License Request
Application A
52
Application B
54
51
53
Network
50
57
59
55
Request a license token
License Server
License token
Application C
56
58
Return a license token
60
61

FIG. 4A

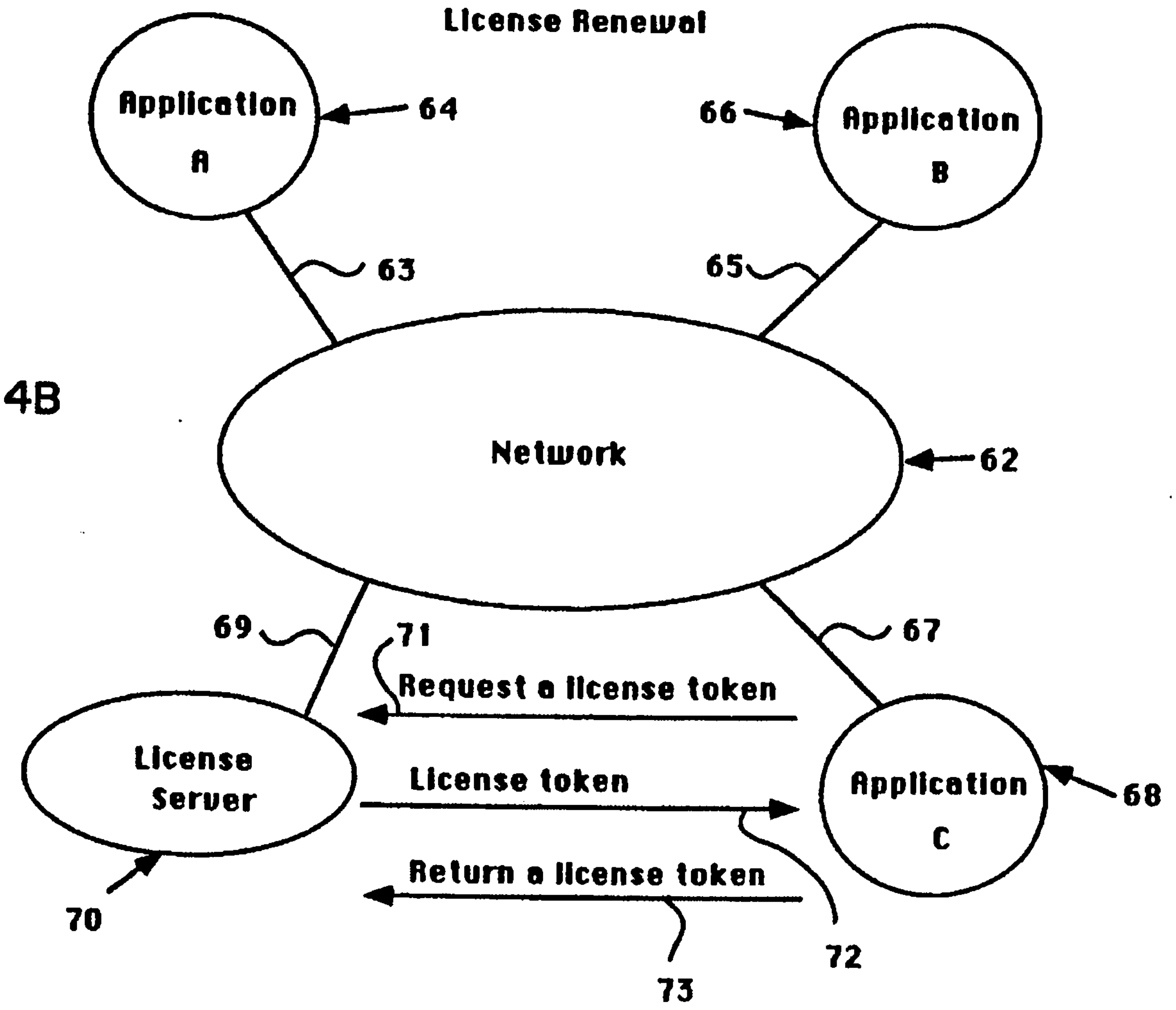

License Renewal
Application A
64
Application B
66
63
65
Network
62
69
71
67
Request a license token
License Server
License token
Application C
68
Return a license token
70
72
73

FIG. 4B

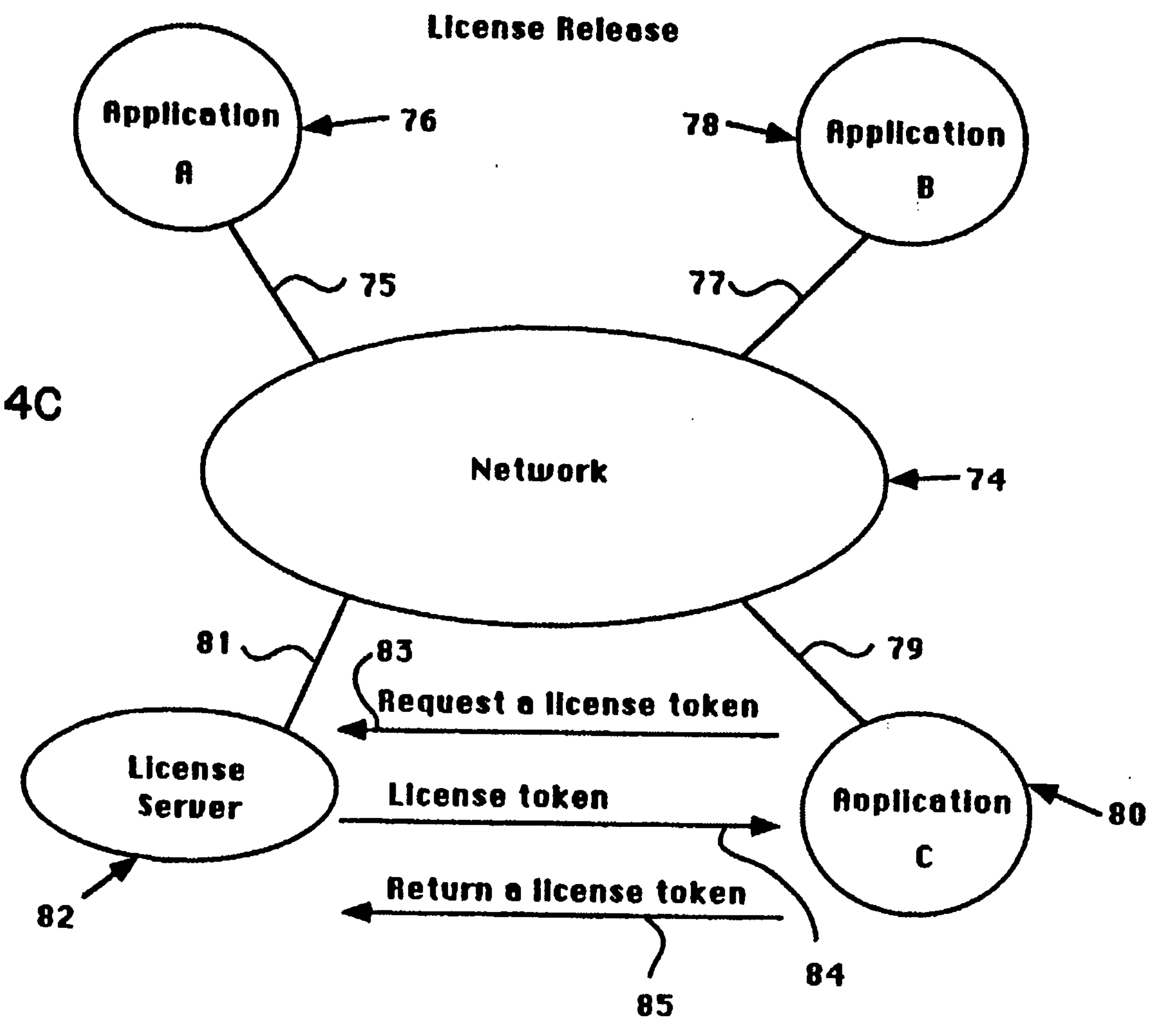

License Release
Application A
76
78
Application B
75
77
Network
74
81
83
79
Request a license token
License Server
License token
Application C
80
82
Return a license token
84
85

FIG. 4C

2236604

METHOD FOR PROTECTING AGAINST THE UNAUTHORIZED USE OF SOFTWARE IN A COMPUTER NETWORK ENVIRONMENT

BACKGROUND OF THE INVENTION

1. FIELD OF THE INVENTION

The present invention relates to a method for protecting against the unauthorized use of a software application in a computer network environment.

2. ART BACKGROUND

A computer network is typically an interconnection of machines or agents over links or cables. The open access characteristics of a computer network presents opportunities for the unauthorized copying of software, thus eroding the licensing revenue potential of software developers. Traditionally, either the entire network must be licensed (commonly referred to as a site license), or each node where the software is run must be licensed (commonly referred to as a node license). A node refers to a single machine, agent or system in a computer network. A license is an authorization given by a software developer to a customer to use a software application in a specific manner.

A site license lets all users at a designated location or network use the software application, regardless of their position on the network. This flat-fee approach is an overkill for a low usage software application. A node license not only ties a software application to a particular machine in a network, but also is not cost effective for the infrequent use of a software application. See, for example, U.S. Patent No. 4,688,169. Furthermore, if new users of licensed nodes wish to use the software application, they are often required to purchase additional licenses.

An alternative to a site license or a node license is the concept of a concurrent usage license. A concurrent usage license restricts the number of users allowed to use a software application at any given time, regardless of their location on the network. Just as renters check out available copies of a movie video from a video rental store, users on a network check out a software application from an agent on a first-come-first-serve basis. Thus, a concurrent usage license charges a fee for the use of a software application proportional to its actual use.

Methods to license a software application for concurrent use in a network environment are currently offered by Highland Software, Inc. and Apollo Computer, Inc. See, M. Olson and P. Levine, "Concurrent Access Licensing", *Unix Review*, September 1988, Vol. 6, No. 9. In general, the license for a software application is stored in a database controlled by a license server. A license server is a program that not only stores the license, but also verifies the user's credentials before checking out the license to the authenticated user. To protect against the authorized use, these methods to license concurrent usage rely on secured communications such as public/private key encryption. Under public/private key encryption, each user of the system has two keys, one of which is generally known to the public, and the other which is private. The private transformation using the private key is related to the public one using the public key but the private key cannot be computationally determined from the public key. See Denning, D., *Cryptography and Data Security*, Addison-Wesley, 1982. The encryption key is hidden in the license server to encrypt the database of licenses. Well designed public/private key encryption schemes are difficult to crack, especially if the license server is located in a trusted environment. A trusted environment is one whose access is limited to users having the proper credentials. However, a license server is more likely to be located at a customer's site and hence in an hostile environment. It follows that the license server is vulnerable to sophisticated intruders. Once the private key is decrypted, all sensitive information on the license server such as licenses are compromised.

It is therefore an object of the present invention to provide a more secure method to protect against the unauthorized use of software in a concurrent use licensing environment.

SUMMARY OF THE INVENTION

The present invention provides to the software application the verification and license check out functions which are normally performed by a license server. The preferred embodiment of the present invention comprises a computer network including a plurality of agents running at least one license server and at least one software application. The license server controls a database of an agent containing the license information for the software application. The license information is contained in a license token, and is stored in the database controlled by the license server. The license token is a special bit pattern or packet which is encrypted by the software vendor of the application software. The software application communicates with the license server through a licensing library. The licensing library is a collection of library routines that the software application invokes to request or renew a license from the license server. Before a software application obtains a license, the license token must be decoded by a license access module. The license access module, which is linked with the software application and the licensing library is a program that decodes the license token from a vendor specific format to a licensing library format.

When an user wishes to run a software application, the licensing library invokes a call to request a license token from the license server. In contrast to the prior art where the license server either grants or denies the request after verifying the user's credentials, the license server in the preferred embodiment of the present invention finds the correct license token for the software application and transmits the license token to the licensing library. The license access module attached to the licensing library decodes the licensing token. Routines in the licensing library coupled to the software application verify the license information before checking out the license and updating the license token. The license access module encodes the updated license token before returning it to the license server.

Because the verification and check out function of a license token are performed by a software application, the software application rather than the license server becomes the point of attack by unauthorized users. Reverse engineering the license access module is less rewarding than attacking the license server because the license access module reveals the contents of a fraction of a database of licenses. By the time most attackers crack the license access module, the software vendors would most likely introduce newer versions of the software application and new license access modules for them. Thus the present invention provides a more secure method for protecting against the unauthorized use of a software application in a computer network environment without modifying the underlying computer network.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a network environment employing the present invention.

Figure 2 describes the architecture of a network licensing scheme employing the preferred embodiment of the present invention.

Figure 3 describes the installation of a license token in the preferred embodiment of the present invention.

Figure 4a illustrates the use of a license token to request a license from a license server in the preferred embodiment of the present invention.

Figure 4b illustrates the use of a license token to renew a license from a license server in the preferred embodiment of the present invention.

Figure 4c illustrates the use of a license token to release a license from a license server in the preferred embodiment of the present invention.

NOTATION AND NOMENCLATURE

The detailed description that follows is presented largely in terms of algorithms and symbolic representations of operations on data bits and data structures within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art.

An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. These steps are those requiring physical manipulation of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It proves convenient at times, principally for reasons of common usage, to refer to these signals as bit patterns, values, elements, symbols, characters, data packages, or the like. It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities.

Further, the manipulations performed are often referred to in terms, such as adding or comparing, that are commonly associated with mental operations performed by a human operator. No such capability of a human operator is necessary, or desirable in most cases, in any of the operations described herein that form part of the present invention; the operations are machine operations. Useful machines for performing the operations of the present invention include general purpose digital computers or other similar devices. In all cases there should be borne in mind the distinction between the method of operations in operating a computer and the method of computation itself. The present invention relates to method steps for operating a computer in processing electrical or other (e.g. mechanical, chemical) physical signals to generate other desired physical signals.

The present invention also relates to an apparatus for performing these operations. This apparatus may be specially constructed for the required purposes, or it may comprise a general purpose computer as selectively activated or reconfigured by a computer program stored in the computer. The algorithms presented herein are not inherently related to any particular computer or other apparatus. In particular, various general purpose machines may be used with programs written in accordance with the teachings herein, or it may prove more convenient to construct a more specialized apparatus to perform the required method steps. The required structure for a variety of these machines will appear from the description given below.

DETAILED DESCRIPTION OF THE INVENTION

The following detailed description is divided into several sections. The first of these sections describes a general network environment for accessing a database of licensed software programs. Subsequent sections discuss the details of a method for protecting against the unauthorized use of a software application.

I. General Network Environment

Referring to **Figure 1**, computer network environment comprises a plurality of data processing devices identified generally by numerals 10 through 10^n (illustrated as 10, 10' and 10^n). These data processing devices may include terminals, personal computers, workstations, minicomputer, mainframes and even supercomputers. For the purposes of this Specification, all data processing devices which are coupled to the present invention's network are collectively referred to as "agents". It should be understood that the agents may be manufactured by different vendors and may also use different operating systems such as MS-DOS, UNIX, OS/2, MAC OS and others. Particular examples of suitable agents include machines manufactured by Sun Microsystems, Inc., Mountain View, Calif. Each of the agents has an input device such as a keyboard 11, 11' and 11^n or a mouse 12, 12' and 12^n. As shown, agents 10 through 10^n (illustrated as 10, 10' and 10^n) are interconnected for data transfer to one another by a common cable 13. It will be appreciated by one skilled in the art that the common cable 13 may comprise any shared media, such as coaxial cable, fiber optics, radio channel and the like. Furthermore, the network resulting from the interconnection of the cable 13 and agents 10 through 10^n (illustrated as 10, 10' and 10^n) may assume a variety of topologies, such as ring, star, bus, and may also include a collection of smaller networks linked by gateways or bridges.

Referring again to **Figure 1** is a license service 14. The license service 14 is a resource shared by every agent connected to the network. In the preferred embodiment of the present invention, the license service 14 comprises license servers 15 through 15^m (illustrated as 15, 15' and 15^m) and databases 17 through 17^m (illustrated as 17, 17' and 17^m), where m is less than or equal to n. A license server is a program that runs on an agent with a memory storage capability. Each license server 15 (illustrated as 15, 15' and 15^m) communicates with a database 17 stored in memory on the agent over an interface 16 (illustrated as 16, 16' and 16^m). As will be described in detail below, the database 17 stores licensing information for various software applications which are purchased and authorized to run in the computer network environment. The license server is not limited to run on a specific agent, but can operate on any agent including the agent on which the user is to operate the application. Thus, any agent connected to the network may function as a license server as well as a device on which a user may operate application software. As will be described below, the license server does not perform verification of licenses of application software; rather the license server is passive and provides storing, locking, logging, and crash recovering function for the application software.

Figure 2 illustrates the architecture of a network licensing scheme of the present invention. The architecture comprises a database 18, database interface 19, license server 20, licensing library 24, License access module 27, license administration tool 21, license service binder 29, and license production tool 34.

The database 18 stores licensing information and application usage data. Preferably the database 18 comprises a plurality of records which contain the following information:

Database Element	Description
Unique Key Table	Keys for all other tables
Vendor Table	Vendor's ID and name
Product Table	Product number and name
Version Table	Version number and date
License Table	License #, exp date, total units
License Token Table	Stores encoded license token
Unit Group Table	A group's allocation of license
Group List Table	Name of the group
Allowed Users Table	Credentials of allowed users
Current License Use Table	Applications using a license
Lock Table	Locked records in database
Authorized administrator Table	Login names of administrators
License Operation Log Table	Administrator's log information
License Usage Log Table	Request handle plus Client Log
License Queue Log Table	License wait queue
Application Message Log Table	Application specific messages

A database interface 19 provides communication between the license server 20 and the database 18 in order to prevent concurrent access to the same database record by multiple users which can cause the data in the record to become corrupted. Thus, only the owner of the lock can read from and write to the locked record during the usage of the application.

The license server 20 operates on an agent and interfaces the database 18 to license administration tool 21, licensing library 24 and license service binder 29. The license server 20 communicates with the license administration tool 21, licensing library 24 and license service binder 29 via an interface 23. Preferably the interface 23 is a remote procedure call mechanism which permits a process operating on one device or agent connected to the network to request a resource or service from a remote device or agent connected to the network. See A. Birrell and B. Nelson, "Implementing Remote Procedure Calls," *ACM Transaction on Computer Systems*, February 1984, Vol. 2, No. 1.

Multiple license servers may reside on multiple agents. Preferably the license server 20 operates in a background mode of the agent such that its operation is transparent to a user of that agent. More particularly, as will be described below, the license server 20 provides the following functions: 1) servicing the requests from the licensing library 24 for license token; (2) maintaining a wait queue for requests to the database 18 when no licensing units are available; (3) generating `locks` for exclusive access to database 18; and (4) providing access to information in the database 18.

The licensing library 24 is a set of library routines which enable the application 26 to request licensing service from the license server 20. Upon receiving the request for service from the licensing library 24, the license server 20 retrieves a license token from the database 18 and transmits it to the licensing library 24. The licensing library 24 is linked with the application 26 and communicates with the license server 20 over a path 28 with, preferably, a remote procedure call mechanism 23. Among the major library calls in the licensing library 24 is the application's request for a license from the license server 20. Other important library calls include the request to renew and to release a license. The use of the license token to accomplish the request for the various licensing service will be described in detail below.

The license access module (LAM) 27 is prepared by the software vendor 24 to decode the license token. Once decoded, the application 26 via routines in the licensing library verifies the licensing information in the license token and determines whether a license may be checked out. The LAM 27 also encodes the license token before the application returns it to the database 18 via license server 20. The license access module 27 is described in further detail below.

The license administration tool 21 is utilized by the network administrator to perform administrative functions relevant to the concurrent usage of a software application. The license administration tool 21 may run on any agent connected to the computer network. The license administration tool 21 is primarily used to install the license token into the database 18 through the license server 20. The functionality of the license administration tool 21 includes: (1) starting or terminating a license server, (2) accessing a database controlled by a license server; and (3) generating and printing reports on license usage.

The application 26 may not access the database 18 directly; rather, the request for a license is made through the licensing library 24 to the license server 20 over a path 28. Most network licensing schemes employ secured communication between the licensing library 24 and the license server 20. In contrast, the present invention uses the license access module (LAM) 27 the license library 24 and a plurality of license tokens to protect against the unauthorized use of software application in a computer network.

Referring once again to **Figure 2**, a license service binder 29 is shown coupled to the license server 20 over a path 30. The license service binder 29 is invoked by means known in the art, such as a network service program. The license service binder 29 locates all agents that are designated as servers on the network, and keeps track of which server is servicing which application. The license service binder 29 contacts each server on its table of available servers and requests a list of products it serves. Finally the license service binder 29 writes the contents of the table of available license servers and the list of products into a binding file 32 over a path 31. In **Figure 2**, the binding file 32 is coupled to the licensing library 24 over a path 33. The application 26 queries the binding file 32 to see which license server can service its request for a license.

A license production tool 34 is used by the software vendor to create a license token for transmittal to the network administrator. Receiving the license token, the network administrator installs it with the license administration tool 21 into the database 18 through license server 20.

II. License Token

Referring to Figure 3, the creation of a license token in a computer network employing the preferred embodiment of the present invention will be described. A computer network 38 is shown coupled with a license administration tool 39 and a single license server 44. The license server 44 communicates with a database 45. Applications 41, 42, and 43 are shown requesting licensing service from the license server 44. When a customer purchases a license for an application, such as a CAD/CAM program for its research and development department, the software vendor creates a license token with a license production tool, and delivers the license token to the customer's network administrator. A license token is a special bit pattern or packet representing a license to use a software application. The network administrator installs the license token 46 into the database of the license server using the license administration tool 39. Unlike the token used in a token ring which is passed from agent to agent, a license token in the preferred embodiment of the present invention is passed only between a license server and a licensing library for a predetermined amount of time. The predetermined amount of time corresponds to the time the license token is checked out of the license server. Currently, the license token is checked out to an application for no more than ten seconds, and the license token is returned as quickly as possible to the issuing license server. The license token 46 contains information encrypted in the vendor's format such as ,vendor identification, product and version numbers as well as the number of license units purchased for the license token. A license unit corresponds to the license weighting for an agent connected to the computer network. For example, powerful workstations could require more license units to use a software application than an average personal computer.

The software vendor produces a license token using a license production tool 40. A path 47 illustrates how a license token 46' makes its way to a license administration tool 39 at the customer's site. There, the system administrator installs the license token 46' as license token 46 into the license database 45 of the license server 44. A path 48 indicates the transfer of the license token 46' from the license administration tool 39 to the license server 44 and into the database 45 as license token 46. The license server 44 is now ready to entertain requests from applications 41, 42, and 43 for a license to use the application corresponding to token 46 as well as other applications represented in its database 45.

It should be understood that each network may have a plurality of license servers and each license server may have in its database a plurality of license tokens for a variety of software applications. Referring again to Figure 3, if application A 41 requests and checks out the license token 46 for less than ten seconds, applications B and C 42, 43 would be unable to check out the license token 46 if their requests were made during the same time application 41 is checking out a license from the license token 46 because of the locking mechanism provided by database interface 19. Thus, to achieve concurrent license usage in network 38, it is preferred that the network administrator installs more than one license server. To minimize the task of recovering from license server crashes, it is also preferred that the system administrator spreads the license units for any one application among a plurality of strategically located license servers. For instance, if a network has four license servers, the network administrator may want to allocate the twenty license units for a particular popular application among four license tokens with five license units for each license token. In the event one license server crashes or the license token is checked out, the other three license servers may provide licensing service to other applications.

Figure 4a illustrates the use of a license token to request for a license. As shown, a network 50 is provided, and is coupled to Applications 52, 54 and 56 respectively. Application 56 succeeded in requesting a license token from the license server 58 in step 59. The license token is transmitted to application 56 in step 60. When done, Application 56 returns the license token to the license server 58 in step 61. Aside from the license request function performed with the license token as shown in **Figure 4a**, the license token is also used in other critical stages of the licensing process. For example, an user may wish to run an application beyond the initial allocated time. As shown in **Figure 4b**, Application 68 makes a license renewal request 71 from the license server 70 with license token 72. Similarly, in **Figure 4c** the user makes a license release request 83 when the application no longer needs the license units. As such, the user updates the license token 84 by returning the updated license token to the license server 82 in step 85.

III. License Access Module

In **Figure 2**, a license access module (LAM) 27 is linked with the application 26 and the licensing library 24 to form the executable code that software vendors ship to the customers. The license access module 27 decodes and encodes the encrypted license token as it is being passed between the license server and the licensing library 24. Thus the level of security of an application from unauthorized use depends heavily upon how secure the license access module is.

Conventional network licensing schemes use public/private key encryption to encode sensitive information. Such a scheme is effective if the license server is in a trusted environment. However, the customer has the same access to any agent in a network, including the license server. The security of the licensing scheme can be compromised by a user who decrypts the license server's private key. Once the unauthorized user determines the server's private key, he can decrypt all sensitive information on the license server. Should all license servers use the same key, as is frequently done, then all the security of the applications served by all the license servers will be compromised.

The license access module 27 first translates a license token from a vendor specific format to a format usable by the licensing library 24. The license access module accomplishes the translation in two modules. One module translates or decodes a license token from a vendor specific format to a licensing library format. The second module translates or encodes the updated license token from the licensing library format to the vendor specific format. The second module is invoked anytime the licensing library updates the information in a license token.

Upon receiving the license token in the licensing library format, the licensing library invokes routines which verify the correctness of the license by reviewing the following license information stored in the token: (1) flag, (2) maintenance contract date, (3) host name and domain, (4) product name, (5) host id number, (6) license serial number, and (7) expiration date of license. This is compared to the information maintained by the application. If the information matches, the license is verified. After completing the verification process, a routine in the licensing library is initiated which checks out the license by decrementing the license units in license token by the number of licensing units being checked out.

The decoding and encoding routines allow software vendors to implement their own security mechanism to protect their licenses from unauthorized use even though they reside at the customer's site.

Below is an example of a sample application using the licensing library and the license access module written in C language:

```
#define LIC_RENEWAL_TIME (60)             /set renewal time for this session/
#define EST_LIC_RENEWAL_TIME (LIC_RENEWAL_TIME x .9)

NL_vendor_id NL_Vendor_id = 1223;                 /set vendor #/
NL_prod_num NL_Prod_num = "02"                    /set product #/
NL_version NL_Version = ( 12/20/88, "1.0" );      /set version id #/

--
      status = NL_init (vendor_id, NULL, &job_id);  /initialize license service/
      if (status != NL_NO_ERROR)      /accept job id if no error/
            {
            fprintf (stderr, "nl_init failed - error =
            %d\n", status );                     /error message if error and
                                                 return/
            return;
            }

      units = 3;
      code_funcs.encode_p = nl_encode;   /pointer to encode function/
      code_funcs.decode_p = nl_decode;   /pointer to decode function/
      if (signal (SIGALRM), alarm_intr ) == (void *) -1)    /set alarm if no
                                                          error/
            {
            perror ("Cannot set SIGALRM"); /otherwise, error message/
            return;
            }
      status = NL_request (job_id, NL_Prod_num,  /request a license/
      &NL_Version,
      units, LIC_RENEWAL_TIME, NL_L2_SRCH,
      &code_funcs, NULL,
      &req_handle, NULL, &app_info);

      if (status != NL_NO_ERROR)         /no error, license checked
            {                            out from license server/
            fprintf (stderr, "nl_request failed - error =
            %d\n", status);              /otherwise, error message/
            return;
            }

      /*
       * We got a license                /license request successful/
       */

      alarm (EST_LIC_RENEWAL_TIME);      /set alarm for license renewal
                                          time/
--
      Application Runs                   /runs application/
--
      status = NL_release (req_handle);  /request to release a license/
      if (status != NL_NO_ERROR)
            {
            fprintf (stderr, "nl_release failed - error =     /otherwise, error
          %d\n", status);                                   messages/
          return;
          }

      Int
      alarm_intr ()
          {

      status = NL_confirm (req_handle,       /renew licensing unit with
      LIC_RENEWAL_TIME, NULL);                licensing server/
    /*
     * Verify vendor private information
    */
          }

      If (status!= NL_NO_ERROR)
      fprintf (stderr, "nl_confirm failed - error =       /otherwise, error
      %\n", status);                                      message/
          {
                puts ("license renewed")                  /successful license
          }                                               renewal/
```

The sample application given above is accompanied by self-explanatory annotation to the right margin of the codes. Of particular interest are code_func.encode_p and code_func.decode_p. Encode_p and decode_p are pointers to the software vendor's encode and decode routines, respectively. Taking the pointers in the code_func variable, the licensing library can use the pointers to invoke the decoding and encoding routines in the license access module. The three major licensing library routines, request for a license (NL_request), release a license (NL_release) and renew a license (NL_confirm) invoke the decoding and encoding routines. For example of a license access module, see Appendix 1.

In implementing the license access module, the license server becomes merely a repository for license tokens. The licensing library coupled to the application performs the procedure of authenticating the license token prior to granting a license and therefore access to run the application.

Because the level of security of the system is dictated by the license access module, the software vendors are free to make the license access module as simple or as complex as they desire. In particular, they are free to adopt any of the encryption schemes as part of their encryption routines. If the security mechanism is broken, and the encryption known to others, then the software vendors can easily remedy the situation by releasing a new version of the product with a new license access module.

While the present invention has been particularly described with reference to Figures 1-4 as well as Appendix 1, and with emphasis on certain language in implementing a method to protect against the unauthorized use of software application in a computer network environment, it should be understood that they are for illustration only and should not be taken as limitation upon the invention. In addition, it is clear that the method of the present invention has utility in any application run in a computer network environment. It is contemplated that many changes and modifications may be made, by one skilled in the art, without departing from the spirit and scope of the invention disclosed above.

CLAIMS

1. In a computer network environment including a plurality of software applications licensed to run on at least one network of agents, said applications located on said agents wherein use of the application on a particular agent is permitted upon the grant of a license, said license being requested by a user from said agent of said applications, a system for protecting against the unauthorized use of said applications comprising:

license token means for storing licensing information of said applications; license server means connected to said agents for communicating with said applications, said license server means having a database which stores said license token means, said license server means further retrieving said license token means from said database upon a request for a license by said applications, said license server means further transmitting said license token means to said applications;

license access means connected to said agents for decoding and encoding said license token means from said license server means, said license access means being integrated with said applications, said license access means receiving said license token means from said license server means; and

licensing library means connected to said agents for verifying said decoded license token means before access to said license is granted, said licensing library means being integrated with said applications.

2. The system as defined in claim 1, wherein each said license token means containing licensing information for at least one version of each said applications.

3. The system as defined in claim 1, wherein the contents of said license token means is encrypted.

4. The system as defined in claim 1, wherein said license token means is passed between said license server means and said licensing library means for a predetermined time period.

5. The license token means as defined in claim 4, wherein during said predetermined time period, only one said applications may check out one said license token means.

6. The system as defined in claim 1, wherein said license server means receives said request for a license from said applications, said license server searches in said database for a license token means storing the license requested by said application before retrieving said license token means.

7. The system as defined in claim 1, wherein said license access means decodes the contents of said license token means before said licensing library means verifies said license token means.

8. The system as defined in claim 1, wherein said license access means encodes said license token means after said licensing library verifies said license token means and prior to returning said license token means to said license server means.

9. The system as defined in claim 1, wherein said licensing library verifies said license token means by

comparing the licensing information stored in said license token means with the licensing information maintained by said application.

10. The system as defined in claim 1, wherein said licensing library means checks out said license of said application in response to a positive comparison of the license information.

11. The licensing library means as defined in claim 10, wherein said license for said application being checked out after said licensing library verifies said license token means.

12. In a computer network environment including a plurality of software applications licensed to run on at least one network of agents, said applications located on said agents wherein use of the application on a particular agent is permitted upon the grant of a license, said license being requested by a user from said agent of said applications, a system for protecting against the unauthorized use of said applications comprising:

license token means for storing licensing information of said applications;

license server means connected to said agents for communicating with said applications, said license server means having a database which stores said license token means, said license server means further retrieving said license token means from said database upon a request for a license by said applications, said license server means further transmitting said license token means to said applications;

license access means connected to said application and accessible from said agents for decoding and encoding said license token means from said license server means, said license access means being integrated with said applications;

licensing library means connected to said application and accessible from said agents for verifying said decoded license token means before access to said license is granted, said licensing library means being integrated with said applications; and

license binding means connected to said license server means and to said licensing library means for constructing a binding file, said binding file informing said licensing library means which of said license server means may grant a license to said application.

13. The system as defined in claim 12, wherein said licensing library means are located on the same agents as said applications.

14. The system as defined in claim 12, wherein said license sever means are located on the same agents as said licensing library means.

15. The system as defined in claim 12, wherein each said license token means contains licensing information for at least one version of each of said applications.

16. The system as defined in claim 12, wherein the contents of said license means is encrypted.

17. The system as defined in claim 12, wherein said license token means is passed between said license server

means and said licensing library means for a predetermined time period.

18. The license token means as defined in claim 17, wherein, during said predetermined time period, only one of said applications may check out one said license token means.

19. The system as defined in claim 12, wherein said license server means further transmit said license token means to said licensing library means.

20. The system as defined in claim 12, wherein said license access means decodes the contents of said license token means before said licensing library means verifies said license token means.

21. The system as defined in claim 12, wherein said license access means encodes said license token means after said licensing library verifies said license token means and prior to returning said license token means to said license server means.

22. The system as defined in claim 12, wherein said license binding means constructs said binding file by contracting each said license server means to request for a list of applications it serves, said binding file containing said list of applications available from said license server means.

23. In a computer network environment including a plurality of software applications licensed to run on at least one network of agents, said applications located on

said agents wherein use of the application on a particular agent is permitted upon the grant of a license, said license being requested by a user from said agent of said applications, a system for protecting against the unauthorized use of said applications substantially as hereinbefore described with reference to the accompanying drawings.

Published 1991 at The Patent Office, State House, 66/71 High Holborn, London WC1R 4TP. Further copies may be obtained from
Sales Branch, Unit 6, Nine Mile Point, Cwmfelinfach, Cross Keys, Newport, NP1 7HZ. Printed by Multiplex techniques ltd, St Mary Cray, Kent.

(12) UK Patent Application (19) GB (11) 2 309 364 (13) A

(43) Date of A Publication 23.07.1997

(21) Application No 9700921.1

(22) Date of Filing 17.01.1997

(30) Priority Data
(31) 08588848 (32) 19.01.1996 (33) US

(71) Applicant(s)
Northern Telecom Limited

(Incorporated in Canada - Quebec)

World Trade Center Of Montreal,
380 St Antoine Street West, 8th Floor, Montreal,
Quebec H2Y 3Y4, Canada

(72) Inventor(s)
David Allan
Liam Casey
Adrian Jones

(74) Agent and/or Address for Service
M C Dennis
Nortel Patents, London Road, HARLOW, Essex,
CM17 9NA, United Kingdom

(51) INT CL^6
H04L 9/30

(52) UK CL (Edition O)
H4P PDCSC
U1S S2204 S2208 S2209

(56) Documents Cited
EP 0328232 A2 WO 95/23468 A1

(58) Field of Search
UK CL (Edition O) H4P PDCSA PDCSC
INT CL^6 H04L 9/30 9/32
Online:WPI,INSPEC

(54) Public/private key encryption/decryption

(57) In a hybrid fiber-coax distribution network, communications between a central station and particular end stations are encrypted using a working key (WK) of a symmetric encryption scheme. The central station has a public and private key (PPK) of a PPK encryption scheme, and some of the end stations can also each have a respective PPK. To provide secure communications for each end station, if the end station has a PPK, then the respective WK is generated in the central station and communicated, encrypted using the end station's public key (PK), to the end station. Otherwise, the WK is generated in the end station and communicated, encrypted using the central station's PK, to the central station. An individual identifier for each end station, and a cryptographic signature at least for end stations not having a PPK, can be communicated to the central station for authentication of the end stations.

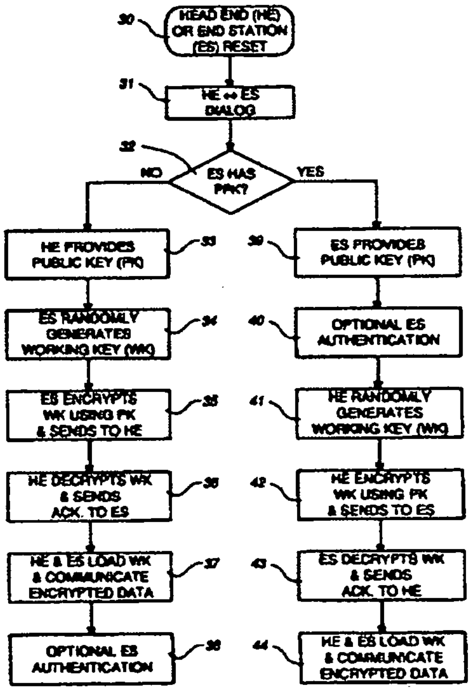

30
HEAD END (HE) OR END STATION (ES) RESET
31
HE ↔ ES DIALOG
32
ES HAS PPK?
NO
YES
HE PROVIDES PUBLIC KEY (PK)
33
ES RANDOMLY GENERATES WORKING KEY (WK)
34
ES ENCRYPTS WK USING PK & SENDS TO HE
35
HE DECRYPTS WK & SENDS ACK. TO ES
36
HE & ES LOAD WK & COMMUNICATE ENCRYPTED DATA
37
OPTIONAL ES AUTHENTICATION
38
39
ES PROVIDES PUBLIC KEY (PK)
40
OPTIONAL ES AUTHENTICATION
41
HE RANDOMLY GENERATES WORKING KEY (WK)
42
HE ENCRYPTS WK USING PK & SENDS TO ES
43
ES DECRYPTS WK & SENDS ACK. TO HE
44
HE & ES LOAD WK & COMMUNICATE ENCRYPTED DATA

Fig. 2

GB 2 309 364 A

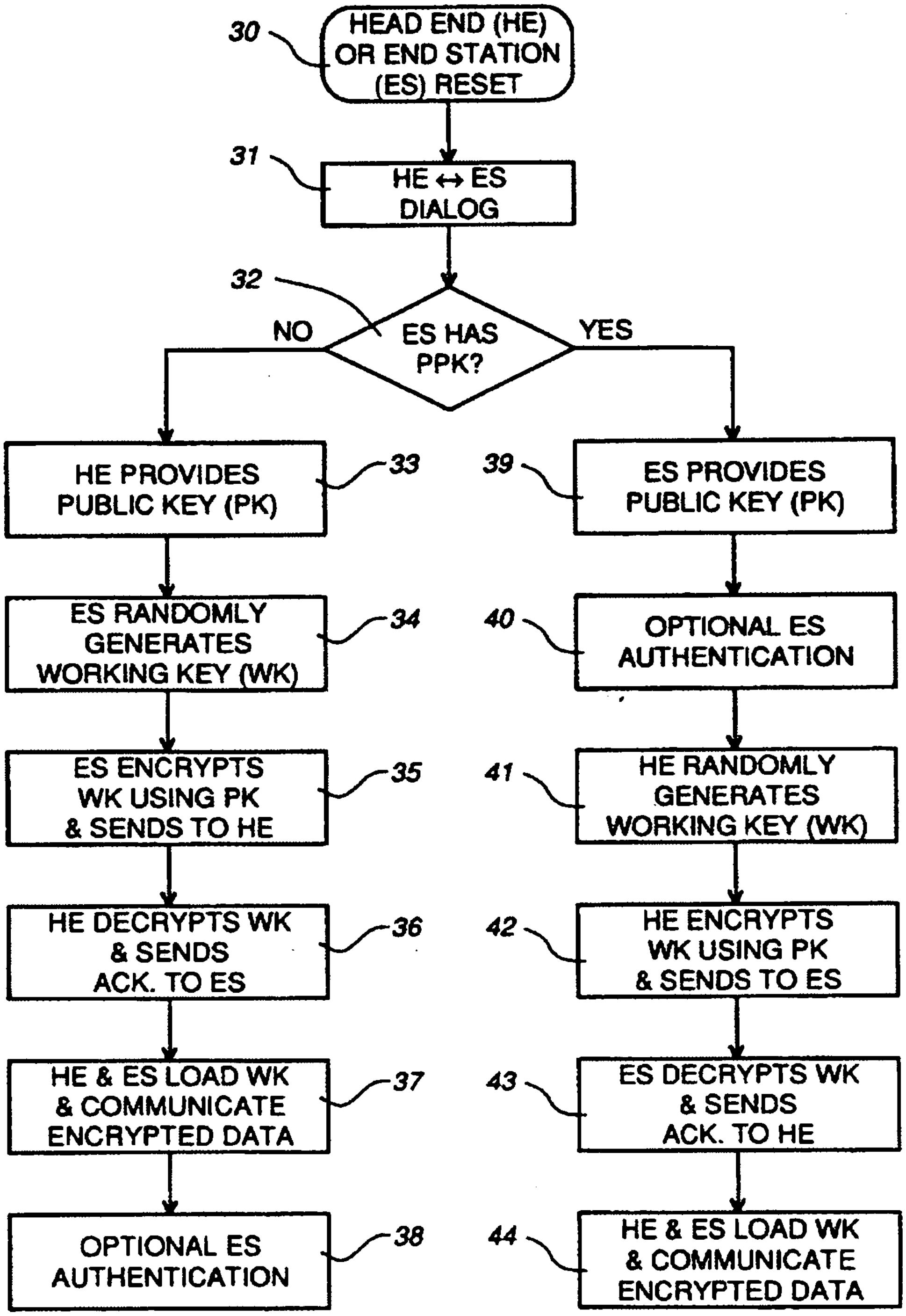

30
HEAD END (HE) OR END STATION (ES) RESET
31
HE ↔ ES DIALOG
32
ES HAS PPK?
NO
YES
HE PROVIDES PUBLIC KEY (PK)
33
ES RANDOMLY GENERATES WORKING KEY (WK)
34
ES ENCRYPTS WK USING PK & SENDS TO HE
35
HE DECRYPTS WK & SENDS ACK. TO ES
36
HE & ES LOAD WK & COMMUNICATE ENCRYPTED DATA
37
OPTIONAL ES AUTHENTICATION
38
39
ES PROVIDES PUBLIC KEY (PK)
40
OPTIONAL ES AUTHENTICATION
41
HE RANDOMLY GENERATES WORKING KEY (WK)
42
HE ENCRYPTS WK USING PK & SENDS TO ES
43
ES DECRYPTS WK & SENDS ACK. TO HE
44
HE & ES LOAD WK & COMMUNICATE ENCRYPTED DATA

Fig. 2

FACILITATING SECURE COMMUNICATIONS IN A DISTRIBUTION NETWORK

This invention relates to methods of facilitating secure communications in a distribution network, such as for example a coaxial cable or hybrid fiber-coax (HFC) network.

Background of the Invention

A distribution network, such as an HFC network in which data is communicated to subscriber end stations via optical fiber and coaxial distribution cables, is a point-to-multipoint network in which data addressed to and intended for any particular subscriber is also inevitably supplied via the network to other subscribers. If the data is not scrambled or encrypted, it can be easily monitored by these other subscribers, leading to a loss of subscriber privacy and a loss of revenues for data suppliers when the data (e.g. television programs) is supplied for a fee. Accordingly, it is important to provide a desired level of security in the data communications in a distribution network.

While various encryption and decryption schemes are known, these have a number of disadvantages associated with them in the environment of a distribution network. A significant factor in this respect is the cost and security of subscriber end stations. As a distribution network will contain large numbers of subscriber end stations, it is commercially necessary that the cost of each end station be kept relatively low. It is therefore desirable to avoid incorporating expensive security schemes in the subscriber end stations. However, subscriber end stations are also easily subject to theft, tampering, and duplication, so that complicated schemes have been considered necessary to provide adequate security.

For example, a security scheme can be implemented using an encryption key which can be stored in the subscriber end station. To prevent access to the encryption key, the store in the subscriber end station, and data lines to and from this store, must also be made physically secure. This leads to extra complexity and costs. Different subscribers may have differing security and privacy needs, which makes it desirable for the network to accommodate differing security schemes and end station costs.

A further security-related desirable aspect of a distribution network is an ability for authentication of subscriber end stations, typically using a unique end station identity which can be physically incorporated (e.g. hard wired) into the end station during manufacture.

Encryption schemes can be divided into those involving public and private keys (PPK) and those involving symmetric keys. In PPK schemes, a first station can distribute its public key, in accordance with which a second station can encrypt data and send the encrypted data to the first station, which decrypts the data using its private key. Because the private key is retained at the first station, and is not practically discoverable by other parties, PPK schemes are considered to be secure. However, the encryption and decryption processes are relatively slow, so that such schemes are not practical for encryption of real-time high-speed data, such as television program signals, for which distribution networks are primarily intended.

In symmetric key schemes, a single key, referred to as a working key, is used by both of first and second stations to encrypt and decrypt data being communicated between the stations. The nature of the working key is such that encryption of real-time high-speed data, such as television program signals, is practical. However, these schemes require that the working key be present in both stations, and make it desirable for the working key to be periodically changed or updated. Thus symmetric key schemes require generation of a working key in one of the stations or in a third station referred to as a key distribution agent, and communication of the working key to the other station(s).

This communication itself presents a risk of the working key being insecure, and this risk increases with the frequency with which the working key is updated. It is also known to avoid this risk by using a PPK scheme for communication of a working key, and then to use the working key for data encryption.

An object of this invention is to provide a method of facilitating secure communications in a distribution network.

Summary of the Invention

One aspect of this invention provides a method of facilitating secure communications using encryption and decryption processes in a distribution network comprising a central station and a plurality of addressable end stations, in which communications from the central station addressed to and intended for a particular end station are delivered via the network to a plurality of end stations, wherein the central station has, and one or more of the end stations can each have, a respective public and private key (PPK) of a PPK encryption scheme, comprising the steps of:

(a) determining in communications between the central station and an end station whether the end station has a PPK, if so proceeding with step (b) and if not proceeding with step (c);

(b) at the central station, determining the public key (PK) of the end station, generating a working key (WK) for encryption of communications to the end station, encrypting the WK using the PK of the end station, and communicating the encrypted WK to the end station; at the end station, decrypting the WK using the private key of the end station; and proceeding with step (d);

(c) at the end station, determining the public key (PK) of the central station, generating a working key (WK) for encryption of communications to the central station, encrypting the WK using the PK of the central station, and communicating the encrypted WK to the central station; at the central station, decrypting the WK using the private key of the central station; and proceeding with step (d);

(d) using the WK to encrypt at the central station, and to decrypt at the end station, communications from the central station to the end station.

Another aspect of this invention provides a method of facilitating secure communications in a distribution network comprising a central station and a plurality of addressable end stations, in which communications from the central station addressed to and intended for a particular end station are delivered via the network to a plurality of end stations, wherein the central station has a public and private key (PPK) of a PPK encryption scheme and each end station has an individual identity (ID) and an individual cryptographic signature encrypted using a private key of a predetermined PPK encryption scheme, comprising the steps of: communicating the ID of an end station to the central station; at the end station, generating a working key (WK) for encryption of communications between the end station and the central station and encrypting the WK using the public key of the central station; communicating the encrypted WK from the end station to the central station; at the central station, decrypting the encrypted WK using the private key of the central station; communicating the cryptographic signature of the end station to the central station; and at the central station, decrypting the cryptographic signature using a public key of the predetermined PPK scheme for authentication of the end station.

Brief Description of the Drawings

The invention will be further understood from the following description with reference to the accompanying drawings, in which:

Fig. 1 illustrates parts of a distribution network to which the invention is applied; and

Fig. 2 is a flow chart illustrating steps of a method for facilitating secure communications in the network in accordance with the invention.

Detailed Description

The invention is described below in the context of a hybrid fiber-coax (HFC) distribution network in which signals are distributed from a central station or head end (HE) to a large number of subscriber end stations (ES) via optical fibers and coaxial cables in known manner. An example of such a network is described in Warwick United States Patent No. 5,408,259 issued April 18, 1995 and entitled "Data Modulation Arrangement For Selectively Distributing Data". Typically in such a network digital data communications are provided between any ES and the HE using asynchronous transfer mode (ATM) cells which are communicated in both directions, i.e. downstream from the HE to the ES and upstream from the ES to the HE, using suitable modulation schemes and carrier frequencies outside the bands used for analog television signals also carried on the coaxial cables. However, it is observed that the invention is equally applicable to other forms of distribution network.

Referring to Fig. 1, there is illustrated parts of a distribution network in which many end stations, only two of which are shown and are referenced 10 and 12, are connected via branched cables 14 of the distribution network to a head end 16, via which the end stations have access to a network (not shown) which for example supplies digital television program signals subscribed to by end station subscribers. The cables 14 can comprise both optical fiber and coaxial cables forming a hybrid fiber-coax arrangement, on which the digital signals can be communicated in known manner using ATM cells.

As can be appreciated from the illustration in Fig. 1, signals communicated by the head end 16 and intended for any particular end station will actually be delivered via the cables 14 to all of the end stations. For secure and/or private communication of the signals, the head end 16 includes an encryption engine 18 which encrypts the signals in accordance with a working key known only by the head end and the intended end station, which also includes an encryption engine 20 which decrypts the signals for use. These working keys are similarly used for communications in the opposite direction, from the end station to the head end 14. The working keys of this symmetric key encryption scheme are provided in the head end and the end station in a manner which is described in detail below.

The end stations 10 and 12 are of two types, with differing levels of security to enable different security needs of subscribers to be accommodated. The end station 12 represents a relatively secure end station, which includes its own public and private keys of a PPK encryption scheme. As explained in the introduction, such an end station has a relatively high complexity and cost, because of the need for secure storage of the keys and operation of the PPK encryption. Other end stations, which do not have their own public and private keys and accordingly can be provided at a much lower cost, are represented by the end station 10. The network as a whole may have an arbitrary mix of these two types of end station.

Each end station 10 or 12 also has an individual, unique identity number, which is stored (e.g. hard wired) into the ES during its manufacture. This is referred to as a global ID (identity). The global IDs of all of the end stations are stored in a database 22, which can be colocated with the head end 16 or separately from it and with which the head end 16 communicates via a path 24. The head end 16 also has its own public and private keys of a PPK encryption scheme.

Fig. 2 shows steps of a process which is followed in order to set up secure communications between the head end 16 and one of the end stations 10 or 12. This process takes place between the head end and the respective end station without involvement of any other node such as a central key distribution agent, and is described below as being initiated in each case following any reset (e.g. following a power-up) of either the head end 16 or the respective end station. Consequently, the working key which is used for encrypting the communications between the head end and the end station is changed on any reset. However, the same process can alternatively or additionally be carried out on demand, and/or periodically to provide periodic changes of the working key. It is also observed that the encrypted communications take place between the encryption engines 18 in the head end 16 and 20 in the respective end station 10 or 12, and communications on the network access side of the head end 16 are not subject to the same encryption.

In Fig. 2, a block 30 represents a reset of the head end (HE) or end station (ES), in response to which, as shown by a block 31 in Fig. 2, a dialog or handshake is carried out between the HE and the ES to establish communications between them. These communications are effected using unencrypted ATM cells using addresses of the end station and the head end. As a part of this dialog, as shown by a block 32 in Fig. 2 the head end 16 interrogates the end station to determine whether or not the end station has its own public and private keys. If not, i.e. if the end station is an end station 10 as described above, then the process continues with successive blocks 33 to 38 in Fig. 2. If the interrogation establishes that the end station is an end station 12 having its own public and private keys, then the process instead continues with blocks 39 to 44 in Fig. 2.

In the former case of an end station 10, as shown by the block 33 the head end 16 communicates its public key (PK) to the end station 10; this communication can form part of the dialog block 31. The end station 10 randomly generates (block 34) a working key (WK) for communicating signals in a symmetric key encryption scheme, and encrypts (block 35) this working key in accordance with the supplied public key, sending the encrypted working key in a message to the head end 16. The head end 16 decrypts (block 36) the encrypted working key from this message in accordance with its private key, which is not known to others so that the communication of the working key from the end station 10 to the head end 16 is secure, and optionally but preferably sends an acknowledgement to the end station 10. As shown by the block 37, the head end 16 and the end station 10 then load their encryption engines 18 and 20 respectively with the working key, and thereafter (until this process is repeated, for example in response to a subsequent reset at either end) communications between them take place with data encrypted in accordance with the working key. An optional additional step represented by the block 38 provides for authentication of the end station 10 in a manner described below.

Conversely, in the latter case of an end station 12, as shown by the block 39 the end station 12 communicates its public key (PK) to the head end 16; this communication can form part of the dialog block 31. An optional authentication step for the end station

12 can be carried out by the head end 16 as represented by the block 40 in a manner described below. The head end 16 randomly generates (block 41) a working key (WK) for communicating signals in a symmetric key encryption scheme, and encrypts (block 42) this working key in accordance with the supplied public key of the end station 12, sending the encrypted working key in a message to the end station 12. The end station 12 decrypts (block 43) the encrypted working key from this message in accordance with its private key, which is not known to others so that the communication of the working key from the head end 16 to the end station 12 is secure, and optionally but preferably sends an acknowledgement to the head end 18. As shown by the block 44, the head end 16 and the end station 12 then load their encryption engines 18 and 20 respectively with the working key, and thereafter (until this process is repeated, for example in response to a subsequent reset at either end) communications between them take place with data encrypted in accordance with the working key.

It can be seen from the above description that, in the relatively secure but more expensive situation in which the end station 12 includes its own public and private keys, these are used for communicating a working key generated in the head end, whereas in the other case the end station 10 generates the working key and this is communicated to the head end using the latter's public key.

The optional step of authentication of the end station 12 in the block 40 as described above can make use of the global ID of the end station 12 together with data in the database 22, in which the public key of the end station 12 is stored in association with this global ID. As part of the dialog block 31, the end station communicates its global ID to the head end 16. In the step 40, therefore, the head end 16 can communicate via the path 24 with the database 22 to confirm that the public key which it has received from the end station 12 in the step 39 matches that stored in the database 22 for this end station's global ID, the subsequent steps 41 to 44 only being followed if this authentication step is successful.

Alternatively, or in addition, the optional end station authentication step of block 40 can comprise the steps of the head end sending an unencrypted message to the end station 12 with a request that it be cryptographically signed. In accordance with this request, the end station 12 produces a digest of the message using a known hashing function (thereby reducing the data to be encrypted), encrypts this digest in accordance with its private key, and sends the encrypted message digest to the head end 16. The head end 16 then decrypts this in accordance with the public key of the end station, retrieved from the database 22, to confirm the digest of its original message which the head end also produces using the hashing function.

It can be seen that, alternatively, the steps represented by the blocks 39 and 40 in Fig. 2 could be replaced by a single step in which the head end 16 determines the public key of the end station 12 from the database 22 in accordance with the global ID of the end station 12 supplied in the dialog 31, without any authentication of the end station or any communication of the public key from the end station 12.

The above sequences provide a particularly strong or secure authentication of the end station 12. For the end station 10 which does not have its own public and private keys, a weaker but still valuable authentication can be provided as shown by the block 38. The authentication block 38 is shown in Fig. 2 as the final block in the process because this enables the exchange of data in the authentication process to be encrypted in accordance with the working key, but this authentication step could alternatively be provided anywhere else in the sequence of steps from the blocks 31 to 37.

For this optional authentication step, the end station 10 is manufactured (e.g. hard wired) with not only its global ID, but also a cryptographic signature. Conveniently, the end station 10 is manufactured with a certificate comprising data including the global ID of the end station and the public key of the manufacturer and a cryptographic signature comprising an encryption, in accordance with the private key of the manufacturer, of a digest of that data produced using a known hashing function. The public key of the manufacturer can also or instead be stored in the database 22. The optional end station authentication step of the block 38 comprises a communication of the cryptographic signature from the end station 10 to the head end 16 (as explained above this could be a part of the dialog 31 or any later step, but the encryption after the block 37 obstructs public observation in the network of cryptographic signatures). The head end 16 then confirms the authenticity of the end station 10 by decrypting the cryptographic signature using the manufacturer's public key, producing a digest from the same data (global ID and public key, both of which can be communicated in the dialog step 31 or later) and the known hashing function, and matching these.

This is a relatively weak authentication, in that identical copies of the end station 10, including duplicated data and cryptographic signatures, could operate at different times on the network without this being detected. However, simultaneous operation of two or more such duplicates would be detected by the fact that two or more end stations would be supplying the same global ID which is supposedly unique. Thus even such a weak authentication is valuable especially in detecting illicit large-scale duplication of end stations.

The processes in accordance with the invention as described above provide a number of significant advantages over known configurations. In particular, requirements for secure storage of public and private keys are minimized in the network as a whole, and eliminated for the end stations 10 which can accordingly be provided at relatively lower cost. At the same time, end stations 12 with greater security can be provided, and the head end 16 can operate simultaneously with both types of end station. This, combined with optional authentication of the end stations as described above, enables different degrees of security to be easily provided in the network in accordance with service requirements.

Furthermore, renewal of the working keys at reset is simpler than providing time-based schedules for changing encryption keys, and key exchanges take place only between the head end and the end station which use the keys, thereby enhancing security compared with distribution of keys from a key distribution agent. In addition, all of the data flowing between the head end and any particular end station 10 or 12, between successive resets, can be encrypted using a single working key, thereby simplifying the encryption and decryption processes. However, it is observed that different working keys could be generated, communicated, and used in the same manner as described above for encrypting and decrypting different types of information, or different services, for a single end station 10 or 12.

Although particular embodiments of the invention have been described in detail, it should be appreciated that numerous modifications, variations, and adaptations may be made without departing from the scope of the invention as defined in the claims.

WHAT IS CLAIMED IS:

1. A method of facilitating secure communications using encryption and decryption processes in a distribution network comprising a central station and a plurality of addressable end stations, in which communications from the central station addressed to and intended for a particular end station are delivered via the network to a plurality of end stations, wherein the central station has, and one or more of the end stations can each have, a respective public and private key (PPK) of a PPK encryption scheme, comprising the steps of:

(a) determining in communications between the central station and an end station whether the end station has a PPK, if so proceeding with step (b) and if not proceeding with step (c);

(b) at the central station, determining the public key (PK) of the end station, generating a working key (WK) for encryption of communications to the end station, encrypting the WK using the PK of the end station, and communicating the encrypted WK to the end station; at the end station, decrypting the WK using the private key of the end station; and proceeding with step (d);

(c) at the end station, determining the public key (PK) of the central station, generating a working key (WK) for encryption of communications to the central station, encrypting the WK using the PK of the central station, and communicating the encrypted WK to the central station; at the central station, decrypting the WK using the private key of the central station; and proceeding with step (d);

(d) using the WK to encrypt at the central station, and to decrypt at the end station, communications from the central station to the end station.

2. A method as claimed in claim 1 wherein each end station has an individual identity (ID) and step (a) includes the step of communicating the ID of the end station to the central station.

3. A method as claimed in claim 2 wherein in step (b) the PK of the end station is determined by the central station from a database using the ID of the end station.

4. A method as claimed in claim 1, 2, or 3 wherein step (b) further comprises an end station authentication step comprising the steps of communicating an unencrypted message from the central station to the end station, producing an encrypted message at the end station using the private key of the end station, communicating the encrypted message to the central station, decrypting the message at the central station using the PK of the end station, and comparing the decrypted message with the original message.

5. A method as claimed in claim 4 wherein in step (b) the end station authentication step is carried out before the step of communicating the encrypted WK to the end station.

6. A method as claimed in any of claims 1 to 5 wherein in step (b) the PK of the end station is communicated to the central station from the end station.

7. A method as claimed in claims 2 and 6 wherein in step (b) the PK of the end station is verified by the central station from a database using the ID of the end station.

8. A method as claimed in any of claims 1 to 7 wherein a plurality of end stations which do not have a PPK each have an individual cryptographic signature encrypted using a private key of a predetermined PPK scheme, step (a) or (c) includes the step of communicating the cryptographic signature of the end station to the central station, and step (c) further comprises an end station authentication step comprising, at the central station, decrypting the cryptographic signature using a public key of the predetermined PPK scheme.

9. A method as claimed in claims 2 and 8 wherein the individual cryptographic signature comprises an encryption of data derived from the ID of the respective end station.

10. A method as claimed in claim 8 or 9 wherein the predetermined PPK scheme uses a private key and a public key of a source of the end station.

11. A method as claimed in claim 8, 9, or 10 wherein the cryptographic signature is communicated to the central station in step (c).

12. A method as claimed in claim 11 and including the steps of encrypting the cryptographic signature at the end station, and decrypting the encrypted cryptographic signature at the central station, using the WK.

13. A method as claimed in any of claims 1 to 12 and further comprising the step of using the WK to encrypt at the end station, and to decrypt at the central station, communications from the end station to the central station.

14. A method of facilitating secure communications in a distribution network comprising a central station and a plurality of addressable end stations, in which communications from the central station addressed to and intended for a particular end station are delivered via the network to a plurality of end stations, wherein the central station has a public and private key (PPK) of a PPK encryption scheme and each end station has an individual identity (ID) and an individual cryptographic signature encrypted using a private key of a predetermined PPK encryption scheme, comprising the steps of:

communicating the ID of an end station to the central station;

at the end station, generating a working key (WK) for encryption of communications between the end station and the central station and encrypting the WK using the public key of the central station;

communicating the encrypted WK from the end station to the central station;

at the central station, decrypting the encrypted WK using the private key of the central station;

communicating the cryptographic signature of the end station to the central station; and

at the central station, decrypting the cryptographic signature using a public key of the predetermined PPK scheme for authentication of the end station.

15. A method as claimed in claim 14 wherein the individual cryptographic signature comprises an encryption of data derived from the ID of the respective end station.

16. A method as claimed in claim 14 or 15 wherein the predetermined PPK scheme uses a private key and a public key of a source of the end station.

17. A method as claimed in claim 14, 15, or 16 wherein the step of communicating the cryptographic signature of the end station to the central station comprises the steps of encrypting the cryptographic signature at the end station using the WK, communicating the encrypted cryptographic signature from the end station to the central station, and decrypting the encrypted cryptographic signature at the central station using the WK.

18. A method of facilitating secure communications in a distribution network, substantially as hereinbefore described with reference to Figs 1 and 2 of the accompanying drawings.

The Patent Office

12

Application No: GB 9700921.1
Claims searched: 1-13

Examiner: Mr B J Spear
Date of search: 19 March 1997

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.O): H4P (PDCSC)

Int Cl (Ed.6): H04L 9/30

Other: Online: WPI, INSPEC

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
	NONE	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
		E	Patent document published on or after, but with priority date earlier than, the filing date of this application.
&	Member of the same patent family		

The Patent Office

i2

Application No:	GB 9700921.1	**Examiner:**	Mr B J Spear
Claims searched:	14-17	**Date of search:**	21 May 1997

Patents Act 1977
Further Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.O): H4P (PDCSA)

Int Cl (Ed.6): H04L 9/32

Other: Online: WPI, INSPEC

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	EP0328232A2 (Fischer)	-
A	WO 95/23468A1 (Merdan)	-

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
		E	Patent document published on or after, but with priority date earlier than, the filing date of this application.
&	Member of the same patent family		

09

(12) UK Patent Application (19) GB (11) 2 316 503 (13) A

(43) Date of A Publication 25.02.1998

(21) Application No 9617596.3

(22) Date of Filing 22.08.1996

(71) Applicant(s)
ICL Personal Systems Oy

(Incorporated in Finland)

PO Box 458, SF-00101 Helsinki, Finland

(72) Inventor(s)
Tapani Lindgren

(74) Agent and/or Address for Service
S M Dupuy
International Computers Limited, Cavendish Road, STEVENAGE, Hertfordshire, SG1 2DY, United Kingdom

(51) INT CL^6
G06F 1/00

(52) UK CL (Edition P)
G4A AAP

(56) Documents Cited
GB 2236604 A **EP 0332304 A2** **WO 93/11480 A1**
US 5375206 A **US 4924378 A**

(58) Field of Search
UK CL (Edition O) **G4A AAP**
INT CL^6 **G06F**

(54) Software licence management

(57) A software licence management method and system is for a computer system including at least one server (1,5) and particularly for a plurality of computers connected via a network. Before a service (2) can offer functionality to a user it has to check that the user has a licence for that service. A licensing subsystem (3) has associated with it a ticket database (4) that hold tickets corresponding to existing licences. Tickets, if available, are issued to a service on request, thereby verifying the existence of a licence. The receipt of a ticket allows a service to offer functionality.

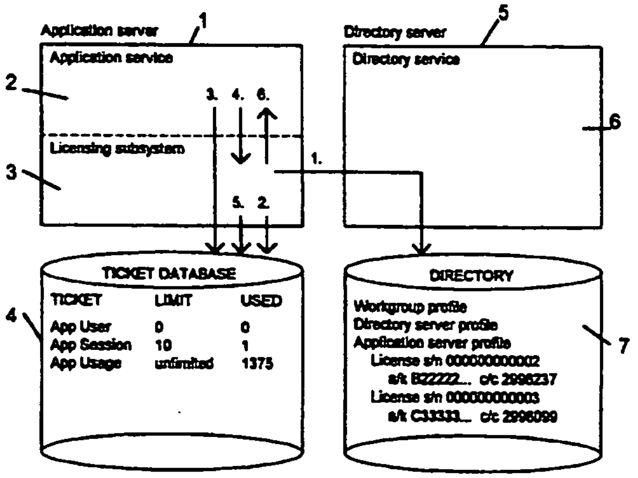

Application server
1
Application service
2
3. 4. 6.
Licensing subsystem
3
1.
5. 2.
Directory server
5
Directory service
6
TICKET DATABASE
TICKET LIMIT USED
App User 0 0
App Session 10 1
App Usage unlimited 1375
4
DIRECTORY
Workgroup profile
Directory server profile
Application server profile
License s/n 000000000002
s/t B22222... c/c 2996237
License s/n 000000000003
s/t C33333... c/c 2996099
7

FIG 1

GB 2 316 503 A

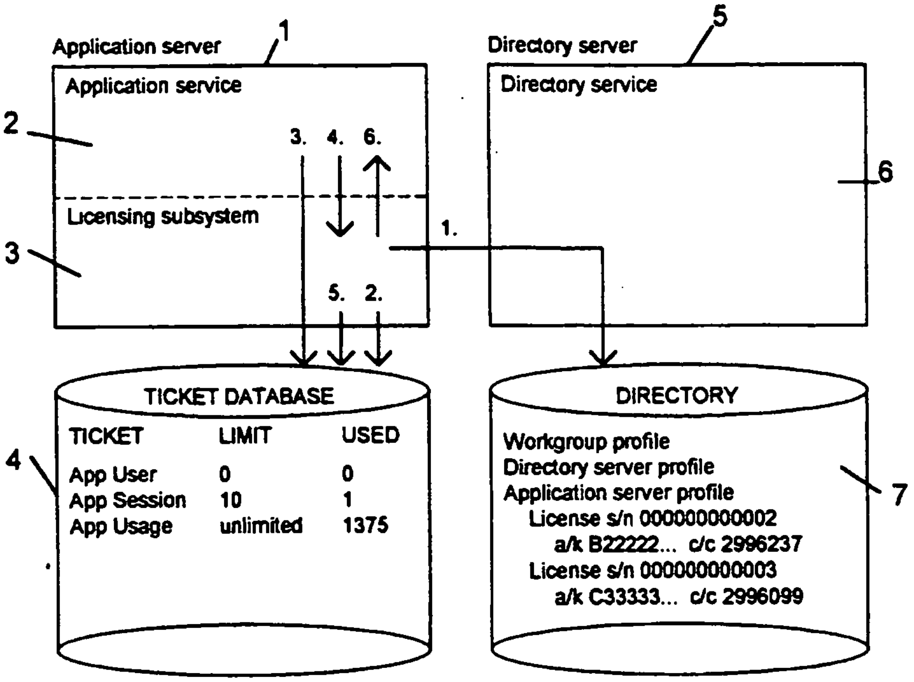

Application server
1
Application service
2
3. 4. 6.
Licensing subsystem
3
1.
5. 2.
Directory server
5
Directory service
6
TICKET DATABASE
TICKET LIMIT USED
4
App User 0 0
App Session 10 1
App Usage unlimited 1375
DIRECTORY
Workgroup profile
Directory server profile
Application server profile
License s/n 000000000002
a/k B22222... c/c 2996237
License s/n 000000000003
a/k C33333... c/c 2996099
7

FIG 1

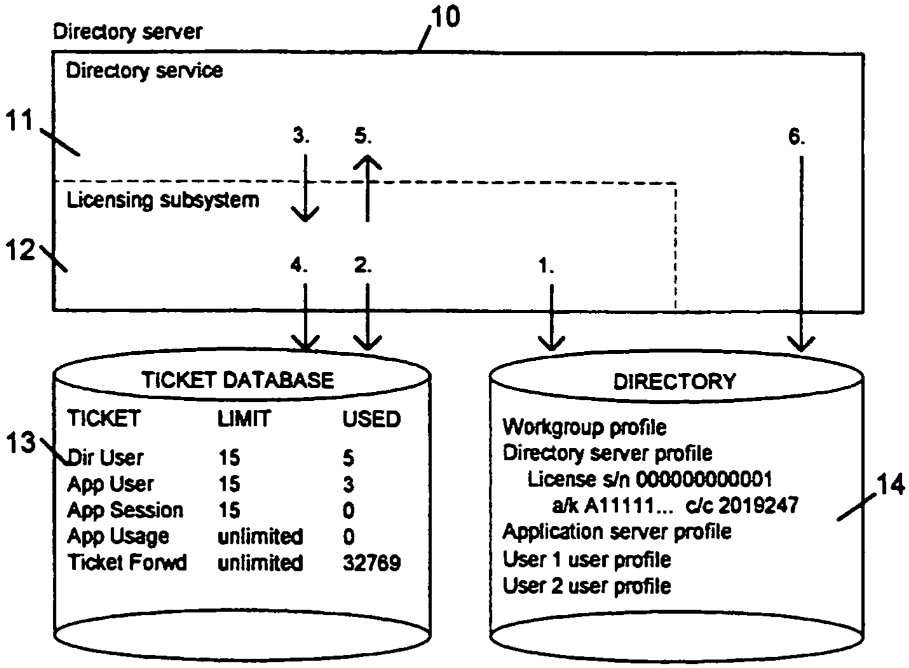

Directory server
10
Directory service
11
3.
5.
6.
Licensing subsystem
12
4.
2.
1.
TICKET DATABASE
TICKET
LIMIT
USED
13
Dir User
15
5
App User
15
3
App Session
15
0
App Usage
unlimited
0
Ticket Forwd
unlimited
32769
DIRECTORY
Workgroup profile
Directory server profile
License s/n 000000000001
a/k A11111... c/c 2019247
Application server profile
User 1 user profile
User 2 user profile
14

FIG 2

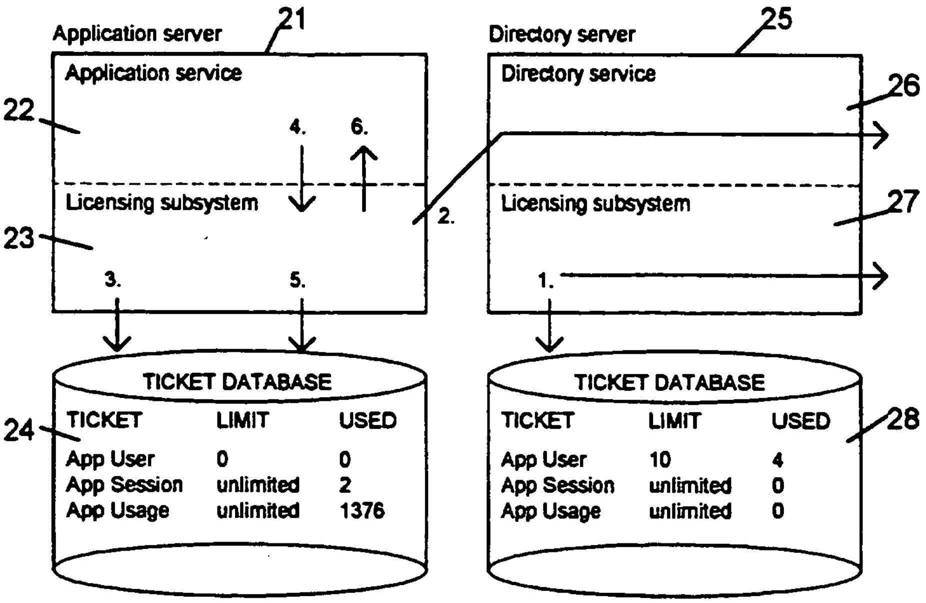

Application server
21
Application service
22
4.
6.
Licensing subsystem
23
3.
5.
2.
Directory server
25
Directory service
26
Licensing subsystem
27
1.
TICKET DATABASE
24
TICKET LIMIT USED
App User 0 0
App Session unlimited 2
App Usage unlimited 1376
TICKET DATABASE
28
TICKET LIMIT USED
App User 10 4
App Session unlimited 0
App Usage unlimited 0

FIG 3

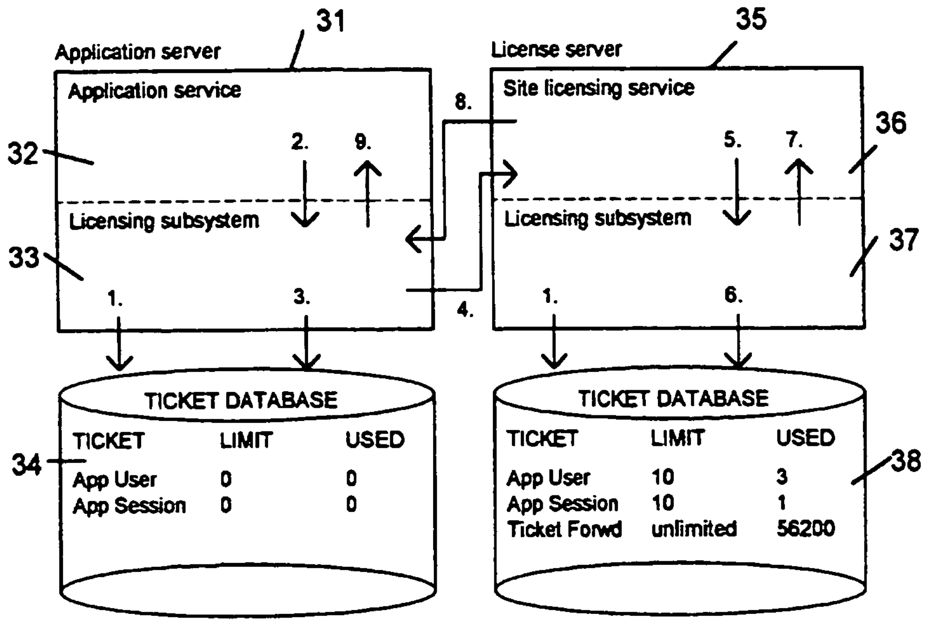

31
Application server
Application service
32
2.
9.
Licensing subsystem
33
1.
3.
8.
4.
35
License server
Site licensing service
36
5.
7.
Licensing subsystem
37
1.
6.
TICKET DATABASE
TICKET LIMIT USED
34
App User 0 0
App Session 0 0
TICKET DATABASE
TICKET LIMIT USED
App User 10 3
App Session 10 1
Ticket Forwd unlimited 56200
38

FIG 4

2316503

SOFTWARE LICENCE MANAGEMENT

This invention relates to software licence management and in particular to licence management for software running on a plurality of computers connected via a network.

Conventionally, licences have been provided by software vendors as separate licences for individual workstations or as a single licence for a number of workstations. Various schemes have been proposed in order to try and make unlicensed software unusable, in particular pirated (illegal) copies of software. Other schemes have been proposed such as in order to achieve low initial software costs but licensing royalties consistent with the extent of use, in order not to deter low-usage users from purchasing particular forms of software, and thus to reduce piracy, whilst still enabling a vendor to collect higher dues from high-usage users.

The present invention is, particularly, concerned with a distributed system consisting of various server and client programs running on various computers which are connected via a local or wide area network, and an object is to provide server software licensing which ensures that all software running in the network has been purchased legally.

According to one aspect of the present invention there is provided a software licence management method for use with a computer system including at least one server, the method being such that before a service can offer functionality to a user, the said service shall verify that the user has a licence for said service, and wherein the computer system further includes a licensing subsystem with which are associated service tickets corresponding to existing licences, the method including the steps of the said service requesting a respective service ticket from the licensing subsystem prior to offering functionality to the user, and the licensing subsystem issuing a said service ticket to the said service, if one is available, thereby verifying the licence exists and allowing the said service to offer functionality.

According to another aspect of the present invention there is provided a computer system including at least one server and a software licence management system, the management system being such that before a service can offer functionality to a user, the service shall verify that the user has a licence for said service, the management system including a licencing subsystem with which are associated service tickets corresponding to existing licences, and the management system being such that a said service ticket is issued to a service, if one is available, upon request by the service, thereby verifying existence of a licence and allowing the said service to offer functionality.

Embodiments of the invention will now be described with reference to the accompanying drawings, in which:

Figure 1	Illustrates obtaining a session or usage ticket for an application,
Figure 2	Illustrates obtaining a user ticket for a directory server,
Figure 3	Illustrates independent licence sharing, and
Figure 4	Illustrates licence sharing with a site licencing service,

Various terms used in the following will first be defined. For the purposes of the description the software is considered to relate to a Groupware Office system which provides various facilities including mail, for example.

Definitions

"Server"	An instance of server software running on a server computer. Usually only one such instance runs on any one computer. Each "server" implements one or more collections of related functions called "function sets", examples of which are directory, mail, library etc. The "directory function set" includes functions to access a database that contains information about the Groupware Office system.
"Client"	Any piece of software that connects to the "server" using a "client-server protocol" to access the functions offered by the "server". A "client" may be a program run by a user on a workstation, or a part of any other program.
"Session"	An instance of client-server dialogue between one "client and one "server". Each "session" allows the "client" to use the functions of one or more "function sets".
"Directory Server"	A server that implements the directory function set.
"Mail Server"	A server that implements the mail function set. [A server may be a directory server and a mail server simultaneously.]
"User"	A person (actual or virtual) listed in the database of a directory server.

"User profile"	The information pertaining to one user stored in a directory database entry, such as the name of the user, user authentication information, the list of servers and function sets the user is permitted to access, etc.
"Server profile"	The name and network address of a server and the list of services offered by it, as stored in a directory database entry.
"Service profile"	Information stored in a directory database entry about one service in one server. If the same type of service is offered by more than one server, each instance has its own profile.
"Site profile"	Information stored in a directory database entry about one site.
"Site"	A set of servers connected to a single directory server. Each server belongs to exactly one site, and each site has exactly one directory server. Other servers in the site are optional, usually unlimited in number, and sometimes called member servers or application servers.
"Enterprise"	A set of sites that share their directory databases. The directory servers in each site replicate directory information to other directory servers in the enterprise. Each directory server contains both "local" and "external" information. One of the directory servers, the "enterprise directory server" controls the others, which are

	"site directory servers".
"Subsystems"	Collections of programs and/or subroutines that perform a set of interrelated functions. Some subsystems implement a function set within a server program, while others run independently as stand-alone applications. Many subsystems are collections of common subroutines called by other subsystems. Client programs are also subsystems.
"Subsystem id"	A respective unique number identifying every type of subsystem. Some systems use two ids, a "real subsystem id" when dealing with licensing issues and an "alias subsystem id" when performing a task on behalf of a virtual entity, such as "generic gateway no 9". Not every subsystem software needs to be purchased individually. Most collections of subroutines can be used freely by other subsystems.
"Services"	Those subsystems that need to be explicitly purchased. Service types may include directory service, mail service, fax gateway, X.400 gateway, enterprise option, library service, power library option, etc. Each service is located in one server, either as a function set of the server program or a standalone application running in the same computer. Many services of the same type can exist in different servers.

The software licence management system of the present invention proceeds from the premise that before offering any usable functionality to the users, services shall verify from a "licensing subsystem" that a licence for the service exists. To achieve that, it is proposed that the licensing subsystem holds "service tickets" and a service requests a permission to offer its functions to the user by requesting a corresponding "service ticket" be provided from the licensing subsystem. Each service knows that kind of tickets are needed to fulfil the service's functionality. The licensing subsystem has to keep track of the available licences and of the service tickets it has issued. A service ticket may be considered as partially the equivalent of a password in that one must be provided before a service can operate.

A "licence" is a permission to use one or more services within certain limits. Typically these limits are "license duration", which specifies the maximum length of the period when the licence may be used (the "active" period) and the "licence size", which specifies the maximum number of users of the licence. The interpretation of "number of users" varies from service to service. It may, for example, mean the number of users in the local directory that are allowed to use the service, or the number of concurrent sessions that are connected to the service. Licence duration and/or size may also be unlimited.

When a customer purchases a Groupware, for example, software product which employs the software licence management method and system of the invention from a supplier, as well as the media containing the software itself and associated documentation, there is obtained a single licence to one or more services. Each said product has a unique serial number. The license is supplied in the form of a licence agreement document on which the licence information is printed. This licence information consists of the serial number of the product and an "activation key" for the licence. The licence size and duration and the included services are encoded into the activation key.

The software license management method and system of the invention is such that the Groupware software may be copied and installed by the customer without any technical restrictions, but before any of the services can be used, a corresponding licence must be installed and activated. Licence installation consists of entering the license information (serial number and activation key) into the server profile of a server in the directory server's database, ie in the site directory, in the server profile of the server in question. Licence activation consists of setting the active period of the installed licence so that service tickets can be issued. Typically licence installation and licence activation are performed simultaneously by the server setup program. The license information is stored in the site directory, in the server profile of the server in question.

As will be appreciated, there also exists "evaluation licences" which allow a prospective customer to use a service for a short trial period before actually purchasing it. These licences typically have a very short duration and a relatively small size. The product serial numbers associated with such evaluation licences are not necessarily unique, since the licence information may be distributed on CD-ROMs or via public networks.

As mentioned above, each software product contains just one licence, although that licence may include a large number of services, for example, enough to build a complete Groupware Office site with all of the basic services. Alternatively, the licence may include just one service. Product with that kind of licence could be used to expand the capacity or functionality of an existing Groupware Office System.

The core of the process of designing a product is, therefore, determining what services will be included and the size and duration of the licence. This information is encoded into a number, the covert code, which may be a 7-digit number, for example. The building of the covert code is discussed in more detail hereinafter.

The amount of information that can be encoded into the covert code is limited by the size of the code. Therefore, there are some necessary restrictions on what kinds of licences are possible. The most obvious limitation is that the size and duration can only take certain discrete values. Also, the same size and duration will apply to all services covered by the licence. Another restriction is that only the most common groups of services can be combined freely into a multi-service licence. Other services will have to be licensed individually.

The covert code, which specifies the properties of the software licence, is thus a part of the product description in the logistics database. When the product is manufactured it has the unique serial number, referred to above, assigned to it. The actuation key for the license is calculated as a function of the serial number and the covert code using a secret algorithm. The serial number and activation key may be printed on a label, which is attached to the licence agreement document.

When creating a site, a customer must have a licence that includes a site creation ticket. This licence is installed for the directory server. The customer may also install additional licences for the directory server and for other servers. Each licence may apply to one or more services. Some licences are valid only in that server for which they are installed, whilst other licences may be shared with other servers at the same site (see later). Shared licences would usually be installed in the server profile of the directory server, although optionally, and with some restrictions, another server may be designated as the licence server. The serial number of the first licence installed in a directory server's profile can be used to identify the site uniquely. Thus the directory server is computer number 1. Other servers in the site will use the same site id but differing computer numbers for identification.

When a service program is about to execute an action which requires that a customer possesses a licence for that service, the service program must first obtain a corresponding service ticket from the licensing subsystem. The actions concerned are ones which are potentially profitable for the customer and may, for example, include namely:

setting up a new Groupware site;

creating a new user account;

setting up an instance of the mail service;

enabling mail usage for a user and creating a user mailbox;

starting a session between a mail UI client and the mail server;

sending a mail message;

relaying a mail message to an X.400 mail network.

Each kind of action requires a specific kind of service ticket. To obtain a ticket the service needs to specify the ticket type and the number of tickets. The service tickets are only identified by ticket type. There is a licensing subsystem in each server and it counts the number of different tickets in all available licences and keeps track of how many licences of each type are being used in the server.

The steps involved in obtaining various licences will now be described in greater detail. With respect to Figure 1 there will, firstly, be described the case of an application service running in a separate server from the directory server obtaining a session or usage ticket.

Figure 1 illustrates schematically an application server 1, providing an application service 2 and having a licensing subsystem 3, with an associated ticket database 4, a directory server 5 providing a directory service 6 and having an associated directory 7. The ticket database 4 has stored therein details of ticket types, the limit, if any, of the number of such tickets which are available and the number of used tickets for each type. The ticket types as illustrated are "App User" (Application User), "App Session", "App Usage". The directory 7 has stored therein, the "Workgroup profile", the "Directory server profile", the Application server profile. In the example illustrated, the application server has two associated licenses whose serial numbers (s/n) are 000000000002 and 000000000003, respectively, whose activation keys (a/n) are of the form B22222... and C33333..., respectively, and whose covert codes (c/c) are 2996237 and 2996099, for example, respectively.

When the licensing subsystem 3 on the application server starts, it fetches the application server's server profile from the directory service 6, 7 using the directory API (Application Programming Interface) (Step 1 in Figure 1). The licensing subsystem 3 analyses the licences and updates the limits of each ticket type in the local ticket database 4. The numbers of used tickets are not modified at this time, the old accumulated values being maintained (step 2).

When the application service 2 starts, and before any user logs in, it tells the licensing subsystem 3 to set the number of used session tickets to zero. This frees any session tickets that may have been left unreturned at the end of a session because of a system crash etc. (Step 3). The application service 2 then requests a service ticket (session or usage) from the licensing subsystem 3, since without a ticket it cannot proceed. (Step 4). The licensing subsystem checks the ticket availability in the local ticket database 4 and updates the used ticket count (step 5), to take into account the requested ticket, before issuing the ticket to the application service (step 6), which then proceeds since it has determined that there exists the appropriate licence.

In the embodiment of Figure 2, the procedure whereby a directory service obtains a ticket for adding a user to a directory is illustrated.

A directory server 10 provides a directory service 11 and includes a licensing subsystem 12 with an associated ticket database 13, the directory service 11 having an associated directory 14. The ticket database 13 has stored therein details of ticket types, the limit, if any, of the number of such tickets which are available, and the number of used tickets for each type. The ticket types are illustrated as "Dir User" (Directory User), "App User", "App Session", "App Usage" and "Ticket Forwd" (Ticket Forwarding). The directory 14 has stored therein the "Workgroup profile", the "Directory server profile", the "Application Server profile" and the user profile of two users, User 1 and User 2. The Directory server has a license serial number (s/n) 000000000001, with an actuation key (a/) of the form A11111..., and a corresponding covert code (c/c) 2019247, for example.

When the licensing subsystem 12 on the directory server 10 starts, it fetches the directory server's server profile directly from the directory 14 (step 1). The licensing subsystem 12 analyses the licenses and updates the limits of each ticket type in the ticket database 13. The numbers of used tickets are not modified at this time, the old accumulated values being maintained (step 2).

When it is desired to add a new user to the directory, the directory service 11 requests a user ticket from the licensing subsystem 12 (step 3). The licensing subsystem 12 checks ticket availability in the local ticket database 13 and updates the used ticket count (step 4) to take into account the requested ticket. The licensing subsystem 12 issues the requested user ticket to the directory service 11 (step 5). The directory service then adds the new user to the directory 14, that is it adds its user profile.

To ensure consistency, the directory service 11 may periodically count the number of users in the directory 14 and tell the licensing subsystem 12 to set the used ticket count accordingly.

When a licence is installed, the start time of its active period will be fixed. By default this is the same as the installation time, but any time in the past or in the future may be specified. If the licence has a limited period, the end time will also be set. The licence will be active whenever the current time is after the start time and before the end time.

A customer may wish to deactivate a licence so that it cannot be used. Thus can be done at any time by altering the end time of the licence with the server setup program. The end time may be altered freely, as long as the active period does not exceed the licence duration.

Once installed, limited-duration licences are fixed, ie they cannot be removed, except by remaining the entire site directory, or moved to another server, and their start time may not be altered. These restrictions, however, do not apply to unlimited-duration licences. They may be removed, reinstalled, moved or altered freely. The only restriction that remains is that a licence may only be installed for one server at a time.

A further restriction applies to the licence that has been used to create a site. This licence cannot be removed or deactivated, except by removing the entire site.

The licensing method described with reference to Figures 1 and 2 applies only to local licences, ie the tickets included in a license can only be issued in one server, the server whose server profile contains the licence. Often there is a need to share a single licence between two or more servers, so that tickets can be issued in all of them. Most commonly, the user tickets for an application are needed in the directory server, and session and usage tickets in the application servers.

If a licence includes an unlimited number of a certain kind of service ticket, sharing the licence is not very complicated. Any server can read the licences in any other server's profile. If the licensing subsystem in a server can verify that another server's licence contains an unlimited supply of freely shareable tickets, it will deduce that these tickets may be issued without limit in any server, independently of other servers. This is independent license sharing.

Not all licences are necessarily shareable, even if they contain an unlimited number of tickets. Whether each licence is shareable or not is a licence-specific property, which is coded in the covert code together with other licence properties.

The first implementation of the licensing subsystem capable of independent licence sharing will not scan every server profile for available licences. It will only scan its own server profile and the directory server's profile. Therefore, all licenses that are meant to be shared, should be installed for the directory server.

If the tickets to be shared are limited in number, the situation is more complicated. For each "pool" of shareable tickets, there must be a single process that is responsible for keeping track of their usage. It has to co-ordinate the activities of the licensing subsystems in various servers and make sure that no ticket is issued more than once. To achieve this a site licensing service can be implemented. This is an extension to the licensing subsystem that allows the licensing subsystems of various servers to communicate using a client-server protocol. The site licensing service, together with the licensing subsystem in the same server, control the usage of tickets installed for that server. Another server's licensing subsystem may connect to the site licensing service and ask the latter to obtain a service ticket on its behalf.

Licenses that are shareable by independent sharing would also be shareable by the site licensing service, with the addition that also limited-number tickets could be shared. Some types of licences will still be unshareable, since shareability is a licence-specific property. The licensing service could itself require a licence. A site licensing service could be expanded to support also client licensing and enterprise-wide licence sharing.

An example of independent licence sharing will now be described with reference to Figure 3 in which an application server 21 provides an application service 22 and includes a licensing subsystem 23 with an associated ticket database 24. A directory server 25 provides a directory service 26 and includes a licensing subsystem 27, with a associated ticket database 28, and a directory (not shown but containing information of the type illustrated in Figures 1 and 2). The ticket databases 24 and 28 have details of ticket type, limit and usage as indicated.

The licensing system 27 on the directory server 25 fetches the server profile from the directory (not shown), analyses the licences therein, and updates the ticket limits (step 1). The licensing system 23 on the application server 21 fetches the application server's server profile from the directory (not shown) using the directory API. It also fetches the directory server's server profile (step 2).

The Application server's licensing subsystem 23 analyses the licences in the server's own profile. In this case there are none, since the example is concerned with licence sharing. The licensing subsystem 23 then analyses the directory server's licences. Because there are unlimited session and usage tickets in a shareable licence, the local limit is also set to unlimited. The user ticket limit is set to 0, because they are limited (10 according to ticket database 28) and limited tickets cannot be shared with this method (step 3).

The application service 22 then requests an application session ticket from its licensing subsystem 23 (step 4). The ticket is granted because there are an unlimited supply of them. The used ticket count is updated in the local ticket database 24 (step 5), although it is only needed for statistics as the number is unlimited. The session ticket is then issued to the application service 22, which then proceeds since it has determined that there exists an appropriate licence.

License sharing in the case of a site licensing service will now be described with reference to Figure 4, in which an application server 31 provides an application service 32 and includes a licensing subsystem 33 with an associated ticket database 34. A license server 35 provides a site licensing service 36 and includes a licensing subsystem 37 with an associated ticket database 38.

The licensing subsystems 33 and 27 of the servers 31 and 35, fetch their corresponding server profiles from a directory (not shown), analyse installed licences and store the ticket limits in the local databases 34 and 38 (step 1). The application server 31 need not have any licences.

The application service 32 requests a service ticket, for example an application session ticket, from the local licensing subsystem 33 (step 2). The local licensing subsystem 33 in the application server 31 will first attempt to issue the ticket locally, but this will fail as there are no licences installed for the application server 31, as indicated by the lack of available tickets in the ticket database 34 (step 3). The licensing subsystem 33 in the application server 31 will then connect to the site licensing service 36 using the client-server protocol and request the ticket remotely (step 4). The site licensing service 36 requests the ticket from the local licensing subsystem 37 and it also request a ticket-forwarding ticket (step 5). The licensing subsystem 37 of the license server 35 checks ticket availability and updates the used ticket counts in the ticket database 38 (step 6). The tickets are issued to the site licensing service 36 (step 7) which forwards the application ticket to the client ie licensing subsystem 33 (step 8), which as a result issues the application ticket to the application service 32, allowing that to proceed (step 9).

Whenever a licensing subsystem issues a service ticket, or a ticket is returned such as because it is an unused ticket (any number can be requested) or because it is a session ticket, which are required to be returned at the end of a session, the transaction can, optionally, be logged to a log file which is separate from other log files in the system.

The information in this separate log file may be used to implement a pay-by-usage licensing scheme (delayed billing). Logging can be enabled or disabled by an administrator. Each server has its own log file and all kinds of tickets issued in the server will be logged the same way. Logging parameters for each kind of ticket could be specified for certain types of licences, although such a licence could not be shared by the independent sharing method.

The proposed licensing method allows for introducing new services while retaining compatibility with old licences. The licensing subsystems will initially support some types of licences and service tickets that are not yet connected to any particular service. New services can be assigned to these items without making any modifications to existing administration programs and the licensing subsystem. The method could be extended further by adding new license/ticket combinations to the licensing subsystem, although all existing combinations would need to be kept unchanged. This would involve updating the licensing subsystem in all servers where the new services would be used. Older subsystems would not accept the new kind of licenses not issue tickets for the new services. The licenses and tickets could be defined statically, as they are now, although there could be other possibilities.

As discussed above, the covert code specifies the licence duration, licence size and included services. An example of a covert code comprises a 7-digit decimal number, with the digits numbered from right to left, starting from zero eg in number 6543210, digit no 0 is "0", digit no 1 is "1" etc.

Licence duration may be encoded in the last digit ie digit 0, as follows:

Digit No 0	Licence Duration
"0"	10 days
"1"	1 month (31 days)
"2"	3 months (92 days)
"3"	6 months (184 days)
"4"	1 year (366 days)
"5"	2 years
"6"	3 years
"7"	Unlimited (small size)
"8"	Unlimited (medium size)
"9"	Unlimited (large size)

Licence size may be coded in the next-to-last digit, digit no 1. However, its interpretation may depend on the licence duration. Limited duration licences may be one of, for example, 30 different sizes; duration digits "7", "8" or "9" select small, medium or large licence sizes respectively.

Digit No 1 Licence size for each duration type

	Limited	Unlimited Small	Unlimited Medium	Unlimited Large
"0"	1	1	60	400
"1"	2	2	80	500
"2"	5	5	100	600
"3"	10	10	125	800
"4"	15	15	150	1000
"5"	20	20	175	1200
"6"	30	25	200	1500
"7"	50	30	225	2000
"8"	100	40	250	3000
"9"	Unlimited	50	300	Unlimited

The services that are included in a licence may be encoded into four digits, digits no 2 to no 5, of the covert code. These digits are called the service code. The licence may apply to one kind of service tickets only, to a group of related service tickets that are used by one service, or to a group of selected services. The service code can be chosen to represent a particular name of service, such as "basic directory service", "basic mail service", "basic calendar service", in any desired manner but it will indicate what types of tickets are included and how many licence service tickets are included for each type of service.

The digit no 6, the most significant digit, may be used to specify a particular product line. In the examples shown in the drawings the covert codes all commence with the number 2, indicating they relate to the same product line.

Any number of licences may be installed in the server profile of any server. The activation key is verified, and the covert code calculated from the serial number and the activation key at license installation time. The mapping of covert code to service ticket is, preferably, not stored in the directory, rather it is recalculated by a licensing subsystem every time it starts up. All tickets of the same type are indistinguishable. The licensing subsystems do not keep track of individual tickets issued.

Any number of identical tickets may be obtained at once by a service from the corresponding licensing subsystem, providing of course that they are available. Tickets can be returned if they are not used.

The licensing subsystem does not force services to obtain tickets rather it is the service's responsibility to offer services only to legal users and without obtaining a respective ticket, a service which requires a licence will not function.

Session tickets are associated with client-server sessions.

Unless a service wants to allow unlimited usage, it should obtain a session ticket whenever a session starts. Determining when each session starts and ends is the responsibility of the service. Session tickets may not be applicable to all services. It is important that session tickets are returned when the sessions end, otherwise they will be unusable, at least until the licencing subsystem is resynchronised. This is achieved at server start up, when there are no sessions in existence, by setting the used session ticket count to zero.

When a user is given the right to use a service, the associated user ticket should be obtained first. Because in a currently preferred embodiment, users are created and user rights given by the directory service, the licenses that include user tickets should be installed into the directory server. The directory service is the only service that requests user tickets and it is responsible for maintaining consistency of the used ticket counts. It periodically counts all users in the directory and their user rights and sets the number of tickets in use as appropriate.

Some kinds of tickets are "consumable" e.g. for sending mail messages, and these will not be returned unless, for example, the message is cancelled.

Clearly if an originally purchased licence becomes inadequate, due for example to an increased number of users, then supplemental licences can be purchased which when installed will increase the number of available tickets for a service. Additional functionality can of course also be purchased subsequently, in order to add new features to a system, and the appropriate software and licence installed in an appropriate server.

It is considered that with the above description of the licence management system and method proposed by the invention, a software developer will have difficulty producing the corresponding code for licence management for a particular software product written in a particular language, and hence no further description is considered necessary in this respect.

CLAIMS

1. A software licence management method for use with a computer system including at least one server, the method being such that before a service can offer functionality to a user, the said service shall verify that the user has a licence for said service, and wherein the computer system further includes a licensing subsystem with which are associated service tickets corresponding to existing licences, the method including the steps of the said service requesting a respective service ticket from the licensing subsystem prior to offering functionality to the user, and the licensing subsystem issuing a said service ticket to the said service, if one is available, thereby verifying the licence exists and allowing the said service to offer functionality.

2. A method as claimed in Claim 1, including the step of installing licence information comprising a licence serial number and a licence activation key into the computer system, the activation key containing encoded details of the licensed services, and wherein the computer system calculates, from the serial number and the activation key, information including the types of service tickets associated with a particular licence, the numbers of service tickets, and the duration of the licence.

3. A method as claimed in Claim 2 wherein the licensing subsystem maintains a log of the numbers of the maximum available and issued service tickets.

4. A method as claimed in Claim 2 or Claim 3, wherein a covert code is calculated by the computer system from the serial number and activation key and wherein mapping of the covert code to service tickets is calculated by

the licencing subsystem each time it is started.

5. A method as claimed in any one of the preceding claims wherein the computer system comprises a plurality of computers connected in a network and wherein a said server comprises a directory server, providing a directory service and including a respective licencing subsystem, together with a directory database and a ticket database, wherein stored in the directory database are directory server profile details, licence details and user profile details, and wherein the ticket database includes details of service tickets available in accordance with the respective licence details and issued, and wherein adding a user to the computer system includes the steps of starting the directory server licensing subsystem, the directory server licensing subsystem fetching the directory server profile with licence details from the directory database and updating the ticket database, the requesting of a user service ticket by the directory service from the licensing subsystem, the checking of ticket availability in the ticket database by the licensing subsystem, the issuing of a ticket by the licensing subsystem to the directory service, and the adding to the directory database of the new user's profile by the directory service.

6. A method as claimed in any one of Claims 1 to 4 wherein the computer system comprises a plurality of computers connected in a network, wherein a said server comprises an application server providing an application service and including a respective licensing subsystem with a respective ticket database, and another said server comprises a directory server providing a directory service and with a respective directory database, wherein stored in the directory database are directory server profile details, application server profile details and licence details, and wherein the ticket

database includes details of service tickets available in accordance with the licence details and issued, and wherein obtaining a use ticket for the application service includes the steps of starting the application server licensing subsystem, the subsystem fetching the application server profile and licence details from the directory database and updating the ticket database accordingly, starting the application service without providing funtionality, the requesting by the application service of a user service ticket from the licensing subsystem, the checking of ticket availability in the ticket database by the licensing subsystem, and the issuing of a service ticket to the application service by the licensing subsystem, the application service then providing its functionality to a user.

7. A method as claimed in any one of Claims 1 to 4 and for independent licence sharing, wherein the computer system comprises a plurality of computers connected in a network, wherein a said server comprises an application server providing an application service and including a respective licensing subsystem with a respective ticket database, and another said server comprises a directory server providing a directory service and including a respective licensing subsystem with a respective ticket base and with a respective directory database, wherein stored in the directory database are directory server profile details, application server profile details and shareable licence details, the number of service tickets being unlimited, wherein the directory server ticket database includes details of service tickets available in accordance with the shareable licence details and issued, and wherein the application server ticket database includes details of service tickets issued, and wherein obtaining a service ticket for the application service includes the steps of the directory server licensing system fetching the server profile from the

directory database, analysing the shareable licence details and updating the corresponding ticket types and ticket limits in the directory server ticket database, the application server licensing subsystem fetching the application server and the directory server profiles and shareable licence details from the directory database and analysing them and updating the corresponding ticket types in the application server ticket database, starting the application service without providing functionality, the requesting by the application service of a service ticket from the application server licensing system, the granting of a service ticket, and the issuing of the service ticket to the application service by the application server licensing system, the application service then providing its functionality to a user.

8. A method as claimed in any one of Claims 1 to 4 and for licence sharing with site licensing, wherein the computer system comprises a plurality of computers connected in a network, wherein a said server comprises an application server providing an application service and including a respective licensing subsystem with a respective ticket database, another said server comprises a site licensing server providing a site licensing service and including a respective licensing subsystem with a respective ticket database, and a further said server comprises a directory server providing a directory service and having a directory database, wherein stored in the directory database are directory server profile details, site licensing server profile details, application server profile details and licence details, wherein the site licensing subsystem ticket database includes details of service tickets available in accordance with the licence details and issued, and wherein obtaining a service ticket for the application service when the application server has no

respective licence includes the steps of the licensing subsystems fetching their corresponding server profiles from the directory database, analysing the installed licence details and the site licensing server updating the respective ticket database, starting the application service without providing functionality, the requesting by the application service of a service ticket from the site licensing service, the requesting of a service ticket and a ticket-forwarding ticket by the site licensing service from its licensing subsystem, the checking of ticket availability and the issuing of the service and ticket-forwarding tickets to the site licensing service, the forwarding of the service ticket to the application server licensing subsystem, and the issuing of the service ticket to the application service, the application service then providing its functionality to a user.

9. A computer system including at least one server and a software licence management system, the management system being such that before a service can offer functionality to a user, the service shall verify that the user has a licence for said service, the management system including a licencing subsystem with which are associated service tickets corresponding to existing licences, and the management system being such that a said service ticket is issued to a service, if one is available, upon request by the service, thereby verifying existence of a licence and allowing the said service to offer functionality.

10. A computer system as claimed in Claim 9, wherein the management system includes means for calculating from an input licence serial number and input licence activation key, information including the types of service tickets associated with a particular licence, the numbers of service tickets and the duration of the licence, said

information being encoded in the activation key.

11. A computer system as claimed in Claim 10, and including a log in which are stored the numbers of the maximum available and issued service tickets.

12. A computer system as claimed in Claim 9 or Claim 10, and wherein the calculating means include means for calculating a covert code from the serial number and activation key, and the licensing subsystem including means for mapping the covert code into service tickets each time the licensing subsystem is started.

13. A computer system as claimed in any one of Claims 9 to 12 and comprising a plurality of computers connected in a network, wherein a said server comprises a directory server, providing a directory service and including a respective licensing subsystem, together with a directory database and a ticket database, wherein stored in the directory database are directory server profile details, licence details and user profile details, and wherein the ticket database includes details of service tickets available in accordance with respective licence details and issued.

14. A computer system as claimed in any one of Claims 9 to 12 and comprising a plurality of computers connected in a network, wherein a said server comprises an application server providing an application service and including a respective licensing subsystem with a respective ticket database, and another server comprises a directory server providing a directory service with a respective directory database, wherein stored in the directory database are directory server profile details, application server profile details and licence details, and wherein the ticket database includes details of service tickets available in accordance with the licence

details and issued.

15. A computer system as claimed in Claim 14 and for independent licence sharing, wherein the directory server includes a respective directory licensing subsystem and a respective ticket database, shareable licence details, for which the number of service tickets available is unlimited, being stored in the directory database, the directory ticket database including details of service tickets available in accordance with the shareable licence details and issued, and the application server ticket database including details of service tickets issued.

16. A computer system as claimed in Claim 14 and for licence sharing with site licensing, and including another said server comprising a site licensing server providing a site licensing service and including a respective licensing subsystem with a respective ticket database, the directory database also including site licensing server profile details, and wherein the site licensing ticket database includes details of service tickets available in accordance with the licence detailed and issued.

17. A software licence management method substantially as herein described with reference to an as illustrated in Figure 1, Figure 2, Figure 3, or Figure 4, of the accompanying drawings.

18. A computer system including at least one server and a software licence management system substantially as herein described with reference to and as illustrated in Figure 1, or Figure 2, or Figure 3, or Figure 4 of the accompanying drawings.

The Patent Office

29

Application No: GB 9617596.3
Claims searched: 1-18
Examiner: Mike Davis
Date of search: 26 September 1996

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.O): G4A (AAP)

Int Cl (Ed.6): G06F

Other:

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
X	GB 2236604 A	(SUN MICROSYSTEMS)	1,9 at least
X	EP 0332304 A2	(DIGITAL EQUIPMENT)	•
X	WO 93/11480 A1	(INTERGRAPH)	•
X	US 5375206	(HUNTER ET AL)	•
X	US 4924378	(HERSHEY ET AL)	•

X Document indicating lack of novelty or inventive step
Y Document indicating lack of inventive step if combined with one or more other documents of same category.
& Member of the same patent family
A Document indicating technological background and/or state of the art.
P Document published on or after the declared priority date but before the filing date of this invention.
E Patent document published on or after, but with priority date earlier than, the filing date of this application.

An Executive Agency of the Department of Trade and Industry

```
E26         1  PN=BR 9810991
E27         1  PN=BR 9810992
E28         1  PN=BR 9810993
E29         1  PN=BR 9810994
E30         1  PN=BR 9810995
E31         1  PN=BR 9810996
E32         1  PN=BR 9810997
E33         1  PN=BR 9810998
E34         1  PN=BR 9810999
E35         1  PN=BR 9811000
E36         1  PN=BR 9811001
E37         1  PN=BR 9811002
E38         1  PN=BR 9811004
E39         1  PN=BR 9811005
E40         1  PN=BR 9811006
E41         1  PN=BR 9811007
E42         1  PN=BR 9811008
E43         1  PN=BR 9811009
E44         1  PN=BR 9811010
E45         1  PN=BR 9811011
E46         1  PN=BR 9811012
E47         1  PN=BR 9811013
E48         1  PN=BR 9811014
E49         1  PN=BR 9811015
E50         1  PN=BR 9811016

Enter P or PAGE for more
```

? s e3

```
      S1        1  PN='BR 9810967'
```

? t 1/7/1

1/7/1
DIALOG(R)File 351: Derwent WPI
(c) 2008 The Thomson Corporation. All rights reserved.

0009253575 *Drawing available*
WPI Acc no: 1999-181268/199915
Related WPI Acc No: 1996-465320; 1997-363998; 1998-363180; 1999-154174; 1999-154175; 1999-154176; 1999-154177; 1999-154178; 1999-154179; 1999-243551; 2002-060946; 2002-499082; 2002-705909; 2002-722051; 2002-722052; 2003-677663; 2003-898213; 2004-155029; 2004-478232; 2004-579235; 2004-623798; 2005-809338; 2007-015228
XRPX Acc No: N1999-133079
method for decrypting an instance of service that has been decrypted with short-term key
Patent Assignee: SCIENTIFIC-ATLANTA INC (SCAT)
Inventor: AKINS G L; PALGON M S; PINDER H G; WASILEWSKI A J; AKINS G

Patent Family (8 patents, 79 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
WO 1999009743	A2	19990225	WO 1998US16079	A	19980731	199915	B

AU 199915816	A	19990308	AU 199915816	A	19980731	199929	E
EP 1000511	A2	20000517	EP 1998960147	A	19980731	200028	E
			WO 1998US16079	A	19980731		
BR 199810967	A	20011030	BR 199810967	A	19980731	200173	E
			WO 1998US16079	A	19980731		
EP 1000511	B1	20011114	EP 1998960147	A	19980731	200175	E
			WO 1998US16079	A	19980731		
DE 69802540	E	20011220	DE 69802540	A	19980731	200207	E
			EP 1998960147	A	19980731		
			WO 1998US16079	A	19980731		
JP 2003521820	W	20030715	WO 1998US16079	A	19980731	200347	E
			JP 2000510276	A	19980731		
JP 2005253109	A	20050915	JP 2000510276	A	19980731	200560	E
			JP 2005120425	A	20050418		

Priority Applications (no., kind, date): US 199754575 P 19970801; US 1998126921 A 19980731

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes	
WO 1999009743	A2	EN	113	29		
National Designated States,Original	AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GE GH GM HR HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG UZ VN YU ZW					
Regional Designated States,Original	AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW NL OA PT SD SE SZ UG ZW					
AU 199915816	A	EN			Based on OPI patent	WO 1999009743
EP 1000511	A2	EN			PCT Application	WO 1998US16079
					Based on OPI patent	WO 1999009743
Regional Designated States,Original	DE FR GB IT NL					
BR 199810967	A	PT			PCT Application	WO 1998US16079
					Based on OPI patent	WO 1999009743
EP 1000511	B1	EN			PCT Application	WO 1998US16079
					Based on OPI patent	WO 1999009743
Regional Designated States,Original	DE FR GB IT NL					
DE 69802540	E	DE			Application	EP 1998960147
					PCT Application	WO 1998US16079
					Based on OPI patent	EP 1000511

				Based on OPI patent	WO 1999009743
JP 2003521820	W	JA	136	PCT Application	WO 1998US16079
				Based on OPI patent	WO 1999009743
JP 2005253109	A	JA	59	Division of application	JP 2000510276

Alerting Abstract WO A2
NOVELTY - The method involves receiving a second message in a receiver together with the instance of the service. The second message includes a key derivation value that is used with a long-term key to obtain the short-term key to decrypt the instance of the service.
DESCRIPTION - A control word is combined into an encrypted coded message (ECM) (107) with other service-related information. The ECM (107) is authenticated by Control Word Encrypt & Message Authenticate function (204) which produces a message authentication code using a keyed-hash value derived from the message content combined with a secret which can be shared with the receiving set-top box (113). This secret is preferably part or all of a multisession key (MSS) (208). The message authentication code is appended to the rest of the ECM (107). The CAW (202) is always encrypted before being sent along with the other parts of the ECM to MX (200). This encryption is preferably a symmetric cipher such as the Triple-DES algorithm using two distinct 56-bit keys (which taken together comprise MSS (208).
USE - The invention concerns systems for protecting information and more particularly concerns systems for protecting information that is transmitted by a wired or wireless medium against unauthorized access.
ADVANTAGE - The service distribution organizations require access restrictions which are both more secure and more flexible than those in conventional systems
DESCRIPTION OF DRAWINGS - The drawing is a block diagram of service instance encryption techniques.
107 encrypted coded message
204 Control Word Encrypt & Message Authenticate function
200 MX

Title Terms /Index Terms/Additional Words: METHOD; INSTANCE; SERVICE; SHORT; TERM; KEY

Class Codes

International Patent Classification

IPC	Class Level	Scope	Position	Status	Version Date
H04L-009/08			Main		"Version 7"
H04H-001/00; H04N-007/167; H04N-007/173			Secondary		"Version 7"
H04H-0001/00	A	I	L	R	20060101
H04L-0009/08	A	I	L	R	20060101
H04N-0005/00	A	I		R	20060101
H04N-0007/16	A	I		R	20060101
H04N-0007/167	A	I		R	20060101
H04N-0007/173	A	I	F	R	20060101

H04H-0001/00	C	I	L	R	20060101
H04L-0009/08	C	I	F	R	20060101
H04N-0005/00	C	I		R	20060101
H04N-0007/16	C	I		R	20060101
H04N-0007/167	C	I		R	20060101
H04N-0007/173	C	I	L	R	20060101

File Segment: EPI;
DWPI Class: W02; W03
Manual Codes (EPI/S-X): W02-F05A1B; W03-A16C3A

Original Publication Data by Authority

Australia

Publication No. AU 199915816 A (Update 199929 E)
Publication Date: 19990308
Assignee: SCIENTIFIC-ATLANTA INC; US (SCAT)
Language: EN
Application: AU 199915816 A 19980731 (Local application)
Priority: US 199754575 P 19970801
US 1998126921 A 19980731
Related Publication: WO 1999009743 A (Based on OPI patent)
Current IPC: H04H-1/00(R,I,M,JP,20060101,20051220,A,L) H04H-1/00 (R,I,M,JP,20060101,20051220,C,L) H04L-9/08(R,I,M,JP,20060101,20051220,A,L) H04L-9/08 (R,I,M,JP,20060101,20060310,C,F) H04N-5/00(R,I,M,EP,20060101,20051008,A) H04N-5/00 (R,I,M,EP,20060101,20051008,C) H04N-7/16(R,I,M,EP,20060101,20051008,A) H04N-7/16 (R,I,M,EP,20060101,20051008,C) H04N-7/167(R,I,M,EP,20060101,20051008,A) H04N-7/167 (R,I,M,EP,20060101,20051008,C) H04N-7/173(R,I,M,JP,20060101,20051220,A,F) H04N-7/173 (R,I,M,JP,20060101,20051220,C,L)

Brazil

Publication No. BR 199810967 A (Update 200173 E)
Publication Date: 20011030
Assignee: SCIENTIFIC-ATLANTA INC (SCAT)
Inventor: WASILEWSKI A J
AKINS G L
PALGON M S
PINDER H G
Language: PT
Application: BR 199810967 A 19980731 (Local application)
WO 1998US16079 A 19980731 (PCT Application)
Priority: US 199754575 P 19970801
US 1998126921 A 19980731
Related Publication: WO 1999009743 A (Based on OPI patent)

Current IPC: H04H-1/00(R,I,M,JP,20060101,20051220,A,L) H04H-1/00 (R,I,M,JP,20060101,20051220,C,L) H04L-9/08(R,I,M,JP,20060101,20051220,A,L) H04L-9/08 (R,I,M,JP,20060101,20060310,C,F) H04N-5/00(R,I,M,EP,20060101,20051008,A) H04N-5/00 (R,I,M,EP,20060101,20051008,C) H04N-7/16(R,I,M,EP,20060101,20051008,A) H04N-7/16 (R,I,M,EP,20060101,20051008,C) H04N-7/167(R,I,M,EP,20060101,20051008,A) H04N-7/167 (R,I,M,EP,20060101,20051008,C) H04N-7/173(R,I,M,JP,20060101,20051220,A,F) H04N-7/173 (R,I,M,JP,20060101,20051220,C,L)

Germany

Publication No. DE 69802540 E (Update 200207 E)
Publication Date: 20011220
Assignee: SCIENTIFIC-ATLANTA INC; US (SCAT)
Language: DE
Application: DE 69802540 A 19980731 (Local application)
EP 1998960147 A 19980731 (Application)
WO 1998US16079 A 19980731 (PCT Application)
Priority: US 199754575 P 19970801
US 1998126921 A 19980731
Related Publication: EP 1000511 A (Based on OPI patent)
WO 1999009743 A (Based on OPI patent)

EPO

Publication No. EP 1000511 A2 (Update 200028 E)
Publication Date: 20000517
Assignee: SCIENTIFIC-ATLANTA, INC., One Technology Parkway South, Norcross, Georgia 30092, US
Inventor: AKINS, Glendon, L., III, 2510 Windward Lane N.E., Gainesville, GA 30501, US
PALGON, Michael, S., 1196 Poplar Grove Drive, Atlanta, GA 30306, US
PINDER, Howard, G., 4317 Stilson Circle, Norcross, GA 30092, US
WASILEWSKI, Anthony, J., 10680 Wren Ridge Road, Alpharetta, GA 30022, US
Agent: Kugele, Bernhard, NOVAPAT INTERNATIONAL SA, 9, Rue du Valais, 1202 Geneve, CH
Language: EN
Application: EP 1998960147 A 19980731 (Local application)
WO 1998US16079 A 19980731 (PCT Application)
Priority: US 199754575 P 19970801
US 1998126921 A 19980731
Related Publication: WO 1999009743 A (Based on OPI patent)
Designated States: (Regional Original) DE FR GB IT NL
Original IPC: H04N-7/167(A)
Current IPC: H04H-1/00(R,I,M,JP,20060101,20051220,A,L) H04H-1/00 (R,I,M,JP,20060101,20051220,C,L) H04L-9/08(R,I,M,JP,20060101,20051220,A,L) H04L-9/08 (R,I,M,JP,20060101,20060310,C,F) H04N-5/00(R,I,M,EP,20060101,20051008,A) H04N-5/00 (R,I,M,EP,20060101,20051008,C) H04N-7/16(R,I,M,EP,20060101,20051008,A) H04N-7/16 (R,I,M,EP,20060101,20051008,C) H04N-7/167(R,I,M,EP,20060101,20051008,A) H04N-7/167 (R,I,M,EP,20060101,20051008,C) H04N-7/173(R,I,M,JP,20060101,20051220,A,F) H04N-7/173 (R,I,M,JP,20060101,20051220,C,L)
Original Abstract:

A cable television system provides conditional access to services. The cable television system includes a headend from which service "instances", or programs, are broadcast and a plurality of set top units for receiving the instances and selectively decrypting the instances for display to system subscribers. The service instances are encrypted using public and/or private keys provided by service providers or central authorization agents. Keys used by the set tops for selective decryption may also be public or private in nature, and such keys may be reassigned at different times to provide a cable television system in which piracy concerns are minimized.

Publication No. EP 1000511 B1 (Update 200175 E)
Publication Date: 20011114
Assignee: Scientific-Atlanta, Inc., 5030 Sugarloaf Parkway, Lawrenceville, GA 30044, US
Inventor: AKINS, Glendon, L., III, 2510 Windward Lane N.E., Gainesville, GA 30501, US
PALGON, Michael, S., 1196 Poplar Grove Drive, Atlanta, GA 30306, US
PINDER, Howard, G., 4317 Stilson Circle, Norcross, GA 30092, US
WASILEWSKI, Anthony, J., 10680 Wren Ridge Road, Alpharetta, GA 30022, US
Agent: Kugele, Bernhard, NOVAPAT INTERNATIONAL SA, 9, Rue du Valais, 1202 Geneve, CH
Language: EN
Application: EP 1998960147 A 19980731 (Local application)
WO 1998US16079 A 19980731 (PCT Application)
Priority: US 199754575 P 19970801
US 1998126921 A 19980731
Related Publication: WO 1999009743 A (Based on OPI patent)
Designated States: (Regional Original) DE FR GB IT NL
Original IPC: H04N-7/167(A)
Current IPC: H04H-1/00(R,I,M,JP,20060101,20051220,A,L) H04H-1/00 (R,I,M,JP,20060101,20051220,C,L) H04L-9/08(R,I,M,JP,20060101,20051220,A,L) H04L-9/08 (R,I,M,JP,20060101,20060310,C,F) H04N-5/00(R,I,M,EP,20060101,20051008,A) H04N-5/00 (R,I,M,EP,20060101,20051008,C) H04N-7/16(R,I,M,EP,20060101,20051008,A) H04N-7/16 (R,I,M,EP,20060101,20051008,C) H04N-7/167(R,I,M,EP,20060101,20051008,A) H04N-7/167 (R,I,M,EP,20060101,20051008,C) H04N-7/ 173(R,I,M,JP,20060101,20051220,A,F) H04N-7/173 (R,I,M,JP,20060101,20051220,C,L)
Claim:

1. Verfahren der Entschlusselung einer Diensteeinheit (325), die mit einem gegebenen Kurzzeitschlussel (319) verschlusselt wurde, wobei das Verfahren in einem Empfanger (333) ausgefuhrt wird, der ein Offentlich/Privat-Schlusselpaar besitzt, und das Verfahren durch die folgenden Schritte **gekennzeichnet** ist:
    - im Empfanger eine erste Nachricht (315) zu empfangen, deren Inhalt einen ersten Langzeitschlussel (309) einschliesst und unter Verwendung des offentlichen Schlussels (312) fiir den Empfanger (333) verschlusselt wurde;
    - den privaten Schlussel (337) zur Entschlusselung des Inhalts zu verwenden;
    - den ersten Schlussel (309) zu speichern;
    - im Empfanger (333) zusammen mit der verschlusselten Diensteeinheit (329) eine zweite Nachricht (323) zu empfangen, wobei die zweite Nachricht (323) einen Indikator fur einen zweiten Kurzzeitschlussel (319) einschliesst;
    - den Indikator und den ersten Schlussel (309) zu benutzen, um den zweiten Schlussel zu erhalten; worin der zweite Schlussel dem gegebenen Schlussel (319), mit dem der Dienst verschlusselt wurde, gleichwertig ist, und
    - den zweiten Schlussel zur Entschlusselung der empfangenen Diensteeinheit zu

verwenden.

1. A method of decrypting an instance of a service (325) that has been encrypted with a given short-term key (319), the method being carried out in a receiver (333) that has a public key-private key pair and the method being **characterised by** the following steps:
   - receiving a first message (315) in the receiver whose contents include a first long-term key (309), the contents having been encrypted using the public key (312) for the receiver (333);
   - using the private key (337) to decrypt the contents;
   - storing the first key (309);
   - receiving a second message (323) in the receiver (333) together with the encrypted instance of the service (329), the second message (323) including an indicator for a second short-term key (319);
   - using the indicator an the first key (309) to obtain the second key; wherein the second key is equivalent to the given key (319) that encrypted the service, and
   - using the second key to decrypt the received instance of the service.

1. Procede de decryptage d'une instance d'un service (326) qui etait cryptee avec une cle a court terme donnee (319), le procede etant execute dans un recepteur (333) qui comporte une paire de cle publique-cle privee et le procede etant **caracterise par** les etapes suivantes:
   - recevoir un premier message (315) dans le recepteur dont le contenu comprend une premiere cle a long terme (309), le contenu ayant ete crypte en utilisant la cle publique (312) pour le recepteur (333),
   - utiliser la cle privee (337) pour decrypter le contenu,
   - memoriser la premiere cle (309),
   - recevoir un second message (323) dans le recepteur (333) en meme temps que l'instance cryptee du service (329), le second message (323) comprenant un indicateur pour une seconde cle a court terme (319),
   - utiliser l'indicateur et la premiere cle (309) pour obtenir la seconde cle, dans lequel
   - la seconde cle est equivalente a la cle donnee (319) qui a crypte le service, et
   - utiliser la seconde cle pour decrypter l'instance recue du service.

Japan

Publication No. JP 2003521820 W (Update 200347 E)
Publication Date: 20030715
Language: JA (136 pages)
Application: WO 1998US16079 A 19980731 (PCT Application)
JP 2000510276 A 19980731 (Local application)
Priority: US 199754575 P 19970801
US 1998126921 A 19980731
Related Publication: WO 1999009743 A (Based on OPI patent)

Original IPC: H04L-9/08(A) H04H-1/00(B) H04N-7/167(B) H04N-7/173(B)
Current IPC: H04L-9/08(A) H04H-1/00(B) H04N-7/167(B) H04N-7/173(B)

Publication No. JP 2005253109 A (Update 200560 E)
Publication Date: 20050915
CONDITIONAL ACCESS SYSTEM
Assignee: SCIENTIFIC-ATLANTA INC (SCAT)
Inventor: AKINS GLENDON L III
PALGON MICHAEL S
PINDER HOWARD G
WASILEWSKI ANTHONY J
Language: JA (59 pages)
Application: JP 2000510276 A 19980731 (Division of application)
JP 2005120425 A 20050418 (Local application)
Priority: US 199754575 P 19970801
US 1998126921 A 19980731
Original IPC: H04L-9/08(A)
Current IPC: H04H-1/00(R,I,M,JP,20060101,20051220,A,L) H04H-1/00 (R,I,M,JP,20060101,20051220,C,L) H04L-9/08(R,I,M,JP,20060101,20051220,A,L) H04L-9/08 (R,I,M,JP,20060101,20060310,C,F) H04N-5/00(R,I,M,EP,20060101,20051008,A) H04N-5/00 (R,I,M,EP,20060101,20051008,C) H04N-7/16(R,I,M,EP,20060101,20051008,A) H04N-7/16 (R,I,M,EP,20060101,20051008,C) H04N-7/167(R,I,M,EP,20060101,20051008,A) H04N-7/167 (R,I,M,EP,20060101,20051008,C) H04N-7/173(R,I,M,JP,20060101,20051220,A,F) H04N-7/173 (R,I,M,JP,20060101,20051220,C,L)

WIPO

Publication No. WO 1999009743 A2 (Update 199915 B)
Publication Date: 19990225
CONDITIONAL ACCESS SYSTEM
RESEAU D'ACCES CONDITIONNEL
Assignee: SCIENTIFIC-ATLANTA, INC., Intellectual Property Dept., One Technology Parkway South, Norcross, GA 30092, US Residence: US Nationality: US (SCAT)
Inventor: AKINS, Glendon, L., III, 2510 Windward Lane N.E., Gainesville, GA 30501, US
PALGON, Michael, S., 1196 Poplar Grove Drive, Atlanta, GA 30306, US
PINDER, Howard, G., 4317 Stilson Circle, Norcross, GA 30092, US
WASILEWSKI, Anthony, J., 10680 Wren Ridge Road, Alpharetta, GA 30022, US
Agent: GARDNER, Kelly, A., Scientific-Atlantic, Inc., Intellectual Property Dept., One Technology Parkway South, Norcross, GA 30092, US
Language: EN (113 pages, 29 drawings)
Application: WO 1998US16079 A 19980731 (Local application)
Priority: US 199754575 P 19970801
US 1998126921 A 19980731
Designated States: (National Original) AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GE GH GM HR HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG UZ VN YU ZW
(Regional Original) AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW NL OA PT SD SE SZ UG ZW
Original IPC: H04N-7/167(A)
Current IPC: H04H-1/00(R,I,M,JP,20060101,20051220,A,L) H04H-1/00

(R,I,M,JP,20060101,20051220,C,L) H04L-9/08(R,I,M,JP,20060101,20051220,A,L) H04L-9/08 (R,I,M,JP,20060101,20060310,C,F) H04N-5/00(R,I,M,EP,20060101,20051008,A) H04N-5/00 (R,I,M,EP,20060101,20051008,C) H04N-7/16(R,I,M,EP,20060101,20051008,A) H04N-7/16 (R,I,M,EP,20060101,20051008,C) H04N-7/167(R,I,M,EP,20060101,20051008,A) H04N-7/167 (R,I,M,EP,20060101,20051008,C) H04N-7/173(R,I,M,JP,20060101,20051220, A,F) H04N-7/173 (R,I,M,JP,20060101,20051220,C,L)

Original Abstract:

A cable television system provides conditional access to services. The cable television system includes a headend from which service "instances", or programs, are broadcast and a plurality of set top units for receiving the instances and selectively decrypting the instances for display to system subscribers. The service instances are encrypted using public and/or private keys provided by service providers or central authorization agents. Keys used by the set tops for selective decryption may also be public or private in nature, and such keys may be reassigned at different times to provide a cable television system in which piracy concerns are minimized.

Un reseau de television par cable assure un acces conditionnel a des services. Le reseau de television par cable comprend une tete de reseau a partir de laquelle on diffuse les "instances" de service ou programmes. Ce reseau comprend aussi une pluralite d'unites decodeurs concues pour recevoir les instances et dechiffrer selectivement les instances qui vont s'afficher pour les abonnes du reseau. Les instances de service sont chiffrees par des cles publiques et/ou privees fournies par des fournisseurs de service ou des agents d'autorisation centraux.Les cles utilisees par les decodeurs permettant un dechiffrement selectif peuvent aussi etre publiques ou privees et de telles cles peuvent etre reaffectees a differents moments pour assurer un reseau de television par cable dans lequel les risques de piratage sont minimises.

?

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 934 765 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
11.08.1999 Bulletin 1999/32

(51) Int. Cl.^{6}: **A63F 9/22**

(21) Application number: **98400285.7**

(22) Date of filing: **09.02.1998**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(60) Divisional application:
98202314.5

(71) Applicant:
CANAL+ Société Anonyme
75711 Paris Cedex 15 (FR)

(72) Inventors:
- **Agasse, Bernard**
  95610 Eragny/Oise (FR)
- **Bayassi, Mulham**
  75015 Paris (FR)

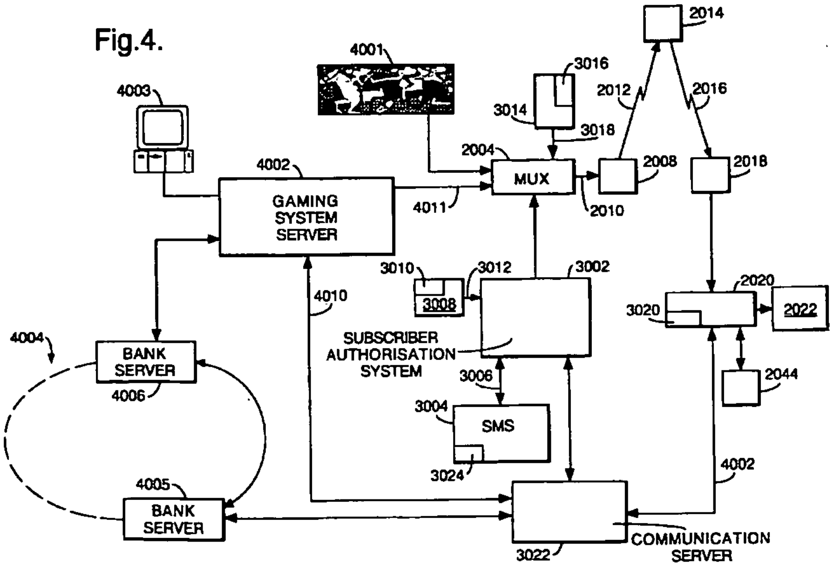
(74) Representative:
Cozens, Paul Dennis et al
Mathys & Squire
100 Grays Inn Road
London WC1X 8AL (GB)

Remarks:
The application is published incomplete as filed (Article 93 (2) EPC).

(54) Interactive gaming system

(57) An interactive gaming and audiovisual transmission system comprising a central gaming computer 4002 for processing gaming data, a decoder 2020 adapted to receive gaming data from the central gaming computer 4002 together with transmitted audiovisual data, the decoder further including a card reading device for interacting with a user's bank card in order to credit a gaming account held by the central gaming computer means in response to a transfer of credit from the user's bank account.

Fig.4.


4001
4003
3016
2014
2012
2016
3014
3018
2004
4002
MUX
2008
2018
GAMING SYSTEM SERVER
4011
2010
3010
3012
3002
2020
3008
4010
3020
2022
4004
SUBSCRIBER AUTHORISATION SYSTEM
BANK SERVER
3006
2044
4006
3004
SMS
3024
4002
4005
BANK SERVER
COMMUNICATION SERVER
3022

EP 0 934 765 A1

Printed by Xerox (UK) Business Services
2.16.7/3.6

Description

[0001] The present invention relates to an interactive gaming and digital audiovisual transmission system, in particular a gaming and digital television transmission system.

[0002] Broadcast transmission of digital data is well-known in the field of pay TV systems, where scrambled audiovisual information is sent, usually by a satellite or satellite/cable link, to a number of subscribers, each possessing a decoder capable of descrambling the transmitted program for subsequent viewing. Terrestrial digital broadcast systems are also known. Recent systems have also used the broadcast link to transmit other data, in addition to or as well as audiovisual data, such as computer programs or interactive applications to the decoder or to a connected PC.

[0003] The increasing sophistication of such technology, in particular in relation to the receiver/decoder devices used in the systems, has led to an increase in the possible services that may be provided thereby. In particular, a number of systems have been proposed using interactive technology to enable a viewer to, for example, participate in a quiz show, or to select further information regarding a product currently being displayed on a shopping channel.

[0004] In the case of gaming applications, a number of largely theoretical systems have been proposed to enable a viewer to gamble a sum of money on the outcome of a sporting event or casino-type game broadcast over a television network. In most of these systems, a viewer is usually obliged to open an initial account with the controlling gaming authority by phoning or mailing a money transfer to the gaming authority before any gambling can be carried out. The disadvantages of this sort of procedure will be apparent.

[0005] Alternative systems are also known, in which the viewer buys credits to be gambled in the form of an electronic purse, i.e. a smart card or the like, the credits in the purse being available for subsequent gaming operations. The card is inserted in the decoder and the credits used thereafter in the subsequent gaming operations. When the contents of the purse are exhausted, the viewer buys a new card or re-charges the card at a suitable sales point. This system again implies a certain infra-structure to be put in place to enable a user to obtain the necessary credits to be gambled.

[0006] The present invention seeks to overcome some or all of the disadvantages of these prior art systems.

[0007] According to the present invention, there is provided an interactive gaming and audiovisual transmission system comprising a central gaming computer means for processing gaming data, a decoder adapted to receive gaming data from the central gaming computer together with transmitted audiovisual data, the decoder further including a card reading device for interacting with a user's bank card in order to credit a gaming account held by the central gaming computer means in response to a transfer of credit from the user's bank account.

[0008] In this way, the present invention enables a user to simply and quickly open and credit a gaming account from the comfort of his home, avoiding the more elaborate payment methods of the known systems.

[0009] The type of bank card used in this transaction may be of the debit or credit type. The card reading device may in particular comprise a smart card reader adapted to interact with a bank card in the form of a smart card.

[0010] Advantageously, the decoder is further equipped with a second card reading device. For example, in the case where the decoder forms part of a television subscription service, the subscriber may be provided with a subscription card in the form of a smart card or the like. The provision of two card reader devices in the decoder permits the decoder to carry out credit transactions on a bank card inserted in one reader whilst the subscription card is held in the second reader.

[0011] In one realisation, the decoder may be adapted to obtain transfer of credit information in the form of an electronic certificate generated by the bank card in response to transaction data submitted by the decoder. This transaction information may include, for example, the details of the bank account of the gaming authority to be credited in the operation, the sum of money to be transferred etc.

[0012] Typically, data is entered by the user into the decoder using a handheld remote control. In the case where a credit transaction is to be carried out, it may be necessary to enter the bank card PIN number using the remote control. In one embodiment, the decoder is provided with a handheld remote control, some or all of the data sent to the decoder being encrypted by the handheld remote control and subsequently decrypted by the decoder. In this way, interception by third parties of sensitive data emitted by the remote control may be avoided.

[0013] Preferably, the decoder is adapted to transmit transfer of credit information from the decoder to a bank server via a network communication link, for example, using an modem integrated in the decoder.

[0014] The decoder may be adapted to directly communicate transfer of credit information to a bank computer. However, preferably, the system further comprises an intermediate communications server, adapted to receive transfer of credit information communicated from the decoder and to forward this information on to a bank server.

[0015] The intermediate communications server may further be adapted to communicate with the central gaming computer means, for example, to inform the central communication means of a transfer of credit instruction being forwarded from the intermediate communication means to a bank computer, so as to permit the gaming computer means to set up an account without having to verify the transaction carried out at an associated bank server.

[0016] The central gaming computer means may equally be adapted to receive and transmit credit information to or from a bank server via a network communication link. This may be necessary, for example, in the case of a win or in order to verify the transfer of funds from the bank account of a user to the gaming authorities bank account before opening a gaming account.

[0017] Preferably, the decoder is adapted to communicate gaming information to the central gaming computer during gaming operation via a network communication link. This may be the same link as used to communicate transfer of credit information to a bank computer, for example, using a modem device integrated in the decoder.

[0018] Some or all of the gaming information communicated from the decoder to the central gaming computer during gaming operation may be encrypted by the decoder. For example, the decoder may be adapted to transmit in encrypted form a code word entered by the user associated with the gaming account of the user held by the central gaming computer.

[0019] The decoder may be adapted to directly communicate information to the central gaming computer during gaming operation. However, preferably, the system further comprises an intermediate communications server, adapted to receive information communicated from the decoder during gaming operation and to forward this information on to the central gaming computer. This may be the same intermediate server as used for the transfer of credit information between the decoder and a bank.

[0020] In the case where gaming information is encrypted by the decoder, the intermediate communications server may be adapted to simply pass this information "as is" to the central gaming computer. However, in one embodiment, the intermediate communications server is adapted to decrypt information received from the decoder and to re-crypt this information for subsequent communication to the central gaming computer. This may be required, for example, in the case where different encryption algorithms are used by the decoder and central gaming computer.

[0021] The intermediate communications server may further be adapted to communicate information to and from other computer devices, for example, computer databases holding TV subscriber information. In this way, the intermediate communications server may obtain directly information regarding the user of the system (name, address etc) to be used in setting up a gaming account, without the user having to re-enter the same information.

[0022] The communication means used to transmit gaming data from the central gaming computer to the decoder may be defined in a number of different ways and by a number of different communication elements. For example, some or all of the gaming data sent from the gaming computer to the decoder may be transmitted via a transmitter means used to transmit audiovisual data to the decoder.

[0023] In addition, or alternatively, some or all of the gaming data sent from the central gaming computer to the decoder may be sent via a network communication link, for example, the same network used to communicate information from the decoder to the central gaming computer during gaming operation.

[0024] In practice, a mixture of these two communication paths may prove optimal, the network path being used for rapid dialogue between the decoder and the gaming computer during real-time operation and the transmission path being used for relatively fixed data, such as screen format display data or the like.

[0025] The present invention also extends to a gaming system for processing gaming data, comprising:

> means for transmitting gaming data to a user's decoder;
> means for receiving data from the user's decoder; and
> means for connection to a bank server holding the user's bank account in order to transfer credit to or from the account.

[0026] The gaming system may include a gaming account held by the gaming system which can be credited in response to the transfer of credit.

[0027] The gaming system may be adapted to communicate with the decoder and the bank server via a communications server. If so, the gaming system may be adapted to receive encrypted information from the communications server.

[0028] The present invention also provides a interactive gaming and audiovisual transmission system comprising a gaming system as aforementioned, said user's decoder, and said bank server.

[0029] As mentioned above the system may be used to permit gaming in relation to various events. For example, the central gaming computer may be adapted to generate a computer game (computer blackjack or the like), the computer generated images being transmitted via the audiovisual link to the decoder.

[0030] However, as will be appreciated, the combination of gaming and audiovisual systems makes the present invention particularly adapted to permit gaming in relation to televised sports, such as horse racing or the like. In one embodiment, the present invention comprises a central gaming computer adapted to provide gaming data related to a real-time sporting event, the decoder being adapted to receive both gaming data and associated audiovisual data of the event.

[0031] In the context of the present application the term ⟨⟨audiovisual transmission system⟩⟩ refers to all transmission systems for transmitting or broadcasting primarily audiovisual or multimedia digital data. The present invention is particularly, but not exclusively, applicable to a broadcast digital television system.

[0032] In this application the term ⟨⟨ smart card ⟩⟩ is used to mean any conventional chip-based card device possessing, for example, microprocessor and/or memory storage. Also included in this term are chip devices having alternative physical forms, for example key-shaped devices such as are often used in TV decoder systems.

[0033] In the present application, the term "decoder" is used to apply to an integrated receiver/decoder for receiving and decrypting an encrypted transmission, the receiver and decoder elements of such a system as considered separately, as well as to a receiver capable of receiving non-encrypted broadcasts. The term equally covers decoders including additional functions, such as web browsers, together with decoder systems integrated with other devices, for example, integrated VHS/decoder devices or the like.

> Figure 1 shows the overall architecture of a digital television system, as may be incorporated in the gaming system of the present invention;
>
> Figure 2 shows the conditional access system of the television system of Figure 1;
>
> Figure 3 shows the structure of the decoder of Figures 1 and 2;
>
> Figure 4 shows a gaming system incorporating the television system of Figures 1 and 2; and
>
> Figure 5 shows a flow diagram of the logical steps involved in a gaming transaction

Digital Television System

[0034] An overview of a digital television broadcast and reception system 1000 adaptable to the present invention is shown in Figure 1. The system includes a mostly conventional digital television system 2000, which uses the known MPEG-2 compression system to transmit compressed digital signals. In more detail, the MPEG-2 compressor 2002 in a broadcast centre receives a digital signal stream (typically a stream of video signals). The compressor 2002 is connected to a multiplexer and scrambler 2004 by linkage 2006. The multiplexer 2004 receives a plurality of further input signals, assembles one or more transport streams and transmits compressed digital signals to a transmitter 2008 of the broadcast centre via linkage 2010, which can of course take a wide variety of forms including telecom links.

[0035] The transmitter 2008 transmits electromagnetic signals via uplink 2012 towards a satellite transponder 2014, where they are electronically processed and broadcast via notional downlink 2016 to earth receiver 2018, conventionally in the form of a dish owned or rented by the end user. The signals received by receiver 2018 are transmitted to an integrated receiver/decoder 2020 owned or rented by the end user and connected to the end user's television 2022. The receiver/decoder 2020 decodes the compressed MPEG-2 signal into a television signal for the television set 2022.

[0036] A conditional access system 3000 is connected to the multiplexer 2004 and the receiver/decoder 2020, and is located partly in the broadcast centre and partly in the decoder. It enables the end user to access digital television broadcasts from one or more broadcast suppliers. A smart card, capable of decrypting messages relating to commercial offers (that is, on or several television programmes sold by the broadcast supplier), can be inserted into the receiver/decoder 2020. Using the decoder 2020 and smart card, the end user may purchase events in either a subscription mode or a pay-per-view-mode.

[0037] An interactive system 4000, also connected to the multiplexer 2004 and the receiver/decoder 2020 and again located partly in the broadcast and partly in the decoder, enables the end user to interact with various applications via a modemmed back channel 4002. Such interactive applications may include an interactive shopping service, a quiz application, an interactive programme guide etc.

[0038] In point of fact, whilst the interactive system 4000 has been represented as a discrete logical block, the physical elements of this system, such as the server or servers used to handle communications between the receiver/decoder and central servers, may be elements shared with the conditional access system 3000. This will become clear in the description of the gaming system of Figure 4.

Conditional Access System

[0039] With reference to Figure 2, the conditional access system 3000 includes a Subscriber Authorization System (SAS) 3002. The SAS 3002 is connected to one or more Subscriber Management Systems (SMS) 3004, one SMS for each broadcast supplier, by a respective TCP-IP link 3006 (although other types of linkage could alternatively be used). Alternatively, one SMS could be shared between two broadcast suppliers, or one supplier could use two SMSs, and so on.

[0040] First encrypting units in the form of ciphering units 3008 utilising ⟨⟨ mother ⟩⟩ smart cards 3010 are connected to the SAS by linkage 3012. Second encrypting units again in the form of ciphering units 3014 utilising mother smart cards 3016 are connected to the multiplexer 2004 by linkage 3018. The receiver/decoder 2020 receives a ⟨⟨ daughter ⟩⟩ smart card 3020. It is connected directly to the SAS 3002 by Communications Servers 3022 via the modemmed back channel 4002. The SAS sends amongst other things subscription rights to the daughter smart card on request.

[0041] The smart cards contain the secrets of one or more commercial operators. The ⟨⟨ mother ⟩⟩ smart card encrypts different kinds of messages and the ⟨⟨ daughter ⟩⟩ smart cards decrypt the messages, if they have the rights to do so.

[0042] The first and second ciphering units 3008 and 3014 comprise a rack, an electronic VME card with software stored on an EEPROM, up to 20 electronic cards and one smart card 3010 and 3016 respectively, for each electronic card, one (card 3016) for encrypting the ECMs and one (card 3010) for encrypting the EMMS.

[0043] Also shown in Figure 2 is a handheld remote control used by the viewer to control and program functions of the receiver/decoder 2020.

Multiplexer and Scrambler

[0044] With reference to Figures 1 and 2, in the broadcast centre, the digital video signal is first compressed (or bit rate reduced), using the MPEG-2 compressor 2002. This compressed signal is then transmitted to the multiplexer and scrambler 2004 via the linkage 2006 in order to be multiplexed with other data, such as other compressed data.

[0045] The scrambler generates a control word CW used in the scrambling process and included in the MPEG-2 stream in the multiplexer 2004. The control word CW is generated internally and enables the end user's integrated receiver/decoder 2020 to descramble the programme. Access criteria, indicating how the programme is commercialised, are also added to the MPEG-2 stream. The programme may be commercialised in either one of a number of ⟨⟨ subscription ⟩⟩ modes and/or one of a number of ⟨⟨ Pay Per View ⟩⟩ (PPV) modes or events.

[0046] In the subscription mode, the end user subscribes to one or more commercial offers, of ⟨⟨ bouquets ⟩⟩, thus getting the rights to watch every channel inside those bouquets. In the preferred embodiment, up to 960 commercial offers may be selected from a bouquet of channels. In the Pay Per View mode, the end user is provided with the capability to purchase events as he wishes. This can be achieved by either pre-booking the event in advance (⟨⟨ pre-book mode ⟩⟩), or by purchasing the event as soon as it is broadcast (⟨⟨ impulse mode ⟩⟩).

[0047] Both the control word CW and the access criteria are used to build an Entitlement Control Message (ECM); this is a message sent in relation with a scrambled program. The message contains a control word (which allows for the descrambling of the program) and the access criteria of the broadcast program. The access criteria and control word are transmitted to the second encrypting unit 3014 via the linkage 3018. In this unit an ECM is generated, encrypted with an exploitation key Cex and transmitted on to the multiplexer and scrambler 2004.

Programme Transmission

[0048] The multiplexer 2004 receives encrypted EMMs from the SAS 3002, encrypted ECMs from the second encrypting unit 3014 and compressed programmes from the compressor 2002. The multiplexer 2004 scrambles the programmes and communicates the scrambled programmes, the encrypted EMM (if present) and the encrypted ECMs to a transmitter 2008 of the broadcast centre via linkage 2010. The transmitter 2008 transmits electromagnetic signals towards the satellite transponder 2014 via uplink 2012.

Programme Reception

[0049] The satellite transponder 2014 receives and processes the electromagnetic signals transmitted by the transmitter 2008 and transmits the signals on to the earth receiver 2018, conventionally in the form of a dish owned or rented by the end user, via downlink 2016. The signals received by receiver 2018 are transmitted to the integrated receiver/decoder 2020 owned or rented by the end user and connected to the end user's television set 2022. The receiver/decoder 2020 demultiplexes the signals to obtain scrambled programmes with encrypted EMMs and encrypted ECMs.

[0050] If the programme is not scrambled the receiver/decoder 2020 decompresses the data and transforms the signal into a video signal for transmission to television set 2022.

[0051] If the programme is scrambled, the receiver/decoder 2020 extracts the corresponding ECM from the MPEG-2 stream and passes the ECM to the ⟨⟨ daughter ⟩⟩ smart card 3020 of the end user. This slots into a housing in the receiver/decoder 2020. The daughter smart card 3020 controls whether the end user has the right to decrypt the ECM and to access the programme. If not, a negative status is passed to the receiver/decoder 2020 to indicate that the programme cannot be descrambled. If the end user does have the rights, the ECM is decrypted and the control word extracted. The decoder 2020 can then descramble the programme using this control word. The MPEG-2 stream is decompressed and translated into a video signal onward transmission to television set 2022.

Subscriber Management System (SMS)

[0052] A Subscriber Management System (SMS) 3004 includes a database 3024 which manages, amongst others, all of the end user files, commercial offers (such as tariffs and promotions), subscriptions, PPV details, and data regarding end user consumption and authorization. The SMS may be physically remote from the SAS

[0053] Each SMS 3004 transmits messages to the SAS 3002 via respective linkage 3006 to enable modifications to or creations of Entitlement Management Messages (EMMs) to be transmitted to end users.

[0054] The SMS 3004 also transmits messages to the SAS 3002 which imply no modifications or creations of EMMs but imply only a change in an end user's state (relating to the authorization granted to the end user when ordering products or to the amount that the end user will be charged).

Entitlement Management Messages and Entitlement Control Messages

[0055] ECMs or Entitlement Control Messages are encrypted messages embedded in the data stream of a transmitted program and which contain the control word necessary for descrambling of part or all of a program. Authorisation of a given receiver/decoder is controlled by EMMs or Entitlement Management Messages, transmitted on a less frequent basis and which supply an authorised receiver/decoder with the exploitation key necessary to decode the ECM.

[0056] An EMM is a message dedicated to an individual end user (subscriber), or a group of end users. A group may contain a given number of end users. This organisation as a group aims at optimising the bandwidth; that is, access to one group can permit the reaching of a great number of end users.

[0057] Various specific types of EMM may be used. Individual EMMs are dedicated to individual subscribers, and are typically used in the provision of Pay Per View services. So-called ⟨⟨ Group ⟩⟩ subscription EMMs are dedicated to groups, of say, 256 individual users, and are typically used in the administration of some subscription services. This EMM has a group identifier and a subscribers' group bitmap

[0058] For security reasons, the control word CW embedded in an encrypted ECM changes on average every 10 seconds or so. In contrast, the exploitation key Cex used by the receiver to decode the ECM is changed every month or so by means of an EMM. The exploitation key Cex is encrypted using a personalised key corresponding to the identity of the subscriber or group of subscribers recorded on the smart card. If the subscriber is one of those chosen to receive an updated exploitation key Cex, the card will decrypt the message using its personalised key to obtain that month's exploitation key Cex.

[0059] The operation of EMMs and ECMs will be well-known to one skilled in the art and will not be described here in any more detail.

Receiver/Decoder Structure

[0060] Referring to Figure 3, the elements of a receiver/decoder 2020 or set-top box for use in a digital broadcast system and adapted to be used in the present invention will now be described. As will be understood, the elements of this decoder are largely conventional and their implementation will be within the capabilities of one skilled in the art.

[0061] As shown, the decoder 2020 is equipped with several interfaces for receiving and transmitting data, in particular an MPEG tuner and demultiplexer 2040 for receiving broadcast MPEG transmissions, a serial interface 2041, a parallel interface 2042, and a modem 2028 for sending and receiving data via the telephone network. In this embodiment, the decoder also includes a first and second smart card reader 2030 and 2031, the first reader 2030 for accepting a subscription smart card containing decryption keys associated with the system and the second reader 2031 for accepting bank and other cards. As will be described, the use of a two-slot decoder, adapted to read bank cards, is an important aspect in the implementation of the gaming system of Figure 4.

[0062] The decoder also includes a receiver 2043 for receiving infra-red control signals from the handset remote control 2044 and a Peritel output for sending audiovisual signals to a television 2022 connected to the decoder. In certain cases it may be desired that the infra-red signals transmitted from the handset 2044 to receiver 2043 are subject to a simple scrambling/descrambling process to ensure that no useful information may be obtained by any third party monitoring the transmission.

[0063] Such algorithms will not be described in any detail, but may comprise, for example a symmetric algorithmic key known to both handset 2044 and receiver/decoder 2020. This may be varied from time to time, for example, by means of a modulating random number chosen by the receiver/decoder 2020 and displayed by the television 2022, the user then programming the handset 2044 with this number to ensure that the handset scrambles entered data using an encryption algorithm key equivalent to that used the receiver/decoder to decrypt the received infra-red signals.

[0064] Processing of digital signals received via the interfaces and generation of digital output signals is handled by a central control unit 2045. The software architecture of the control unit within the decoder may correspond to that used in a known decoder and will not be described here in any detail. It may be based, for example, on a virtual machine interacting via an interface layer with a lower level operating system implemented in the hardware components of the decoder. In terms of the hardware architecture, the decoder will be equipped with a processor, memory elements such as ROM, RAM, FLASH memory etc. as in known decoders.

[0065] Applications processed by the control unit 2045 may be resident applications stored in the ROM or FLASH of the decoder or applications broadcast and downloaded via the MPEG interface 2 of the decoder. Applications can include program guide applications, games, interactive services, teleshopping applications, as well as initiating applications to enable the decoder to be immediately operational upon start-up and applications for configuring the decoder. Applications are stored in memory locations in the decoder and represented as resource files comprising graphic object description files, unit files, variables block files, instruction sequence files, application files, data files etc.

[0066] Conventionally, applications downloaded into the decoder via the broadcast link are divided into modules, each module corresponding to one or more MPEG tables. Each MPEG table may be divided into a number of sections. For data transfer via the serial and parallel ports, modules are also split into tables and sections, the size of the section depending on the channel used.

[0067] In the case of broadcast transmission, modules are transported in the form of data packets within respective types of data stream, for example, the video data stream, the audio data stream, a text data stream. In accordance with MPEG standards each packet is preceded by a Packet Identifier (PID) of 13 bits, one PID for every packet transported in the MPEG stream. A programme map table (PMT) contains a list of the different streams and defines the content of each stream according to the respective PID. A PID may alert the device to the presence of applications in the data stream, the PID being identified by the PMT table.

Gaming System Architecture

[0068] Referring now to Figure 4, there will now be described the elements and functioning of a gaming system according to an embodiment of the present invention. The gaming system includes the elements of the digital television system described and shown in Figures 1 and 2, which have been assigned the same reference numerals . Some elements, such as the digital compressor 2002 shown in Figure 1, have been omitted in order to focus on those aspects of the system which are pertinent to the present invention.

[0069] As shown, the gaming system additionally comprises a source of audiovisual information 4001 regarding the event which will form the subject of betting etc within the system. In the present case, the event has been represented as a horse race, and the present system is indeed particular adapted to gaming activities centred around televised live action sporting events. However, as will be understood, the present system may equally used to permit gambling in relation to other events, such as casino-type games, as well as computer generated games, pre-recorded events etc.

[0070] The system further comprises a central gaming computer means in the form of a gaming system server 4002, together with associated operating terminal or terminals 4003, adapted to generate odds, calculate winnings etc in relation to the gaming event. The gaming server 4002 is adapted to communicate with a receiver/decoder 2020 via the intermediate communication server or servers 3022. The connection between the gaming server 4002 and communication server 3022 may be implemented by an X25 Transpac link or via a dedicated line. The network link for the server is indicated broadly at 4010.

[0071] As described above, the communication server 3022 communicates with the receiver/decoder 2020 by means of a telephone link using the in-built modem of the receiver/decoder.

[0072] The gaming server may be equally adapted to send information to the receiver/decoder 2020 via a satellite link, indicated broadly at 4011, by injection of information into the multiplexer 2004 for subsequent integration in the transmitted MPEG stream.

[0073] As will be understood, all communications from the receiver/decoder 2020 to the gaming server 4002 are via the receiver/decoder modem and communication server 3022. In the case of communications from the gaming server 4002 to the receiver/decoder 2020, the choice of communication channel and communication means (MPEG satellite transmission or communication server/modem connection) may depend on the nature of the information to be transmitted.

[0074] Typically, the satellite link 4011 will be used to send data or information that may be updated on a daily basis or which may be received by any number of receiver/decoders in the park (odds for tomorrow's races etc). In particular, the satellite link may be used to download the application that needs to be installed in the receiver/decoder to enable the receiver/decoder to function in the gaming system.

[0075] In contrast, the modem link 4010 may be preferred for data that changes on a minute-by-minute basis or that is specific to a particular user (results of last race, current state of the account of the user etc).

[0076] In addition to handling gaming activities resulting from bets placed via the receiver/decoder 2020, for example as programmed in using the remote control 2044, the gaming server 4002 may also be adapted to manage bets to be placed by other input means, for example as placed by a phone service or as received by a "Minitel" type system, as used in France and other countries.

[0077] The gaming system server 4002 is additionally connected to a bank server network 4003 comprising one or more bank servers 4005, 4006. The bank server network may correspond to an existing network used to handle electronic payment transactions. The level of security and encryption in the communications between each of the elements of the gaming system will be described in more detail below in relation to the operation of the system.

Gaming System Operation

[0078] As mentioned in the introduction of the present application, gaming systems used in interactive television systems proposed to date have tended to use relatively laborious methods for settling accounts between the viewer and the central gaming authority, requiring the viewer either to pay by a conventional method (cheque, telephone credit transfer etc) or to physically purchase an "electronic purse" in the form of a smart card or key containing a number of pre-paid credits that may be gambled.

[0079] The present embodiment differs from such systems in proposing a system architecture that enables a viewer to pay by means of a credit or debit card inserted in the decoder and by entering data into the system by means of the hand-held remote control. As mentioned above, the provision of a decoder provided with two distinct card readers 2030, 2031 enables the decoder to simultaneously hold a subscription card containing the viewers access rights (eg to the gaming channel) as well as interacting with a credit/debit card inserted in the decoder.

[0080] In order to comply with regulations concerning the use of credit/debit cards in gambling transactions, two different types of transactions need to be distinguished: (i) opening or re-crediting an account managed by the gaming system server and (ii) gambling the sums in this account.

Opening an account

[0081] In the present case, the card reader 2031 functions in a similar manner to a standard card reader used in banking terminals and the like to read and write data on a smart card presented in the reader. As with all card readers used in the banking field, communication between the terminal (in this case the decoder) and external servers is prohibited during the time that the card is being accessed by the terminal, i.e. for the time that the memory zones on the card are "open".

[0082] In order to open and credit an account with the gaming system server, the following steps are carried out during a first phase:

a) Using the handheld remote control, and as guided by the application loaded in the receiver/decoder, the user selects the option "open an account" and enters the sum of money that he wishes to transfer to this account.

b) After having introduced his credit card into the card reader slot 2031, the viewer is invited to enter his personal PIN code. The user has a maximum of two opportunities to enter the code, after which the receiver/decoder will refuse to accept any further entries and the transaction will be abandoned.

Note that in the case of sensitive information communicated to the receiver/decoder by the handset (in particular the PIN code) the data entered by the user on the key pad of the handset may be scrambled before transmission between the handset and decoder so as to prevent interception of this information by any third party. See above.

c) Assuming the code is correct, the smart card downloads certain information in response to a request from the receiver/decoder, including details of the last transactions, to enable the decoder to verify that the sum of transactions during a certain period is within, for example, the transaction limit of the card holder for that period.

d) The receiver/decoder then passes to the smart card information regarding the current transaction including the amount of the transaction, the date and time of the transaction, the details of the bank account to be credited in the transaction and so on. (The details of the account to be credited can be obtained by the decoder prior to the interrogation of the card from the gaming system server or the intermediate communications system server).

e) In the conventional manner, the smart card then calculates a first numeric certificate using this information, which is communicated to the receiver/decoder. The receiver/decoder writes the present transaction in the card and a second numeric certificate is calculated and communicated to the receiver/decoder. The memory zones of the smart card are then closed off.

The generation of a pair of numeric certificates is a specific security measure associated with the use of a receiver/decoder as transaction terminal.

Once the above steps have been carried out, the system then moves to a second phase involving communication between the receiver/decoder 2020, the intermediate communication server 3022 and the bank server 4005.

f) Before transferring any information, the receiver/decoder 2020 verifies the identity of the communication server 3022 by means of a public/private key system (eg using the RSA algorithm). In particular, the receiver/decoder generates a random number, which is transmitted to the server for encryption by a private key and returned to the receiver/decoder, which checks the encrypted value using the equivalent public key.

A simple handshake signal may also be provided by the decoder 2020 to identify itself to the server 3022.

g) Assuming the identity of the communication server is verified, the receiver/decoder 2020 sends to the communication server 3022 the details of the transaction to be carried out, including the first and second numeric certificate generated by the smart card.

h) The communication server 3022 then sends the transaction details to the first bank server 4005, which verifies the account of the user, and authorises (or not) the transaction and sends an acknowledgement of the transaction to the communication server. The transfer of money between the user's account and that of the central gaming authority will then be handled within the bank network 4004.

i) Once the communication server 3022 has received acknowledgement of the acceptation of the monetary transfer, a message will be sent to the receiver/decoder 2020 of the completion of the transfer and the operation will proceed to the next phase.

Note that the same steps a) to i) as used in the first two phases will also be carried out in the event that the user wishes to increase the credit in an existing gaming account.

The next phase in the opening of a gaming account involves communication between the receiver/decoder 2020, the communication server 3022 (and the SAS and SMS servers 3002, 3004) and the gaming server 4002. The information communicated between these servers is largely non-sensitive and may be communicated in clear, with the exception of the code word chosen by the user to obtain access to his gaming account.

j) Using the information (name, address etc) on the user held in the SAS and SMS servers 3002, 3004, the communication server prepares a request for opening of an account with the gaming system server 4002. This information has been gathered in the SMS server during the original procedure carried out when the user originally subscribed to the television service. The user is thus spared the inconvenience of repeating all this information when subscribing to the gaming service.

Note that in the event that SMS database reveals, for example, that the subscriber is in debt with the television service, the communication server may abort the opening of an account with the gaming service. This extra verification step may be carried out earlier, for example, at step g).

k) In one embodiment, the communication server 3022 may send the subscriber information to the receiver/decoder 2020 where it is displayed on the television 2022 for verification by the user. Once verified, the information is sent to the gaming system server 4002 where a gambling account is created by the server 4002.

l) The account information (account number etc) is then sent from the gaming server 4002, via the communication server 3022, to the receiver/decoder 2022. The user is then invited to choose a suitable code word for the account which will be demanded by the system at every opening of a gaming session. As for the PIN number, the infrared signal containing this information and sent between the remote control and the decoder may be scrambled by the remote to avoid interception and descrambled by decoder.

m) The code word is then encrypted by a public key of a public/private key pair held in the receiver/decoder 2020 and sent to the communication server 3022, where it is decrypted by the corresponding private key. In this case, for example, the same RSA key pair as used for the verification of the communication server may be used.

n) The code word is then re-encrypted by the communication server 3022 and sent to the gaming system server 4002 where it is decrypted and assigned to the user's account. In this case, a symmetric key algorithm, such as DES, may be advantageously used, for example, to permit two-way encrypted communication between the communication server 3022 and gaming server 4002.

Gambling with an existing gaming account

[0083] Once the user has set up and credited a gaming account with the gaming server 4002, all future gambling transactions will be handled between the receiver/decoder 2020 and the gaming system server 4002. At the start of every gaming session, the system server 4002 will demand the user's assigned code word, which will be communicated between the receiver/decoder and the gaming server, via the communications server, as described above.

[0084] For simplicity, and in order to permit a relatively rapid dialogue, all questions and responses between the user and the gaming system in order to place a bet and receive the results are preferably passed via the telephone/modem link and the communication server 3022. Certain data, such as the format of the screens displayed by the receiver/decoder in gaming mode and/or slowly changing or universal data (details of that day's races, the horses taking part etc) may be passed via the satellite uplink in order to take advantage of the bandwidth of this channel.

[0085] Other embodiments, in which data is shared between the two communication channels in alternative ways may nevertheless be envisaged, for example, where all communication from the receiver/decoder to the gaming system server passes via the modem link, whilst all communications from the server to the receiver/decoder pass via the satellite link.

[0086] As mentioned above, the present system may be used with a number of interactive gaming applications, for example, with computer games such as blackjack, poker or the like, in which the user places a bet on the outcome of a game managed by the gaming server. However, in view of the use of television broadcast technology, the system is particularly adapted to permit gaming in relation to live action sporting events, such as televised horse, dog or camel racing.

[0087] Figure 5 is a flow diagram of the steps involved in the placing of a bet in relation to one or more broadcast horse races. In the present case, the bet is to be placed in respect of the present day's races, i.e. in "real time", and the odds quoted for the horses may depend on the time at which the bet is taken. In alternative embodiments, bets may be placed the day or week before the race or races in question.

[0088] Firstly, at step 5000, the user enters his code word and opens a betting session. At steps 5001 and 5002, he chooses the racecourse he is interested in and one of the races running at that racecourse, respectively. Depending on which race is running, the user may be offered a number of different standard types of bet, from a simple bet to more complex bets, including main and side bets.

[0089] As will be appreciated, the bet types offered may be determined according to the wishes of the gaming authority and may be based on any of the usual types of bet offered for an event of this type.

[0090] At step 5003, the user chooses the type of bet he wishes to place. In the case of a simple bet on one horse, the next step will be step 5004 where the user chooses the formula of the bet, ie whether the horse will win or be placed in the first three or four positions. At step 5005, the user chooses the horse he wishes to bet on.

[0091] In the case of a complex bet, the user then chooses from a combination of win, place or win/place at step 5007 and from one of a number of types of bet (single, combined, reduced field, full field) at step 5007. The user may decide, for example to choose one horse to win and/or one horse to be placed in the top three or four. Other combinations may be made presented to reflect the choice of bet normally available. At step 5008 the user chooses the horses he wishes to bet on.

[0092] At step 5009 the user chooses his stake, i.e. the sum to be extracted from the money deposited in his gaming account. At step 5010 confirmation of the stake to be gambled is demanded. At this time, the system may also indicate the overall odds for the bet or bets placed and the sum of money to be won. Assuming that the user confirms the bet, the bet is registered at step 5011.

[0093] Following the results of the race, the gaming system server calculates the winnings or losses for the user. These will be subtracted or added automatically to his gaming account. The user may demand at any time the position of his account.

[0094] In the event that the user eventually wishes to close the account or to transfer some of his winnings to his bank account, a message to this end may be sent by the user from the receiver/decoder 2020 to the gaming system server 4002 (Figure 4). At that time, the server 4002 will communicate with the bank server 4006 to organise a credit transfer to the user's bank account. Since the identity and bank details of the owner of the receiver/decoder are already known, the server will only transfer money from the gaming account of the user to the bank account originally used in the setting up of the gaming account.

[0095] It will be understood that the present invention has been described above purely by way of example, and modifications of detail can be made within the scope of the invention.

[0096] Each feature disclosed in the description, and (where appropriate) the claims and drawings may be provided independently or in any appropriate combination.

[0097] In the aforementioned preferred embodiments, certain features of the present invention have been implemented using computer software. However, it will of course be clear to the skilled man that any of these features may be implemented using hardware. Furthermore, it will be readily understood that the functions performed by the hardware, the computer software, and such like are performed on or using electrical and like signals.

Claims

1. An interactive gaming and audiovisual transmission system comprising a central gaming computer means for processing gaming data, a decoder adapted to receive gaming data from the central gaming computer together with transmitted audiovisual data, the decoder further including a card reading device for interacting with a user's bank card in order to credit a gaming account held by the central gaming computer means in response to a transfer of credit from the user's bank account.

2. An interactive gaming and audiovisual transmission system as claimed in claim 1, in which the decoder is equipped with a card reading device in the form of a smart card reader.

3. An interactive gaming and audiovisual transmission system as claimed in claim 1 or 2, in which the decoder is further equipped with a second card reading device

4. An interactive gaming and audiovisual transmission system as claimed in any preceding claim in which the decoder is adapted to obtain transfer of credit information in the form of an electronic certificate generated by the bank card in response to transaction data submitted by the decoder.

5. An interactive gaming and audiovisual transmission system as claimed in any preceding claim in which the decoder is provided with a handheld remote control, some or all of the data sent to the decoder being encrypted by the handheld remote control and subsequently decrypted by the decoder.

6. An interactive gaming and audiovisual transmission system as claimed in any preceding claim in which the decoder is adapted to transmit transfer of credit information from the decoder to a bank server via a network communication link.

21. A gaming system as claimed in Claim 19 or 20, adapted to communicate with the decoder and the bank server via a communications server.

22. A gaming system as claimed in Claim 21, adapted to receive encrypted information from the communications server.

23. A gaming system as claimed in any of Claims 19 to 22, adapted to transmit gaming data related to a real-time sporting event.

24. An interactive gaming and audiovisual transmission system comprising a gaming system as claimed in any of Claims 19 to 23, said user's decoder, and said bank server.

Fig.1.

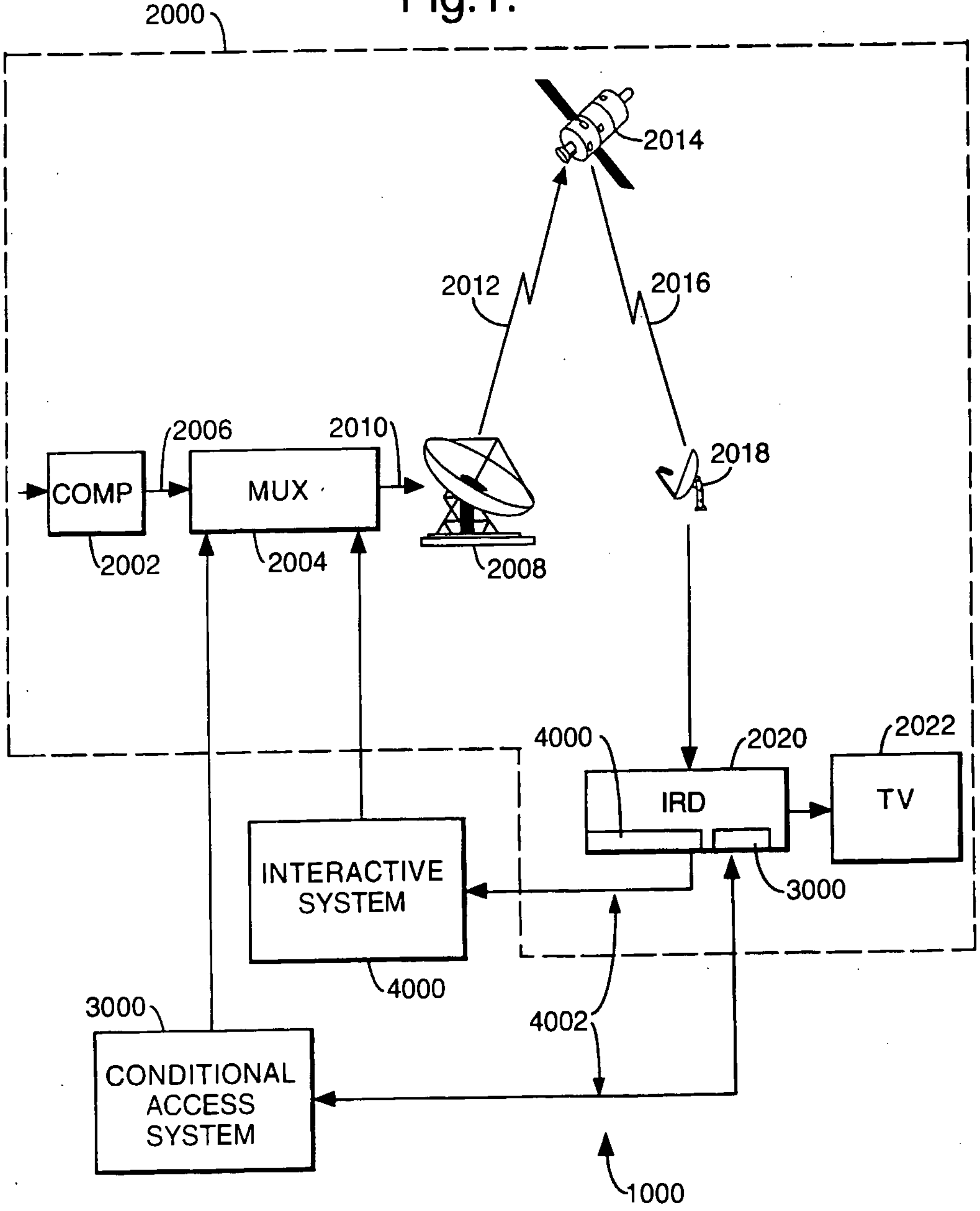

2000
2014
2012
2016
2006
2010
COMP
MUX
2018
2002
2004
2008
4000
2020
2022
IRD
TV
3000
INTERACTIVE SYSTEM
4000
3000
4002
CONDITIONAL ACCESS SYSTEM
1000

Fig.2.

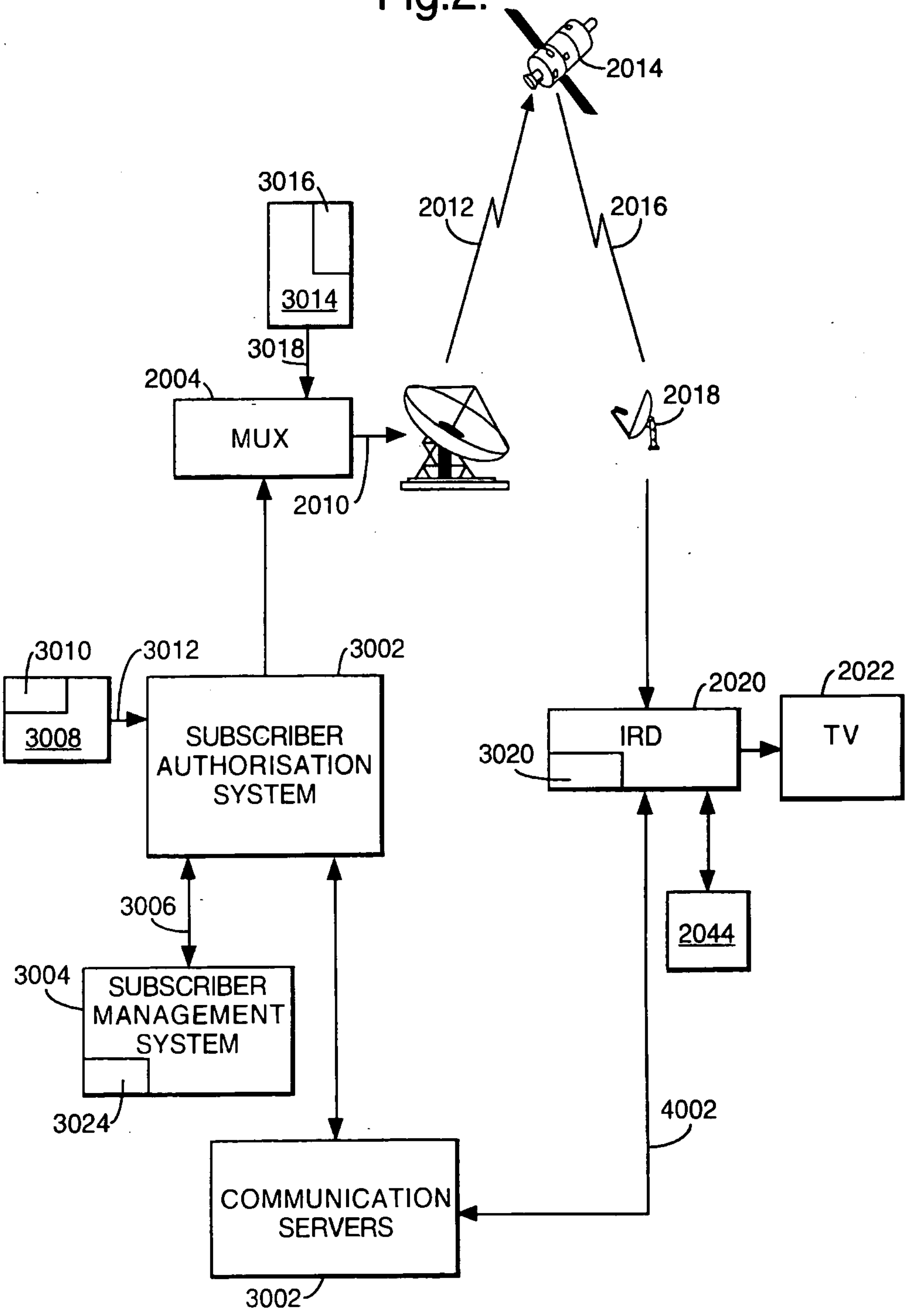

2014
2012
2016
3016
3014
3018
2004
MUX
2010
2018
3010
3012
3002
3008
SUBSCRIBER AUTHORISATION SYSTEM
2020
2022
3020
IRD
TV
2044
3006
3004
SUBSCRIBER MANAGEMENT SYSTEM
3024
4002
COMMUNICATION SERVERS
3002

Fig.3.

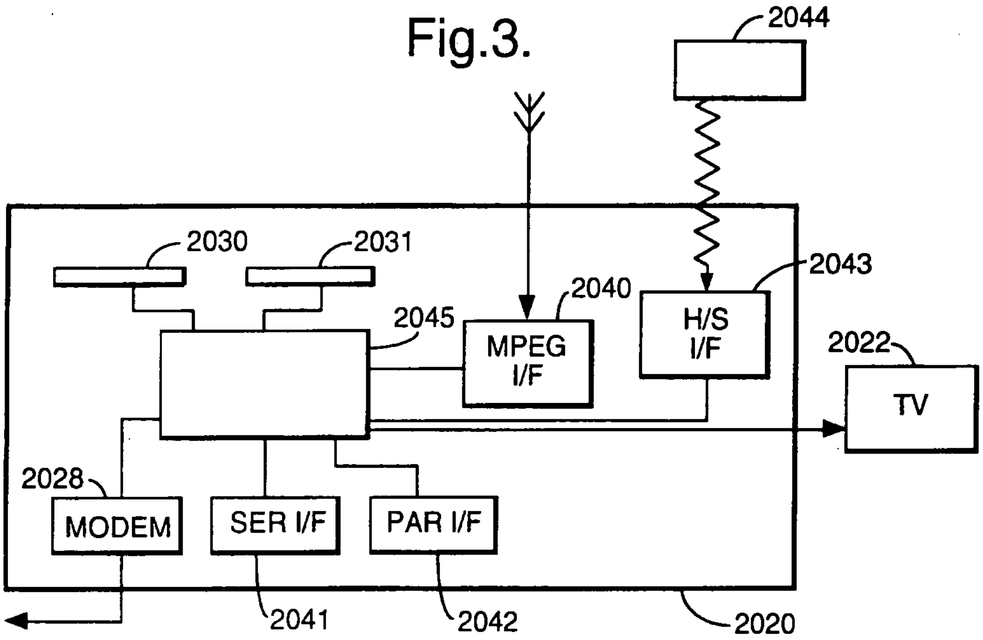
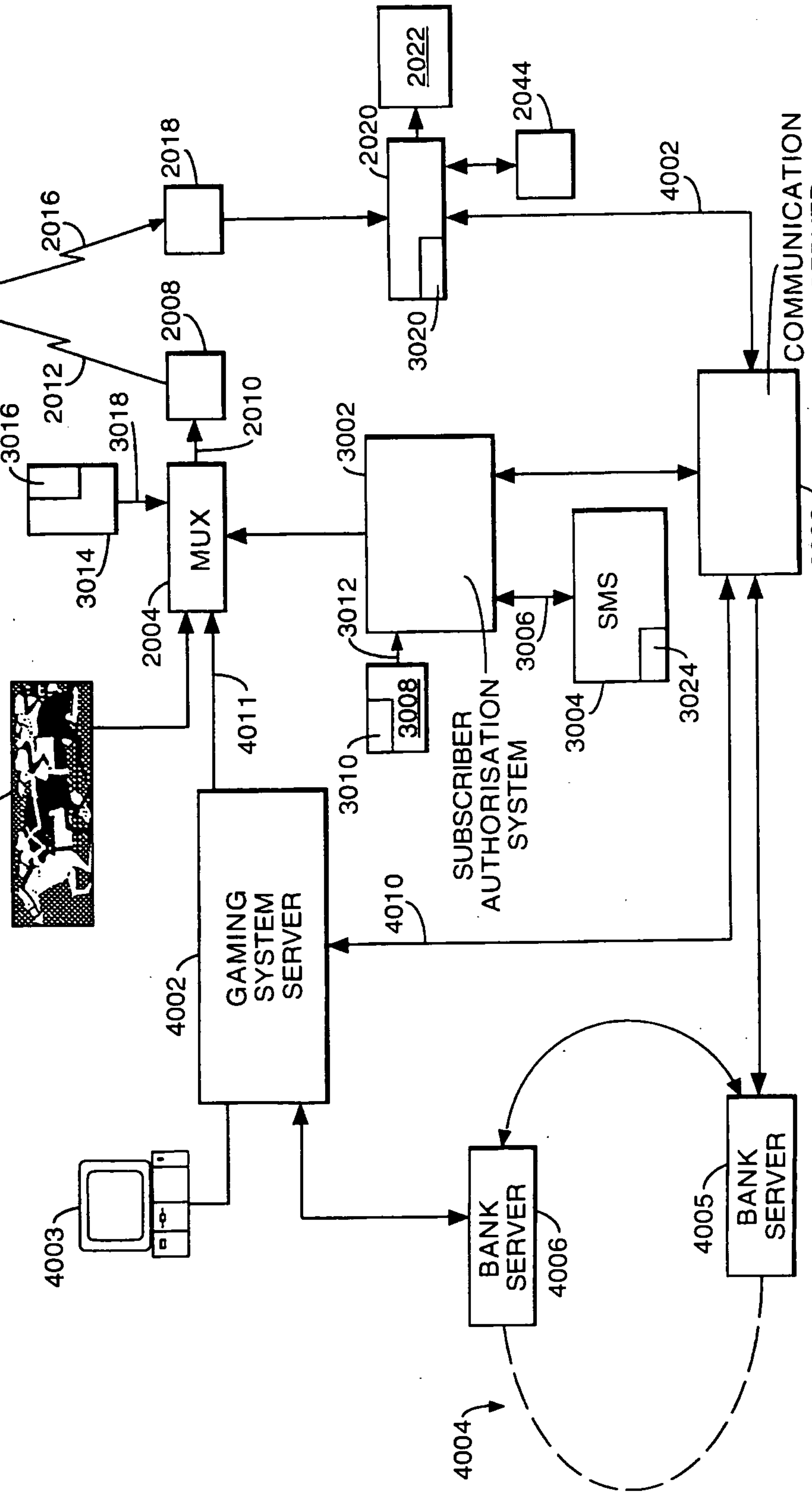

2044
2030
2031
2043
2040
2045
MPEG I/F
H/S I/F
2022
TV
2028
MODEM
SER I/F
PAR I/F
2041
2042
2020

Fig.4.


2014
2016
2018
2022
2044
2020
2012
2008
3020
4002
COMMUNICATION SERVER
3016
3018
2010
3002
MUX
3014
2004
3012
3006
SMS
3022
3008
3010
3004
3024
4001
4011
SUBSCRIBER AUTHORISATION SYSTEM
4010
GAMING SYSTEM SERVER
4002
4003
BANK SERVER
4006
BANK SERVER
4005
4004

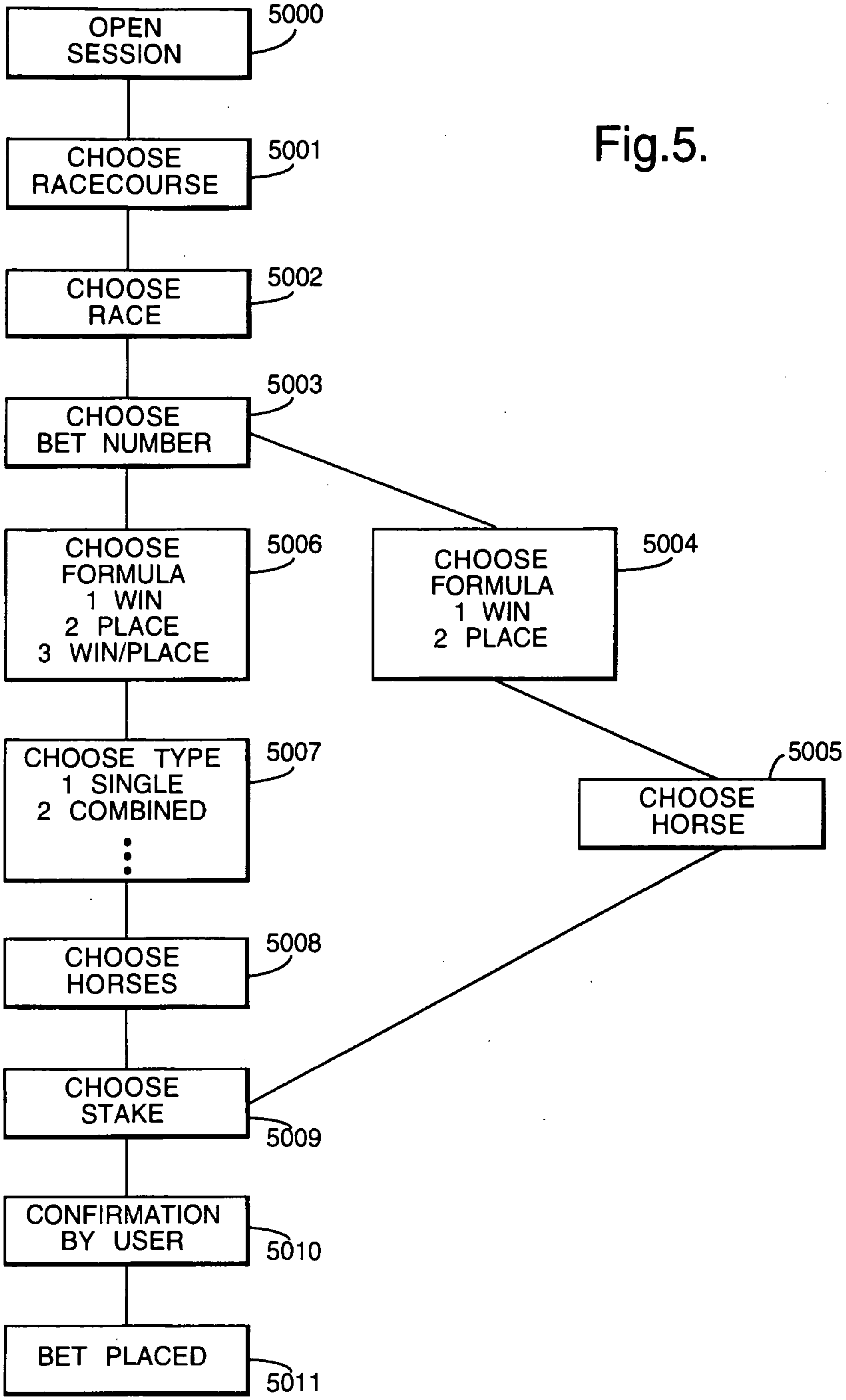

OPEN SESSION
5000
CHOOSE RACECOURSE
5001
CHOOSE RACE
5002
CHOOSE BET NUMBER
5003
CHOOSE FORMULA 1 WIN 2 PLACE 3 WIN/PLACE
5006
CHOOSE FORMULA 1 WIN 2 PLACE
5004
CHOOSE TYPE 1 SINGLE 2 COMBINED
5007
CHOOSE HORSE
5005
CHOOSE HORSES
5008
CHOOSE STAKE
5009
CONFIRMATION BY USER
5010
BET PLACED
5011

Fig.5.

European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 98 40 0285

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	US 5 539 822 A (LETT DAVID B) 23 July 1996	1-3,10, 12, 15-19,23	A63F9/22
Y		4-9,11, 14,20, 21,24	
A		5	
	* column 7, line 13 - line 33 * * column 9, line 58 - column 10, line 11 * * column 11, line 22 - line 42 * * column 15, line 11 - line 44 * * column 18, line 44 - column 20, line 33 * * figures 3E,3I * ---		
X	US 4 815 741 A (SMALL MAYNARD E) 28 March 1989 * column 4, line 41 - column 5, line 5 * * column 5, line 27 - line 38 * ---	1,19	
Y	US 5 634 848 A (TSUDA YOICHIRO ET AL) 3 June 1997	6-9,14, 20,21,24	TECHNICAL FIELDS SEARCHED (Int.Cl.6)
A		1,19	A63F H04N G07F G06F
	* column 1, line 42 - line 64 * * column 3, line 6 - column 4, line 2 * * column 8, line 44 - column 9, line 21 * ---		
Y	WO 95 01060 A (LINCOLN MINT HONG KONG LTD) 5 January 1995 * page 1, line 19 - line 29 * * page 3, line 17 - page 4, line 35 * * page 12, line 36 - page 13, line 35 * * page 14, line 31 - page 15, line 2 * * page 18, line 22 - line 27 * * page 20, line 8 - line 17 * * page 27, line 21 - page 28, line 6 * * page 29, line 4 - line 23 * * page 40, line 13 - page 42, line 2 * -----	4,5,11	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	26 August 1998	Sindic, G

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 946 022 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
29.09.1999 Bulletin 1999/39

(51) Int. Cl.^{6}: **H04L 12/58**, H04L 29/06,
H04L 12/22

(21) Application number: 99105140.0

(22) Date of filing: 26.03.1999

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 26.03.1998 JP 7983798
18.06.1998 JP 17193098
07.08.1998 JP 22486198
05.11.1998 JP 31517298

(71) Applicant:
Nippon Telegraph and Telephone Corporation
Tokyo (JP)

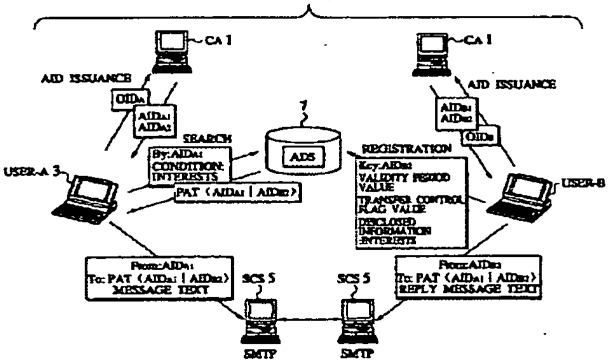
(72) Inventors:
- Hisada, Yusuke
  Nippon Telegraph Telephone Corp
  Shinjuku-ku, Tokyo 163-14 (JP)
- Ono, Satoshi
  Nippon Telegraph Telephone Corp
  Shinjuku-ku, Tokyo 163-14 (JP)
- Ichikawa, Haruhisa
  Nippon Telegraph Telephone Corp
  Shinjuku-ku, Tokyo 163-14 (JP)

(74) Representative: HOFFMANN - EITLE
Patent- und Rechtsanwälte
Arabellastrasse 4
81925 München (DE)

(54) Email access control scheme for communication network using identification concealment mechanism

(57) An email access control scheme capable of resolving problems of the real email address and enabling a unique identification of the identity of the user while concealing the user identification is disclosed. A personalized access ticket containing a sender's identification and a recipient's identification in correspondence is to be presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email. Then, accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket at a secure communication service. Also, an official identification of each user by which each user is uniquely identifiable by a certification authority, and an anonymous identification of each user containing at least one fragment of the official identification are defined, and each user is identified by the anonymous identification of each user in communications for emails on a communication network.

FIG.1


CA 1
CA 1
AID ISSUANCE
AID ISSUANCE
OIDA
AIDA1
AIDA2
AIDB1
AIDB2
OIDB
7
ADS
SEARCH
By:AIDA1
CONDITION:
INTERESTS
PAT (AIDA1 | AIDB2)
REGISTRATION
Key:AIDB2
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
DISCLOSED INFORMATION
:INTERESTS
USER-A 3
USER-B 3
From:AIDA1
To: PAT (AIDA1 | AIDB2)
MESSAGE TEXT
From:AIDB2
To: PAT (AIDA1 | AIDB2)
REPLY MESSAGE TEXT
SCS 5
SCS 5
SMTP
SMTP

EP 0 946 022 A2

Printed by Xerox (UK) Business Services
2.16.7/3.6

Description

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

[0001] The present invention relates to an email access control scheme for controlling transmission and reception of emails by controlling accesses for communications from other users whose identifications on the communication network are concealed while concealing an identification of a recipient on the communication network.

DESCRIPTION OF THE BACKGROUND ART

[0002] In conjunction with the spread of the Internet, the SPAM and the harassment using emails are drastically increasing. The SPAM is a generic name for emails or news that are unilaterally sent without any consideration to the recipient's time consumption, economical and mental burdens. The SPAM using emails are also known as UBE (Unsolicited Bulk Emails) or UCE (Unsolicited Commercial Emails).

[0003] The SPAM is sent indiscriminately regardless of the recipient's age, sex, interests, etc., so that the SPAM often contains an uninteresting or unpleasant content for the recipient. Moreover, the time consumption load and the economical load required for receiving the SPAM is not so small. For the business user, the SPAM can cause the lowering of the working efficiency as it becomes hard to find important mails that are buried among the SPAM. Also, as the SPAM is sent to a huge number of users, the SPAM wastes the network resources and in the worst case the SPAM can cause the overloading. As a result, there case be cases where mails that are important for the user may be lost. Also, the SPAM is sent either anonymously or by pretending someone else so that there is a need to provide some human resources to handle complaints.

[0004] On the other hand, the harassment is an act for keep sending mails with unpleasant contents for the user continually on the purpose of causing mental agony or exerting economical and time consumption burdens to the specific user. Similarly as the SPAM, the harassment mails are sent by pretending an actual or virtual third person, so that the identification of the sender is quite difficult. Also, there are cases where a large capacity mail is sent or a large amount of mails are sent in short period of time so that there is a danger of causing the system breakdown.

[0005] In order to deal with the SPAM and the harassment, the mail system is required to satisfy the following requirements.

- Security
  It is necessary to detect the pretending by the sender and refuse the delivery from the pretending sender.
- Strength
  It is necessary to limit the mail capacity in order to circumvent the system breakdown due to the large capacity mail. It is also necessary to limit the number of transmissions in order to circumvent the system breakdown due to the large amount transmission.
- Compatibility
  It is necessary not to require a considerable change to the implementation of the existing mail system.
- Handling
  It is necessary not to require a considerable change to the handling of the existing mail system.

The MTA (message Transfer Agent) such as sendmail and qmail detects the forgery of the envelope information and the header information and refuses the delivery. The MTA also refuses mail receiving from a mail server which is a source of the SPAM by referring to the so called black list such as MAPS RBL. The MTA also detects the transmission using someone else's real email address and refuses the delivery by carrying out the signature verification using PGP, S/MIME, TLS, etc. The MTA also limits the message length by partial deletion of the message text.

One of the causes of the SPAM and the harassment is the real email address, and the real email address is associated with the following problems.

- User's identity can be guessed from real email address:
  The real email address contains an information useful in guessing the identity so that it can be used in selecting the harassment target. For example, the place of employment can be identified from the real domain. Also, the name and the sex can be guessed from the user name.
- Real email address can be guessed from user's identity:
  The real email address has a universal format of [user name]@[domain name] so that the real email address can be guessed if the user's identity is known, without an explicit knowledge of the real email address itself. For example, if the user's real name is known, the candidates for the user name can be enumerated. Also, if the user's affiliation is known, the candidates for the domain name can be enumerated. Even in the case where the user name is given by a character string which is totally unrelated to the real name, if the naming rule for the user name is known, the user name can be guessed by trial and error transmissions.
- Real email address is transferrable:
  The real email address can be transferred from one person to another, so that mails can be transmitted even if the real email address is not taught by the holder himself. The transfer of real email

address through mails includes the following cases. By specifying the other's real email address in the cc: line of the mail, that real email address can be transferred to all the recipients specified in the To: line of the mail. Also, by forwarding the mail that contains the real email address of the recipient specified in the To: line in the message text to a third person, that real email address can be transferred to the third person.

- Real email address is hard to cancel:

  It is difficult to cancel the real email address because if the real email address is cancelled it becomes impossible to read not only the SPAM and the harassment mails but also the important mails as well.

[0006] Cypherpunk remailers and Mixmaster remailers which are collectively known as Anonymous remailers use a scheme for delivering mails after encrypting the real email address and the real domain of the sender. This scheme is called the reply block. The encryption and decryption of the reply block uses a public key and a secret key of the Anonymous remailer so that it is difficult to identify the real email address and the real domain of the sender for any users other than the sender.

[0007] The Anonymous remailers also make it difficult to transfer the real email address because it is difficult to identify the real email address. However, the reply block is transferrable, so that reply mails can be returned to the sender from users other than the recipient.

[0008] AS-Node and nym.alias.net which are collectively known as Pseudonymous servers use mail transmission and reception using a pseudonym account uniquely corresponding to the real email address of the user. The pseudonym account can be arbitrarily created at the user side so that the user can have a pseudonym account from which the real email address is hard to guess. In addition, by the use of the reply block, it is also possible to conceal the real email address and the real domain of the user to the Pseudonymous server. By combining these means, it can be made difficult to identify the real email address and the real domain of the sender for any users other than the sender. Also, the pseudonym account is cancellable so that there is no need to cancel the real email address.

[0009] The Pseudonymous servers also make it difficult to transfer the real email address because it is difficult to identify the real email address. However, the pseudonym account is transferrable so that reply mails can be returned to the sender from users other than the recipient.

[0010] In addition, in order to protect a recipient from the SPAM and the harrassment, it is also necessary to reject a connection request from a sender who are exercising such action. For this reason, it is necessary for the communication system to be capable of uniquely identifying the identity of the sender.

[0011] In view of these factors, the communication system is required to be capable of uniquely identifying the identity of the user while concealing the real email address of the user (that is while guaranteeing the anonymity of the user), but in the conventional communication system, it has been difficult to meet both of these requirements simultaneously.

[0012] In order to identify the identity of the user in the mail system, the real email address of that user is necessary. On the other hand, the Anonymous remailers deliver a mail after either encrypting or deleting the real email address of the sender in order to guarantee the anonymity of the sender. In order to identify the identity of the sender under this condition, it is necessary to trace the delivery route of the mail using the traffic analysis. However, the Anonymous remailers may delay the mail delivery or interchange the delivery orders of mails. Also, The Mixmaster remailers deliver the mail by dividing it into plural blocks. For this reason, it is difficult to trace the delivery route by the traffic analysis, and therefore the identification of the identity of the sender is also difficult.

[0013] The Pseudonymous servers also utilize the Anonymous remailers for the mail delivery, so that it is possible to guarantee the anonymity of the sender but it is also difficult to uniquely identify the identity of the sender.

[0014] On the other hand, the German Digital Signature Law allows entry of a pseudonym instead of a real name into a digital certificate for generating the digital signature to be used in communication services. The digital certificate is uniquely assigned to the user so that the identity of the user can be uniquely identified even if the pseudonym is entered. Also, the right for naming the pseudonym is given to the user side so that it is possible to enter the pseudonym from which it is difficult to guess the real name.

SUMMARY OF THE INVENTION

[0015] It is therefore an object of the present invention to provide an email access control scheme in a communication network which is capable of resolving the above described problems of the real email address which is one of the causes of the SPAM and the harassment.

[0016] It is another object of the present invention to provide an email access control scheme in a communication network which is capable of enabling a unique identification of the identity of the user while concealing the user identification.

[0017] In order to resolve the problems associated with the transfer and the cancellation of the real email address, the present invention employs the email access control scheme using a personalized access ticket (PAT). In order to resolve the problem associated with the transfer of the real email address, the destination is specified by the PAT which contains both the real email address of the sender and a real email address of the recipient. Also, in order to resolve the problem associated with the cancellation of the real email address, a validity period is set in the PAT by a Trusted Third Party. Then, the mail delivery from the sender who presented the PAT with the expired validity period will be refused. Also, instead of cancelling the real email address, the PAT is registered at a secure storage device managed by a secure communication service.

[0018] In other words, the present invention controls accesses in units in which the real email address of the sender and the real email address of the recipient is paired. For this reason, even when the real email address is transferred, it is possible to avoid receiving mails from users to which the real email address has been transferred as long as the PAT is not acquired by these users.

[0019] Also, in the present invention, it is possible to refuse receiving mails without cancelling the real email address because the mail delivery from the sender who presented the PAT with the expired validity period or the PAT that is registered in a database by the recipient will be refused.

[0020] Also, in the present invention, the mail receiving can be resumed without re-acquiring the real email address because the mail receiving can be resumed by deleting the PAT from the above described storage device.

[0021] Also, in the present invention, the time consumption and economical loads required for the mail receiving or downloading at the user side can be reduced because the transmission of mails are refused at the server side.

[0022] In addition, the present invention employs the email access control scheme using an official identification (OID) and an anonymous identification (AID) in order to make it possible to identify the identity of the user while guaranteeing the anonymity of the user.

[0023] Namely, in the present invention, a certificate in which the personal information is signed by a secret key of the Trusted Third Party is assigned to each user in order to uniquely identify each user. This certificate will be referred to as OID. Also, a certificate which contains fragments of the OID information is assigned to each user as a user identifier on a communication network in order to make it possible to identify the identity while guaranteeing the anonymity of the user. This certificate will be referred to as AID.

[0024] Also, in the present invention, the OID is reconstructed by judging the identity of a plurality of AIDs in order to identify the identity of the user. Also, the AID is contained in the PAT and the PAT is authenticated at a secure communication service (SCS) in order to resolve the problems associated with the transfer and the cancellation of the AID.

[0025] Also, in the present invention, the AID is managed in a directory which is accessible for search by unspecified many and which outputs the PAT containing the AID as a destination, in order to meet the user side demand for being able to admit accesses from unspecified many without revealing the own identity.

[0026] In this way, in the present invention, the identity of the user can be concealed in the mail transmission and reception because the AID only contains fragments of the OID. Also, the identity of the user can be concealed from unspecified many even when the AID is registered at the directory service which is accessible from unspecified many.

[0027] Also, in the present invention, the identity of the user can be identified probabilistically by reconstructing the OID by judging the identity of a plurality of AIDs. For this reason, it is possible to provide a measure against the SPAM and the harassment without revealing the identity.

[0028] Also, in the present invention, it is possible to admit accesses from unspecified many without revealing the identity, by managing the AID rather than the real email address at the directory and outputting the PAT containing the AID as a destination at the directory.

[0029] More specifically, according to one aspect of the present invention there is provided a method of email access control, comprising the steps of: receiving a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, at a secure communication service for connecting communications between the sender and the receiver; and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket at the secure communication service.

[0030] Also, in this aspect of the present invention, at the controlling step the secure communication service authenticates the personalized access ticket presented by the sender, and refuses a delivery of the email when the personalized access ticket presented by the sender has been altered.

[0031] Also, in this aspect of the present invention, the personalized access ticket is signed by a secret key of a secure processing device which issued the personalized access ticket, and at the controlling step the secure communication service authenticates the personalized access ticket by verifying a signature of the secure processing device in the personalized access ticket using a public key of the secure processing device.

[0032] Also, in this aspect of the present invention, at the receiving step the secure communication service also receives the sender's identification presented by the sender along with the personalized access ticket, and at the controlling step the secure communication service checks whether the sender's identification presented by the sender is contained in the personalized access ticket presented by the sender, and refuses a delivery of the email when the sender's identification presented by the sender is not contained in the personalized access ticket presented by the sender.

[0033] Also, in this aspect of the present invention, the personalized access ticket also contains a validity period indicating a period for which the personalized access ticket is valid, and at the controlling step the secure communication service checks the validity period contained in the personalized access ticket presented by the sender and refuses a delivery of the email when the personalized access ticket presented by the sender contains the validity period that has already been expired.

[0034] Also, in this aspect of the present invention, the validity period of the personalized access ticket is set by a trusted third party.

[0035] Also, in this aspect of the present invention, the method can further comprise the step of: issuing the personalized access ticket to the sender at a directory service for managing an identification of each registrant and a disclosed information of each registrant which has a lower secrecy than a personal information, in a state which is accessible for search by unspecified many, in response to search conditions specified by the sender, by using an identification of a registrant whose disclosed information matches the search conditions as the recipient's identification and the sender's identification specified by the sender along with the search conditions.

[0036] Also, in this aspect of the present invention, the method can further comprise the step of: registering in advance the personalized access ticket containing an identification of a specific user from which a delivery of emails to a specific registrant is to be refused as the sender's identification and an identification of the specific registrant as the recipient's identification, at the secure communication service; wherein the controlling step the secure communication service refuses a delivery of the email from the sender when the personalized access ticket presented by the sender is registered therein in advance at the registering step.

[0037] Also, in this aspect of the present invention, the method can further comprise the step of: deleting the personalized access ticket registered at the secure communication service upon request from the specific registrant who registered the personalized access ticket at the registering step.

[0038] Also, in this aspect of the present invention, the personalized access ticket also contains a transfer control flag indicating whether or not the sender should be authenticated by the secure communication service, and at the controlling step, when the transfer control flag contained in the personalized access ticket indicates that the sender should be authenticated, the secure communication service authenticates the sender's identification presented by the sender and refuses a delivery of the email when an authentication of the sender's identification fails.

[0039] Also, in this aspect of the present invention, the authentication of the sender's identification is realized by a challenge/response procedure between the sender and the secure communication service.

[0040] Also, in this aspect of the present invention, the transfer control flag of the personalized access ticket is set by a trusted third party.

[0041] Also, in this aspect of the present invention, the sender's identification and the recipient's identification in the personalized access ticket can be given by real email addresses of the sender and the recipient.

[0042] Also, in this aspect of the present invention, the sender's identification and the recipient's identification in the personalized access ticket can be given by anonymous identifications of the sender and the recipient, where an anonymous identification of each user contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority.

[0043] Also, in this aspect of the present invention, the anonymous identification of each user is an information containing the at least one fragment of the official identification of each user which is signed by the certification authority using a secret key of the certification authority.

[0044] Also, in this aspect of the present invention, the official identification of each user is a character string uniquely assigned to each user by the certification authority and a public key of each user which are signed by a secret key of the certification authority.

[0045] Also, in this aspect of the present invention, the method can further comprise the step of: probabilistically identifying an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

[0046] Also, in this aspect of the present invention, an anonymous identification of each user that contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority and a link information of each anonymous identification by which each anonymous identification can be uniquely identified can be defined, and the sender's identification and the recipient's identification in the personalized access ticket can be given by a link information of the anonymous identification of the sender and a link information of the anonymous identification of the recipient.

[0047] Also, in this aspect of the present invention, the link information of each anonymous identification is an identifier uniquely assigned to each anonymous identification by the certification authority.

[0048] Also, in this aspect of the present invention, the method can further comprise the step of: probabilistically identifying an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

[0049] Also, in this aspect of the present invention, the personalized access ticket can contain a single sender's identification and a single recipient's identification in 1-to-1 correspondence.

[0050] Also, in this aspect of the present invention, the personalized access ticket can contain a single sender's identification and a plurality of recipient's identifications in 1-to-N correspondence, where N is an integer greater than 1.

[0051] Also, in this aspect of the present invention, one identification among the single sender's identification and the plurality of recipient's identifications is a holder identification for identifying a holder of the personalized access ticket while other identifications among the single sender's identification and the plurality of recipient's identifications are member identifications for identifying members of a group to which the holder belongs.

[0052] Also, in this aspect of the present invention, the method can further comprise the step of: issuing an identification of each user and an enabler of the identification of each user indicating a right to change the personalized access ticket containing the identification of each user as the holder identification, to each user at a certification authority, such that prescribed processing on the personalized access ticket can be carried out at a secure processing device only by a user who presented both the holder identification contained in the personalized access ticket and the enabler corresponding to the holder identification to the secure processing device.

[0053] Also, in this aspect of the present invention, the certification authority issues the enabler of the identification of each user as an information indicating that it is the enabler and the identification of each user itself which are signed by a secret key of the certification authority.

[0054] Also, in this aspect of the present invention, the prescribed processing includes a generation of a new personalized access ticket, a merging of a plurality of personalized access tickets, a splitting of one personalized access ticket into a plurality of personalized access tickets, a changing of the holder of the personalized access ticket, changing of a validity period of the personalized access ticket, and a changing of a transfer control flag of the personalized access ticket.

[0055] Also, in this aspect of the present invention, a special identification and a special enabler corresponding to the special identification which are known to all users can be defined such that the generation of a new personalized access ticket and the changing of the holder of the personalized access ticket can be carried out by the holder of the personalized access ticket by using the special identification and the special enabler without using an enabler of a member identification.

[0056] Also, in this aspect of the present invention, the special identification is defined to be capable of being used only as the holder identification of the personalized access ticket.

[0057] Also, in this aspect of the present invention, a special identification which is known to all users can be defined such that a read only attribute can be set to the personalized access ticket by using the special identification.

[0058] Also, in this aspect of the present invention, at the controlling step, when the access right of the sender with respect to the recipient is verified according to the personalized access ticket, the secure communication service takes out the recipient's identification from the personalized access ticket by using the sender's identification presented by the sender, converts the mail by using a taken out recipient's identification into a format that can be interpreted by a mail transfer function for actually carrying out a mail delivery processing, and gives the mail after conversion to the mail transfer function by attaching the personalized access ticket.

[0059] According to another aspect of the present invention there is provided a method of email access control, comprising the steps of: defining an official identification of each user by which each user is uniquely identifiable by a certification authority, and an anonymous identification of each user containing at least one fragment of the official identification; and identifying each user by the anonymous identification of each user in communications for emails on a communication network.

[0060] Also, in this aspect of the present invention, the anonymous identification of each user is an information containing the at least one fragment of the official identification of each user which is signed by the certification authority using a secret key of the certification authority.

[0061] Also, in this aspect of the present invention, the official identification of each user is a character string uniquely assigned to each user by the certification authority and a public key of each user which are signed by a secret key of the certification authority.

[0062] Also, in this aspect of the present invention, the method can further comprise the steps of: receiving a personalized access ticket containing a sender's anonymous identification and a recipient's anonymous identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, at a secure communication service for connecting communications between the sender and the receiver; and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket at the secure communication service.

[0063] Also, in this aspect of the present invention, the method can further comprises the step of: probabilistically identifying an identity of the sender at the secure communication service by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

[0064] Also, in this aspect of the present invention, the defining step can also define a link information of each anonymous identification by which each anonymous identification can be uniquely identified, and each anonymous identification can also contain the link information of each anonymous identification.

[0065] Also, in this aspect of the present invention, the link information of each anonymous identification is an identifier uniquely assigned to each anonymous identification by the certification authority.

[0066] Also, in this aspect of the present invention, the method can further comprises the steps of: receiving a personalized access ticket containing a link information of a sender's anonymous identification and a link information of a recipient's anonymous identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, at a secure communication service for connecting communications between the sender and the receiver; and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket at the secure communication service.

[0067] Also, in this aspect of the present invention, the method can further comprises the step of: probabilistically identifying an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

[0068] According to another aspect of the present invention there is provided a communication system realizing email access control, comprising: a communication network to which a plurality of user terminals are connected; and a secure communication service device for connecting communications between the sender and the receiver on the communication network, by receiving a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket.

[0069] Also, in this aspect of the present invention, the secure communication service device authenticates the personalized access ticket presented by the sender, and refuses a delivery of the email when the personalized access ticket presented by the sender has been altered.

[0070] Also, in this aspect of the present invention, the system further comprises: a secure processing device for issuing the personalized access ticket which is signed by a secret key of the secure processing device; wherein the secure communication service device authenticates the personalized access ticket by verifying a signature of the secure processing device in the personalized access ticket using a public key of the secure processing device.

[0071] Also, in this aspect of the present invention, the secure communication service device also receives the sender's identification presented by the sender along with the personalized access ticket, checks whether the sender's identification presented by the sender is contained in the personalized access ticket presented by the sender, and refuses a delivery of the email when the sender's identification presented by the sender is not contained in the personalized access ticket presented by the sender.

[0072] Also, in this aspect of the present invention, the personalized access ticket also contains a validity period indicating a period for which the personalized access ticket is valid, and the secure communication service device checks the validity period contained in the personalized access ticket presented by the sender and refuses a delivery of the email when the personalized access ticket presented by the sender contains the validity period that has already been expired.

[0073] Also, in this aspect of the present invention, the system further comprises: a trusted third party for setting the validity period of the personalized access ticket.

[0074] Also, in this aspect of the present invention, the system can further comprise: a directory service device for managing an identification of each registrant and and a disclosed information of each registrant which has a lower secrecy than a personal information, in a state which is accessible for search by unspecified many, and issuing the personalized access ticket to the sender in response to search conditions specified by the sender, by using an identification of a registrant whose disclosed information matches the search conditions as the recipient's identification and the sender's identification specified by the sender along with the search conditions.

[0075] Also, in this aspect of the present invention, the secure communication service device can register in advance the personalized access ticket containing an identification of a specific user from which a delivery of emails to a specific registrant is to be refused as the sender's identification and an identification of the specific registrant as the recipient's identification, and refuse a delivery of the email from the sender when the personalized access ticket presented by the sender is registered therein in advance.

[0076] Also, in this aspect of the present invention, the secure communication service device can delete the personalized access ticket registered therein upon request from the specific registrant who registered the personalized access ticket.

[0077] Also, in this aspect of the present invention, the personalized access ticket also contains a transfer control flag indicating whether or not the sender should be authenticated by the secure communication service, and when the transfer control flag contained in the personalized access ticket indicates that the sender should be authenticated, the secure communication service device authenticates the sender's identification presented by the sender and refuses a delivery of the email when an authentication of the sender's identification fails.

[0078] Also, in this aspect of the present invention, the authentication of the sender's identification is realized by a challenge/response procedure between the sender and the secure communication service device.

[0079] Also, in this aspect of the present invention, the system further comprises a trusted third party for setting the transfer control flag of the personalized access ticket.

[0080] Also, in this aspect of the present invention, the sender's identification and the recipient's identification in the personalized access ticket can be given by real email addresses of the sender and the recipient.

[0081] Also, in this aspect of the present invention, the system can further comprise: a certification authority device for issuing an anonymous identification of each user which contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by the certification authority device; wherein the sender's identification and the recipient's identification in the personalized access ticket can be given by anonymous identifications of the sender and the recipient.

[0082] Also, in this aspect of the present invention, the anonymous identification of each user is an information containing the at least one fragment of the official identification of each user which is signed by the certification authority device using a secret key of the certification authority device.

[0083] Also, in this aspect of the present invention, the official identification of each user is a character string uniquely assigned to each user by the certification authority device and a public key of each user which are signed by a secret key of the certification authority device.

[0084] Also, in this aspect of the present invention, the secure communication service device can probabilistically identify an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

[0085] Also, in this aspect of the present invention, the system can further comprise: a certification authority device for issuing an anonymous identification of each user which contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by the certification authority device and a link information of each anonymous identification by which each anonymous identification can be uniquely identified; wherein the sender's identification and the recipient's identification in the personalized access ticket can be given by a link information of the anonymous identification of the sender and a link information of the anonymous identification of the recipient.

[0086] Also, in this aspect of the present invention, the link information of each anonymous identification is an identifier uniquely assigned to each anonymous identification by the certification authority device.

[0087] Also, in this aspect of the present invention, the secure communication service device can probabilistically identify an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

[0088] Also, in this aspect of the present invention, the personalized access ticket can contain a single sender's identification and a single recipient's identification in 1-to-1 correspondence.

[0089] Also, in this aspect of the present invention, the personalized access ticket can contain a single sender's identification and a plurality of recipient's identifications in 1-to-N correspondence, where N is an integer greater than 1.

[0090] Also, in this aspect of the present invention, one identification among the single sender's identification and the plurality of recipient's identifications is a holder identification for identifying a holder of the personalized access ticket while other identifications among the single sender's identification and the plurality of recipient's identifications are member identifications for identifying members of a group to which the holder belongs.

[0091] Also, in this aspect of the present invention, the system can further comprises: a certification authority device for issuing to each user an identification of each user and an enabler of the identification of each user indicating a right to change the personalized access ticket containing the identification of each user as the holder identification; and a secure processing device at which prescribed processing on the personalized access ticket can be carried out only by a user who presented both the holder identification contained in the personalized access ticket and the enabler corresponding to the holder identification to the secure processing device.

[0092] Also, in this aspect of the present invention, the certification authority device issues the enabler of the identification of each user as an information indicating that it is the enabler and the identification of each user itself which are signed by a secret key of the certification authority device.

[0093] Also, in this aspect of the present invention, the prescribed processing includes a generation of a new personalized access ticket, a merging of a plurality of personalized access tickets, a splitting of one personalized access ticket into a plurality of personalized access tickets, a changing of the holder of the personalized access ticket, changing of a validity period of the personalized access ticket, and a changing of a transfer control flag of the personalized access ticket.

[0094] Also, in this aspect of the present invention, a special identification and a special enabler corresponding to the special identification which are known to all users can be defined such that the generation of a new personalized access ticket and the changing of the holder of the personalized access ticket can be carried out by the holder of the personalized access ticket by using the special identification and the special enabler without using an enabler of a member identification.

[0095] Also, in this aspect of the present invention, the special identification is defined to be capable of being used only as the holder identification of the personalized access ticket.

[0096] Also, in this aspect of the present invention, a special identification which is known to all users can be defined such that a read only attribute can be set to the personalized access ticket by using the special identification.

[0097] Also, in this aspect of the present invention, when the access right of the sender with respect to the recipient is verified according to the personalized access ticket, the secure communication service device takes out the recipient's identification from the personalized access ticket by using the sender's identification presented by the sender, converts the mail by using a taken out recipient's identification into a format that can be interpreted by a mail transfer function for actually carrying out a mail delivery processing, and gives the mail after conversion to the mail transfer function by attaching the personalized access ticket.

[0098] According to another aspect of the present invention there is provided a communication system realizing email access control, comprising: a certification authority device for defining an official identification of each user by which each user is uniquely identifiable by the certification authority device, and an anonymous identification of each user which contains at least one fragment of the official identification; and a communication network on which each user is identified by the anonymous identification of each user in communications for emails on the communication network.

[0099] Also, in this aspect of the present invention, the anonymous identification of each user is an information containing the at least one fragment of the official identification of each user which is signed by the certification authority device using a secret key of the certification authority device.

[0100] Also, in this aspect of the present invention, the official identification of each user is a character string uniquely assigned to each user by the certification authority device and a public key of each user which are signed by a secret key of the certification authority device.

[0101] Also, in this aspect of the present invention, the system can further comprises: a secure communication service device for connecting communications between the sender and the receiver on the communication network, by receiving a personalized access ticket containing a sender's anonymous identification and a recipient's anonymous identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket.

[0102] Also, in this aspect of the present invention, the secure communication service device can probabilistically identify an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

[0103] Also, in this aspect of the present invention, the certification authority device can also define a link information of each anonymous identification by which each anonymous identification can be uniquely identified, and each anonymous identification can also contain the link information of each anonymous identification.

[0104] Also, in this aspect of the present invention, the link information of each anonymous identification is an identifier uniquely assigned to each anonymous identification by the certification authority device.

[0105] Also, in this aspect of the present invention, the system can further comprise: a secure communication service device for connecting communications between the sender and the receiver on the communication network, by receiving a personalized access ticket containing a link information of a sender's anonymous identification and a link information of a recipient's anonymous identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket.

[0106] Also, in this aspect of the present invention, the secure communication service device can probabilistically identify an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of link informations of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

[0107] According to another aspect of the present invention there is provided a secure communication service device for use in a communication system realizing email access control, comprising: a computer hardware; and a computer software for causing the computer hardware to connect communications between the sender and the receiver, by receiving a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket.

[0108] Also, in this aspect of the present invention, the computer software causes the computer hardware to authenticate the personalized access ticket presented by the sender, and refuse a delivery of the email when the personalized access ticket presented by the sender has been altered.

[0109] Also, in this aspect of the present invention, the personalized access ticket is signed by a secret key of a secure processing device which issued the personalized access ticket, and the computer software causes the computer hardware to authenticate the personalized access ticket by verifying a signature of the secure processing device in the personalized access ticket using a public key of the secure processing device.

[0110] Also, in this aspect of the present invention, the computer software causes the computer hardware to also receive the sender's identification presented by the sender along with the personalized access ticket, check whether the sender's identification presented by the sender is contained in the personalized access ticket presented by the sender, and refuse a delivery of the email when the sender's identification presented by the sender is not contained in the personalized access ticket presented by the sender.

[0111] Also, in this aspect of the present invention, the personalized access ticket also contains a validity period indicating a period for which the personalized access ticket is valid, and the computer software causes the computer hardware to check the validity period contained in the personalized access ticket presented by the sender and refuse a delivery of the email when the personalized access ticket presented by the sender contains the validity period that has already been expired.

[0112] Also, in this aspect of the present invention, the computer software can cause the computer hardware to register in advance the personalized access ticket containing an identification of a specific user from which a delivery of emails to a specific registrant is to be refused as the sender's identification and an identification of the specific registrant as the recipient's identification, at the secure communication service device, and refuse a delivery of the email from the sender when the personalized access ticket presented by the sender is registered at the secure communication service device in advance.

[0113] Also, in this aspect of the present invention, the computer software can cause the computer hardware to delete the personalized access ticket registered at the secure communication service device upon request from the specific registrant who registered the personalized access ticket.

[0114] Also, in this aspect of the present invention, the personalized access ticket also contains a transfer control flag indicating whether or not the sender should be authenticated by the secure communication service device, and when the transfer control flag contained in the personalized access ticket indicates that the sender should be authenticated, the computer software causes the computer hardware to authenticate the sender's identification presented by the sender and refuse a delivery of the email when an authentication of the sender's identification fails.

[0115] Also, in this aspect of the present invention, the computer software causes the computer hardware to realize the authentication of the sender's identification by a challenge/response procedure between the sender and the secure communication service device.

[0116] Also, in this aspect of the present invention, the sender's identification and the recipient's identification in the personalized access ticket can be given by anonymous identifications of the sender and the recipient, where an anonymous identification of each user contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority, and the computer software can also cause the computer hardware to probabilistically identify an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

[0117] Also, in this aspect of the present invention, an anonymous identification of each user that contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority and a link information of each anonymous identification by which each anonymous identification can be uniquely identified can be defined, the sender's identification and the recipient's identification in the personalized access ticket can be given by a link information of the anonymous identification of the sender and a link information of the anonymous identification of the recipient, and the computer software can also cause the computer hardware to probabilistically identify an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

[0118] Also, in this aspect of the present invention, when the access right of the sender with respect to the recipient is verified according to the personalized access ticket, the computer software causes the computer hardware to take out the recipient's identification from the personalized access ticket by using the sender's identification presented by the sender, convert the mail by using a taken out recipient's identification into a format that can be interpreted by a mail transfer function for actually carrying out a mail delivery processing, and give the mail after conversion to the mail transfer function by attaching the personalized access ticket.

[0119] According to another aspect of the present invention there is provided a secure processing device for use in a communication system realizing email access control, comprising: a computer hardware; and a computer software for causing the computer hardware to receive a request for a personalized access ticket from a user, and issue a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is signed by a secret key of the secure processing device.

[0120] According to another aspect of the present invention there is provided a directory service device for use in a communication system realizing email access control, comprising: a computer hardware; and a computer software for causing the computer hardware to manage an identification of each registrant and a disclosed information of each registrant which has a lower secrecy than a personal information, in a state which is accessible for search by unspecified many, and issue a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, to the sender in response to search conditions specified by the sender, by using an identification of a registrant whose disclosed information matches the search conditions as the recipient's identification and the sender's identification specified by the sender along with the search conditions.

[0121] According to another aspect of the present invention there is provided a certification authority device for use in a communication system realizing email access control, comprising: a computer hardware; and a computer software for causing the computer hardware to issue to each user an official identification of each user by which each user is uniquely identifiable by the certification authority device, and an anonymous identification of each user which contains at least one fragment of the official identification.

[0122] According to another aspect of the present invention there is provided a certification authority device for use in a communication system realizing email access control, comprising: a computer hardware; and a computer software for causing the computer hardware to issue to each user an identification of each user and an enabler of the identification of each user indicating a right to change any personalized access ticket that contains the identification of each user as a holder identification, where the persnalized access ticket generally contains a sender's identification and a plurality of recipient's identifications in correspondence, and one of the sender's identification and the recipient's identifications is a holder identification.

[0123] According to another aspect of the present invention there is provided a secure processing device for use in a communication system realizing email access control, comprising: a computer hardware; and a computer software for causing the computer hardware to receive from a user a request for prescribed processing on a personalized access ticket containing a sender's identification and a plurality of recipient's identifications in correspondence, where one of the sender's identification and the recipient's identifications is a holder identification, and execute the prescribed processing on the personalized access ticket when the user presented both the holder identification contained in the personalized access ticket and an enabler corresponding to the holder identification which indicates a right to change the personalized access ticket containing the identification of the user as the holder identification.

[0124] According to another aspect of the present invention there is provided a computer usable medium having computer readable program code means embodied therein for causing a computer to function as a secure communication service device for use in a communication system realizing email access control, the computer readable program code means includes: first computer readable program code means for causing said computer to receive a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email; and second computer readable program code means for causing said computer to control accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket, so as to connect communications between the sender and the receiver on the communication network.

[0125] Also, in this aspect of the present invention, the second computer readable program code means causes said computer to authenticate the personalized access ticket presented by the sender, and refuse a delivery of the email when the personalized access ticket presented by the sender has been altered.

[0126] Also, in this aspect of the present invention, the personalized access ticket is signed by a secret key of a secure processing device which issued the personalized access ticket, and the second computer readable program code means causes said computer to authenticate the personalized access ticket by verifying a signature of the secure processing device in the personalized access ticket using a public key of the secure processing device.

[0127] Also, in this aspect of the present invention, the first computer readable program code means causes said computer to also receive the sender's identification presented by the sender along with the personalized access ticket, and the second computer readable program code means causes said computer to check whether the sender's identification presented by the sender is contained in the personalized access ticket presented by the sender and refuse a delivery of the email when the sender's identification presented by the sender is not contained in the personalized access ticket presented by the sender.

[0128] Also, in this aspect of the present invention, the personalized access ticket also contains a validity period indicating a period for which the personalized access ticket is valid, and the second computer readable program code means causes said computer to check the validity period contained in the personalized access ticket presented by the sender and refuse a delivery of the email when the personalized access ticket presented by the sender contains the validity period that has already been expired.

[0129] Also, in this aspect of the present invention, the second computer readable program code means can cause said computer to register in advance the personalized access ticket containing an identification of a specific user from which a delivery of emails to a specific registrant is to be refused as the sender's identification and an identification of the specific registrant as the recipient's identification, at the secure communication service device, and refuse a delivery of the email from the sender when the personalized access ticket presented by the sender is registered at the secure communication service device in advance.

[0130] Also, in this aspect of the present invention, the second computer readable program code means can cause said computer to delete the personalized access ticket registered at the secure communication service device upon request from the specific registrant who registered the personalized access ticket.

[0131] Also, in this aspect of the present invention, the personalized access ticket also contains a transfer control flag indicating whether or not the sender should be authenticated by the secure communication service device, and when the transfer control flag contained in the personalized access ticket indicates that the sender should be authenticated, the second computer readable program code means causes said computer to authenticate the sender's identification presented by the sender and refuse a delivery of the email when an authentication of the sender's identification fails.

[0132] Also, in this aspect of the present invention, the second computer readable program code means causes said computer to realize the authentication of the sender's identification by a challenge/response procedure between the sender and the secure communication service device.

[0133] Also, in this aspect of the present invention, the sender's identification and the recipient's identification in the personalized access ticket can be given by anonymous identifications of the sender and the recipient, where an anonymous identification of each user contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority, and the second computer readable program code means can also cause said computer to probabilistically identify an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

[0134] Also, in this aspect of the present invention, an anonymous identification of each user that contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority and a link information of each anonymous identification by which each anonymous identification can be uniquely identified can be defined, the sender's identification and the recipient's identification in the personalized access ticket can be given by a link information of the anonymous identification of the sender and a link information of the anonymous identification of the recipient, and the second computer readable program code means can also cause said computer to probabilistically identify an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

[0135] Also, in this aspect of the present invention, when the access right of the sender with respect to the recipient is verified according to the personalized access ticket, the second computer readable program code means causes said computer to take out the recipient's identification from the personalized access ticket by using the sender's identification presented by the sender, convert the mail by using a taken out recipient's identification into a format that can be interpreted by a mail transfer function for actually carrying out a mail delivery processing, and give the mail after conversion to the mail transfer function by attaching the personalized access ticket.

[0136] According to another aspect of the present invention there is provided a computer usable medium having computer readable program code means embodied therein for causing a computer to function as a secure processing device for use in a communication system realizing email access control, the computer readable program code means includes: first computer readable program code means for causing said computer to receive a request for a personalized access ticket from a user; and second computer readable program code means for causing said computer to issue the personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is signed by a secret key of the secure processing device.

[0137] According to another aspect of the present invention there is provided a computer usable medium having computer readable program code means embodied therein for causing a computer to function as a directory service devicer for use in a communication system realizing email access control, the computer readable program code means includes: first computer readable program code means for causing said computer to manage an identification of each registrant and a disclosed information of each registrant which has a lower secrecy than a personal information, in a state which is accessible for search by unspecified many, and second computer readable program code means for causing said computer to issue a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, to the sender in response to search conditions specified by the sender, by using an identification of a registrant whose disclosed information matches the search conditions as the recipient's identification and the sender's identification specified by the sender along with the search conditions.

[0138] According to another aspect of the present invention there is provided a computer usable medium having computer readable program code means embodied therein for causing a computer to function as a certification authority device for use in a communication system realizing email access control, the computer readable program code means includes: first computer readable program code means for causing said computer to issue to each user an official identification of each user by which each user is uniquely identifiable by the certification authority device; and second computer readable program code means for causing said computer to issue to each user an anonymous identification of each user which contains at least one fragment of the official identification.

[0139] According to another aspect of the present invention there is provided a computer usable medium having computer readable program code means embodied therein for causing a computer to function as a certification authority device for use in a communication system realizing email access control, the computer readable program code means includes: first computer readable program code means for causing said computer to issue to each user an identification of each user; and second computer readable program code means for causing said computer to issue to each user an enabler of the identification of each user indicating a right to change any personalized access ticket that contains the identification of each user as a holder identification, where the persnalized access ticket generally contains a sender's identification and a plurality of recipient's identifications in correspondence, and one of the sender's identification and the recipient's identifications is a holder identification.

[0140] According to another aspect of the present invention there is provided a computer usable medium having computer readable program code means embodied therein for causing a computer to function as a secure processing device for use in a communication system realizing email access control, the computer readable program code means includes: first computer readable program code means for causing said computer to receive from a user a request for prescribed processing on a personalized access ticket containing a sender's identification and a plurality of recipient's identifications in correspondence, where one of the sender's identification and the recipient's identifications is a holder identification; and second computer readable program code means for causing said computer to execute the prescribed processing on the personalized access ticket when the user presented both the holder identification contained in the personalized access ticket and an enabler corresponding to the holder identification which indicates a right to change the personalized access ticket containing the identification of the user as the holder identification.

[0141] Other features and advantages of the present invention will become apparent from the following description taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0142]

Fig. 1 is a diagram showing an overall configuration of a communication system according to the first embodiment of the present invention.

Fig. 2 is a diagram showing exemplary data structures of an official identification, an anonymous identification, and a 1-to-1 personalized access ticket according to the first embodiment of the present invention.

Fig. 3 is a flow chart for an anonymous identification generation processing at a certification authority according to the first embodiment of the present invention.

Fig. 4 is a flow chart for a personalized access ticket generation processing at an anonymous directory service according to the first embodiment of the present invention.

Fig. 5 is a flow chart for a mail access control processing at a secure communication service according to the first embodiment of the present invention.

Fig. 6 is a flow chart for an anonymous identification identity judgement processing at a secure communication service according to the first embodiment of the present invention.

Fig. 7 is a diagram showing exemplary data structures of data used in the anonymous identification identity judgement processing of Fig. 6.

Fig. 8 is a diagram showing exemplary data structures of an official identification, an anonymous identification, and a 1-to-N personalized access ticket according to the second embodiment of the present invention.

Fig. 9 is a diagram showing exemplary data structures of an anonymous identification and an enabler according to the second embodiment of the present invention.

Fig. 10 is a diagram showing a definition of a processing rule (MakePAT) used in the second embodiment of the present invention.

Fig. 11 is a diagram showing a definition of a processing rule (MergePAT) used in the second embodiment of the present invention.

Fig. 12 is a diagram showing a definition of a processing rule (SplitPAT) used in the second embodiment of the present invention.

Fig. 13 is a diagram showing a definition of a processing rule (TransPAT) used in the second embodiment of the present invention.

Fig. 14 is a first exemplary system configuration that can be used in the second embodiment of the present invention.

Fig. 15 is a second exemplary system configuration that can be used in the second embodiment of the present invention.

Fig. 16 is a third exemplary system configuration that can be used in the second embodiment of the present invention.

Fig. 17 is a fourth exemplary system configuration that can be used in the second embodiment of the present invention.

Fig. 18 is a fifth exemplary system configuration that can be used in the second embodiment of the present invention.

Fig. 19 is a sixth exemplary system configuration that can be used in the second embodiment of the present invention.

Fig. 20 is a seventh exemplary system configuration that can be used in the second embodiment of the present invention.

Fig. 21 is a flow chart showing an overall processing flow of MakePAT, MergePAT or TransPAT processing according to the second embodiment of the present invention.

Fig. 22 is a flow chart showing an overall processing flow of SplitPAT processing according to the second embodiment of the present invention.

Fig. 23 is a flow chart for an anonymous identification list generation processing (for MakePAT, MergePAT, SplitPAT and TransPAT) according to the second embodiment of the present invention.

Fig. 24 is an enabler authenticity verification processing (for MakePAT, MergePAT, SplitPAT and TransPAT) according to the second embodiment of the present invention.

Fig. 25 is a diagram showing an exemplary data structure of Null-AID used in the third embodiment of the present invention.

Fig. 26 is a diagram showing an exemplary data structure of Enabler of Null-AID used in the third embodiment of the present invention.

Fig. 27 is a diagram showing a first exemplary application of the third embodiment of the present invention.

Fig. 28 is a diagram showing a second exemplary application of the third embodiment of the present invention.

Fig. 29 is a diagram showing an exemplary data structure of God-AID used in the fourth embodiment of the present invention.

Fig. 30 is a diagram showing a first exemplary application of the fourth embodiment of the present invention.

Fig. 31 is a diagram showing a second exemplary application of the fourth embodiment of the present invention.

Fig. 32 is a flow chart for a member anonymous identification checking processing according to the fifth embodiment of the present invention.

Fig. 33 is a diagram showing an overall configuration of a communication system according to the sixth embodiment of the present invention.

Fig. 34 is a diagram showing exemplary data structures of an official identification, a link information attached anonymous identification, and a link specifying 1-to-1 personalized access ticket according to the sixth embodiment of the present invention.

Fig. 35 is a flow chart for a link information attached anonymous identification generation processing at a certification authority according to the sixth embodiment of the present invention.

Fig. 36 is a flow chart for a link specifying 1-to-1 personalized access ticket generation processing at an anonymous directory service according to the sixth embodiment of the present invention.

Fig. 37 is a flow chart for a mail access control processing at a secure communication service according to the sixth embodiment of the present invention.

Fig. 38 is a flow chart for an anonymous identification identity judgement processing at a secure communication service according to the sixth embodiment of the present invention.

Fig. 39 is a diagram showing exemplary data structures of data used in the anonymous identification identity judgement processing of Fig. 38.

Fig. 40 is a diagram showing exemplary data structures of an official identification, a link information attached anonymous identification, and a link specifying 1-to-N personalized access ticket according to the seventh embodiment of the present invention.

Fig. 41 is a diagram showing exemplary data structures of a link information attached anonymous identification and an enabler according to the seventh embodiment of the present invention.

Fig. 42 is a first exemplary system configuration that can be used in the seventh embodiment of the present invention.

Fig. 43 is a second exemplary system configuration that can be used in the seventh embodiment of the present invention.

Fig. 44 is a third exemplary system configuration that can be used in the seventh embodiment of the present invention.

Fig. 45 is a fourth exemplary system configuration that can be used in the seventh embodiment of the present invention.

Fig. 46 is a fifth exemplary system configuration that can be used in the seventh embodiment of the present invention.

Fig. 47 is a sixth exemplary system configuration that can be used in the seventh embodiment of the present invention.

Fig. 48 is a seventh exemplary system configuration that can be used in the seventh embodiment of the present invention.

Fig. 49 is a flow chart for a link specifying anonymous identification list generation processing (for MakePAT, MergePAT, SplitPAT and TransPAT) according to the seventh embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0143] Referring now to Fig. 1 to Fig. 7, the first embodiment of the email access control scheme according to the present invention will be described in detail.

[0144] The email access control scheme of the present invention enables bidirectional communications between a sender and a recipient appropriately while maintaining anonymity of a sender and a recipient on a communication network. Basically, this is realized by disclosing only information indicative of characteristics of recipients in a state of concealing true identifiers of the recipients, and assigning limited access rights with respect to those who wish to carry out communications while maintaining the anonymity according to the disclosed information.

[0145] More specifically, an Anonymous Identification (abbreviated hereafter as AID) that functions as a role identifier in which a personal information is concealed is assigned to a user, and this AID is disclosed on the network in combination with an information indicative of characteristics of the user such as his/her interests, age, job, etc., which cannot be used in identifying the user on the network but which can be useful for a sender in judging whether or not it is worth communicating with that user.

[0146] Also, the sender can search out a recipient with whom he/she wishes to communicate by reading or searching through the disclosed information. Namely, in the case where the sender wishes to communicate with a recipient while maintaining his/her own anonymity, the sender specifies the AID of that recipient and acquires a Personalized Access Ticket (abbreviated hereafter as PAT). The PAT contains the AIDs of the sender and the recipient as well as information regarding a transfer control flag and a validity period. The transfer control flag is used in order to determine whether a Secure Communication Service (abbreviated hereafter as SCS) to be described below carries out the authentication with respect to the sender. Namely, when the transfer control flag is set ON, the SCS will carry out the authentication such as signature verification for example, with respect to the sender at a time of the connection request. On the other hand, when the transfer control flag is set OFF, the SCS will give the connection request to a physical communication network to which the SCS is connected, without carrying out the authentication. In other words, the transfer control is used in order to verify whether or not the AID is properly utilized by the user to whom it is allocated by a Certification Authority (abbreviated hereafter as CA).

[0147] In the communication network realizing the email access control scheme of the present invention, the assignment of AIDs with respect to users, the maintenance of information disclosed in combination with AIDs, the issuance of PATs, and the email access control based on PATs are realized by separate organizations. This is because it is more convenient to realize them by separate organizations from a perspective of maintaining the security of the entire network, since security levels to be maintained in relation to respective actions are different. Note however that the maintenance of the disclosed information and the issuance of PATs may be realized by the same organization.

[0148] Fig. 1 shows an overall configuration of a communication system in this first embodiment, which is directed to the email service on Internet or Intranet.

[0149] In Fig. 1, the CA (Certification Authority) 1 has a right to authenticate an Official Identification (abbreviated hereafter as OID) that identifies each individual and a right to issue AIDs, and functions to generate AIDs from OIDs and allocate AIDs to users 3.

[0150] The SCS (Secure Communication Service) 5 judges whether or not to admit a connection in response to a connection request by an email from a user 3, according to the PAT (Personalized Access Ticket) presented from a user 3. The SCS 5 also rejects a connection request by an email according to a request from a user 3. The SCS 5 also judges the identity of OIDs according to a request from a user 3.

[0151] An Anonymous Directory Service (abbreviated hereafter as ADS) 7 is a database for managing the AID, the transfer control flag value, the validity period value, and the disclosed information (such as interests, which can be regarded as requiring a lower secrecy compared with a personal information such as name, telephone number, and real email address) of each user 3. The ADS 7 has a function to generate the PAT from the AID of a user 3 who presented search conditions, the AID of a user 3 who has been registering the disclosed information that matches the search conditions in the ADS 7, the transfer control flag value given from a user 3 or administrators of the ADS, and the validity period value given from a user 3 or administrators of the ADS, and then allocate the PAT to a user 3 who presented the search conditions.

[0152] First, a series of processing from generating the AID from the OID according to a request from a user until allocating the AID to that user will be described.

[0153] Fig. 2 shows exemplary formats of the OID, the AID, and the PAT. As shown in a part (a) of Fig. 2, the OID is an information comprising an arbitrary character string according to a rule by which the CA 1 can uniquely identify the user and a public key, which is signed by the CA 1 using a secret key of the CA 1.

[0154] Also, as shown in a part (b) of Fig. 2, the AID is an information comprising fragments of the OID and their position information, redundant character strings, and an SCS information given by an arbitrary character string (host name, real domain name, etc.) by which a host or a domain that is operating the SCS 5 can be uniquely identified on the network, which is signed by the CA 1 using the secret key of the CA 1.

[0155] Also, as shown in a part (c) of Fig. 2, the PAT is an information comprising the transfer control flag, AID_g, AID_1, and the validity period, which is signed by the ADS 7 using a secret key of the ADS 7. Here, the transfer control flag value is defined to take either 0 or 1. Also, the validity period is defined by any one or combination of the number of times for which the PAT is available, the absolute time (UTC) by which the PAT becomes unavailable, the absolute time (UTC) by which the PAT becomes available, and the relative time (lifetime) since the PAT becomes available until it becomes unavailable.

[0156] Note that, as will be explained in the subsequent embodiments described below, in addition to the 1-to-1 PAT which sets one sender and one recipient in correspondence as described above, the present invention can also use a 1-to-N PAT which sets one sender and N recipients, as well as a link specifying PAT which specifies the AID by a link information that is capable of specifying the AID instead of specifying the AID itself in the PAT. The link specifying PAT can be either a link specifying 1-to-1 PAT or a link specifying 1-to-N PAT depending on the correspondence relationship between the sender and the recipients as described above. Namely, the PAT of the present invention can be given in four types: 1-to-1 PAT, 1-to-N PAT, link specifying 1-to-1 PAT, and link specifying 1-to-N PAT.

[0157] Next, a procedure by which the user 3 requests the AID to the CA 1 will be described. The user 3 generates a pair of a secret key and a public key. Then, the user 3 and the CA 1 carries out the bidirectional authentication using the OID of the user 3 and the certificate of the CA 1, and the user 3 transmits the public key to the CA 1 by arbitrary means. Here, there can be cases where communications between the user 3 and the CA 1 are to be encrypted.

[0158] Next, a procedure by which the CA 1 issues the AID to the user 3 in response to a request for the AID as described above will be described. Upon receiving the public key from the user 3, the CA 1 generates the AID. Then, the CA 1 transmits the AID to the user 3 by arbitrary means. Upon receiving the AID from the CA 1, the user 3 stores the received AID into its storage device. Here, there can be cases where communications between the user 3 and the CA 1 are to be encrypted.

[0159] Next, the AID generation processing at the CA will be described with reference to Fig. 3.

[0160] In the procedure of Fig. 3, the CA 1 generates an information of a length equal to the total length L of the OID, and sets this information as a tentative AID (step S911). Then, in order to carry out the partial copying of the OID, values of parameters p_i and ℓ_i for specifying a copying region are determined using arbitrary means such as random number generation respectively (step S913). Here, L is equal to the total length L of the OID, and ℓ_i is an arbitrarily defined value within a range in which a relationship of $0 \leq \ell_i \leq L$ holds. Then, an information in a range between a position p_i to a position $p_i + \ell_i$ from the top of the OID is copied to the same positions in the tentative AID (step S915). In other words, this OID fragment will be copies to a range between a position p_i and a position $p_i + \ell_i$ from the top of the tentative AID. Then, the values of p_i and ℓ_i are written into a prescribed range in the tentative AID into which the OID has been partially copied, in a form encrypted by an arbitrary means (step S917). Then, an SCS information given by an arbitrary character string (host name, real domain, etc.) that can uniquely identify a host or a domain that is operating the SCS 5 on the network is written into a prescribed range in the tentative AID into which these values are written (step S919). Then, the tentative AID into which the above character string is written is signed using a secret key of the CA 1 (step S921).

[0161] Next, a procedure for registering the AID of a user-B 3 and the disclosed information into the ADS 7 will be described. First, the bidirectional authentication by arbitrary means using the AID of the user-B 3 and the certificate of the ADS 7 is carried out between the user-B 3 who is a registrant and the ADS 7. Then, the user-B 3 transmits the transfer control flag value, the validity period value, and the disclosed information such as interests to the ADS 7. Then, the ADS 7 stores the transfer control flag value, the validity period value, and the entire disclosed information in relation to the AID of the user-B 3 in its storage device. Here, there can be cases where communications between the user-B 3 who is the registrant and the ADS 7 are to be encrypted.

[0162] Next, a procedure by which a user-A 3 searches through the disclosed information that is registered in the ADS 7 will be described. First, the bidirectional authentication by arbitrary means using the AID of the user-A 3 and the certificate of the ADS 7 is carried out between the user-A 3 who is a searcher and the

ADS 7. Then, the user-A 3 transmits arbitrary search conditions to the ADS 7. Then, the ADS 7 presents all the received search conditions to its storage device, and extracts the AID of a registrant which satisfies these search conditions. Then, the ADS 7 generates the PAT from the AID of the user-A 3, the AID of the registrant who satisfied all the search conditions, the transfer control flag value, and the validity period value. Then, the ADS 7 transmits the generated PAT to the user-A 3. Here, there can be cases where communications between the user-A 3 who is a searcher and the ADS 7 are to be encrypted. Note that the 1-to-1 PAT is generated as a search result of the ADS 7.

[0163] Next, the 1-to-1 PAT generation processing at the ADS 7 will be described with reference to Fig. 4.

[0164] First, an information of a prescribed length is generated, and this information is set as a tentative PAT (step S1210). Then, the AID of the user-A 3 who is a searcher and the AID of the user-B 3 who is a registrant are copied into a prescribed region of the tentative PAT (step S1215). Then, the transfer control flag value and the validity period value are written into respective prescribed regions of the tentative PAT into which the AIDs are copied (step S1217). Then, the tentative PAT into which these values are written is signed using a secret key of the ADS 7 (step S1219).

[0165] Next, the transfer control using the 1-to-1 PAT will be described. The transfer control is a function for limiting accesses to a user who has a proper access right from a third person to whom the PAT has been transferred or who has eavesdropped the PAT (a user who originally does not have the access right).

[0166] The ADS 7 and the user-B 3 of the registrant AID can prohibit a connection to the user-B 3 from a third person who does not have the access right, by setting a certain value in to the transfer control flag of the PAT.

[0167] When the transfer control flag value is set to be 1, the sender's AID is authenticated between the SCS 5 and the sender according to an arbitrary challenge/response process, so that even if the sender gives both the sender's AID and the PAT to another user other than the sender, that another user will not be able to make a connection to the registrant of the ADS 7 through the SCS 5.

[0168] On the other hand, when the transfer control flag value is set to be 0, no challenge/response process will be carried out between the SCS 5 and the sender, so that if the sender gives both the sender's AID and the PAT to another user other than the sender, that another user will also be able to make a connection to the registrant of the ADS 7 through the SCS 5.

[0169] Next, the email access control method at the SCS 5 will be described with reference to Fig. 5.

[0170] The sender specifies "[sender's AID]@[real domain of SCS of sender]" in From: line, and "[PAT]@[real domain of SCS of sender]" in To: line.

[0171] The SCS 5 acquires a mail received by an MTA (Message Transfer Agent) such as SMTP (Simple Mail Transfer Protocol), and executes the processing of Fig. 5 as follows.

(1) The signature of the PAT is verified using a public key of the ADS 7 (step S1413).

When the PAT is found to have been altered (step S1415 YES), the mail is discarded and the processing is terminated (step S1416).

When the PAT is found to have been not altered (step S1415 NO), the following processing (2) is executed.

(2) The search is carried out by presenting the sender's AID to the PAT (steps S1417, S1419, S1421).

When an AID that completely matches with the sender's AID is not contained in the PAT (step S1423 NO), the mail is discarded and the processing is terminated (step S1416).

When an AID that completely matches with the sender's AID is contained in the PAT (step S1423 YES), the following processing (3) is executed.

(3) The validity period value of the PAT is evaluated (steps S1425, S1427).

When the PAT is outside the validity period (step S1427 NO), the mail is discarded and the processing is terminated (step S1416).

When the PAT is within the validity period (step S1427 YES), the following processing (4) is executed.

(4) Whether or not to authenticate the sender is determined by referring to the transfer control flag value of the PAT (steps S1431, S1433).

When the value is 1 (step S1433 YES), the challenge/response authentication between the SCS 5 and the sender is carried out, and the signature of the sender is verified (step S1435). When the signature is valid, the recipient is specified and the PAT is attached (step S1437). When the signature is invalid, the mail is discarded and the processing is terminated (step S1416).

When the value is 0 (step S1433 NO), the recipient is specified and the PAT is attached without executing the challenge/response authentication (step S1437).

[0172] Next, an exemplary challenge/response authentication between the SCS 5 and the sender will be described.

[0173] First, the SCS 5 generates an arbitrary information such as a timestamp, for example, and transmits the generated information to the sender.

[0174] Then, the sender signs the received information using a secret key of the sender's AID and transmits it along with a public key of the sender's AID.

[0175] The SCS 5 then verifies the signature of the received information using the public key of the sender's AID. When the signature is valid, the recipient is specified and the PAT is attached. When the signature is invalid, the mail is discarded and the processing is terminated.

[0176] Next, a method for specifying the recipient at the SCS 5 will be described. First, the SCS 5 carries out the search by presenting the sender's AID to the PAT, so as to acquire all the AIDs which do not completely match the sender's AID. All these acquired AIDs will be defined as recipient's AIDs hereafter. Then, for every recipient's AID, the real domain of SCS of recipient is taken out from the recipient's AID. Then, the recipient is specified in a format of "[recipient's AID]@[real domain of SCS of recipient]". Finally, the SCS 5 changes the sender from a format of "[sender's AID]@[real domain of SCS of sender]" to a format of "sender's AID".

[0177] Next, a method for attaching the PAT at the SCS 5 will be described. The SCS 5 attaches the PAT to an arbitrary position in the mail. The SCS 5 gives the mail to the MTA after specifying the sender and the recipient and attaching the PAT.

[0178] Note that all the processings described above are the same in the case of the 1-to-N PAT.

[0179] Next, a method of receiving refusal with respect to the PAT at the SCS 5 will be described.

[0180] Receiving refusal setting: The bidirectional authentication is carried out by an arbitrary means between the user and the SCS 5. Then, the user transmits a registration command, his/her own AID, and arbitrary PATs to the SCS 5. Then, the SCS 5 verifies the signature of the received AID. If the signature is invalid, the processing of the SCS 5 is terminated. If the signature is valid, the SCS 5 next verifies the signature of each received PAT using a public key of the ADS. Those PATs with the invalid signature are discarded by the SCS 5. When the signature is valid, the SCS 5 carries out the search by presenting the received AID to each PAT. For each of those PATs which contain the AID that completely matches with the received AID, the SCS 5 presents the registration command and the PAT to the storage device such that the PAT is registered into the storage device. Those PATs which do not contain the AID that completely matches with the received AID are discarded by the SCS 5 without storing them into the storage device. Here, there can be cases where communications between the user and the SCS 5 are to be encrypted.

[0181] Receiving refusal execution: The SCS 5 carries out the search by presenting the PAT to the storage device. When a PAT that completely matches the presented PAT is registered in the storage device, the mail is discarded. When a PAT that completely matches the present PAT is not registered in the storage device, the mail is not discarded.

[0182] Receiving refusal cancellation: The bidirectional authentication is carried out by an arbitrary means between the user and the SCS 5. Then, the user presents his/her own AID to the SCS 5. Then, the SCS 5 verifies the signature of the received AID. If the signature is invalid, the processing of the SCS 5 is terminated. If the signature is valid, the SCS 5 next presents the presented AID as a search condition to the storage device and acquire all the PATs that contain the presented AID, and then presents all the acquired PATs to the user. Then, the user selects all the PATs for which the receiving refusal is to be cancelled by referring to all the PATs presented from the SCS 5, and transmits all the selected PATs along with a deletion command to the SCS 5. Upon receiving the deletion command and all the PATs for which the receiving refusal is to be cancelled, the SCS 5 presents the deletion command and all the PATs received from the user to the storage device, such that all the received PATs are deleted from the storage device.

[0183] Note that the method of receiving refusal with respect to the 1-to-N PAT at the SCS 5 is the same as the method of receiving refusal with respect to the 1-to-1 PAT described above.

[0184] Note also the the case of returning of a mail from the user-B to the user-A is the same as in the case of transmitting a mail from the user-A to the user-B.

[0185] Next, the judgement of identity will be described with reference to Fig. 6 and Fig. 7.

(1) An initial value of a variable OID_M is defined as a bit sequence with a length equal to the total length L of the OID and all values equal to "0". Also, an initial value of a variable OID_V is defined as a bit sequence with a length equal to the total length of the OID and all values equal to "0" (step S2511).

(2) One AID is selected from a set of processing target AIDs, and the following bit processing is carried out (step S2513).

(a) Values of variables AID_M and AID_V are determined according to the position information contained in the AID (step S2515). Here, AID_M is defined as a bit sequence with a length equal to the total length L of the OID and a value of a position at which the OID information is defined is "1" while a value of a position at which the OID information is not defined is "0" (see Fig. 7). Also, AID_V is defined as a bit sequence with a length equal to the total length L of the OID and a value of a position at which the OID information is defined is an actual value of the OID information while a value of a position at which the OID information is not defined is 0 (see Fig. 7).

(b) AND processing of OID_M and AID_M is carried out and its result is substituted into a variable OVR_M (step S2517).

(c) AND processing of OVR_M and AID_M as well as AND processing of OVR_M and OID_M are carried out and their results are compared (step S2519). When they coincide, OR processing of OID_M and AID_M is carried out and its result is substituted into OID_M (step S2521), while OR processing of OID_V and AID_V is also carried out and its result is substituted into OID_M (step S2523). On the other hand, when they do not coincide, the processing proceeds to the step S2525.

(d) An AID to be processed next is selected from a set of processing target AIDs. When at least one another AID is contained in the set, the steps S2513 to S2523 are executed for that another AID. When no other AID is contained in the set, the processing proceeds to the step S2527.

(e) Values of OID_M and OID_V are outputted (step S2527).

[0186] The value of OID_M that is eventually obtained indicates all positions of the OID information that can be recovered from the set of processing target AIDs. Also, the value of OID_V that is eventually obtained indicates all the OID information that can be recovered from the set of processing target AID. In other words, by using the values of OID_M and OID_V, it is possible to obtain the OID albeit probabilistically when the value of OID_V is used as a search condition, and it is possible to quantitatively evaluate a precision of the above search by a ratio OID_M/L with respect to the total length L of the OID.

[0187] As described above, in this first embodiment, the CA 1 which is a Trusted Third Party with high secrecy and credibility generates the AID in which the personal information is concealed, from the OID that contains the highly secret personal information such as name, telephone number, real email address, etc., according to a user request, and issues the AID to the user. By identifying the user by this AID on the communication network as well as in various services provided on the communication network, it becomes possible to provide both the anonymity guarantee and the identity guarantee for the user. In other words, it becomes possible for the user to communicate with another user without revealing the own real name, telephone number, email address, etc., to that another user, and it also becomes possible to disclose the disclosed information to unspecified many through the ADS 7 as will be described below.

[0188] The user registers the disclosed information, that is an information which is supposed to have a low secrecy compared with the personal information at the ADS 7. In the case of searching the disclosed information and the registrant AID, the searcher presents the AID of the searcher and arbitrary search conditions to the ADS 7. The ADS 7 then extracts the registrant AID that satisfies these search conditions, and generates the PAT from the AID of the searcher and the AID of the registrant who satisfied the search conditions, the transfer control flag value, and the validity period value.

[0189] In this 1-to-1 PAT, the transfer control flag value and the validity period value are set as shown a part (c) of Fig. 2, and by setting up this validity period in advance, it is possible to limit connections from the sender.

[0190] It is also possible to prohibit connections from a third person who does not have the access right, by using the transfer control flag value. Namely, when the transfer control flag value is set to be 1, the sender's AID is authenticated between the SCS 5 and the sender according to an arbitrary challenge/response process, so that even if the sender gives both the sender's AID and the PAT to another user other than the sender, that another user will not be able to make a connection to the registrant of the ADS 7 through the SCS 5. On the other hand, when the transfer control flag value is set to be 0, no challenge/response process will be carried out between the SCS 5 and the sender, so that if the sender gives both the sender's AID and the PAT to another user other than the sender, that another user will also be able to make a connection to the registrant of the ADS 7 through the SCS 5.

[0191] It is also possible to make a connection request to the communication network such that a call for which the recipient is specified by the 1-to-1 PAT will be received by the recipient's AID or the sender's AID defined within the PAT. In addition, it is also possible to refuse receiving calls with the 1-to-1 PAT selected by the recipient among calls which are specified by the 1-to-1 PAT. It is also possible to cancel the receiving refusal of the calls with the 1-to-1 PAT selected by the recipient. In addition, as a measure against the sender who repeats the personal attach using a plurality of sender's AIDs by taking an advantage of the anonymity, it is possible to judge the identity of the OID from these plurality of sender's AIDs and it is possible to extract that OID at some probability.

[0192] Next, with references to Fig. 8 to Fig. 24, the second embodiment of the email access control scheme according to the present invention will be described in detail.

[0193] In contrast to the first embodiment described above which is directed to the case where a sender and a recipient are set in 1-to-1 correspondence, this second embodiment is directed to the case where a sender and recipients are set in 1-to-N correspondence and a generation of a new PAT and a content change of the existing PAT can be made by the initiative of a user. Here, the sender is either a holder of the PAT or a member of the PAT. Similarly, the recipient is either a holder of the PAT or a member of the PAT.

[0194] In general, a membership of a group communication (mailing list, etc.) is changing dynamically so that it is necessary for a host of the group communication to manage information on a point of contact such as telephone number, email address, etc., of each member. In contrast, in the case where it is only possible to newly generate a 1-to-1 PAT as in the first embodiment, the management of a point of contact is difficult. For example, it is difficult to manage the group collectively, and even if it is given to the others for the purpose of the transfer control, it does not function as an address of the group communication such as mailing list.

[0195] In this second embodiment, in order to resolve such a problem, it is made possible to carry out a generation of a new 1-to-N PAT and a content change or the existing 1-to-N PAT by the initiative of a user.

[0196] First, the definition of various identifications used in this second embodiment will be described with references to Fig. 8 and Fig. 9.

[0197] As shown in a part (a) of Fig. 8, the OID is an information comprising an arbitrary character string (telephone number, email address, etc.) according to a rule by which the CA 1 can uniquely identify the user and a public key, which is signed by the CA 1.

[0198] Also, as shown in a part (b) of Fig. 8, the AID is an information comprising fragments of the OID and their position information, redundant character strings, and an SCS information given by an arbitrary character string (host name, real domain name, etc.) by which a host or a domain that is operating the SCS 5 can be uniquely identified on the network, which is signed by the CA 1.

[0199] Also, as shown in a part (c) of Fig. 8, the 1-to-N PAT is an information comprising two or more AIDs, a holder index, the validity period, the transfer control flag, and a PAT processing device identifier, which is signed using a secret key of the PAT processing device.

[0200] Here, one of the AIDs is a holder AID of this PAT, where the change of the information contained in the PAT such as an addition of AID to the PAT, a deletion of AID from the PAT, a change of the validity period in the PAT, a change of the transfer control flag value in the PAT, etc., can be made by presenting the holder AID and a corresponding Enabler to the PAT processing device.

[0201] On the other hand, the AIDs other than the holder AID that are contained in the PAT are all member AIDs, where a change of the information contained in the PAT cannot be made even when the member AID and a corresponding Enabler are presented to the PAT processing device.

[0202] The holder index is a numerical data for identifying the holder AID, which is defined to take a value 1 when the holder AID is a top AID in the AID list formed from the holder AID and the member AIDs, a value 2 when the holder AID is a second AID from the top of the AID list, or a value n when the holder AID is an n-th AID from the top of the AID list.

[0203] The transfer control flag value is defined to take either 0 or 1 similarly as in the case of the 1-to-1 PAT.

[0204] The holder AID is defined to be an AID which is written at a position of the holder index value in the AID list. The member AIDs are defined to be all the AIDs other than the holder AID.

[0205] The validity period is defined by any one or combination of the number of times for which the PAT is available, the absolute time (UTC) by which the PAT becomes unavailable, the absolute time (UTC) by which the PAT becomes available, and the relative time (lifetime) since the PAT becomes available until it becomes unavailable.

[0206] The identifier of a PAT processing device (or a PAT processing object on the network) is defined as a serial number of the PAT processing device (or an distinguished name of the PAT processing object on the network). The secret key of the PAT processing device (or the PAT processing object on the network) is defined to be uniquely corresponding to the identifier.

[0207] Also, in this second embodiment, an Enabler is introduced as an identifier corresponding to the AID. As shown in Fig. 9, the Enabler is an information comprising a character string uniquely indicating that it is an Enabler and an AID itself, which is signed by the CA 1.

[0208] Next, the operations for a generation of a new PAT and a content change of the existing PAT will be described. Here, the following operations are defined at a secure PAT processing device on the communication terminal or a PAT processing object on the CA or on a network which is properly requested from the CA (which will also be referred to as a PAT processing device hereafter).

1. Editing of AID list:

   A list of AIDs (referred hereafter as an AID list) contained in the PAT is edited using AIDs and Enabler. Else, the AID list is newly generated.

2. Setting of the validity period and the transfer control flag:

   The validity period value and the transfer control flag value contained in the PAT are changed using an AID and Enabler. Also, a new validity period value and a new transfer control flag value are set in the newly generated AID list.

[0209] A user who presented the holder AID and the Enabler corresponding to this holder AID to the PAT processing device can edit the list of AIDs contained in the PAT. In this case, the following processing rules are used.

(1) Generating a new PAT (MakePAT) (see Fig. 10):

The AID list (ALIST<holder AID | member AID_1, member AID_2, ••••••••, member AID_n>) is newly generated, and the validity period value and the transfer control flag value are set with respect to the generated ALIST.

AID_A + AID_B + Enabler of AID_B + Enabler of AID_A

→ ALIST<AID_A | AID_B>

ALIST<AID_A | AID_B > + Enabler of AID_A

+ validity period value

+ transfer control flag value

→ PAT<AID$_A$ | AID$_B$ >

(2) Merging PATs (MergePAT) (see Fig. 11):

A plurality of ALISTs of the same holder AID are merged and the validity period value and the transfer control flag value are set with respect to the merged ALIST.

ALIST<AID$_A$ | AID$_{B1}$, AID$_{B2}$, • • • • • • • • >

+ ALIST<AID$_A$ | AID$_{C1}$, AID$_{C2}$, • • • • • • • • >

+ Enabler of AID$_A$

→ ALIST<AID$_A$ | AID$_{B1}$, AID$_{B2}$, • • • • • • • •, AID$_{C1}$, AID$_{C2}$, • • • • • • • • >

ALIST<AID$_A$ | AID$_{B1}$, AID$_{B2}$, • • • • • • • •, AID$_{C1}$, AID$_{C2}$, • • • • • • • • >

+ Enabler of AID$_A$ + validity period value

+ transfer control flag value

→ PAT<AID$_A$ | AID$_{B1}$, AID$_{B2}$, • • • • • • • •, AID$_{C1}$, AID$_{C2}$, • • • • • • • • >

(3) Splitting a PAT (SplitPAT) (see Fig. 12):

The ALIST is split into a plurality of ALISTs of the same holder AID, and the respective validity period value and transfer control flag value are set with respect to each one of the split ALISTs.

ALIST<AID$_A$ | AID$_{B1}$, AID$_{B2}$, • • • • • • • •, AID$_{C1}$, AID$_{C2}$, • • • • • • • • >

+ Enabler of AID$_A$

→ ALIST<AID$_A$ | AID$_{B1}$, AID$_{B2}$, • • • • • • • • >

+ ALIST<AID$_A$ | AID$_{C1}$, AID$_{C2}$, • • • • • • • • >

ALIST<AID$_A$ | AID$_{C1}$, AID$_{C2}$, • • • • • • • • >

+ Enabler of AID$_A$ + validity period value

+ transfer control flag value

→ PAT<AID$_A$ | AID$_{C1}$, AID$_{C2}$, • • • • • • • • >

(4) Changing a holder of a PAT (TransPAT) (see Fig. 13):

The holder AID of the ALIST is changed, and the validity period value and the transfer control flag value are set with respect to the changed ALIST.

ALIST<AID$_A$ | AID$_B$ > + ALIST<AID$_A$ | AID$_{C1}$, AID$_{C2}$, • • • • • • • • >

+ Enabler of AID$_A$ + Enabler of AID$_B$

→ ALIST<AID$_B$ | AID$_{C1}$, AID$_{C2}$, • • • • • • • • >

ALIST<AID$_B$ | AID$_{C1}$, AID$_{C2}$, • • • • • • • • >

+ Enabler of AID$_B$ + validity period value

+ transfer control flag value

→ PAT<AID$_B$ | AID$_{C1}$, AID$_{C2}$, • • • • • • • • >

[0210] In the operation for setting the validity period value, in order to permit the setting of the validity period value only to a user who holds both the holder AID and the corresponding Enabler, the following operation is defined.

PAT<AID$_A$ | AID$_B$ > + Enabler of AID$_A$

+ validity period value

→ PAT<AID$_A$ | AID$_B$ >

[0211] In the operation for setting the transfer control flag value, in order to permit the setting of the transfer control flag value only to a user who holds both the holder AID and the corresponding Enabler, the following operation is defined.

PAT<AID$_A$ | AID$_B$ > + Enabler of AID$_A$

+ transfer control flag value

→ PAT<AID$_A$ | AID$_B$ >

[0212] Next, with references to Fig. 14 to Fig. 20, the overall system configuration of this second embodiment will be described. In Fig. 14 to Fig. 20, the user-A who has AID$_A$ allocated from the CA stores AID$_A$ and Enabler of AID$_A$ in a computer of the user-A, and the input/output devices such as floppy disk drive, CD-ROM drive, communication board, microphone, speaker, etc., are connected. Else, AID$_A$ and Enabler of AID$_A$ are stored in a communication terminal (telephone, cellular phone, etc.) which has a storage device and a data input/output function.

[0213] Similarly, the user-B who has AID$_B$ allocated from the CA stores AID$_B$ and Enabler of AID$_B$ in a computer of the user-B, and the input/output devices such as floppy disk drive, CD-ROM drive, communication board, microphone, speaker, etc., are connected. Else, AID$_B$ and Enabler of AID$_B$ are stored in a communication terminal (telephone, cellular phone, etc.) which has a storage device and a data input/output function.

[0214] In the following, a procedure by which the user-A generates PAT<AID_A | AID_B> will be described.

(1) The user-A acquires AID_B and Enabler of AID_B using any of the following means.

- AID_B and Enabler of AID_B are registered at the ADS 7, and it is waited until the user-A acquires them as a search result (Fig. 14).
- AID_B and Enabler of AID_B are directly transmitted to the user-A by the email, signaling, etc. (Figs. 15, 16).
- AID_B and Enabler of AID_B are stored in a magnetic, optic, or electronic medium such as floppy disk, CD-ROM, MO, IC card, etc., and this medium is given to the user-A. Else, it is waited until the user acquires them by reading this medium (Figs. 17, 18).
- AID_B and Enabler of AID_B are printed on a paper medium such as book, name card, etc., and this medium is given to the user-A. Else, it is waited until the user-A acquire them by reading this medium (Figs. 19, 20).

(2) The user-A who has acquired AID_B and Enabler of AID_B by any of the means described in the above (1) issues the MakePAT command to the PAT processing device. This procedure is common to Fig. 14 to Fig. 20, and defined as follows.

(a) The user-A requests the issuance of the MakePAT command by setting AID_A, Enabler of AID_A, AID_B, Enabler of AID_B, the validity period value, and the transfer control flag value into the communication terminal of the user-A.
(b) The communication terminal of the user-A generates the MakePAT command.
(c) The communication terminal of the user-A transmits the generated MakePAT command to the PAT processing device by means such as the email, signaling, etc. (the issuance of the MakePAT command).
(d) The PAT processing device generates PAT<AID_A | AID_B> by processing the received MakePAT command according to Fig. 21 and Fig. 23. More specifically, this is done as follows.

AID_A + AID_B + Enabler of AID_B + Enabler of AID_A

→ ALIST<AID_A | AID_B>

ALIST<AID_A | AID_B> + Enabler of AID_A

+ validity period value + transfer control flag value

→ PAT<AID_A | AID_B>

(e) The PAT processing device transmits the generated PAT<AID_A | AID_B> to the communication terminal of the user-A, or to the communication terminal of the user-B according to the need, by means such as the email, signaling, etc.
(f) The communication terminal of the user-A (or the user-B) stores the received PAT<AID_A | AID_B> in the storage device of the communication terminal of the user-A.

[0215] The merging of PATs (MergePAT, Fig. 21, Fig. 23), the splitting of a PAT (SplitPAT, Fig. 22, Fig. 23), and the changing of a holder of a PAT (TransPAT, Fig. 21, Fig. 23) are also carried out by the similar procedure.

[0216] Next, the procedure of MakePAT, MergePAT and TransPAT will be described with reference to Fig. 21.

(1) The holder AID is specified (step S4411).
(2) All the member AIDs are specified (step S4412).
(3) The AID list is generated from the specified holder AID and all the specified member AIDs (step S4413). More specifically, the specified holder AID and all the specified member AIDs are concatenated using arbitrary means.
(4) A tentative PAT is generated using arbitrary means, similarly as in the case of a tentative AID (step S4414).
(5) The generated AID list is copied to a prescribed region of the generated tentative PAT (step S4415).
(6) The holder index value is written into the tentative pat to which the AID list has been copied (step S4416).
(7) The transfer control flag value is written into the tentative PAT into which the holder index value has been written (step S4417).
(8) The validity period value is written into the tentative PAT into which the transfer control flag value has been written (step S4418).
(9) The PAT processing device identifier is written into the tentative PAT into which the validity period value has been written (step S4419).
(10) The tentative PAT into which the PAT processing device identifier has been written is signed using the secret key of the PAT processing device (step S4420).

[0217] Next, the procedure of SplitPAT will be described with reference to Fig. 22.

(1) The holder AID is specified (step S4511).
(2) All the AIDs to be the member AIDs of the PATs after the splitting are specified (step S4512).
(3) The AID list is generated from the specified holder AID and all the specified member AIDs (step

S4513). More specifically, the specified holder AID and all the specified member AIDs are concatenated using arbitrary means.

(4) A tentative PAT is generated using arbitrary means, similarly as in the case of a tentative AID (step S4514).

(5) The generated AID list is copied to a prescribed region of the generated tentative PAT (step S4515).

(6) The holder index value is written into the tentative pat to which the AID list has been copied (step S4516).

(7) The transfer control flag value is written into the tentative PAT into which the holder index value has been written (step S4517).

(8) The validity period value is written into the tentative PAT into which the transfer control flag value has been written (step S4518).

(9) The PAT processing device identifier is written into the tentative PAT into which the validity period value has been written (step S4519).

(10) The tentative PAT into which the PAT processing device identifier has been written is signed using the secret key of the PAT processing device (step S4520).

(11) In the case of continuing the splitting (step S4521 YES), the procedure returns to (2), and repeats (2) to (10) sequentially.

[0218] Note that, in the procedures of Fig. 21 and Fig. 22, the AID list generation is carried out according to Fig. 23 as follows. Namely, a buffer length is determined first (step S4611) and a buffer is generated (step S4612). Then, the holder AID is copied to a vacant region of the generated buffer (step S4613). Then, the member AID is copied to a vacant region of the resulting buffer (step S4614), and if the next member AID exists (step S4615 YES), the step S4614 is repeated.

[0219] Next, the determination of the holder AID will be described. Each of the MakePAT, the MergePAT, the SplitPAT, and the TransPAT commands is defined to have two or more arguments, where AID, PAT, or Enabler can be specified as an argument. In this case, the PAT processing device specifies the holder AID of the PAT to be outputted after executing each command according to the following rules.

- Case of the MakePAT:

  For the MakePAT command, it is defined that AIDs are to be specified for the first argument to the N-th argument (N = 2, 3, $\cdots\cdots\cdots$) and Enablers are to be specified for the N+1-th and subsequent arguments. For example, they can be specified as follows.

  MakePAT AID_1, AID_2, $\cdots\cdots\cdots$, AID_N, Enabler of AID_1, Enabler of AID_2, Enabler of AID_N

  The PAT processing device interprets the AID of the first argument of the MakePAT command as the holder AID.

  Only when one of the Enablers of the N+1-th and subsequent arguments corresponds to the AID of the first argument, the PAT processing device specifies this AID (that is the AID of the first argument) as the holder AID of the PAT to be outputted after executing the MakePAT command.

- Case of the MergePAT:

  For the MergePAT command, it is defined that PATs are to be specified for the first argument to the N-th argument (N = 2, 3, $\cdots\cdots\cdots$) and Enabler is to be specified for the N+1-th argument. Namely, they can be specified as follows.

  MergePAT PAT_1 PAT_2 $\cdots\cdots\cdots$ PAT_N Enabler of AID

  The PAT processing device interprets the holder AID of the PAT of the first argument of the MergePAT command as the holder AID of the PAT to be outputted after executing the MergePAT command.

  Only when the Enabler of the N+1-th argument corresponds to the holder AID of the PAT of the first argument, the PAT processing device specifies this AID (that is the holder AID of the PAT of the first argument) as the holder AID of the PAT to be outputted after executing the MergePAT command.

- Case of the SplitPAT:

  For the SplitPAT command, it is defined that PAT is to be specified for the first argument, a set of one or more AIDs grouped together by some prescribed symbols (assumed to be parentheses () in this example) are to be specified for the second argument to the N-th argument (N = 3, 4, $\cdots\cdots\cdots$), and Enabler is to be specified for the N+1-th argument. Namely, they can be specified as follows.

  SplitPAT PAT_1 (AID_{11}) (AID_{21} AID_{22}) $\cdots\cdots\cdots$ (AID_{N1} AID_{N2} $\cdots\cdots\cdots$ AID_{NM}) Enabler of AID

  The PAT processing device interprets the holder AID of the PAT of the first argument of the SplitPAT command as the holder AID of the PAT to be outputted after executing the SplitPAT command.

  Only when the Enabler of the N+1-th argument corresponds to the holder AID of the PAT of the first argument, the PAT processing device specifies this AID (that is the holder AID of the PAT of the first argument) as the holder AID of the PAT to be outputted after executing the SplitPAT command.

- Case of the TransPAT:

  For the TransPAT command, it is defined that

PATs are to be specified for the first argument and the second argument, AID is to be specified for the third argument, and Enablers are to be specified for the fourth argument and the fifth argument. Namely, they can be specified as follows.

TransPAT PAT_1 PAT_2 AID Enabler of AID_1 Enabler of AID_2

The PAT processing device interprets the AID of the third argument as the holder AID of the PAT to be outputted after executing the TransPAT command provided that the AID of the third argument of the TransPAT command is contained in the PAT of the second argument.

Only when the Enabler of the fourth argument corresponds to both the PAT of the first argument and the PAT of the second argument and the Enabler of the fifth argument corresponds to the AID of the third argument, the PAT processing device specifies the AID of the third argument as the holder AID of the PAT to be outputted after executing the TransPAT command.

Next, the determination of the member AIDs will be described. The definitions of the MakePAT, the MergePAT, the SplitPAT, and the TransPAT commands are as described above. The PAT processing device specifies the member AIDs of the PAT to be outputted after executing each command according to the following rules.

- Case of the MakePAT:

  Only when the holder AID of the PAT to be outputted after executing the MakePAT command is formally determined, the PAT processing device interprets all the AIDs of the second and subsequent arguments of the MakePAT command as the member AIDs of the PAT to be outputted after executing the MakePAT command.

  The PAT processing device specifies only those AIDs among all the AIDs of the second and subsequent arguments which correspond to the Enablers specified by the N+1-th and subsequent arguments as the member AIDs of the PAT to be outputted after executing the MakePAT command.

- Case of the MergePAT:

  Only when the holder AID of the PAT to be outputted after executing the MergePAT command is formally determined, the PAT processing device specifies the member AIDs of all the PATs specified by the first to N-th arguments of the MergePAT as the member AIDs of the PAT to be outputted after executing the MergePAT command.

- Case of the SplitPAT:

  Only when the holder AID of the PAT to be outputted after executing the SplitPAT command is formally determined, the PAT processing device specifies the member AID of the PAT specified by the first argument of the SplitPAT command as the member AID of the PAT to be outputted after executing the SplitPAT command. At this point, the member AIDs are distributed into different PATs in units of parentheses (). For example, in the case of:

  SplitPAT PAT (AID_{11}) $(AID_{21}$ $AID_{22})$ •••••••• $(AID_{N1}$ AID_{N2} •••••••• $AID_{NM})$ Enabler of AID

  (AID_{11}), $(AID_{21}$ $AID_{22})$ and $(AID_{N1}$ AID_{N2} •••••••• $AID_{NM})$ will be the member AIDs of different PATs having a common holder AID.

- Case of TransPAT:

  Only when the holder AID of the PAT to be outputted after executing the TransPAT command is formally determined, the PAT processing device specifies all the member AIDs remaining after excluding the member AID that is scheduled to be a new holder AID from all the member AIDs of the PAT specified by the first argument of the TransPAT command and the member AIDs of the PAT specified by the second argument as the member AIDs of the PAT to be outputted after executing the TransPAT command.

[0220] Next, the verification of the properness of the Enabler will be described. This verification of the properness of the Enabler is common to the MakePAT, the MergePAT, the SplitPAT and the TransPAT, and carried out according to Fig. 24 as follows.

(1) AID and Enabler are entered (step S5511).
(2) Each of these entered AID and Enabler is verified using the public key of the CA 1 (step S5512). If at least one of them is altered (step S5513 YES), the processing is terminated.
(3) A character string for certifying that it is Enabler is entered (step S5514).
(4) The top field of the Enabler of the step S5511 and the character string of the step S5514 are compared (step S5515). If they do not match (step S5516 NO), the processing is terminated.
(5) If they match (step S5516 YES), the AID of the step S5511 and the AID within the Enabler are compared (step S5517).
(6) A comparison result is outputted (step S5519).

[0221] Next, with references to Fig. 25 to Fig. 28, the third embodiment of the email access control scheme according to the present invention will be described in detail.

[0222] In the generation of a new PAT (MakePAT) and the PAT holder change (TransPAT) of the above described embodiment, it is necessary to give member AIDs and Enablers of member AIDs to the holder of the PAT, but when they are given to the holder, it becomes possible for that holder to participate the group communications hosted by the other holders by using the acquired member AIDs. Namely, there arises a problem that the pretending using the member AIDs become possible. Moreover, if that holder places the acquired member AIDs and Enablers of member AIDs on a medium that is readable by unspecified many, these member AIDs become accessible to anyone so that there arises a problem that the harassment to the users of the member AIDs may occur and the pretending using the member AIDS by a third person also become possible.

[0223] For this reason, in this third embodiment, it is made possible to carry out the MakePAT and the TransPAT without giving the Enablers of member AIDs to the holder.

[0224] To this end, in this third embodiment, the generation of a new PAT and the content change of the existing PAT are carried out by using Null-AID (AID$_{\text{Null}}$) and Enabler of Null-AID (Enabler of AID$_{\text{Null}}$).

[0225] Here, the processing involving the Null-AID obeys all of the following rules:

(a) the processing rules of MakePAT, MergePAT, SplitPAT and TransPAT as in the above described embodiment; and

(b) the rules applicable only to the Null-AID, including:

(i) Null-AID is known to every user, and

(ii) Enabler of Null-AID is known to every user.

[0226] Here, the processing rules as defined in the above described embodiment in the case of this third embodiment will be described.

(1) Making a PAT from plural AIDs (MakePAT):

AID$_{\text{holder}}$ + AID$_{\text{member1}}$ +AID$_{\text{member2}}$ + •••••••• + AID$_{\text{memberN}}$

+ Enabler of AID$_{\text{member1}}$ + Enabler of AID$_{\text{member2}}$ + ••••••••

+ Enabler of AID$_{\text{memberN}}$ + Enabler of AID$_{\text{holder}}$

→ PAT<AID$_{\text{holder}}$ | AID$_{\text{member1}}$, AID$_{\text{member2}}$, ••••••••, AID$_{\text{memberN}}$ >

(2) Merging plural PATs of the same holder (MergePAT):

PAT<AID$_{\text{holder}}$ | AID$_{\text{membera1}}$, AID$_{\text{membera2}}$, ••••••••, AID$_{\text{memberaM}}$ >

+ PAT<AID$_{\text{holder}}$ | AID$_{\text{memberb1}}$, AID$_{\text{memberb2}}$, ••••••••, AID$_{\text{memberbN}}$ >

+ Enabler of AID$_{\text{holder}}$

→ PAT<AID$_{\text{holder}}$ | AID$_{\text{membera1}}$, AID$_{\text{membera2}}$, ••••••••, AID$_{\text{memberaM}}$, AID$_{\text{memberb1}}$, AID$_{\text{memberb2}}$, ••••••••, AID$_{\text{memberbN}}$ >

(3) Splitting a PAT into plural PATs of the same holder (SplitPAT):

PAT<AID$_{\text{holder}}$ | AID$_{\text{membera1}}$, AID$_{\text{membera2}}$, ••••••••, AID$_{\text{memberaM}}$, AID$_{\text{memberb1}}$, AID$_{\text{memberb2}}$, ••••••••, AID$_{\text{memberbN}}$ >

+ Enabler of AID$_{\text{holder}}$

→ PAT<AID$_{\text{holder}}$ | AID$_{\text{membera1}}$, AID$_{\text{membera2}}$, ••••••••, AID$_{\text{memberaM}}$ >

+ PAT<AID$_{\text{holder}}$ | AID$_{\text{memberb1}}$, AID$_{\text{memberb2}}$, ••••••••, AID$_{\text{memberbN}}$ >

(4) Changing a holder AID of a PAT (TransPAT):

PAT<AID$_{\text{holder}}$ | AID$_{\text{membera1}}$, AID$_{\text{membera2}}$, ••••••••, AID$_{\text{memberaM}}$ > + PAT<AID$_{\text{holder}}$ | AID$_{\text{newholder}}$ >

+ Enabler of AID$_{\text{holder}}$ + Enabler of AID$_{\text{newholder}}$

→ PAT<AID$_{\text{newholder}}$ | AID$_{\text{membera1}}$, AID$_{\text{membera2}}$, ••••••••, AID$_{\text{memberaM}}$ >

[0227] The method for specifying the validity period value and the transfer control flag value in the PAT containing the Null-AID is similar to the method for specifying the validity period value and the transfer control flag value in the second embodiment described above. Next, the exemplary processings involving the Null-AID will be described.

(1) Case of producing PAT<AID$_{\text{Null}}$ | AID$_{\text{A}}$ > from AID$_{\text{A}}$ and Enabler of AID$_{\text{A}}$:

(a) According to the above described rules (b)(i) and (b)(ii) of the Null-AID, AID$_{\text{Null}}$ and Enabler of AID$_{\text{Null}}$ are known.

(b) Using MakePAT,

AID$_{\text{Null}}$ + AID$_{\text{A}}$ + Enabler of AID$_{\text{A}}$ + Enabler of AID$_{\text{Null}}$

→ PAT<AID$_{\text{Null}}$ | AID$_{\text{A}}$ >.

(2) Case of producing PAT<AID$_{\text{Null}}$ | AID$_{\text{A}}$, AID$_{\text{B}}$ > from PAT<AID$_{\text{Null}}$ | AID$_{\text{A}}$ > and PAT<AID$_{\text{Null}}$ | AID$_{\text{B}}$ >:

(a) According to the above described rules (b)(i) and (b)(ii) of the Null-AID, AID$_{\text{Null}}$ and Enabler of AID$_{\text{Null}}$ are known.

(b) Using MergePAT,

$$PAT<AID_{Null} \mid AID_A > + PAT<AID_{Null} \mid AID_B >$$

$$+ \text{Enabler of } AID_{Null}$$

$$\rightarrow PAT<AID_{Null} \mid AID_A, AID_B >.$$

(3) Case of producing PAT<AID$_A$ | AID$_B$ > from PAT<AID$_{Null}$ | AID$_A$ >, PAT<AID$_{Null}$ | AID$_B$ > and Enabler of AID$_A$:

(a) According to the above described rules (b)(i) and (b)(ii) of the Null-AID, AID$_{Null}$ and Enabler of AID$_{Null}$ are known.
(b) Using TransPAT,

$$PAT<AID_{Null} \mid AID_A > + PAT<AID_{Null} \mid AID_B >$$

$$+ \text{Enabler of } AID_{Null} + \text{Enabler of } AID_A$$

$$\rightarrow PAT<AID_A \mid AID_B >.$$

[0228] As shown in Fig. 25, the data structure of the Null-AID comprises a character string uniquely indicating that it is Null-AID (a character string defined by the CA, for example), which is signed by the CA using the secret key of the CA.

[0229] Also, as shown in Fig. 26, the data structure of the Enabler of Null-AID comprises a character string uniquely indicating that it is Enabler (a character string defined by the CA, for example) and the Null-AID itself, which is signed by the CA using the secret key of the CA.

[0230] Note that the Null-AID and the Enabler of Null-AID are maintained at secure PAT processing devices and secure PAT certification authority.

[0231] Next, the first exemplary application of this third embodiment will be described with reference to Fig. 27, which includes the following operations.

(1) The user-B (PAT member) generates PAT<AID-$_{Null}$ | AID$_B$ > by executing the above described exemplary processing (1) involving the Null-AID at the secure PAT processing device which is connected with the terminal of the user-B, and gives it to the user-A (PAT holder) by arbitrary means.

(2) The user-A who received PAT<AID$_{Null}$ | AID$_B$ > carries out the following operations at the secure PAT processing device which is connected with the terminal of the user-A.

(a) PAT<AID$_{Null}$ | AID$_A$ > is produced by executing the above described exemplary processing (1) involving the Null-AID.
(b) PAT<AID$_A$ | AID$_B$ > is produced by executing the above described exemplary processing (3) involving the Null-AID.

(3) The user-A gives the generated PAT<AID$_A$ | AID$_B$ > to the user-B by arbitrary means.

[0232] Note that the method for determining the validity period is the same as described above so that it will not be repeated here. Also, the processing involving the Null-AID is the same as described above so that it will not be repeated here.

[0233] In the case of giving PAT<AID$_{Null}$ | AID$_A$, AID$_B$ > to the user-B, the above described exemplary processing (2) involving the Null-AID will be executed in the operation (2) described above.

[0234] Next, the second exemplary application of this third embodiment will be described with reference to Fig. 28, which includes the following operations.

(1) The user-B (PAT member) produces PAT<AID-$_{Null}$ | AID$_B$ > by executing the above described exemplary processing (1) involving the Null-AID at the secure PAT processing device which is connected with the terminal of the user-B, and registers it along arbitrary disclosed information at the ADS.
(2) The user-A produces PAT<AID$_{Null}$ | AID$_A$ > by executing the above described exemplary processing (1) involving the Null-AID at the secure PAT processing device which is connected with the terminal of the user-A, and presents it along arbitrary search conditions to the ADS.
(3) When the personal information of the user-B satisfies the search conditions presented by the user-A, the secure PAT processing device connected with the ADS carries out the following operations.

(a) PAT<AID$_{Null}$ | AID$_A$, AID$_B$ > is produced by executing the above described exemplary processing (2) involving the Null-AID.
(b) The produced PAT<AID$_{Null}$ | AID$_A$, AID$_B$ > is given to the ADS.

(4) The ADS gives PAT<AID$_{Null}$ | AID$_A$, AID$_B$ > produced by the PAT processing device to the user-A.
(5) The user-A who received PAT<AID$_{Null}$ | AID$_A$, AID$_B$ > produces PAT<AID$_A$ | AID$_B$ > by executing the following TransPAT processing at the secure PAT processing device which is connected with the terminal of the user-A.

$$PAT<AID_{Null} \mid AID_A > + PAT<AID_{Null} \mid AID_A, AID_B >$$

$$+ \text{Enabler of } AID_{Null} + \text{Enabler of } AID_A$$

$$\rightarrow PAT<AID_A \mid AID_B >.$$

[0235] Note that the method for determining the validity period is the same as described above so that it will not be repeated here. Also, the processing involving the Null-AID is the same as described above so that it will not be repeated here.

[0236] In the case of generating PAT<AID$_A$ | AID$_B$ > at the PAT processing device connected with the ADS, Enabler of AID$_A$ will be given to that PAT processing device, and the above described exemplary processing (3) involving the Null-AID will be executed in the operation (3) described above.

[0237] In the case of generating PAT<AID$_B$ | AID$_A$ > at the PAT processing device connected with the ADS and giving it to the user-B, Enabler of AID$_B$ will be given to that PAT processing device, and the above described exemplary processing (3) involving the Null-AID will be executed in the operation (3) described above.

[0238] Next, with references to Fig. 29 to Fig. 31, the fourth embodiment of the email access control scheme according to the present invention will be described in detail.

[0239] In the group communication, a situation where it is desired to fix the participants is frequently encountered, but the above described embodiment does not have a function for making it impossible to change the PAT so that the participants cannot be fixed. Namely, in the above described embodiment, whether or not to fix the participants is left to the judgement of the holder of the PAT.

[0240] For this reason, in this fourth embodiment, a read only attribute is set up in the PAT. More specifically, in this fourth embodiment, the read only attribute is set up in the PAT by using God-AID (AID$_{God}$).

[0241] Here, the processing involving the God-AID obeys all of the following rules:

(a) God-AID is known to every user, and

(b) the processing involving God-AID is allowed only in the following cases:

(i) a case where the AID$_{holder}$ is neither AID$_{Null}$ nor AID$_{God}$:

PAT<AID$_{holder}$ | AID$_{member1}$, AID$_{member2}$, ········, AID$_{memberN}$ > + Enabler of AID$_{holder}$

→ PAT<AID$_{god}$ | AID$_{holder}$, AID$_{member1}$, AID$_{member2}$, ········, AID$_{memberN}$ >

(ii) a case where AID$_{holder}$ is AID$_{Null}$:

PAT<AID$_{Null}$ | AID$_{member1}$, AID$_{member2}$, ········, AID$_{memberN}$ >

+ Enabler of AID$_{Null}$

→ PAT<AID$_{god}$ | AID$_{member1}$, AID$_{member2}$, ········, AID$_{memberN}$ >

[0242] As shown in Fig. 29, the data structure of the God-AID comprises a character string uniquely indicating that it is God-AID (a character string defined by the CA, for example), which is signed by the CA using the secret key of the CA. The God-AID is maintained at the secure PAT processing devices and the secure PAT certification authority described above.

[0243] The processings of a PAT that contains the Null-AID are according to Fig. 21 to Fig. 24. When the holder AID is neither Null-AID nor God-AID, the God-AID is appended to the AID list and the holder index value is specified to be a position of the God-AID in the AID list after appending the God-AID. When the holder AID is Null-AID, the Null-AID is deleted from the AID list, the God-AID is appended to the AID list, and then the holder index value is specified to be a position of the God-AID in the AID list after appending the God-AID.

[0244] Next, the exemplary application of this fourth embodiment will be described with reference to Fig. 30.

[0245] In the case of producing PAT<AID$_{God}$ | AID$_A$, AID$_B$ > from PAT<AID$_{Null}$ | AID$_A$ > and PAT<AID$_{Null}$ | AID$_B$ >, the following processing is executed at the secure PAT processing device which is connected with the terminal of the PAT holder (user-A in Fig. 30).

(1) Using MergePAT,

PAT<AID$_{Null}$ | AID$_A$ > + PAT<AID$_{Null}$ | AID$_B$ >

+ Enabler of AID$_{Null}$

→ PAT<AID$_{Null}$ | AID$_A$, AID$_B$ >.

(2) According to the above described rule (a) of the God-AID, AID$_{God}$ is known.

(3) According to the above described rule (b)(ii) of the God-AID,

PAT<AID$_{Null}$ | AID$_A$, AID$_B$ > + Enabler of AID$_{Null}$

→ PAT<AID$_{god}$ | AID$_A$, AID$_B$ >

[0246] The above processing is also executed at the secure PAT processing device connected with a computer (search engine, etc.) of the third person (Fig. 31) or at the secure PAT certification authority.

[0247] Next, with reference to Fig. 32, the fifth embodiment of the email access control scheme according to the present invention will be described in detail.

[0248] When the Null-AID is added as described in the third embodiment, there arises a problem that it becomes possible for the holder of the PAT (the user of the holder AID) to transfer the access right with respect to the member (the user of the member AID) to the third person, and moreover this transfer can be done without a permission of the member, as will be described now.

(1) The holder-A of PAT<AID$_{A}$ | AID$_{B}$ > (for the member-B) produces PAT<AID$_{Null}$ | AID$_{B}$ > by using PAT<AID$_{A}$ | AID$_{B}$ >, AID$_{A}$ and Enabler of AID$_{A}$. Here, it is assumed that the holder-A knows all of AID$_{A}$, Enabler of AID$_{A}$, AID$_{Null}$, and Enabler of AID$_{Null}$ in addition to PAT<AID$_{A}$ | AID$_{B}$ >.

(a) The holder-A produces PAT<AID$_{A}$ | AID$_{Null}$ > using the MakePAT as follows.

AID$_{A}$ + AID$_{Null}$ + Enabler of AID$_{Null}$ + Enabler of AID$_{A}$

→ PAT<AID$_{A}$ | AID$_{Null}$ >

(b) The holder-A produces PAT<AID$_{Null}$ | AID$_{B}$ > using the TransPAT as follows.

PAT<AID$_{A}$ | AID$_{B}$ > + PAT<AID$_{A}$ | AID$_{Null}$ >

+ Enabler of AID$_{A}$ + Enabler of AID$_{Null}$

→ PAT<AID$_{Null}$ | AID$_{B}$ >

After the above described operation (1)(b), the holder-A gives PAT<AID$_{Null}$ | AID$_{B}$ > to the third person-C, the following operation (2) becomes possible.

(2) The third person-C produces PAT<AID$_{C}$ | AID$_{B}$ > by using PAT<AID$_{Null}$ | AID$_{B}$ >. Here, it is assumed that the third person-C knows all of AID$_{C}$, Enabler of AID$_{C}$, AID$_{Null}$, and Enabler of AID$_{Null}$ in addition to PAT<AID$_{Null}$ | AID$_{B}$ >.

(a) The third person-C produces PAT<AID$_{Null}$ | AID$_{C}$ > using the MakePAT as follows.

AID$_{Null}$ + AID$_{C}$ + Enabler of AID$_{C}$ + Enabler of AID$_{Null}$

→ PAT<AID$_{Null}$ | AID$_{C}$ >

(b) The third person-C produces PAT<AID$_{C}$ | AID$_{B}$ > using the TransPAT as follows.

PAT<AID$_{Null}$ | AID$_{B}$ > + PAT<AID$_{Null}$ | AID$_{C}$ >

+ Enabler of AID$_{Null}$ + Enabler of AID$_{C}$

→ PAT<AID$_{C}$ | AID$_{B}$ >

[0249] As a result of the above described operation (2)(b), the third person-C obtains PAT<AID$_{C}$ | AID$_{B}$ > so that accesses to the member-B become possible.

[0250] For this reason, in this fifth embodiment, it is made impossible for the holder of PAT<AID$_{holder}$ | AID$_{member}$ > to produce PAT<AID$_{Null}$ | AID$_{member}$ > from this PAT<AID$_{holder}$ | AID$_{member}$ > as long as the holder does not know Enabler of AID$_{member}$.

[0251] In the third embodiment described above, in order for the PAT holder to produce PAT<AID$_{Null}$ | AID$_{member}$ > without using Enabler of AID$_{member}$, it is necessary to produce PAT<AID$_{holder}$ | AID$_{Null}$ >.

[0252] To this end, in this fifth embodiment, for the Null-AID described in the third embodiment, the following rule is added:

- the Null-AID can be used only as the holder AID of the PAT (the Null-AID cannot be used as the member AID).

  That is, PAT<AID$_{Null}$ | AID$_{member1}$, AID$_{member2}$, ••••••••, AID$_{memberN}$ > is allowed, but PAT<AID$_{holder}$ | AID$_{Null}$, AID$_{member1}$, AID$_{member2}$, ••••••••, AID$_{memberN}$ > is not allowed.

  Each of the secure PAT processing devices and the secure PAT certification authority is additionally equipped with a function for checking whether the Null-AID is contained as the member AID or not. This member AID checking processing is carried out according to Fig. 32 as follows.

  (1) Null-AID and PAT are entered (step S6911).
  (2) All the member AIDs are taken out from the PAT entered at the step S6911 (step S6913).
  (3) Each of the taken out member AIDs is compared with the Null-AID entered at the step S6911 (step S6915).

  If all the member AIDs do not completely match with the Null-AID (step S6917 NO, step S6919 NO), the processing proceeds to the MergePAT, SplitPAT or TransPAT processing (Fig. 21 or Fig. 22) (step S6921).

  If there is a member AID that completely matches with the Null-AID (step S6917 YES), the processing is terminated.

[0253] Next, with reference to Fig. 33 to Fig. 39, the sixth embodiment of the email access control scheme according to the present invention will be described in detail.

[0254] This sixth embodiment differs from the first embodiment described above in that a link information is added to the AID of Fig. 2 used in the first embodiment, as shown in a part (b) of Fig. 34, while a link information of the AID is set instead of the AID itself that is contained in the 1-to-1 PAT of Fig. 2, as shown in a part (c) of Fig. 34, such that the AID is uniquely identified by the link information.

[0255] Note that such an AID to which the link information is added will be referred to as a link information attached AID, and a 1-to-1 PAT having the link information of the AID will be referred to as a link specifying 1-to-1 PAT. Also, the link information is an information capable of uniquely identifying the AID, which is given by a kind of data generally known as identifier such as a serial number uniquely assigned to the AID by the CA for example.

[0256] Fig. 33 shows an overall configuration of a communication system in this sixth embodiment.

[0257] In Fig. 33, the CA (Certification Authority) 1 has a right to authenticate OIDs and a right to issue AIDs, and functions to allocate AIDs to users 3.

[0258] The SCS (Secure Communication Service) 5 transfers emails among the users 3, carries out the receiving refusal and the identity judgement and the extraction of the OID according to the need.

[0259] The ADS (Anonymous Directory Service) 7 is a database for managing the AID, the transfer control flag value, the validity period value, and the disclosed information of each user 3. The ADS 7 has a function to generate the PAT from the AID of a searcher and the AID of a registrant who satisfies the search conditions, and issue it to the searcher.

[0260] A series of processing from generating the AID from the OID according to a request from a user until allocating the AID to that user is basically the same as in the first embodiment, except that the link information is to be added, which will now be described with reference to Fig. 34.

[0261] Fig. 34 shows exemplary formats of the OID, the link information attached AID, and the link specifying 1-to-1 PAT. As shown in a part (a) of Fig. 34, the OID is an information comprising an arbitrary character string according to a rule by which the CA 1 can uniquely identify the user and a public key, which is signed by the CA 1.

[0262] Also, as shown in a part (b) of Fig. 34, the link information attached AID is an information comprising fragments of the OID and their position information, redundant character strings, an SCS information given by an arbitrary character string (host name, real domain name, etc.) by which a host or a domain that is operating the SCS 5 can be uniquely identified on the network, and the link information, which is signed by the CA 1.

[0263] Also, as shown in a part (c) of Fig. 34, the link specifying 1-to-1 PAT is an information comprising the transfer control flag, the link information of AID$_0$, the link information of AID$_1$, and the validity period, which is signed by the ADS 7 using a secret key of the ADS 7.

[0264] A procedure by which the user 3 requests the link information attached AID to the CA 1 is the same as that of the first embodiment. A procedure by which the CA 1 issues the link information attached AID to the user 3 in response to a request for the AID is also the same as that of the first embodiment.

[0265] Next, the link information attached AID generation processing at the CA will be described with reference to Fig. 35.

[0266] In the procedure of Fig. 35, the CA 1 generates an information of a length equal to the total length L of the OID, and sets this information as a tentative AID (step S7211). Then, in order to carry out the partial copying of the OID, values of parameters p_i and ℓ_i for specifying a copying region are determined using arbitrary means such as random number generation respectively (step S7213). Here, L is equal to the total length L of the OID, and ℓ_i is an arbitrarily defined value within a range in which a relationship of $0 \leq \ell_i \leq L$ holds. Then, an information in a range between a position p_i to a position $p_i + \ell_i$ from the top of the OID is copied to the same positions in the tentative AID (step S7215). In other words, this OID fragment will be copies to a range between a position p_i and a position $p_i + \ell_i$ from the top of the tentative AID. Then, the values of p_i and ℓ_i are written into a prescribed range in the tentative AID into which the OID has been partially copied, in a form encrypted by an arbitrary means (step S7217). Then, an SCS information given by an arbitrary character string (host name, real domain, etc.) that can uniquely identify a host or a domain that is operating the SCS 5 on the network is written into a prescribed range in the tentative AID into which these values are written (step S7219). Then, the link information is written (step S7220). Then, the tentative AID into which the above character string and the link information are written is signed using a secret key of the CA 1 (step S7221).

[0267] Next, a procedure for registering the AID of a user-B 3 and the disclosed information into the ADS 7 will be described. First, the bidirectional authentication by arbitrary means using the AID of the user-B 3 and the certificate of the ADS 7 is carried out between the user-B 3 who is a registrant and the ADS 7. Then, the user-B 3 transmits the transfer control flag value, the validity period value, and the disclosed information such as interests to the ADS 7. Then, the ADS 7 stores the transfer control flag value, the validity period value, and the entire disclosed information in relation to the AID of the user-B 3 in its storage device. Here, there can be cases where communications between the user-B 3 who is the registrant and the ADS 7 are to be encrypted.

[0268] Next, a procedure by which a user-A 3 searches through the disclosed information that is registered in the ADS 7 will be described. First, the bidirectional authentication by arbitrary means using the AID of the user-A 3 and the certificate of the ADS 7 is carried out between the user-A 3 who is a searcher and the ADS 7. Then, the user-A 3 transmits arbitrary search conditions to the ADS 7. Then, the ADS 7 presents all the received search conditions to its storage device, and extracts the AID of a registrant which satisfies these search conditions. Then, the ADS 7 generates the link specifying 1-to-1 PAT from the link information of the AID of the user-A 3 and the link information of the AID of the registrant who satisfied the search conditions, the transfer control flag value, and the validity period value. Then, the ADS 7 transmits the generated PAT to the user-A 3. Here, there can be cases where communications between the user-A 3 who is a searcher and the ADS 7 are to be encrypted. Note that the link specifying

1-to-1 PAT is generated as a search result of the ADS 7.

[0269] Next, the link specifying 1-to-1 PAT generation processing at the ADS 7 will be described with reference to Fig. 36.

[0270] First, an information of a prescribed length is generated, and this information is set as a tentative PAT (step S7510). Then, the link information of the AID of the user-A 3 who is a searcher and the link information of the AID of the user-B 3 who is a registrant are copied into a prescribed region of the tentative PAT (step S7516). Then, the transfer control flag value and the validity period value are written into respective prescribed regions of the tentative PAT into which the link informations of the AIDs are copied (step S7517). Then, the tentative PAT into which these values are written is signed using a secret key of the ADS 7 (step S7519).

[0271] Next, the transfer control using the link specifying 1-to-1 PAT will be described. The transfer control is a function for limiting accesses to a user who has a proper access right from a third person to whom the PAT has been transferred or who has eavesdropped the PAT (a user who originally does not have the access right).

[0272] The ADS 7 and the user-B 3 of the registrant AID can prohibit a connection to the user-B 3 from a third person who does not have the access right, by setting a certain value in to the transfer control flag of the PAT.

[0273] When the transfer control flag value is set to be 1, the sender's AID is authenticated between the SCS 5 and the sender according to an arbitrary challenge/response process, so that even if the sender gives both the sender's AID and the PAT to another user other than the sender, that another user will not be able to make a connection to the registrant of the ADS 7 through the SCS 5.

[0274] On the other hand, when the transfer control flag value is set to be 0, no challenge/response process will be carried out between the SCS 5 and the sender, so that if the sender gives both the sender's AID and the PAT to another user other than the sender, that another user will also be able to make a connection to the registrant of the ADS 7 through the SCS 5.

[0275] Next, the email access control method at the SCS 5 will be described with reference to Fig. 37.

[0276] The sender specifies "[sender's AID]@[real domain of SCS of sender]" in From: line, and "[PAT]@[real domain of SCS of sender]" in To: line.

[0277] The SCS 5 acquires a mail received by an MTA (Message Transfer Agent) such as SMTP (Simple Mail Transfer Protocol), and executes the processing of Fig. 37 as follows.

(1) The signature of the PAT is verified using a public key of the ADS 7 (step S7713).

When the PAT is found to have been altered (step S7715 YES), the mail is discarded and the processing is terminated (step S7716).

When the PAT is found to have been not altered (step S7715 NO), the following processing (2) is executed.

(2) The search is carried out by presenting the link information of the sender's AID to the PAT (steps S7717, S7720, S7722).

When a link information that completely matches with the link information of the sender's AID is not contained in the PAT (step S7723 NO), the mail is discarded and the processing is terminated (step S7716).

When a link information that completely matches with the link information of the sender's AID is contained in the PAT (step S7723 YES), the following processing (3) is executed.

(3) The validity period value of the PAT is evaluated (steps S7725, S7727).

When the PAT is outside the validity period (step S7727 NO), the mail is discarded and the processing is terminated (step S7716).

When the PAT is within the validity period (step S7727 YES), the following processing (4) is executed.

(4) Whether or not to authenticate the sender is determined by referring to the transfer control flag value of the PAT (steps S7731, S7733).

When the value is 1 (step S7733 YES), the SCS 5 acquires the sender's AID itself and the public key of the sender's AID by presenting the link information to the CA 1, and then the challenge/response authentication between the SCS 5 and the sender is carried out, and the signature of the sender is verified (step S7735). When the signature is valid, the recipient is specified and the PAT is attached (step S7737). When the signature is invalid, the mail is discarded and the processing is terminated (step S7716).

When the value is 0 (step S7733 NO), the recipient is specified and the PAT is attached without executing the challenge/response authentication (step S7737).

[0278] The challenge/response authentication between the SCS 5 and the sender is the same as that for the 1-to-1 PAT described above.

[0279] Next, a method for specifying the recipient at the SCS 5 will be described. First, the SCS 5 carries out the search by presenting the link information of the sender's AID to the PAT, so as to acquire all the link informations which do not completely match the link information of the sender's AID. Then, the search is carried out by presenting all these acquired link informations to the CA 1 so as to acquire the AIDs. All these acquired AIDs will be defined as recipient's AIDs hereafter. Then, for every recipient's AID, the real domain of SCS of recipient is taken out from the recipient's AID. Then, the recipient is specified in a format of "[recipient's AID]@[real domain of SCS of recipient]". Finally, the SCS 5 changes the sender from a format of "[sender's AID]@[real domain of SCS of sender]" to a format of "sender's AID".

[0280] The method for attaching the PAT at the SCS 5 is the same as that for the 1-to-1 PAT described above.

[0281] Next, a method of receiving refusal with respect to the PAT at the SCS 5 will be described.

[0282] Receiving refusal setting: The bidirectional authentication is carried out by an arbitrary means between the user and the SCS 5. Then, the user transmits a registration command, his/her own AID, and arbitrary PATs to the SCS 5. Then, the SCS 5 verifies the signature of the received AID. If the signature is invalid, the processing of the SCS 5 is terminated. If the signature is valid, the SCS 5 next verifies the signature of each received PAT using a public key of the ADS. Those PATs with the invalid signature are discarded by the SCS 5. When the signature is valid, the SCS 5 takes out the link information from the received AID, and then carries out the search by presenting the taken out link information to each PAT. For each of those PATs which contain the link information that completely matches with the link information of the received AID, the SCS 5 presents the registration command and the PAT to the storage device such that the PAT is registered into the storage device. Those PATs which do not contain the link information that completely matches with the link information of the received AID are discarded by the SCS 5 without storing them into the storage device. Here, there can be cases where communications between the user and the SCS 5 are to be encrypted.

[0283] Receiving refusal execution: The SCS 5 carries out the search by presenting the PAT to the storage device. When a PAT that completely matches the presented PAT is registered in the storage device, the mail is discarded. When a PAT that completely matches the present PAT is not registered in the storage device, the mail is not discarded.

[0284] Receiving refusal cancellation: The bidirectional authentication is carried out by an arbitrary means between the user and the SCS 5. Then, the user presents his/her own AID to the SCS 5. Then, the SCS 5 verifies the signature of the received AID. If the signature is invalid, the processing of the SCS 5 is terminated. If the signature is valid, the SCS 5 next takes out the link information from the presented AID, and presents the taken out link information as a search condition to the storage device and acquire all the PATs that contain the presented link information, and then presents all the acquired PATs to the user. Then, the user selects all the PATs for which the receiving refusal is to be cancelled by referring to all the PATs presented from the SCS 5, and transmits all the selected PATs along with a deletion command to the SCS 5. Upon receiving the deletion command and all the PATs for which the receiving refusal is to be cancelled, the SCS 5 presents the deletion command and all the PATs received from the user to the storage device, such that all the received PATs are deleted from the storage device.

[0285] Note that the method of receiving refusal with respect to the link specifying 1-to-N PAT at the SCS 5 is the same as the method of receiving refusal with respect to the link specifying 1-to-1 PAT described above.

[0286] Next, the judgement of identity will be described with reference to Fig. 38 and Fig. 39.

(1) An initial value of a variable OID_M is defined as a bit sequence with a length equal to the total length L of the OID and all values equal to "0". Also, an initial value of a variable OID_V is defined as a bit sequence with a length equal to the total length of the OID and all values equal to "0" (step S7911).

(2) One link information attached AID is selected from a set of processing target link information attached AIDs, and the following bit processing is carried out (step S7913).

(a) Values of variables AID_M and AID_V are determined according to the position information contained in the link information attached AID (step S7915). Here, AID_M is defined as a bit sequence with a length equal to the total length L of the OID and a value of a position at which the OID information is defined is "1" while a value of a position at which the OID information is not defined is "0" (see Fig. 39). Also, AID_V is defined as a bit sequence with a length equal to the total length L of the OID and a value of a position at which the OID information is defined is an actual value of the OID information while a value of a position at which the OID information is not defined is 0 (see Fig. 39).

(b) AND processing of OID_M and AID_M is carried out and its result is substituted into a variable OVR_M (step S7917).

(c) AND processing of OVR_M and AID_M as well as AND processing of OVR_M and OID_M are carried out and their results are compared (step S7919). When they coincide, OR processing of OID_M and AID_M is carried out and its result is substituted into OID_M (step S7921), while OR processing of OID_V and AID_V is also carried out and its result is substituted into OID_M (step S7923). On the other hand, when they do not coincide, the processing proceeds to the step S7925.

(d) A link information attached AID to be processed next is selected from a set of processing target link information attached AIDs. When at least one another link information attached AID is contained in the set, the steps S7913 to S7923 are executed for that another link information attached AID. When no other link information attached AID is contained in the set, the processing proceeds to the step S7927.

(e) Values of OID_M and OID_V are outputted (step S7927).

[0287] The value of OID_M that is eventually obtained indicates all positions of the OID information that can be recovered from the set of processing target link information attached AIDs. Also, the value of OID_V that is eventually obtained indicates all the OID information that can be recovered from the set of processing target link information attached AID. In other words, by using the values of OID_M and OID_V, it is possible to obtain the OID albeit probabilistically when the value of OID_V is used as a search condition, and it is possible to quantitatively evaluate a precision of the above search by a ratio OID_M/L with respect to the total length L of the OID.

[0288] As described above, in this sixth embodiment, the CA 1 which is a Trusted Third Party with high secrecy and credibility generates the link information attached AID in which the personal information is concealed, from the OID that contains the highly secret personal information such as name, telephone number, real email address, etc., according to a user request, and issues the AID to the user. By identifying the user by this AID on the communication network as well as in various services provided on the communication network, it becomes possible to provide both the anonymity guarantee and the identity guarantee for the user. In other words, it becomes possible for the user to communicate with another user without revealing the own real name, telephone number, email address, etc., to that another user, and it also becomes possible to disclose the disclosed information to unspecified many through the ADS 7 as will be described below.

[0289] The user registers the disclosed information, that is an information which is supposed to have a low secrecy compared with the personal information at the ADS 7. In the case of searching the disclosed information and the registrant AID, the searcher presents the link information attached AID of the searcher and arbitrary search conditions to the ADS 7. The ADS 7 then extracts the registrant link information attached AID that satisfies these search conditions, and generates the link specifying 1-to-1 PAT from the link information of the AID of the searcher and the link information of the AID of the registrant who satisfied the search conditions, the transfer control flag value, and the validity period value.

[0290] In this link specifying 1-to-1 PAT, the transfer control flag value and the validity period value are set as shown a part (c) of Fig. 34, and by setting up this validity period in advance, it is possible to limit connections from the sender.

[0291] It is also possible to prohibit connections from a third person who does not have the access right, by using the transfer control flag value. Namely, when the transfer control flag value is set to be 1, the sender's AID is authenticated between the SCS 5 and the sender according to an arbitrary challenge/response process, so that even if the sender gives both the sender's AID and the PAT to another user other than the sender, that another user will not be able to make a connection to the registrant of the ADS 7 through the SCS 5. On the other hand, when the transfer control flag value is set to be 0, no challenge/response process will be carried out between the SCS 5 and the sender, so that if the sender gives both the sender's AID and the PAT to another user other than the sender, that another user will also be able to make a connection to the registrant of the ADS 7 through the SCS 5.

[0292] It is also possible to make a connection request to the communication network such that a call for which the recipient is specified by the link specifying 1-to-1 PAT will be received by the recipient's AID or the sender's AID specified by the link information of the link specifying 1-to-1 PAT. In addition, it is also possible to refuse receiving calls with the link specifying 1-to-1 PAT selected by the recipient among calls which are specified by the link specifying 1-to-1 PAT. It is also possible to cancel the receiving refusal of the calls with the link specifying 1-to-1 PAT selected by the recipient. In addition, as a measure against the sender who repeats the personal attack using a plurality of sender's AIDs by taking an advantage of the anonymity, it is possible to judge the identity of the OID from these plurality of sender's AIDs and it is possible to extract that OID at some probability.

[0293] Next, with references to Fig. 40 to Fig. 49, the seventh embodiment of the email access control scheme according to the present invention will be described in detail.

[0294] In contrast to the sixth embodiment described above which is directed to the case where a sender and a recipient are set in 1-to-1 correspondence, this seventh embodiment is directed to the case where a sender and recipients are set in 1-to-N correspondence and a generation of a new link specifying 1-to-N PAT and a content change of the existing link specifying 1-to-N PAT can be made by the initiative of a user, similarly as in the second embodiment described above. Here, the sender is either a holder of the PAT or a member of the PAT. Similarly, the recipient is either a holder of the PAT or a member of the PAT.

[0295] As described in the second embodiment, in general, a membership of a group communication (mailing list, etc.) is changing dynamically so that it is necessary for a host of the group communication to manage information on a point of contact such as telephone number, email address, etc., of each member. In contrast, in the case where it is possible to newly generate a 1-to-1 PAT as in the sixth embodiment, the management of a point of contact is difficult. For example, it is difficult to manage the group collectively, and even if it is given to the others for the purpose of the transfer control, it does not function as an address of the group communication such as mailing list.

[0296] In this seventh embodiment, in order to resolve such a problem, it is made possible to carry out a generation of a new link specifying 1-to-N PAT and a content change or the existing link specifying 1-to-N PAT by the initiative of a user.

[0297] First, the definition of various identifications used in this seventh embodiment will be described with references to Fig. 40 and Fig. 41.

[0298] As shown in a part (a) of Fig. 40, the OID is an information comprising an arbitrary character string (telephone number, email address, etc.) according to a rule by which the CA 1 can uniquely identify the user and a public key, which is signed by the CA 1.

[0299] Also, as shown in a part (b) of Fig. 40, the link information attached AID is an information comprising fragments of the OID and their position information, redundant character strings, an SCS information given by an arbitrary character string (host name, real domain name, etc.) by which a host or a domain that is operating the SCS 5 can be uniquely identified on the network, and a link information, which is signed by the CA 1. Note that the AID may be encrypted at the SCS 5 or the CA 1. The link information is the same as in the sixth embodiment.

[0300] Also, as shown in a part (c) of Fig. 40, the link specifying 1-to-N PAT is an information comprising two or more link informations of AIDs, a holder index, the validity period, the transfer control flag, and a PAT processing device identifier, which is signed using a secret key of the PAT processing device.

[0301] Here, one of the link informations of AIDs is the link information of the holder AID of this PAT, where the change of the information contained in the PAT such as an addition of the link information of AID to the PAT, a deletion of the link information of AID from the PAT, a change of the validity period in the PAT, a change of the transfer control flag value in the PAT, etc., can be made by presenting the link information of the holder AID and a corresponding Enabler to the PAT processing device.

[0302] On the other hand, the link informations of AIDs other than the link information of the holder AID that are contained in the PAT are all link information of member AIDs, where a change of the information contained in the PAT cannot be made even when the link information of the member AID and a corresponding Enabler are presented to the PAT processing device.

[0303] The holder index is a numerical data for identifying the link information of the holder AID, which is defined to take a value 1 when the link information of the holder AID is a top link information of AID in the link specifying AID list formed from the link information of the holder AID and the link informations of the member AIDs, a value 2 when the link information of the holder AID is a second link information of AID from the top of the link specifying AID list, or a value n when the link information of the holder AID is an n-th link information of AID from the top of the link specifying AID list.

[0304] The transfer control flag value is defined to take either 0 or 1 similarly as in the case of the link specifying 1-to-1 PAT.

[0305] The link information of the holder AID is defined to be a link information of AID which is written at a position of the holder index value in the link specifying AID list. The link informations of the member AIDs are defined to be all the link informations of AIDs other than the link information of the holder AID.

[0306] The validity period is defined by any one or combination of the number of times for which the PAT is available, the absolute time (UTC) by which the PAT becomes unavailable, the absolute time (UTC) by which the PAT becomes available, and the relative time (lifetime) since the PAT becomes available until it becomes unavailable.

[0307] The identifier of a PAT processing device (or a PAT processing object on the network) is defined as a serial number of the PAT processing device (or an distinguished name of the PAT processing object on the network). The secret key of the PAT processing device (or the PAT processing object on the network) is defined to be uniquely corresponding to the identifier.

[0308] Also, in this second embodiment, an Enabler is introduced as an identifier corresponding to the AID. As shown in Fig. 41, the Enabler is an information comprising a character string uniquely indicating that it is an Enabler and a link information attached AID itself, which is signed by the CA 1.

[0309] Next, the operations for a generation of a new PAT and a content change of the existing PAT will be described. Here, the following operations are defined at a secure PAT processing device on the communication terminal or a PAT processing object on the CA or on a network which is properly requested from the CA (which will also be referred to as a PAT processing device hereafter). These operations are similar to those of the second embodiment described above so that they will be described by referring to Fig. 10 to Fig. 13 but it is assumed that each occurrence of AID in Fig. 10 to Fig. 13 should be replaced by the link information of AID in the following.

1. Editing of link specifying AID list:

A link specifying AID list, which is a list of link informations of AIDs contained in the PAT, is edited using link information attached AIDs and Enabler. Else, the link specifying AID list is newly generated.

2. Setting of the validity period and the transfer control flag:

The validity period value and the transfer control flag value contained in the PAT are changed using a link information attached AID and Enabler. Also, a new validity period value and a new transfer control flag value are set in the newly generated link specifying AID list.

[0310] A user who presented the holder AID and the Enabler corresponding to this holder AID to the PAT processing device can edit the list of link informations of

AIDs contained in the PAT. In this case, the following processing rules are used.

(1) Generating a new PAT (MakePAT) (see Fig. 10):

The link specifying AID list (LALIST<(link)holder AID | (link)member AID$_1$, (link)member AID$_2$, • • • • • • • •, (link)member AID$_n$ >) where (link)AID$_x$ denotes the link information of AID$_x$ is newly generated, and the validity period value and the transfer control flag value are set with respect to the generated LALIST.

(link)AID$_A$ + (link)AID$_B$ + Enabler of AID$_B$

+ Enabler of AID$_A$

→ LALIST<(link)AID$_A$ | (link)AID$_B$ >

LALIST<(link)AID$_A$ | (link)AID$_B$ > + Enabler of AID$_A$

+ validity period value

+ transfer control flag value

→ PAT<(link)AID$_A$ | (link)AID$_B$ >

(2) Merging PATs (MergePAT) (see Fig. 11):

A plurality of LALISTs of the same holder AID are merged and the validity period value and the transfer control flag value are set with respect to the merged LALIST.

LALIST<(link)AID$_A$ | (link)AID$_{B1}$, (link)AID$_{B2}$, • • • • • • • • >

+ LALIST<(link)AID$_A$ | (link)AID$_{C1}$, (link)AID$_{C2}$, • • • • • • • • >

+ Enabler of AID$_A$

→ LALIST<(link)AID$_A$ | (link)AID$_{B1}$, (link)AID$_{B2}$, • • • • • • • •, (link)AID$_{C1}$, (link)AID$_{C2}$, • • • • • • • • >

LALIST<(link)AID$_A$ | (link)AID$_{B1}$, (link)AID$_{B2}$, • • • • • • • •, (link)AID$_{C1}$, (link)AID$_{C2}$, • • • • • • • • >

+ Enabler of AID$_A$ + validity period value

+ transfer control flag value

→ PAT<(link)AID$_A$ | (link)AID$_{B1}$, (link)AID$_{B2}$, • • • • • • • •, (link)AID$_{C1}$, (link)AID$_{C2}$, • • • • • • • • >

(3) Splitting a PAT (SplitPAT) (see Fig. 12):

The LALIST is split into a plurality of LALISTs of the same holder AID, and the respective validity period value and transfer control flag value are set with respect to each one of the split LALISTs.

LALIST<(link)AID$_A$ | (link)AID$_{B1}$, (link)AID$_{B2}$, • • • • • • • •, (link)AID$_{C1}$, (link)AID$_{C2}$, • • • • • • • • >

+ Enabler of AID$_A$

→ LALIST<(link)AID$_A$ | (link)AID$_{B1}$, (link)AID$_{B2}$, • • • • • • • • >

+ LALIST<(link)AID$_A$ | (link)AID$_{C1}$, (link)AID$_{C2}$, • • • • • • • • >

LALIST<(link)AID$_A$ | (link)AID$_{C1}$, (link)AID$_{C2}$, • • • • • • • • >

+ Enabler of AID$_A$ + validity period value

+ transfer control flag value

→ PAT<(link)AID$_A$ | (link)AID$_{C1}$, (link)AID$_{C2}$, • • • • • • • • >

(4) Changing a holder of a PAT (TransPAT) (see Fig. 13):

The holder AID of the LALIST is changed, and the validity period value and the transfer control flag value are set with respect to the changed LALIST.

LALIST<(link)AID$_A$ | (link)AID$_B$ >

+ LALIST<(link)AID$_A$ | (link)AID$_{C1}$, (link)AID$_{C2}$, • • • • • • • • >

+ Enabler of AID$_A$ + Enabler of AID$_B$

→ LALIST<(link)AID$_B$ | (link)AID$_{C1}$, (link)AID$_{C2}$, • • • • • • • • >

LALIST<(link)AID$_B$ | (link)AID$_{C1}$, (link)AID$_{C2}$, • • • • • • • • >

+ Enabler of AID$_B$ + validity period value

+ transfer control flag value

→ PAT<(link)AID$_B$ | (link)AID$_{C1}$, (link)AID$_{C2}$, • • • • • • • • >

[0311] In the operation for setting the validity period value, in order to permit the setting of the validity period value only to a user who holds both the holder AID and the corresponding Enabler, the following operation is defined.

$$PAT<(link)AID_A \mid (link)AID_B> + \text{Enabler of } AID_A$$

$$+ \text{validity period value}$$

$$\rightarrow PAT<(link)AID_A \mid (link)AID_B>$$

[0312] In the operation for setting the transfer control flag value, in order to permit the setting of the transfer control flag value only to a user who holds both the holder AID and the corresponding Enabler, the following operation is defined.

$$PAT<(link)AID_A \mid (link)AID_B> + \text{Enabler of } AID_A$$

$$+ \text{transfer control flag value}$$

$$\rightarrow PAT<(link)AID_A \mid (link)AID_B>$$

[0313] Next, with references to Fig. 42 to Fig. 48, the overall system configuration of this seventh embodiment will be described. In Fig. 42 to Fig. 48, the user-A who has AID_A allocated from the CA stores AID_A and Enabler of AID_A in a computer of the user-A, and the input/output devices such as floppy disk drive, CD-ROM drive, communication board, microphone, speaker, etc., are connected. Else, AID_A and Enabler of AID_A are stored in a communication terminal (telephone, cellular phone, etc.) which has a storage device and a data input/output function.

[0314] Similarly, the user-B who has AID_B allocated from the CA stores AID_B and Enabler of AID_B in a computer of the user-B, and the input/output devices such as floppy disk drive, CD-ROM drive, communication board, microphone, speaker, etc., are connected. Else, AID_B and Enabler of AID_B are stored in a communication terminal (telephone, cellular phone, etc.) which has a storage device and a data input/output function.

[0315] In the following, a procedure by which the user-A generates $PAT<(link)AID_A \mid (link)AID_B>$ will be described.

(1) The user-A acquires AID_B and Enabler of AID_B using any of the following means.

- AID_B and Enabler of AID_B are registered at the ADS 7, and it is waited until the user-A acquires them as a search result (Fig. 42).
- AID_B and Enabler of AID_B are directly transmitted to the user-A by the email, signaling, etc. (Figs. 43, 44).
- AID_B and Enabler of AID_B are stored in a magnetic, optic, or electronic medium such as floppy disk, CD-ROM, MO, IC card, etc., and this medium is given to the user-A. Else, it is waited until the user acquires them by reading this medium (Figs. 45, 46).
- AID_B and Enabler of AID_B are printed on a paper medium such as book, name card, etc., and this medium is given to the user-A. Else, it is waited until the user-A acquire them by reading this medium (Figs. 47, 48).

(2) The user-A who has acquired AID_B and Enabler of AID_B by any of the means described in the above (1) issues the MakePAT command to the PAT processing device. This procedure is common to Fig. 42 to Fig. 48, and defined as follows.

(a) The user A requests the issuance of the MakePAT command by setting AID_A, Enabler of AID_A, AID_B, Enabler of AID_B, the validity period value, and the transfer control flag value into the communication terminal of the user-A.
(b) The communication terminal of the user-A generates the MakePAT command.
(c) The communication terminal of the user-A transmits the generated MakePAT command to the PAT processing device by means such as the email, signaling, etc. (the issuance of the MakePAT command).
(d) The PAT processing device generates $PAT<(link)AID_A \mid (link)AID_B>$ by processing the received MakePAT command according to Fig. 21 and Fig. 49. More specifically, this is done as follows.

$$(link)AID_A + (link)AID_B$$

$$+ \text{Enabler of } AID_B + \text{Enabler of } AID_A$$

$$\rightarrow LALIST<(link)AID_A \mid (link)AID_B>$$

$$LALIST<(link)AID_A \mid (link)AID_B> + \text{Enabler of } AID_A$$

$$+ \text{validity period value} + \text{transfer control flag value}$$

$$\rightarrow PAT<(link)AID_A \mid (link)AID_B>$$

(e) The PAT processing device transmits the generated $PAT<(link)AID_A \mid (link)AID_B>$ to the communication terminal of the user-A, or to the communication terminal of the user-B according to the need, by means such as the email, signaling, etc.
(f) The communication terminal of the user-A (or the user-B) stores the received $PAT<(link)AID_A \mid (link)AID_B>$ in the storage device of the communication terminal of the user-A.

[0316] The merging of PATs (MergePAT, Fig. 21, Fig. 49), the splitting of a PAT (SplitPAT, Fig. 22, Fig. 49), and the changing of a holder of a PAT (TransPAT, Fig. 21, Fig. 49) are also carried out by the similar procedure.

[0317] The procedure of MakePAT, MergePAT and TransPAT is similar to that described above with reference to Fig. 21, except that the AID should be replaced by the link information of the AID and the AID list should be replaced by the link specifying AID list. Also, the procedure of SplitPAT is similar to that described above with reference to Fig. 22, except that the AID should be replaced by the link information of the AID and the AID list should be replaced by the link specifying AID list.

[0318] Here, in the procedures of Fig. 21 and Fig. 22, the link specifying AID list generation is carried out according to Fig. 49 as follows. Namely, a buffer length is determined first (step S9011) and a buffer is generated (step S9012). Then, the link information of the holder AID is copied to a vacant region of the generated buffer (step S9017). Then, the link information of the member AID is copied to a vacant region of the resulting buffer (step S9018), and if the next member AID exists (step S9015 YES), the step S9018 is repeated.

[0319] Next, the determination of the link information of the holder AID will be described. Each of the MakePAT, the MergePAT, the SplitPAT, and the TransPAT commands is defined to have two or more arguments, where AID, PAT, or Enabler can be specified as an argument. In this case, the PAT processing device specifies the link information of the holder AID of the PAT to be outputted after executing each command according to the following rules.

- Case of the MakePAT:

  For the MakePAT command, it is defined that AIDs are to be specified for the first argument to the N-th argument(N = 2, 3, • • • • • • • •) and Enablers are to be specified for the N+1-th and subsequent arguments. For example, they can be specified as follows.

  MakePAT AID_1, AID_2, • • • • • • • •, AID_N, Enabler of AID_1, Enabler of AID_2, • • • • • • • •, Enabler of AID_N

  The PAT processing device interprets the link information of AID of the first argument of the MakePAT command as the link information the holder AID.

  Only when one of the Enablers of the N+1-th and subsequent arguments corresponds to the AID of the first argument, the PAT processing device specifies the link information of this AID (that is the link information of the AID of the first argument) as the link information of the holder AID of the PAT to be outputted after executing the MakePAT command.

- Case of the MergePAT:

  For the MergePAT command, it is defined that PATs are to be specified for the first argument to the N-th argument (N = 2, 3, • • • • • • • •) and Enabler is to be specified for the N+1-th argument. Namely, they can be specified as follows.

  MergePAT PAT_1 PAT_2 • • • • • • • • PAT_N Enabler of AID

  The PAT processing device interprets the link information of the holder AID of the PAT of the first argument of the MergePAT command as the link information of the holder AID of the PAT to be outputted after executing the MergePAT command.

  Only when the Enabler of the N+1-th argument corresponds to the holder AID of the PAT of the first argument, the PAT processing device specifies the link information of this AID (that is the link information of the holder AID of the PAT of the first argument) as the link information of the holder AID of the PAT to be outputted after executing the MergePAT command.

- Case of the SplitPAT:

  For the SplitPAT command, it is defined that PAT is to be specified for the first argument, a set of one or more AIDs grouped together by some prescribed symbols (assumed to be parentheses () in this example) are to be specified for the second argument to the N-th argument (N = 3, 4, • • • • • • • •), and Enabler is to be specified for the N+1-th argument. Namely, they can be specified as follows.

  SplitPAT PAT_1 (AID_{11}) (AID_{21} AID_{22}) • • • • • • • • (AID_{N1} AID_{N2} • • • • • • • • AID_{NM}) Enabler of AID

  The PAT processing device interprets the link information of the holder AID of the PAT of the first argument of the SplitPAT command as the link information of the holder AID of the PAT to be outputted after executing the SplitPAT command.

  Only when the Enabler of the N+1-th argument corresponds to the holder AID of the PAT of the first argument, the PAT processing device specifies the link information of this AID (that is the link information of the holder AID of the PAT of the first argument) as the link information of the holder AID of the PAT to be outputted after executing the SplitPAT command.

- Case of the TransPAT:

  For the TransPAT command, it is defined that PATs are to be specified for the first argument and the second argument, an AID is to be specified for the third argument, and Enablers are to be specified for the fourth argument and the fifth argument. Namely, they can be specified as follows.

  TransPAT PAT_1 PAT_2 AID Enabler of AID_1 Enabler of AID_2

  The PAT processing device interprets the link information of AID of the third argument as the link information of the holder AID of the PAT to be outputted after executing the TransPAT command provided that the link information of AID of the third argument of the TransPAT command is contained in the PAT of the second argument.

Only when the Enabler of the fourth argument corresponds to both the PAT of the first argument and the PAT of the second argument and the Enabler of the fifth argument corresponds to the AID of the third argument, the PAT processing device specifies the link information of the AID of the third argument as the link information of the holder AID of the PAT to be outputted after executing the TransPAT command.

Next, the determination of the link informations of the member AIDs will be described. The definitions of the MakePAT, the MergePAT, the SplitPAT, and the TransPAT commands are as described above. The PAT processing device specifies the link informations of the member AIDs of the PAT to be outputted after executing each command according to the following rules.

- Case of the MakePAT:

  Only when the link information of the holder AID of the PAT to be outputted after executing the MakePAT command is formally determined, the PAT processing device interprets all the link informations of the AIDs of the second and subsequent arguments of the MakePAT command as the link informations of the member AIDs of the PAT to be outputted after executing the MakePAT command.

  The PAT processing device specifies only the link informations of those AIDs among all the AIDs of the second and subsequent arguments which correspond to the Enablers specified by the N+1-th and subsequent arguments as the link informations of the member AIDs of the PAT to be outputted after executing the MakePAT command.

- Case of the MergePAT:

  Only when the link information of the holder AID of the PAT to be outputted after executing the MergePAT command is formally determined, the PAT processing device specifies the link informations of the member AIDs of all the PATs specified by the first to N-th arguments of the MergePAT as the link informations of the member AIDs of the PAT to be outputted after executing the MergePAT command.

- Case of the SplitPAT:

  Only when the link information of the holder AID of the PAT to be outputted after executing the SplitPAT command is formally determined, the PAT processing device specifies the link information of the member AID of the PAT specified by the first argument of the SplitPAT command as the link information of the member AID of the PAT to be outputted after executing the SplitPAT command. At this point, the link informations of the member AIDs are distributed into different PATs in units of parentheses (). For example, in the case of:

  SplitPAT PAT (AID_{11}) (AID_{21} AID_{22}) • • • • • • • • (AID_{N1} AID_{N2} • • • • • • • • AID_{NM}) Enabler of AID

  the link informations of (AID_{11}), (AID_{21} AID_{22}) and (AID_{N1} AID_{N2} • • • • • • • • AID_{NM}) will be the link informations of the member AIDs of different PATs having a common link information of holder AID.

- Case of TransPAT:

  Only when the link information of the holder AID of the PAT to be outputted after executing the TransPAT command is formally determined, the PAT processing device specifies all the link informations of the member AIDs remaining after excluding the link information of the member AID that is scheduled to be a new holder AID from all the link informations of the member AIDs of the PAT specified by the first argument of the TransPAT command and the link informations of the member AIDs of the PAT specified by the second argument as the link informations of the member AIDs of the PAT to be outputted after executing the TransPAT command.

  The verification of the properness of the Enabler in this seventh embodiment is the same as described above with reference to Fig. 24. Also, this verification of the properness of the Enabler is common to the MakePAT, the MergePAT, the SplitPAT and the TransPAT.

[0320] Next, the eighth embodiment of the email access control scheme according to the present invention will be described in detail.

[0321] In this eighth embodiment, the OID is given by a real email address.

[0322] The PAT is an information comprising two or more real email addresses, the holder index, the validity period, the transfer control flag and the PAT processing device identifier (or the identifier of the PAT processing object on the network), which is signed using a secret key of the PAT processing device (or the PAT processing object on the network).

[0323] Here, one of the real email addresses is a holder email address of this PAT, where the change of the information contained in the PAT such as an addition of email address to the PAT, a deletion of email address from the PAT, a change of the validity period in the PAT, a change of the transfer control flag value in the PAT, etc., can be made by presenting the holder email address and an Enabler containing the holder email address to the PAT processing device (or the PAT processing object on the network).

[0324] On the other hand, the email addresses other than the holder email address that are contained in the PAT are all member email addresses, where a change of the information contained in the PAT cannot be made even when the member email address and an Enabler containing the member email address are presented to the PAT processing device (or the PAT processing object on the network).

[0325] The holder index is a numerical data for identifying the holder email address, which is defined to take a value 1 when the holder email address is a top email address in the email address list formed from the holder email address and the member email addresses, a value 2 when the holder email address is a second email address from the top of the email address list, or a value n when the holder email address is an n-th email address from the top of the email address list.

[0326] The transfer control flag value is defined to take either 0 or 1.

[0327] The holder email address is defined to be a real email address which is written at a position specified by the holder index in the email address list. The member email addresses are defined to be all the email addresses other than the holder email address.

[0328] The validity period is defined by any one or combination of the number of times for which the PAT is available, the absolute time (UTC) by which the PAT becomes unavailable, the absolute time (UTC) by which the PAT becomes available, and the relative time (lifetime) since the PAT becomes available until it becomes unavailable.

[0329] The identifier of the PAT processing device (or the PAT processing object on the network) is defined as a serial number of the PAT processing device (or an distinguished name of the PAT processing object on the network). The secret key of the PAT processing device (or the PAT processing object on the network) is defined to be uniquely corresponding to the identifier.

[0330] Also, in this eighth embodiment, an Enabler is defined as an identifier corresponding to the real email address. The Enabler is an information comprising a character string uniquely indicating that it is an Enabler and a real email address itself, which is signed using the secret key of the PAT processing device or the PAT processing object on the network.

[0331] The generation of the PAT in this eighth embodiment is carried out as follows.

[0332] Here, a directory will be described as an example of the PAT processing object on the network. The directory manages the real email address and the disclosed information of the user in correspondence, and outputs the PAT upon receiving the search conditions presented from an arbitrary user.

[0333] The user transmits the real email address and the search conditions to the directory. Then, the directory acquires all the real email addresses which uniquely correspond to the disclosed information that satisfies these search conditions. Then, the directory generates a real email address list from the real email address of the user who presented the search conditions and all the real email addresses acquired as a search result. Then, the directory appends the holder index value, the validity period value, the transfer control flag value, and the distinguished name of the directory to the real email address list. Finally, the directory signs the resulting data using a secret key of the directory, and transmits it as the PAT to the user who presented the search conditions.

[0334] Next, the email access control in this eighth embodiment is carried out as follows.

[0335] The sender specifies the real email address of the sender in From: line, and "[PAT]@[real domain of sender]" in To: line of a mail.

[0336] The SCS acquires an email received by an MTA (Message Transfer Agent) such as SMTP (Simple Mail Transfer Protocol), and carries out the authentication by the following procedure.

> (1) The signature of the PAT is verified using the public key of the PAT.
>
> When the PAT is found to have been altered, the email is discarded and the processing is terminated.
>
> When the PAT is found to have been not altered, the following processing (2) is executed.
>
> (2) The search is carried out by presenting the sender's real email address to the PAT.
>
> When a real email address that completely matches with the sender's real email address is not contained in the PAT, the email is discarded and the processing is terminated.
>
> When a real email address that completely matches with the sender's real email address is contained in the PAT, the following processing (3) is executed.
>
> (3) The validity period value of the PAT is evaluated.
>
> When the PAT is outside the validity period, the email is discarded and the processing is terminated.
>
> When the PAT is within the validity period, the following processing (4) is executed.
>
> (4) Whether or not to authenticate the sender is determined by referring to the transfer control flag value of the PAT.
>
> When the value is 1, the challenge/response authentication between the SCS and the sender is carried out, and the signature of the sender is verified. When the signature is valid, the recipient is specified and the PAT is attached. When the signature is invalid, the email is discarded and the processing is terminated.
>
> When the value is 0, the recipient is specified and the PAT is attached without executing the challenge/response authentication.

[0337] An exemplary challenge/response authentication between the SCS and the sender in this eight embodiment can be carried out as follows.

[0338] First, the SCS generates an arbitrary information such as a timestamp, for example, and transmits the generated information to the sender.

[0339] Then, the sender generates the secret key and the public key, signs the received information using the secret key, and transmits it along with the public key.

[0340] The SCS then verifies the signature of the received information using the public key presented from the sender. When the signature is valid, the recipient is specified and the PAT is attached. When the signature is invalid, the email is discarded and the processing is terminated.

[0341] The specifying of the recipient and the attaching of the PAT at the SCS in this eighth embodiment can be carried out as follows.

[0342] First, the SCS carries out the search by presenting the sender's real email address to the PAT, so as to acquire all the real email addresses which do not completely match the sender's real email address. Then, all these acquired real email addresses are specified as recipient's real email addresses.

[0343] Next, the SCS attaches the PAT to an arbitrary position in the email in order to transmit the PAT to all the recipient's email addresses so as to be able to realize the bidirectional communications. Finally, the SCS gives the email to the MTA.

[0344] The receiving refusal with respect to the PAT at the SCS in this eighth embodiment can be carried out as follows.

[0345] Receiving refusal setting: The bidirectional authentication is carried out by an arbitrary means between the user and the SCS 5. Then, the user transmits a registration command, his/her own real email address, and arbitrary PATs to the SCS 5. Then, the SCS 5 next verifies the signature of each received PAT using a public key of the ADS. Those PATs with the invalid signature are discarded by the SCS 5. When the signature is valid, the SCS 5 carries out the search by presenting the received real email address to each PAT. For each of those PATs which contain the real email address that completely matches with the received real email address, the SCS 5 presents the registration command and the PAT to the storage device such that the PAT is registered into the storage device. Those PATs which do not contain the real email address that completely matches with the received real email address are discarded by the SCS 5 without storing them into the storage device.

[0346] Receiving refusal execution: The SCS 5 carries out the search by presenting the PAT to the storage device. When a PAT that completely matches the presented PAT is registered in the storage device, the mail is discarded. When a PAT that completely matches the present PAT is not registered in the storage device, the mail is not discarded.

[0347] Receiving refusal cancellation: The bidirectional authentication is carried out by an arbitrary means between the user and the SCS 5. Then, the user presents his/her own real email address to the SCS 5. Then, the SCS 5 next presents the presented real email address as a search condition to the storage device and acquire all the PATs that contain the presented real email address, and then presents all the acquired PATs to the user. Then, the user selects all the PATs for which the receiving refusal is to be cancelled by referring to all the PATs presented from the SCS 5, and transmits all the selected PATs along with a deletion command to the SCS 5. Upon receiving the deletion command and all the PATs for which the receiving refusal is to be cancelled, the SCS 5 presents the deletion command and all the PATs received from the user to the storage device, such that all the received PATs are deleted from the storage device.

[0348] The editing of the PAT in this eighth embodiment can be carried out as follows.

[0349] The MakePAT, the MergePAT, the SplitPAT, and the TransPAT processings for the PAT using real email addresses as its elements can be obtained from the the MakePAT, the MergePAT, the SplitPAT, and the TransPAT processings for the PAT using AIDs as its elements described above, by replacing the AID by the real email address and the Enabler of AID by the Enabler of real email address.

[0350] A Null operator is an information comprising a data which is uniquely indicating that it is Null and which has a format of the real email address, which is signed by the secret key of the PAT processing device or the PAT processing object on the network.

[0351] Similarly, the God operator is an information comprising a data which is uniquely indicating that it is God and which has a format of the real email address, which is signed by the secret key of the PAT processing device or the PAT processing object on the network.

[0352] The Enabler of Null operator is an information comprising a data which is uniquely indicating that it is Enabler and the Null operator itself, which is signed by the secret key of the PAT processing device or the PAT processing object on the network.

[0353] The processings involving the Null operator and the God operator can be obtained from the processings for the PAT using AIDs as its elements described above, by replacing the AID by the real email address, the Enabler of AID by the Enabler of real email address, the Null-AID by the Null operator, the God-AID by the God operator, and the Enabler of Null-AID by the Enabler of Null operator.

[0354] As described, according to the present invention, a PAT is used for verifying the access right of a sender and the email access control among users is carried out when the verification result is valid, so that it becomes possible to disclose the information indicative of characteristics of a user while concealing the true identification of a user and carrying out communications appropriately according to this disclosed information while preventing conventionally possible attacks from a third person. In addition, even when a recipient receives an attack from a sender who maliciously utilizes the anonymity, damages of a recipient due to that attack can be minimized.

[0355] Also, according to the present invention, the generation and the content change of the personalized access ticket can be made by the initiative of a user by using an AID assigned to each user and an Enabler defined in correspondence to the AID, so that it becomes possible to appropriately manage information such as that of a point of contact of each member of the group communication (mailing list, etc.) which changes dynamically.

[0356] Also, according to the present invention, a Null-AID and an Enabler of Null-AID can be introduced in order to carry out the generation of a new PAT (MakePAT) and the merging of PATs (MergePAT) without giving the member AID and the Enabler of the member AID to the holder of the PAT, so that it becomes possible to prevent the pretending using the member AID.

[0357] Also, according to the present invention, the Null-AID can be used only as the holder AID of the PAT (the Null-AID cannot be used as the member AID), that is $\mathrm{PAT}<\mathrm{AID}_{\mathrm{Null}} \mid \mathrm{AID}_{\mathrm{member1}}, \mathrm{AID}_{\mathrm{member2}}, \cdots\cdots\cdots, \mathrm{AID}_{\mathrm{memberN}} >$ is allowed, but $\mathrm{PAT}<\mathrm{AID}_{\mathrm{holder}} \mid \mathrm{AID}_{\mathrm{Null}}, \mathrm{AID}_{\mathrm{member1}}, \mathrm{AID}_{\mathrm{member2}}, \cdots\cdots\cdots, \mathrm{AID}_{\mathrm{memberN}} >$ is not allowed, so that the holder of $\mathrm{PAT}<\mathrm{AID}_{\mathrm{holder}} \mid \mathrm{AID}_{\mathrm{member}} >$ cannot produce $\mathrm{PAT}<\mathrm{AID}_{\mathrm{Null}} \mid \mathrm{AID}_{\mathrm{member}} >$ from this $\mathrm{PAT}<\mathrm{AID}_{\mathrm{holder}} \mid \mathrm{AID}_{\mathrm{member}} >$ as long as the holder does not know Enabler of $\mathrm{AID}_{\mathrm{member}}$.

[0358] Also, according to the present invention, a God-AID can be introduced in order to set up a read only attribute to the PAT, so that it becomes possible to fix the participants in the group communication.

[0359] Also, according to the present invention, the link information for uniquely specifying the AID can be introduced and the PAT can be given in terms of the link information such that the PAT does not contain the AID itself, so that it becomes possible to realize the receiving refusal function without using the AID itself.

[0360] It is to be noted that, besides those already mentioned above, many modifications and variations of the above embodiments may be made without departing from the novel and advantageous features of the present invention. Accordingly, all such modifications and variations are intended to be included within the scope of the appended claims.

Claims

1. A method of email access control, comprising the steps of:

   receiving a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, at a secure communication service for connecting communications between the sender and the receiver; and

   controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket at the secure communication service.

2. The method of claim 1, wherein at the controlling step the secure communication service authenticates the personalized access ticket presented by the sender, and refuses a delivery of the email when the personalized access ticket presented by the sender has been altered.

3. The method of claim 2, wherein the personalized access ticket is signed by a secret key of a secure processing device which issued the personalized access ticket, and at the controlling step the secure communication service authenticates the personalized access ticket by verifying a signature of the secure processing device in the personalized access ticket using a public key of the secure processing device.

4. The method of claim 1, wherein at the receiving step the secure communication service also receives the sender's identification presented by the sender along with the personalized access ticket, and at the controlling step the secure communication service checks whether the sender's identification presented by the sender is contained in the personalized access ticket presented by the sender, and refuses a delivery of the email when the sender's identification presented by the sender is not contained in the personalized access ticket presented by the sender.

5. The method of claim 1, wherein the personalized access ticket also contains a validity period indicating a period for which the personalized access ticket is valid, and at the controlling step the secure communication service checks the validity period contained in the personalized access ticket presented by the sender and refuses a delivery of the email when the personalized access ticket presented by the sender contains the validity period that has already been expired.

6. The method of claim 5, wherein the validity period of the personalized access ticket is set by a trusted third party.

7. The method of claim 1, further comprising the step of:

   issuing the personalized access ticket to the sender at a directory service for managing an identification of each registrant and a disclosed information of each registrant which has a lower secrecy than a personal information, in a state which is accessible for search by unspecified many, in response to search conditions specified by the sender, by using an identification of a registrant whose disclosed information matches the search conditions as the recipient's identification and the sender's identification specified by the sender along with the search conditions.

8. The method of claim 1, further comprising the step of:

registering in advance the personalized access ticket containing an identification of a specific user from which a delivery of emails to a specific registrant is to be refused as the sender's identification and an identification of the specific registrant as the recipient's identification, at the secure communication service;
wherein the controlling step the secure communication service refuses a delivery of the email from the sender when the personalized access ticket presented by the sender is registered therein in advance at the registering step.

9. The method of claim 8, further comprising the step of:

deleting the personalized access ticket registered at the secure communication service upon request from the specific registrant who registered the personalized access ticket at the registering step.

10. The method of claim 1, wherein the personalized access ticket also contains a transfer control flag indicating whether or not the sender should be authenticated by the secure communication service, and at the controlling step, when the transfer control flag contained in the personalized access ticket indicates that the sender should be authenticated, the secure communication service authenticates the sender's identification presented by the sender and refuses a delivery of the email when an authentication of the sender's identification fails.

11. The method of claim 10, wherein the authentication of the sender's identification is realized by a challenge/response procedure between the sender and the secure communication service.

12. The method of claim 10, wherein the transfer control flag of the personalized access ticket is set by a trusted third party.

13. The method of claim 1, wherein the sender's identification and the recipient's identification in the personalized access ticket are given by real email addresses of the sender and the recipient.

14. The method of claim 1, wherein the sender's identification and the recipient's identification in the personalized access ticket are given by anonymous identifications of the sender and the recipient, where an anonymous identification of each user contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority.

15. The method of claim 14, wherein the anonymous identification of each user is an information containing the at least one fragment of the official identification of each user which is signed by the certification authority using a secret key of the certification authority.

16. The method of claim 14, wherein the official identification of each user is a character string uniquely assigned to each user by the certification authority and a public key of each user which are signed by a secret key of the certification authority.

17. The method of claim 14, further comprising the step of:

probabilistically identifying an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

18. The method of claim 1, wherein an anonymous identification of each user that contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority and a link information of each anonymous identification by which each anonymous identification can be uniquely identified are defined, and the sender's identification and the recipient's identification in the personalized access ticket are given by a link information of the anonymous identification of the sender and a link information of the anonymous identification of the recipient.

19. The method of claim 1, wherein the link information of each anonymous identification is an identifier uniquely assigned to each anonymous identification by the certification authority.

20. The method of claim 18, further comprising the step of:

p robabilistically identifying an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

21. The method of claim 1, wherein the personalized access ticket contains a single sender's identification and a single recipient's identification in 1-to-1 correspondence.

22. The method of claim 1, wherein the personalized access ticket contains a single sender's identification and a plurality of recipient's identifications in 1-to-N correspondence, where N is an integer greater than 1.

23. The method of claim 22, wherein one identification among the single sender's identification and the plurality of recipient's identifications is a holder identification for identifying a holder of the personalized access ticket while other identifications among the single sender's identification and the plurality of recipient's identifications are member identifications for identifying members of a group to which the holder belongs.

24. The method of claim 23, further comprising the step of:

issuing an identification of each user and an enabler of the identification of each user indicating a right to change the personalized access ticket containing the identification of each user as the holder identification, to each user at a certification authority, such that prescribed processing on the personalized access ticket can be carried out at a secure processing device only by a user who presented both the holder identification contained in the personalized access ticket and the enabler corresponding to the holder identification to the secure processing device.

25. The method of claim 24, wherein the certification authority issues the enabler of the identification of each user as an information indicating that it is the enabler and the identification of each user itself which are signed by a secret key of the certification authority.

26. The method of claim 24, wherein the prescribed processing includes a generation of a new personalized access ticket, a merging of a plurality of personalized access tickets, a splitting of one personalized access ticket into a plurality of personalized access tickets, a changing of the holder of the personalized access ticket, changing of a validity period of the personalized access ticket, and a changing of a transfer control flag of the personalized access ticket.

27. The method of claim 26, wherein a special identification and a special enabler corresponding to the special identification which are known to all users are defined such that the generation of a new personalized access ticket and the changing of the holder of the personalized access ticket can be carried out by the holder of the personalized access ticket by using the special identification and the special enabler without using an enabler of a member identification.

28. The method of claim 27, wherein the special identification is defined to be capable of being used only as the holder identification of the personalized access ticket.

29. The method of claim 26, wherein a special identification which is known to all users is defined such that a read only attribute can be set to the personalized access ticket by using the special identification.

30. The method of claim 1, wherein at the controlling step, when the access right of the sender with respect to the recipient is verified according to the personalized access ticket, the secure communication service takes out the recipient's identification from the personalized access ticket by using the sender's identification presented by the sender, converts the mail by using a taken out recipient's identification into a format that can be interpreted by a mail transfer function for actually carrying out a mail delivery processing, and gives the mail after conversion to the mail transfer function by attaching the personalized access ticket.

31. A method of email access control, comprising the steps of:

defining an official identification of each user by which each user is uniquely identifiable by a certification authority, and an anonymous identification of each user containing at least one fragment of the official identification; and
identifying each user by the anonymous identification of each user in communications for emails on a communication network.

32. The method of claim 31, wherein the anonymous identification of each user is an information containing the at least one fragment of the official identification of each user which is signed by the

certification authority using a secret key of the certification authority.

33. The method of claim 31, wherein the official identification of each user is a character string uniquely assigned to each user by the certification authority and a public key of each user which are signed by a secret key of the certification authority.

34. The method of claim 31, further comprising the steps of:

receiving a personalized access ticket containing a sender's anonymous identification and a recipient's anonymous identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, at a secure communication service for connecting communications between the sender and the receiver; and
controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket at the secure communication service.

35. The method of claim 34, further comprising the step of:

probabilistically identifying an identity of the sender at the secure communication service by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

36. The method of claim 31, wherein the defining step also defines a link information of each anonymous identification by which each anonymous identification can be uniquely identified, and each anonymous identification also contains the link information of each anonymous identification.

37. The method of claim 36, wherein the link information of each anonymous identification is an identifier uniquely assigned to each anonymous identification by the certification authority.

38. The method of claim 36, further comprising the steps of:

receiving a personalized access ticket containing a link information of a sender's anonymous identification and a link information of a recipient's anonymous identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, at a secure communication service for connecting communications between the sender and the receiver; and
controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket at the secure communication service.

39. The method of claim 38, further comprising the step of:

probabilistically identifying an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

40. A communication system realizing email access control, comprising:

a communication network to which a plurality of user terminals are connected; and
a secure communication service device for connecting communications between the sender and the receiver on the communication network, by receiving a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket.

41. The system of claim 40, wherein the secure communication service device authenticates the personalized access ticket presented by the sender, and refuses a delivery of the email when the personalized access ticket presented by the sender has been altered.

42. The system of claim 41, further comprising:

a secure processing device for issuing the personalized access ticket which is signed by a secret key of the secure processing device;
wherein the secure communication service device authenticates the personalized access ticket by verifying a signature of the secure processing device in the personalized access ticket using a public key of the secure process-

ing device.

43. The system of claim 40, wherein the secure communication service device also receives the sender's identification presented by the sender along with the personalized access ticket, checks whether the sender's identification presented by the sender is contained in the personalized access ticket presented by the sender, and refuses a delivery of the email when the sender's identification presented by the sender is not contained in the personalized access ticket presented by the sender.

44. The system of claim 40, wherein the personalized access ticket also contains a validity period indicating a period for which the personalized access ticket is valid, and the secure communication service device checks the validity period contained in the personalized access ticket presented by the sender and refuses a delivery of the email when the personalized access ticket presented by the sender contains the validity period that has already been expired.

45. The system of claim 44, further comprising:

a trusted third party for setting the validity period of the personalized access ticket.

46. The system of claim 40, further comprising:

a directory service device for managing an identification of each registrant and and a disclosed information of each registrant which has a lower secrecy than a personal information, in a state which is accessible for search by unspecified many, and issuing the personalized access ticket to the sender in response to search conditions specified by the sender, by using an identification of a registrant whose disclosed information matches the search conditions as the recipient's identification and the sender's identification specified by the sender along with the search conditions.

47. The system of claim 40, wherein the secure communication service device registers in advance the personalized access ticket containing an identification of a specific user from which a delivery of emails to a specific registrant is to be refused as the sender's identification and an identification of the specific registrant as the recipient's identification, and refuses a delivery of the email from the sender when the personalized access ticket presented by the sender is registered therein in advance.

48. The system of claim 47, wherein the secure communication service device deletes the personalized access ticket registered therein upon request from the specific registrant who registered the personalized access ticket.

49. The system of claim 40, wherein the personalized access ticket also contains a transfer control flag indicating whether or not the sender should be authenticated by the secure communication service, and when the transfer control flag contained in the personalized access ticket indicates that the sender should be authenticated, the secure communication service device authenticates the sender's identification presented by the sender and refuses a delivery of the email when an authentication of the sender's identification fails.

50. The system of claim 49, wherein the authentication of the sender's identification is realized by a challenge/response procedure between the sender and the secure communication service device.

51. The system of claim 49, further comprising a trusted third party for setting the transfer control flag of the personalized access ticket.

52. The system of claim 40, wherein the sender's identification and the recipient's identification in the personalized access ticket are given by real email addresses of the sender and the recipient.

53. The system of claim 40, further comprising:

a certification authority device for issuing an anonymous identification of each user which contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by the certification authority device;
wherein the sender's identification and the recipient's identification in the personalized access ticket are given by anonymous identifications of the sender and the recipient.

54. The system of claim 53, wherein the anonymous identification of each user is an information containing the at least one fragment of the official identification of each user which is signed by the certification authority device using a secret key of the certification authority device.

55. The system of claim 53, wherein the official identification of each user is a character string uniquely assigned to each user by the certification authority device and a public key of each user which are signed by a secret key of the certification authority device.

56. The system of claim 53, wherein the secure communication service device probabilistically identifies an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

57. The system of claim 40, further comprising:

a certification authority device for issuing an anonymous identification of each user which contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by the certification authority device and a link information of each anonymous identification by which each anonymous identification can be uniquely identified; wherein the sender's identification and the recipient's identification in the personalized access ticket are given by a link information of the anonymous identification of the sender and a link information of the anonymous identification of the recipient.

58. The system of claim 57, wherein the link information of each anonymous identification is an identifier uniquely assigned to each anonymous identification by the certification authority device.

59. The system of claim 57, wherein the secure communication service device probabilistically identifies an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

60. The system of claim 40, wherein the personalized access ticket contains a single sender's identification and a single recipient's identification in 1-to-1 correspondence.

61. The system of claim 40, wherein the personalized access ticket contains a single sender's identification and a plurality of recipient's identifications in 1-to-N correspondence, where N is an integer greater than 1.

62. The system of claim 61, wherein one identification among the single sender's identification and the plurality of recipient's identifications is a holder identification for identifying a holder of the personalized access ticket while other identifications among the single sender's identification and the plurality of recipient's identifications are member identifications for identifying members of a group to which the holder belongs.

63. The system of claim 62, further comprising:

a certification authority device for issuing to each user an identification of each user and an enabler of the identification of each user indicating a right to change the personalized access ticket containing the identification of each user as the holder identification; and
a secure processing device at which prescribed processing on the personalized access ticket can be carried out only by a user who presented both the holder identification contained in the personalized access ticket and the enabler corresponding to the holder identification to the secure processing device.

64. The system of claim 63, wherein the certification authority device issues the enabler of the identification of each user as an information indicating that it is the enabler and the identification of each user itself which are signed by a secret key of the certification authority device.

65. The system of claim 63, wherein the prescribed processing includes a generation of a new personalized access ticket, a merging of a plurality of personalized access tickets, a splitting of one personalized access ticket into a plurality of personalized access tickets, a changing of the holder of the personalized access ticket, changing of a validity period of the personalized access ticket, and a changing of a transfer control flag of the personalized access ticket.

66. The system of claim 65, wherein a special identification and a special enabler corresponding to the special identification which are known to all users are defined such that the generation of a new personalized access ticket and the changing of the holder of the personalized access ticket can be carried out by the holder of the personalized access ticket by using the special identification and the special enabler without using an enabler of a member identification.

67. The system of claim 66, wherein the special identification is defined to be capable of being used only as the holder identification of the personalized access ticket.

68. The system of claim 65, wherein a special identification which is known to all users is defined such that a read only attribute can be set to the personalized access ticket by using the special identification.

69. The system of claim 40, wherein when the access right of the sender with respect to the recipient is

verified according to the personalized access ticket, the secure communication service device takes out the recipient's identification from the personalized access ticket by using the sender's identification presented by the sender, converts the mail by using a taken out recipient's identification into a format that can be interpreted by a mail transfer function for actually carrying out a mail delivery processing, and gives the mail after conversion to the mail transfer function by attaching the personalized access ticket.

70. A communication system realizing email access control, comprising:

a certification authority device for defining an official identification of each user by which each user is uniquely identifiable by the certification authority device, and an anonymous identification of each user which contains at least one fragment of the official identification; and

a communication network on which each user is identified by the anonymous identification of each user in communications for emails on the communication network.

71. The system of claim 70, wherein the anonymous identification of each user is an information containing the at least one fragment of the official identification of each user which is signed by the certification authority device using a secret key of the certification authority device.

72. The system of claim 70, wherein the official identification of each user is a character string uniquely assigned to each user by the certification authority device and a public key of each user which are signed by a secret key of the certification authority device.

73. The system of claim 70, further comprising:

a secure communication service device for connecting communications between the sender and the receiver on the communication network, by receiving a personalized access ticket containing a sender's anonymous identification and a recipient's anonymous identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket.

74. The system of claim 73, wherein the secure communication service device probabilistically identifies an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

75. The system of claim 70, wherein the certification authority device also defines a link information of each anonymous identification by which each anonymous identification can be uniquely identified, and each anonymous identification also contains the link information of each anonymous identification.

76. The system of claim 75, wherein the link information of each anonymous identification is an identifier uniquely assigned to each anonymous identification by the certification authority device.

77. The system of claim 75, further comprising:

a secure communication service device for connecting communications between the sender and the receiver on the communication network, by receiving a personalized access ticket containing a link information of a sender's anonymous identification and a link information of a recipient's anonymous identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket.

78. The system of claim 77, wherein the secure communication service device probabilistically identifies an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of link informations of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

79. A secure communication service device for use in a communication system realizing email access control, comprising:

a computer hardware; and

a computer software for causing the computer hardware to connect communications between the sender and the receiver, by receiving a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket.

80. The secure communication service device of claim 79,
wherein the computer software causes the computer hardware to authenticate the personalized access ticket presented by the sender, and refuse a delivery of the email when the personalized access ticket presented by the sender has been altered.

81. The secure communication service device of claim 80,
wherein the personalized access ticket is signed by a secret key of a secure processing device which issued the personalized access ticket, and the computer software causes the computer hardware to authenticate the personalized access ticket by verifying a signature of the secure processing device in the personalized access ticket using a public key of the secure processing device.

82. The secure communication service device of claim 79,
wherein the computer software causes the computer hardware to also receive the sender's identification presented by the sender along with the personalized access ticket, check whether the sender's identification presented by the sender is contained in the personalized access ticket presented by the sender, and refuse a delivery of the email when the sender's identification presented by the sender is not contained in the personalized access ticket presented by the sender.

83. The secure communication service device of claim 79,
wherein the personalized access ticket also contains a validity period indicating a period for which the personalized access ticket is valid, and the computer software causes the computer hardware to check the validity period contained in the personalized access ticket presented by the sender and refuse a delivery of the email when the personalized access ticket presented by the sender contains the validity period that has already been expired.

84. The secure communication service device of claim 79,
wherein the computer software causes the computer hardware to register in advance the personalized access ticket containing an identification of a specific user from which a delivery of emails to a specific registrant is to be refused as the sender's identification and an identification of the specific registrant as the recipient's identification, at the secure communication service device, and refuse a delivery of the email from the sender when the personalized access ticket presented by the sender is registered at the secure communication service device in advance.

85. The secure communication service device of claim 84,
wherein the computer software causes the computer hardware to delete the personalized access ticket registered at the secure communication service device upon request from the specific registrant who registered the personalized access ticket.

86. The secure communication service device of claim 79,
wherein the personalized access ticket also contains a transfer control flag indicating whether or not the sender should be authenticated by the secure communication service device, and when the transfer control flag contained in the personalized access ticket indicates that the sender should be authenticated, the computer software causes the computer hardware to authenticate the sender's identification presented by the sender and refuse a delivery of the email when an authentication of the sender's identification fails.

87. The secure communication service device of claim 86,
wherein the computer software causes the computer hardware to realize the authentication of the sender's identification by a challenge/response procedure between the sender and the secure communication service device.

88. The secure communication service device of claim 79,
wherein the sender's identification and the recipient's identification in the personalized access ticket are given by anonymous identifications of the sender and the recipient, where an anonymous identification of each user contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority, and the computer software also causes the computer hardware to probabilistically identify an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

89. The secure communication service device of claim 79, wherein an anonymous identification of each user that contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority and a link information of each anonymous identification by which each anonymous identification can be uniquely identified are defined, the sender's identification and the recipient's identification in the personalized access ticket are given by a link information of the anonymous identification of the sender and a link information of the anonymous identification of the recipient, and the computer software also causes the computer hardware to probabilistically identify an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

90. The secure communication service device of claim 79,
wherein when the access right of the sender with respect to the recipient is verified according to the personalized access ticket, the computer software causes the computer hardware to take out the recipient's identification from the personalized access ticket by using the sender's identification presented by the sender, convert the mail by using a taken out recipient's identification into a format that can be interpreted by a mail transfer function for actually carrying out a mail delivery processing, and give the mail after conversion to the mail transfer function by attaching the personalized access ticket.

91. A secure processing device for use in a communication system realizing email access control, comprising:

a computer hardware; and
a computer software for causing the computer hardware to receive a request for a personalized access ticket from a user, and issue a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is signed by a secret key of the secure processing device.

92. A directory service device for use in a communication system realizing email access control, comprising:

a computer hardware; and
a computer software for causing the computer hardware to manage an identification of each registrant and a disclosed information of each registrant which has a lower secrecy than a personal information, in a state which is accessible for search by unspecified many, and issue a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, to the sender in response to search conditions specified by the sender, by using an identification of a registrant whose disclosed information matches the search conditions as the recipient's identification and the sender's identification specified by the sender along with the search conditions.

93. A certification authority device for use in a communication system realizing email access control, comprising:

a computer hardware; and
a computer software for causing the computer hardware to issue to each user an official identification of each user by which each user is uniquely identifiable by the certification authority device, and an anonymous identification of each user which contains at least one fragment of the official identification.

94. A certification authority device for use in a communication system realizing email access control, comprising:

a computer hardware; and
a computer software for causing the computer hardware to issue to each user an identification of each user and an enabler of the identification of each user indicating a right to change any personalized access ticket that contains the identification of each user as a holder identification, where the persnalized access ticket generally contains a sender's identification and a plurality of recipient's identifications in correspondence, and one of the sender's identification and the recipient's identifications is a holder identification.

95. A secure processing device for use in a communication system realizing email access control, comprising:

a computer hardware; and
a computer software for causing the computer hardware to receive from a user a request for prescribed processing on a personalized access ticket containing a sender's identification and a plurality of recipient's identifications in correspondence, where one of the sender's identification and the recipient's identifications is a holder identification, and execute the prescribed processing on the personalized access ticket when the user presented both the holder

identification contained in the personalized access ticket and an enabler corresponding to the holder identification which indicates a right to change the personalized access ticket containing the identification of the user as the holder identification.

96. A computer usable medium having computer readable program code means embodied therein for causing a computer to function as a secure communication service device for use in a communication system realizing email access control, the computer readable program code means includes:

first computer readable program code means for causing said computer to receive a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email; and
second computer readable program code means for causing said computer to control accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket, so as to connect communications between the sender and the receiver on the communication network.

97. The computer usable medium of claim 96, the second computer readable program code means causes said computer to authenticate the personalized access ticket presented by the sender, and refuse a delivery of the email when the personalized access ticket presented by the sender has been altered.

98. The computer usable medium of claim 97, wherein the personalized access ticket is signed by a secret key of a secure processing device which issued the personalized access ticket, and the second computer readable program code means causes said computer to authenticate the personalized access ticket by verifying a signature of the secure processing device in the personalized access ticket using a public key of the secure processing device.

99. The computer usable medium of claim 96, wherein the first computer readable program code means causes said computer to also receive the sender's identification presented by the sender along with the personalized access ticket, and the second computer readable program code means causes said computer to check whether the sender's identification presented by the sender is contained in the personalized access ticket presented by the sender and refuse a delivery of the email when the sender's identification presented by the sender is not contained in the personalized access ticket presented by the sender.

100. The computer usable medium of claim 96, wherein the personalized access ticket also contains a validity period indicating a period for which the personalized access ticket is valid, and the second computer readable program code means causes said computer to check the validity period contained in the personalized access ticket presented by the sender and refuse a delivery of the email when the personalized access ticket presented by the sender contains the validity period that has already been expired.

101. The computer usable medium of claim 96, wherein the second computer readable program code means causes said computer to register in advance the personalized access ticket containing an identification of a specific user from which a delivery of emails to a specific registrant is to be refused as the sender's identification and an identification of the specific registrant as the recipient's identification, at the secure communication service device, and refuse a delivery of the email from the sender when the personalized access ticket presented by the sender is registered at the secure communication service device in advance.

102. The computer usable medium of claim 101, wherein the second computer readable program code means causes said computer to delete the personalized access ticket registered at the secure communication service device upon request from the specific registrant who registered the personalized access ticket.

103. The computer usable medium of claim 96, wherein the personalized access ticket also contains a transfer control flag indicating whether or not the sender should be authenticated by the secure communication service device, and when the transfer control flag contained in the personalized access ticket indicates that the sender should be authenticated, the second computer readable program code means causes said computer to authenticate the sender's identification presented by the sender and refuse a delivery of the email when an authentication of the sender's identification fails.

104. The computer usable medium of claim 103, wherein the second computer readable program code means causes said computer to realize the authentication of the sender's identification by a challenge/response procedure between the sender and the secure communication service device.

105. The computer usable medium of claim 96, wherein the sender's identification and the recipient's identification in the personalized access ticket are given by anonymous identifications of the sender and the recipient, where an anonymous identification of each user contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority, and the second computer readable program code means also causes said computer to probabilistically identify an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

106. The computer usable medium of claim 96, wherein an anonymous identification of each user that contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority and a link information of each anonymous identification by which each anonymous identification can be uniquely identified are defined, the sender's identification and the recipient's identification in the personalized access ticket are given by a link information of the anonymous identification of the sender and a link information of the anonymous identification of the recipient, and the second computer readable program code means also causes said computer to probabilistically identify an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

107. The computer usable medium of claim 96, wherein when the access right of the sender with respect to the recipient is verified according to the personalized access ticket, the second computer readable program code means causes said computer to take out the recipient's identification from the personalized access ticket by using the sender's identification presented by the sender, convert the mail by using a taken out recipient's identification into a format that can be interpreted by a mail transfer function for actually carrying out a mail delivery processing, and give the mail after conversion to the mail transfer function by attaching the personalized access ticket.

108. A computer usable medium having computer readable program code means embodied therein for causing a computer to function as a secure processing device for use in a communication system realizing email access control, the computer readable program code means includes:

first computer readable program code means for causing said computer to receive a request for a personalized access ticket from a user; and
second computer readable program code means for causing said computer to issue the personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is signed by a secret key of the secure processing device.

109. A computer usable medium having computer readable program code means embodied therein for causing a computer to function as a directory service devicer for use in a communication system realizing email access control, the computer readable program code means includes:

first computer readable program code means for causing said computer to manage an identification of each registrant and a disclosed information of each registrant which has a lower secrecy than a personal information, in a state which is accessible for search by unspecified many, and
second computer readable program code means for causing said computer to issue a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, to the sender in response to search conditions specified by the sender, by using an identification of a registrant whose disclosed information matches the search conditions as the recipient's identification and the sender's identification specified by the sender along with the search conditions.

110. A computer usable medium having computer readable program code means embodied therein for causing a computer to function as a certification authority device for use in a communication system realizing email access control, the computer readable program code means includes:

first computer readable program code means for causing said computer to issue to each user an official identification of each user by which each user is uniquely identifiable by the certification authority device; and
second computer readable program code means for causing said computer to issue to each user an anonymous identification of each user which contains at least one fragment of the official identification.

111. A computer usable medium having computer readable program code means embodied therein for causing a computer to function as a certification authority device for use in a communication system realizing email access control, the computer readable program code means includes:

first computer readable program code means for causing said computer to issue to each user an identification of each user; and
second computer readable program code means for causing said computer to issue to each user an enabler of the identification of each user indicating a right to change any personalized access ticket that contains the identification of each user as a holder identification, where the persnalized access ticket generally contains a sender's identification and a plurality of recipient's identifications in correspondence, and one of the sender's identification and the recipient's identifications is a holder identification.

112. A computer usable medium having computer readable program code means embodied therein for causing a computer to function as a secure processing device for use in a communication system realizing email access control, the computer readable program code means includes:

first computer readable program code means for causing said computer to receive from a user a request for prescribed processing on a personalized access ticket containing a sender's identification and a plurality of recipient's identifications in correspondence, where one of the sender's identification and the recipient's identifications is a holder identification; and
second computer readable program code means for causing said computer to execute the prescribed processing on the personalized access ticket when the user presented both the holder identification contained in the personalized access ticket and an enabler corresponding to the holder identification which indicates a right to change the personalized access ticket containing the identification of the user as the holder identification.

FIG.1

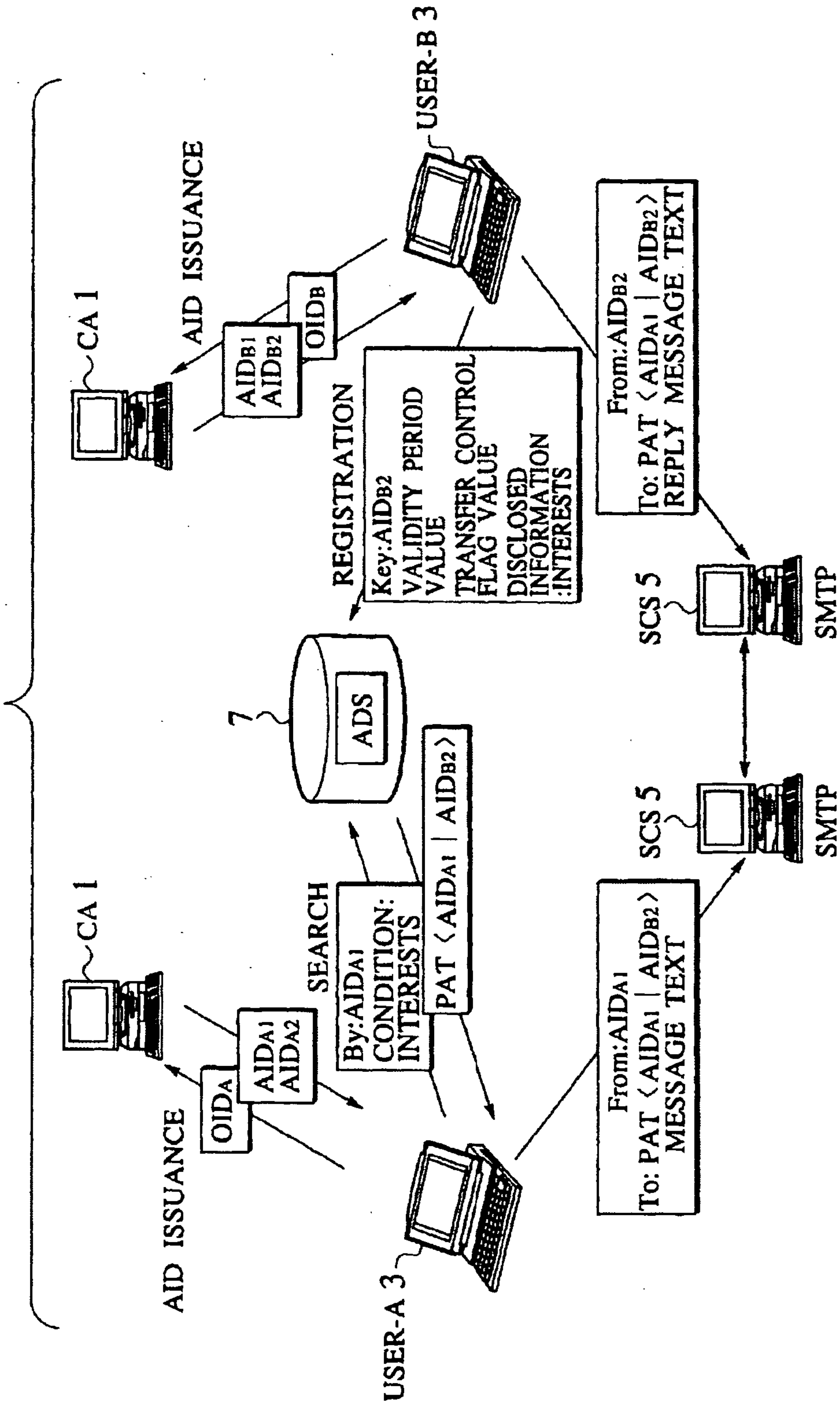

AID ISSUANCE
CA 1
OIDA
AIDA1
AIDA2
SEARCH
By:AIDA1
CONDITION:
INTERESTS
PAT ⟨AIDA1 | AIDB2⟩
USER-A 3
From:AIDA1
To: PAT ⟨AIDA1 | AIDB2⟩
MESSAGE TEXT
SCS 5
SMTP
7
ADS
REGISTRATION
Key:AIDB2
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
DISCLOSED INFORMATION :INTERESTS
SCS 5
SMTP
CA 1
AIDB1
AIDB2
OIDB
AID ISSUANCE
USER-B 3
From:AIDB2
To: PAT ⟨AIDA1 | AIDB2⟩
REPLY MESSAGE TEXT

FIG.2

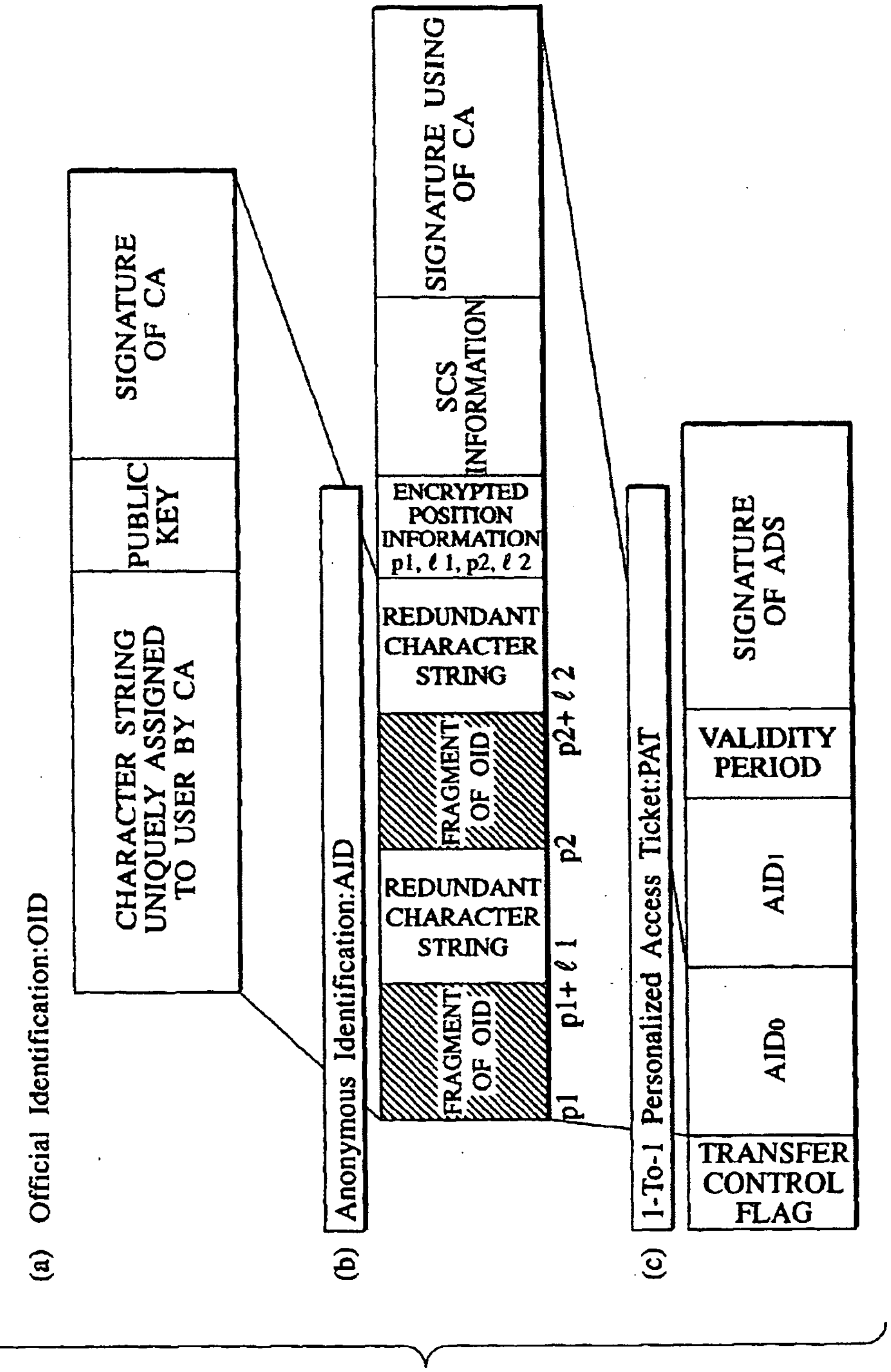

(a) Official Identification:OID
CHARACTER STRING UNIQUELY ASSIGNED TO USER BY CA
PUBLIC KEY
SIGNATURE OF CA
(b) Anonymous Identification:AID
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
ENCRYPTED POSITION INFORMATION p1, ℓ 1, p2, ℓ 2
SCS INFORMATION
SIGNATURE USING OF CA
p1
p1+ ℓ 1
p2
p2+ ℓ 2
(c) 1-To-1 Personalized Access Ticket:PAT
TRANSFER CONTROL FLAG
AID0
AID1
VALIDITY PERIOD
SIGNATURE OF ADS

FIG.3

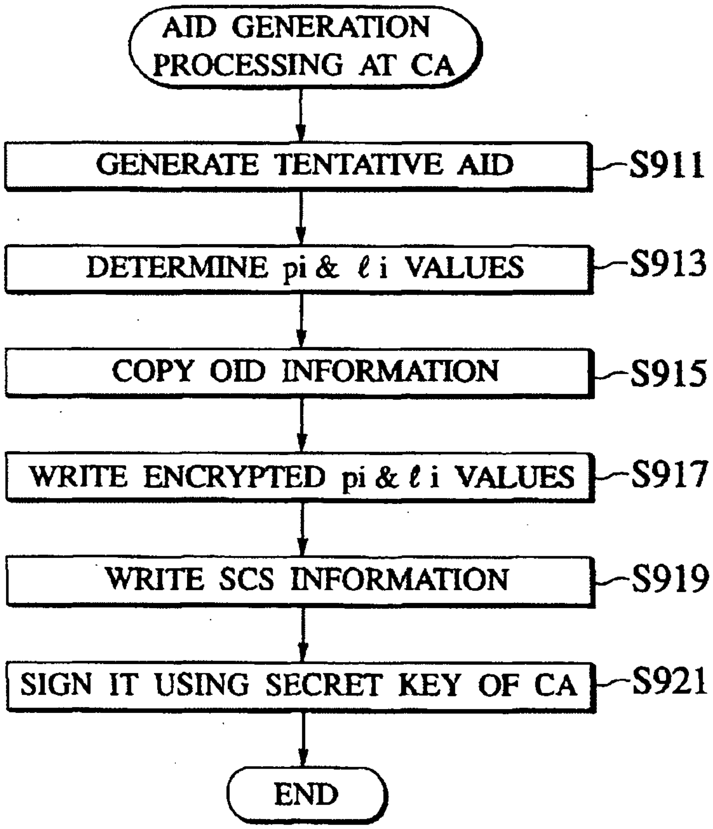

AID GENERATION PROCESSING AT CA
GENERATE TENTATIVE AID
S911
DETERMINE pi & ℓi VALUES
S913
COPY OID INFORMATION
S915
WRITE ENCRYPTED pi & ℓi VALUES
S917
WRITE SCS INFORMATION
S919
SIGN IT USING SECRET KEY OF CA
S921
END

FIG.4

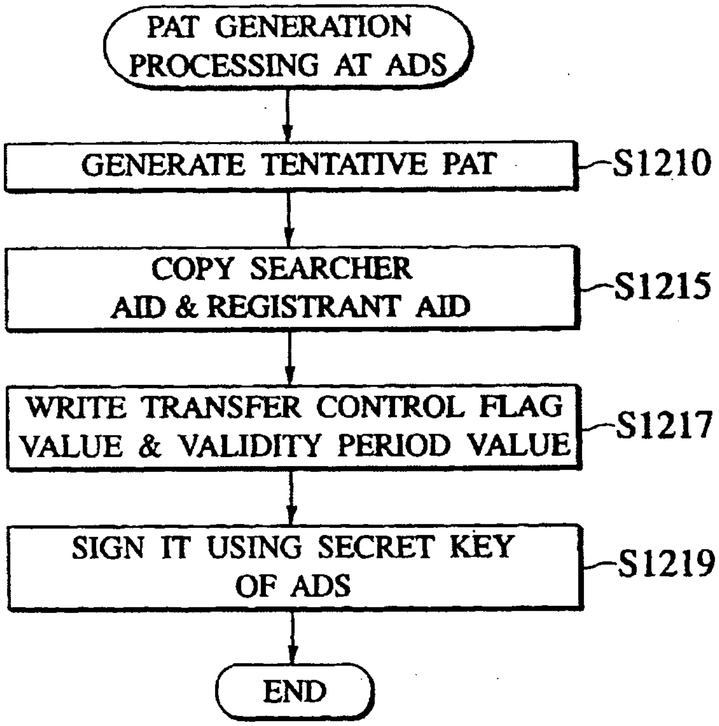

PAT GENERATION PROCESSING AT ADS
GENERATE TENTATIVE PAT
S1210
COPY SEARCHER AID & REGISTRANT AID
S1215
WRITE TRANSFER CONTROL FLAG VALUE & VALIDITY PERIOD VALUE
S1217
SIGN IT USING SECRET KEY OF ADS
S1219
END

FIG.5

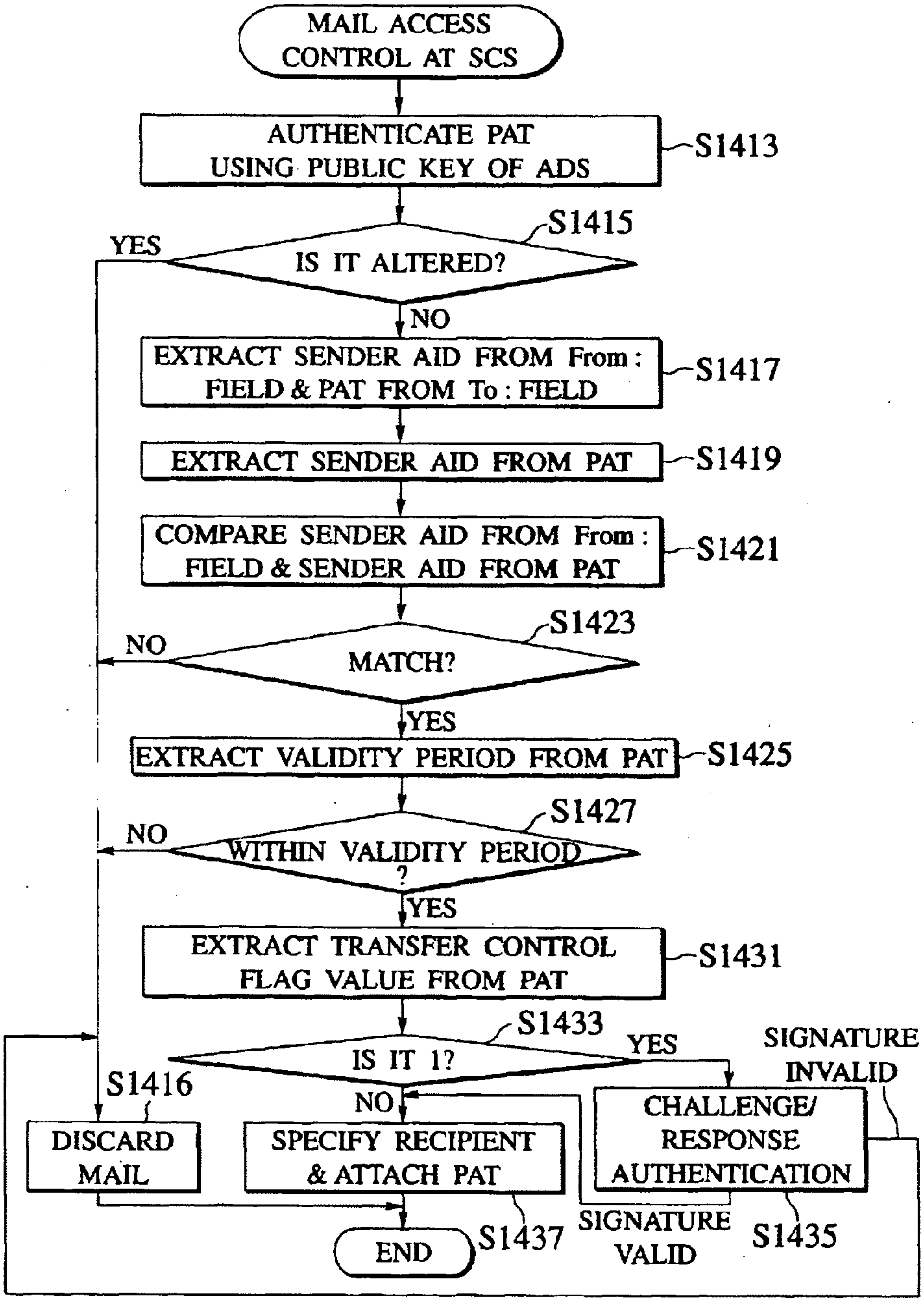

MAIL ACCESS CONTROL AT SCS
AUTHENTICATE PAT USING PUBLIC KEY OF ADS
S1413
S1415
IS IT ALTERED?
YES
NO
EXTRACT SENDER AID FROM From : FIELD & PAT FROM To : FIELD
S1417
EXTRACT SENDER AID FROM PAT
S1419
COMPARE SENDER AID FROM From : FIELD & SENDER AID FROM PAT
S1421
S1423
MATCH?
NO
YES
EXTRACT VALIDITY PERIOD FROM PAT
S1425
S1427
WITHIN VALIDITY PERIOD ?
NO
YES
EXTRACT TRANSFER CONTROL FLAG VALUE FROM PAT
S1431
S1433
IS IT 1?
YES
NO
SIGNATURE INVALID
S1416
DISCARD MAIL
SPECIFY RECIPIENT & ATTACH PAT
CHALLENGE/ RESPONSE AUTHENTICATION
END
S1437
SIGNATURE VALID
S1435

FIG.6

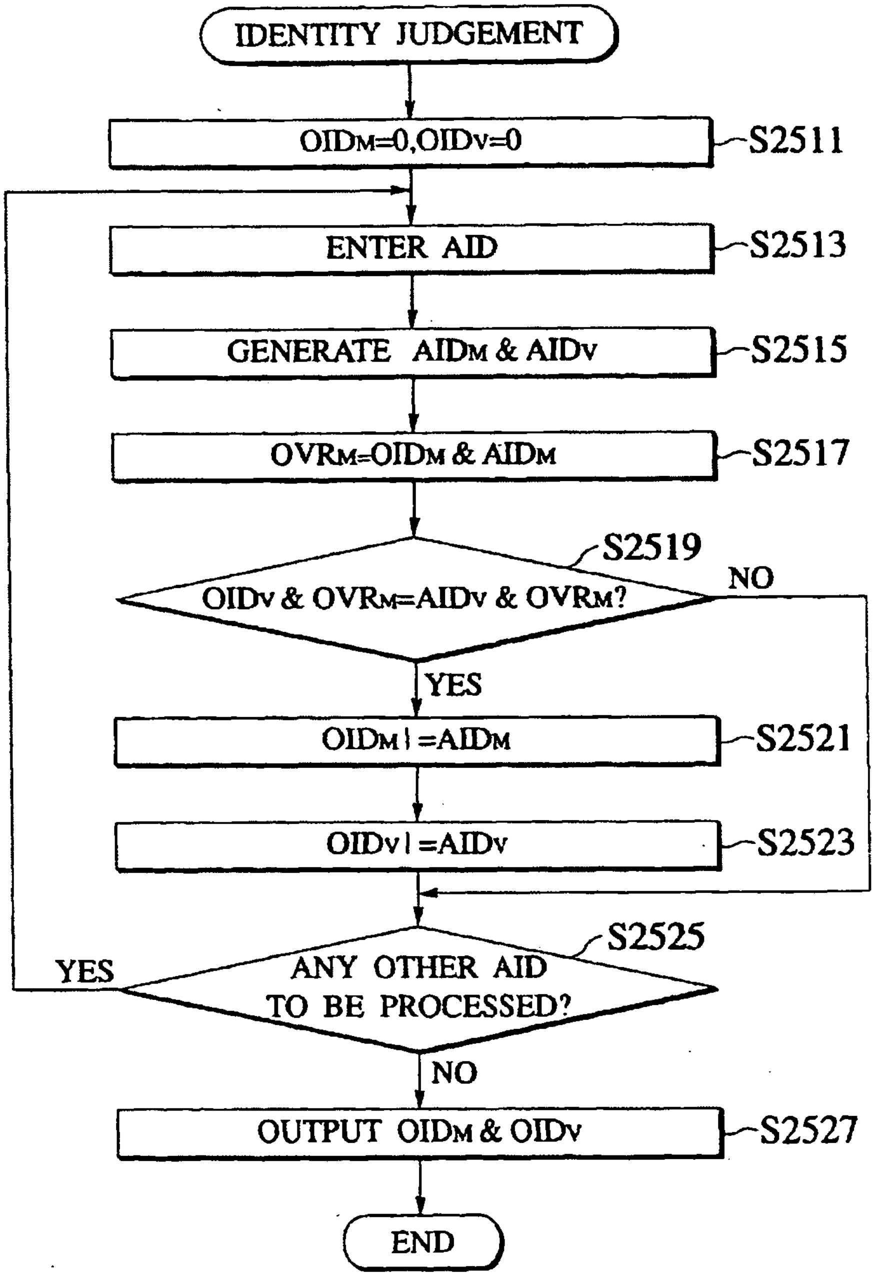

IDENTITY JUDGEMENT
OIDM=0,OIDV=0
S2511
ENTER AID
S2513
GENERATE AIDM & AIDV
S2515
OVRM=OIDM & AIDM
S2517
OIDV & OVRM=AIDV & OVRM?
S2519
NO
YES
OIDM | =AIDM
S2521
OIDV | =AIDV
S2523
ANY OTHER AID TO BE PROCESSED?
S2525
YES
NO
OUTPUT OIDM & OIDV
S2527
END

FIG.7

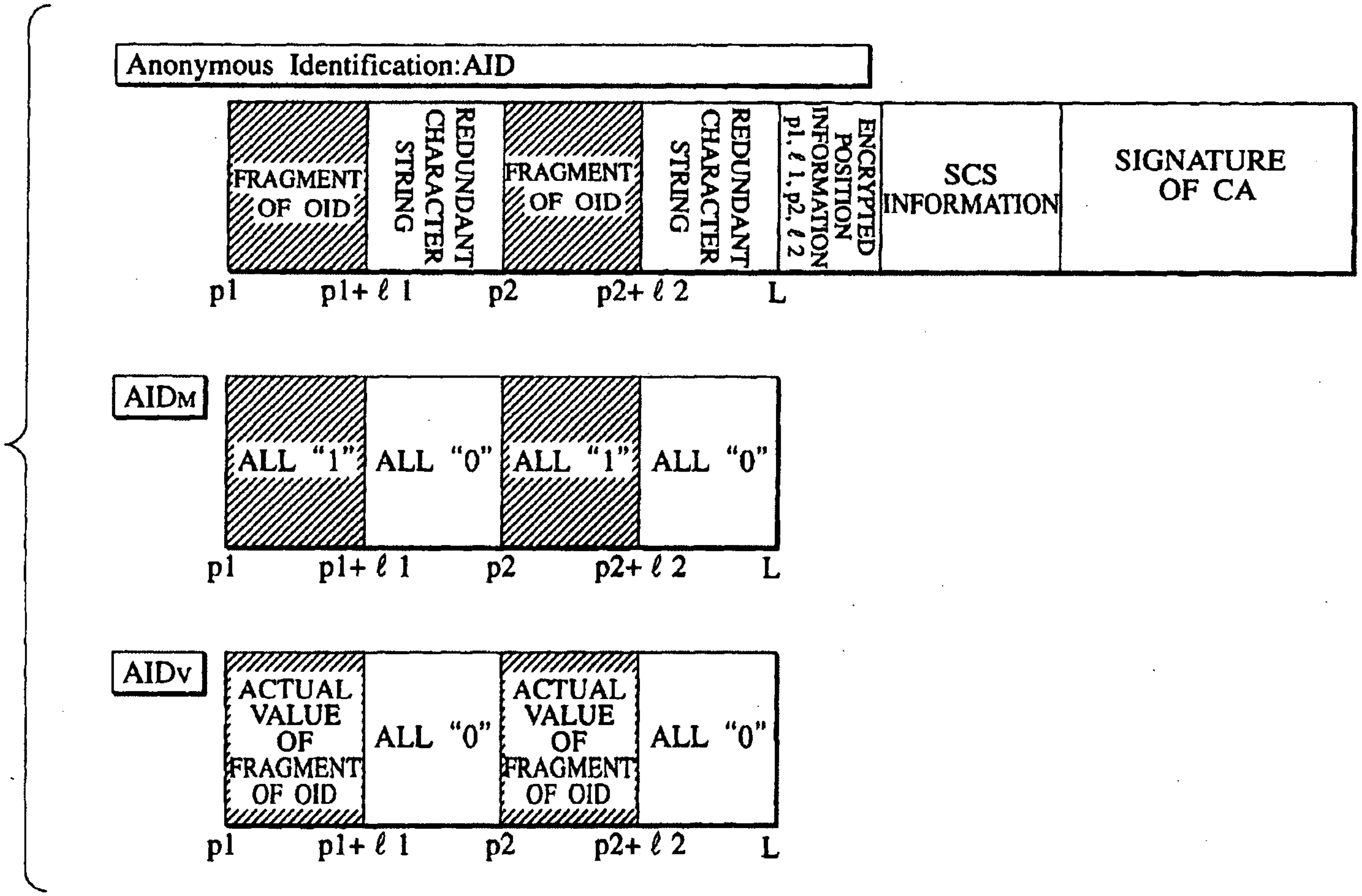

Anonymous Identification:AID
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
ENCRYPTED POSITION INFORMATION p1, ℓ 1, p2, ℓ 2
SCS INFORMATION
SIGNATURE OF CA
p1
p1+ ℓ 1
p2
p2+ ℓ 2
L
AIDM
ALL "1"
ALL "0"
ALL "1"
ALL "0"
p1
p1+ ℓ 1
p2
p2+ ℓ 2
L
AIDV
ACTUAL VALUE OF FRAGMENT OF OID
ALL "0"
ACTUAL VALUE OF FRAGMENT OF OID
ALL "0"
p1
p1+ ℓ 1
p2
p2+ ℓ 2
L

FIG.8

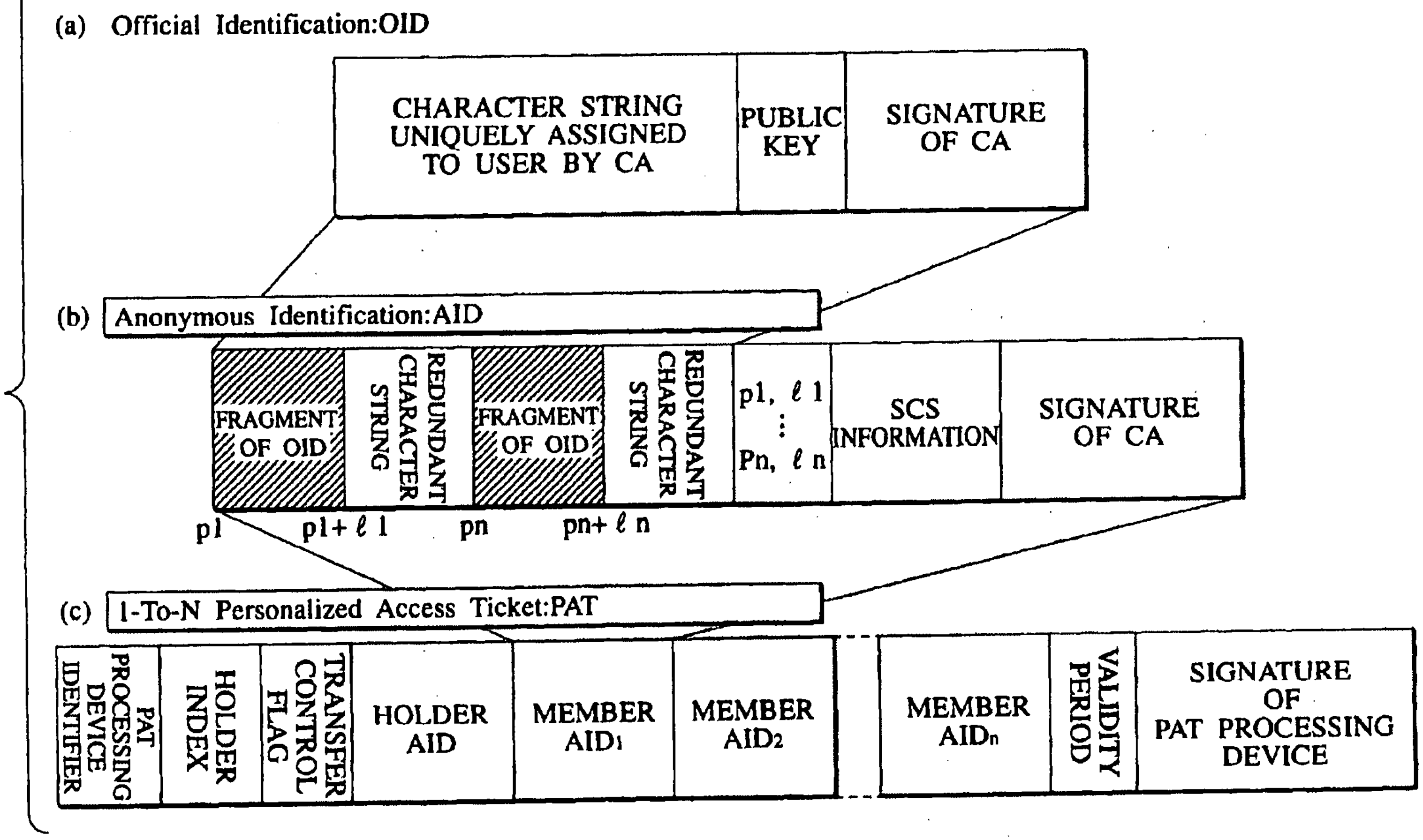

(a) Official Identification:OID
CHARACTER STRING UNIQUELY ASSIGNED TO USER BY CA
PUBLIC KEY
SIGNATURE OF CA
(b) Anonymous Identification:AID
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
p1, ℓ 1
⋮
Pn, ℓ n
SCS INFORMATION
SIGNATURE OF CA
p1
p1+ ℓ 1
pn
pn+ ℓ n
(c) 1-To-N Personalized Access Ticket:PAT
PAT PROCESSING DEVICE IDENTIFIER
HOLDER INDEX
TRANSFER CONTROL FLAG
HOLDER AID
MEMBER AID1
MEMBER AID2
MEMBER AIDn
VALIDITY PERIOD
SIGNATURE OF PAT PROCESSING DEVICE

FIG.9

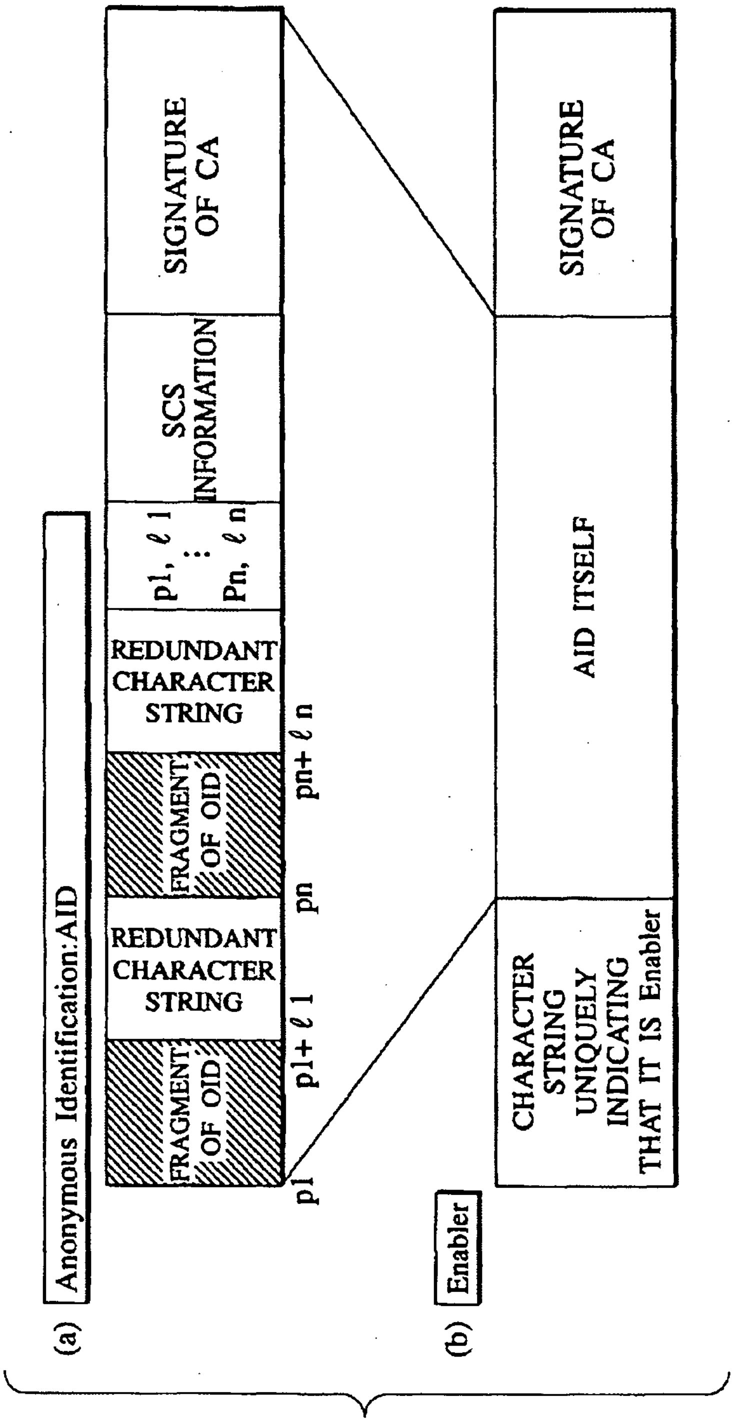

(a)
Anonymous Identification:AID
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
p1, ℓ 1
:
Pn, ℓ n
SCS INFORMATION
SIGNATURE OF CA
p1
p1+ ℓ 1
pn
pn+ ℓ n
(b)
Enabler
CHARACTER STRING UNIQUELY INDICATING THAT IT IS Enabler
AID ITSELF
SIGNATURE OF CA

FIG.10

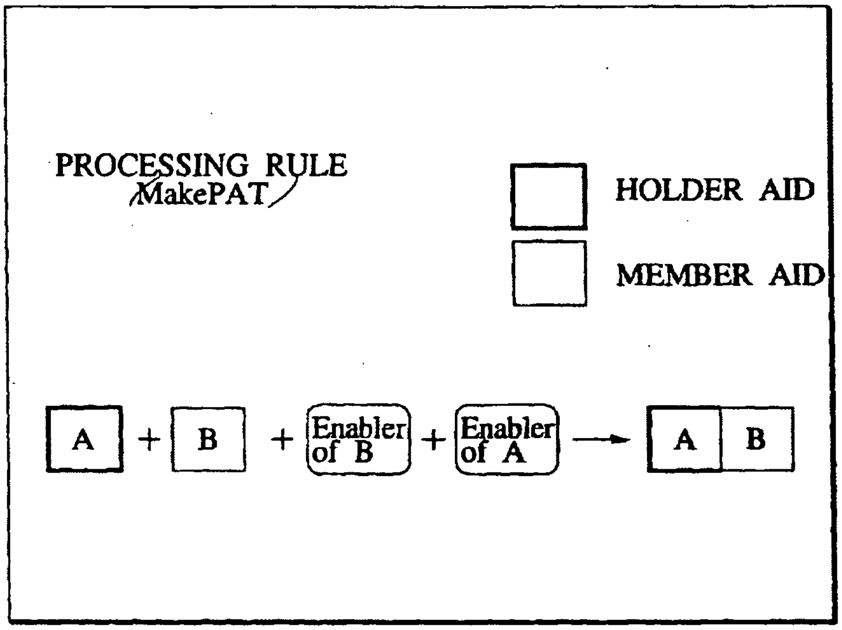

PROCESSING RULE
/MakePAT/
HOLDER AID
MEMBER AID
A
B
Enabler of B
Enabler of A
A
B

FIG.11

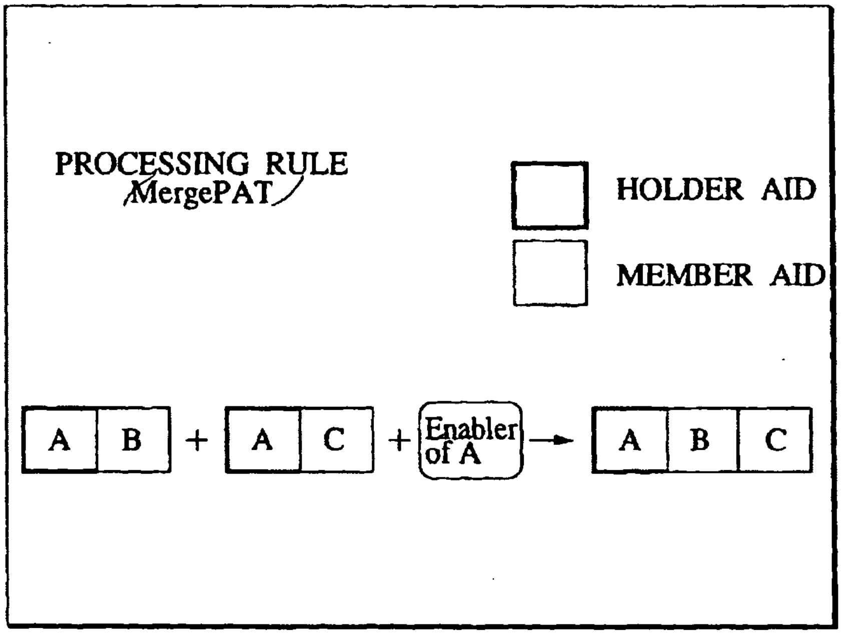

PROCESSING RULE
/MergePAT/
HOLDER AID
MEMBER AID
A
B
A
C
Enabler of A
A
B
C

FIG.12

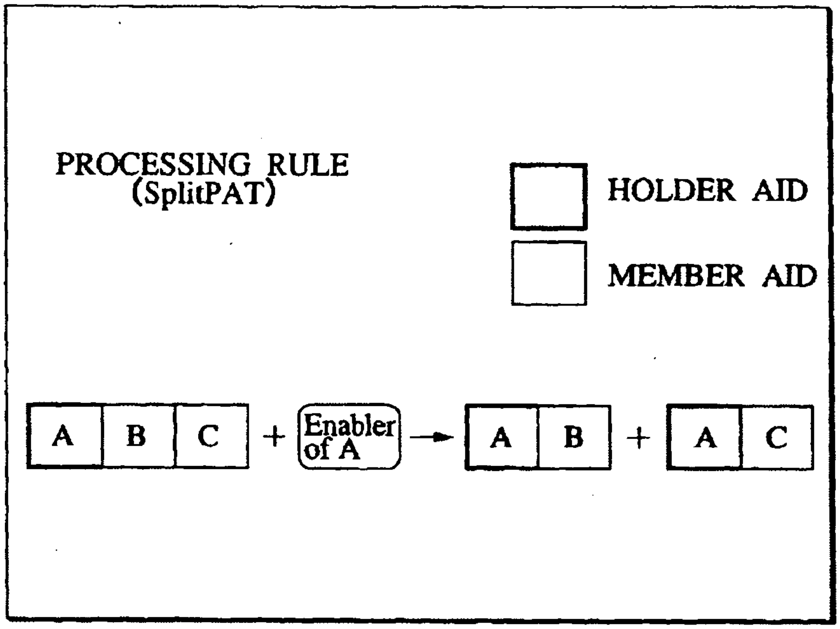

PROCESSING RULE
(SplitPAT)
HOLDER AID
MEMBER AID
A
B
C
+
Enabler of A
A
B
+
A
C

FIG.13

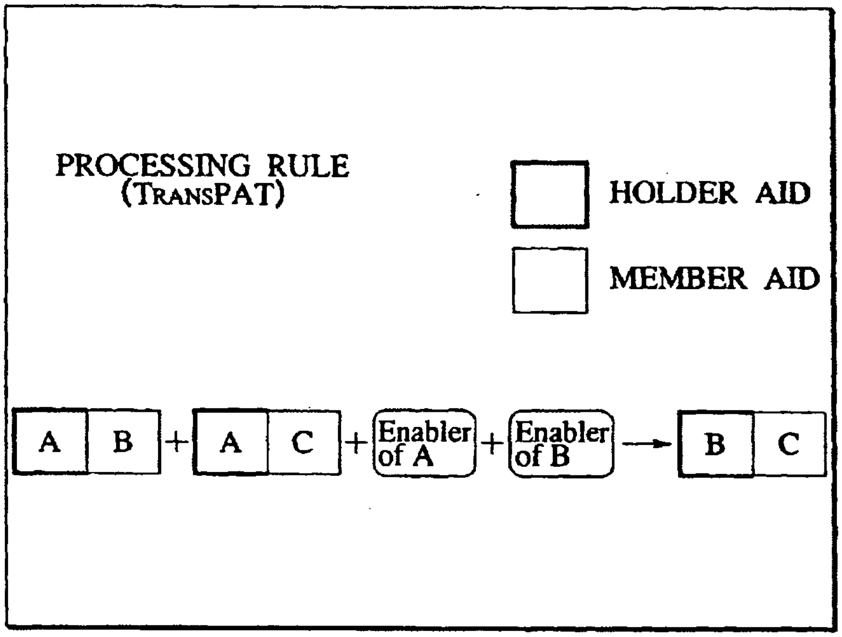

PROCESSING RULE
(TRANSPAT)
HOLDER AID
MEMBER AID
A
B
+
A
C
+
Enabler of A
+
Enabler of B
B
C

FIG.14

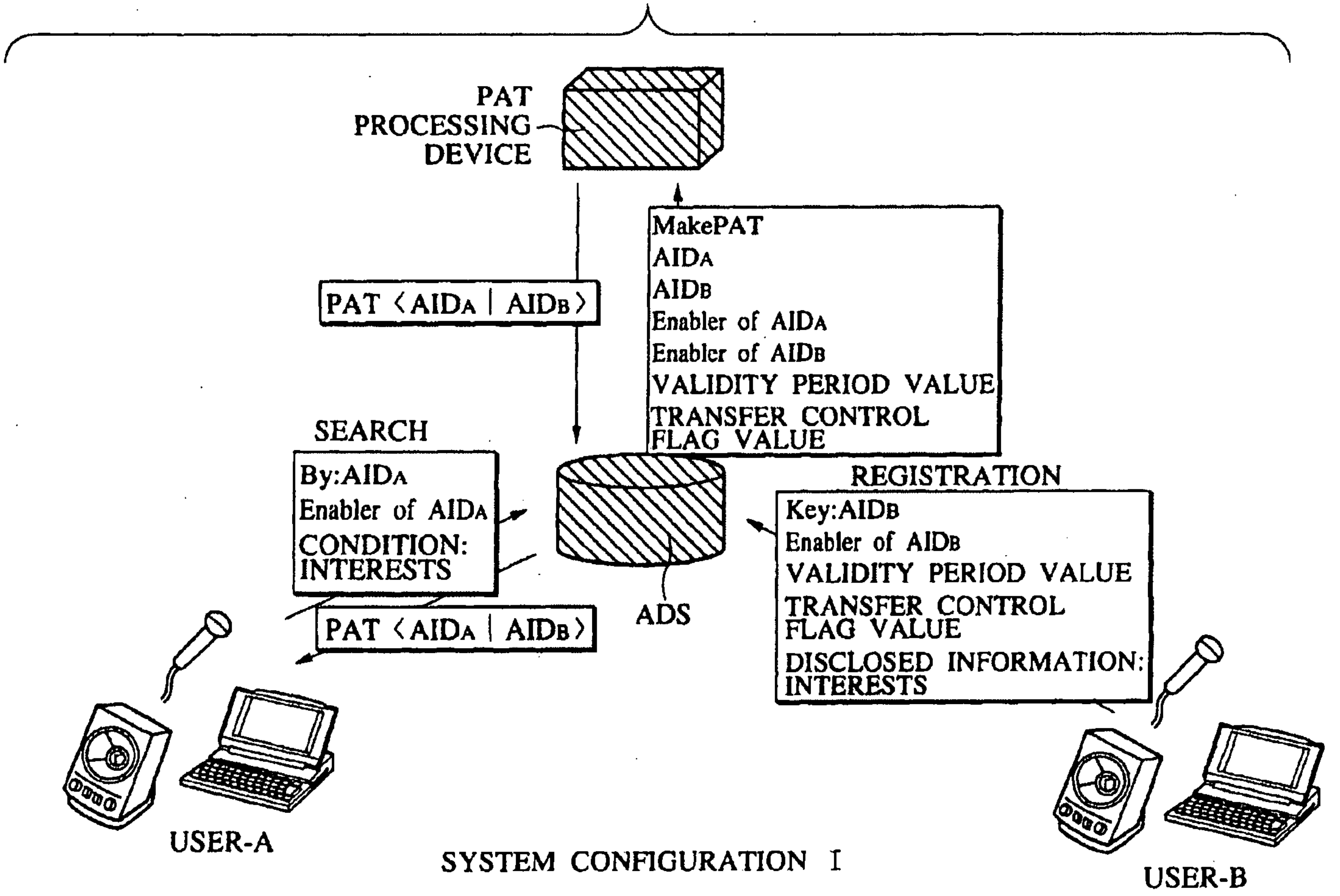

PAT PROCESSING DEVICE
PAT ⟨AID_A | AID_B⟩
MakePAT
AID_A
AID_B
Enabler of AID_A
Enabler of AID_B
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
SEARCH
By:AID_A
Enabler of AID_A
CONDITION: INTERESTS
ADS
REGISTRATION
Key:AID_B
Enabler of AID_B
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
DISCLOSED INFORMATION: INTERESTS
PAT ⟨AID_A | AID_B⟩
USER-A
SYSTEM CONFIGURATION I
USER-B

FIG.15

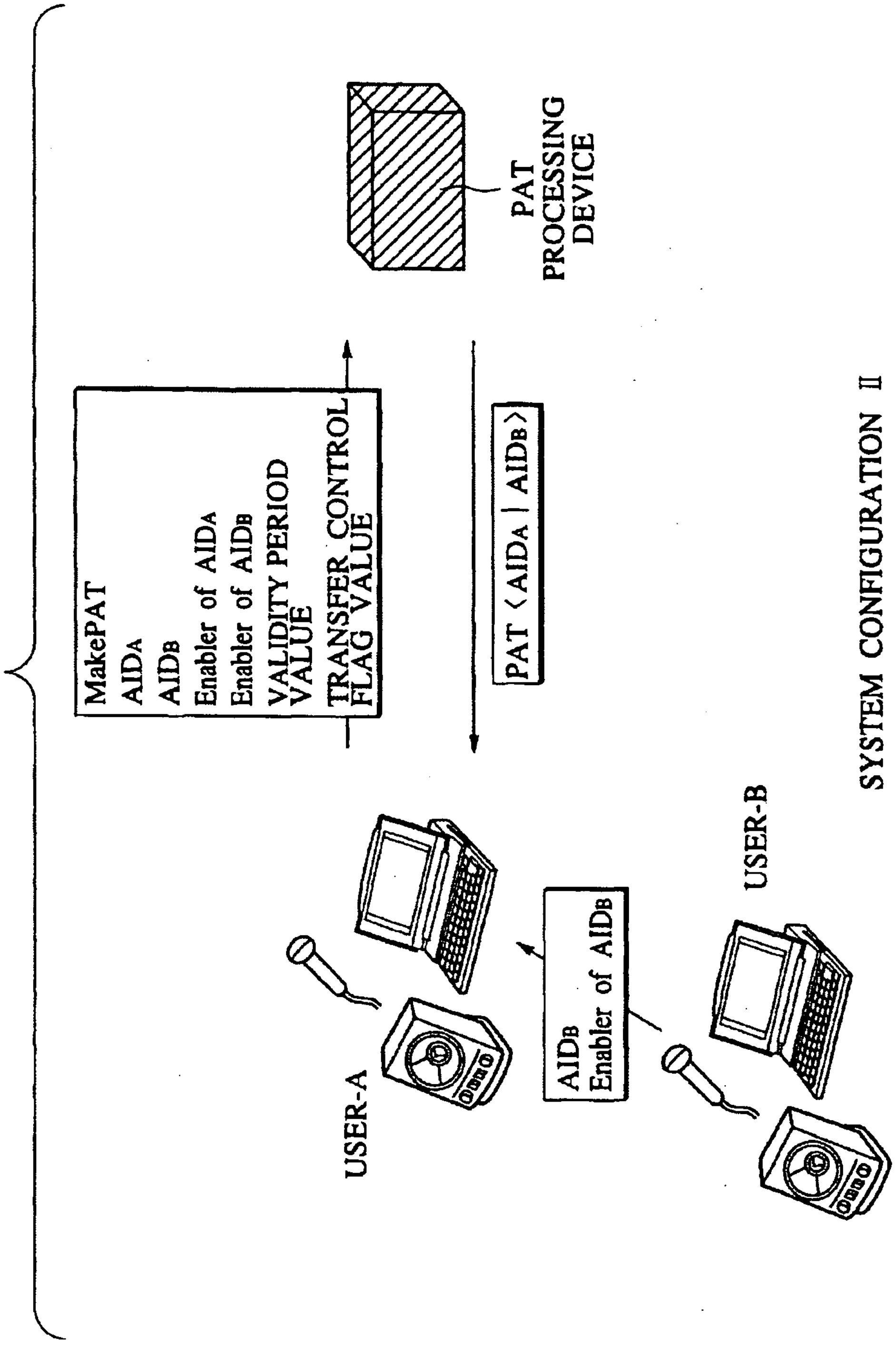
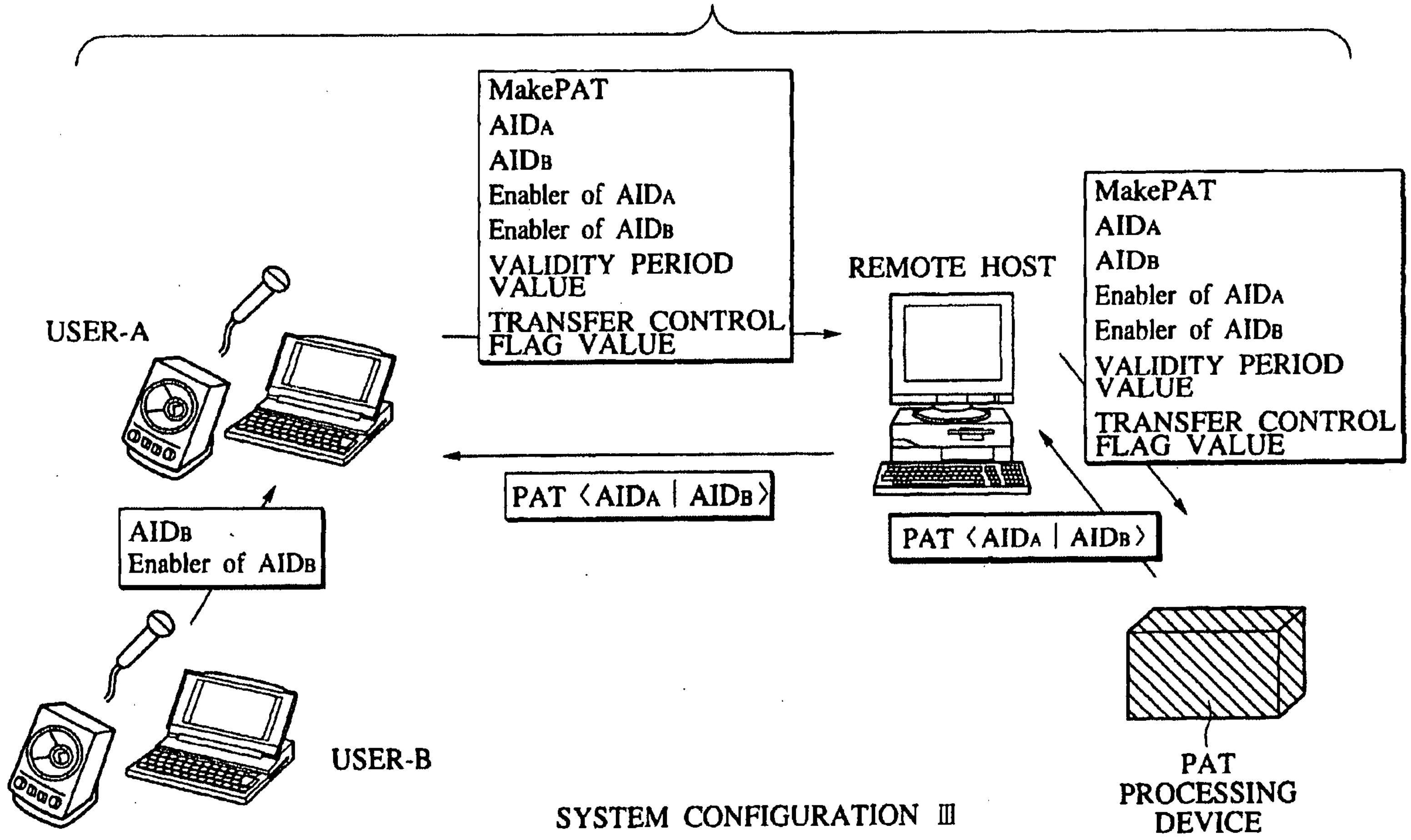
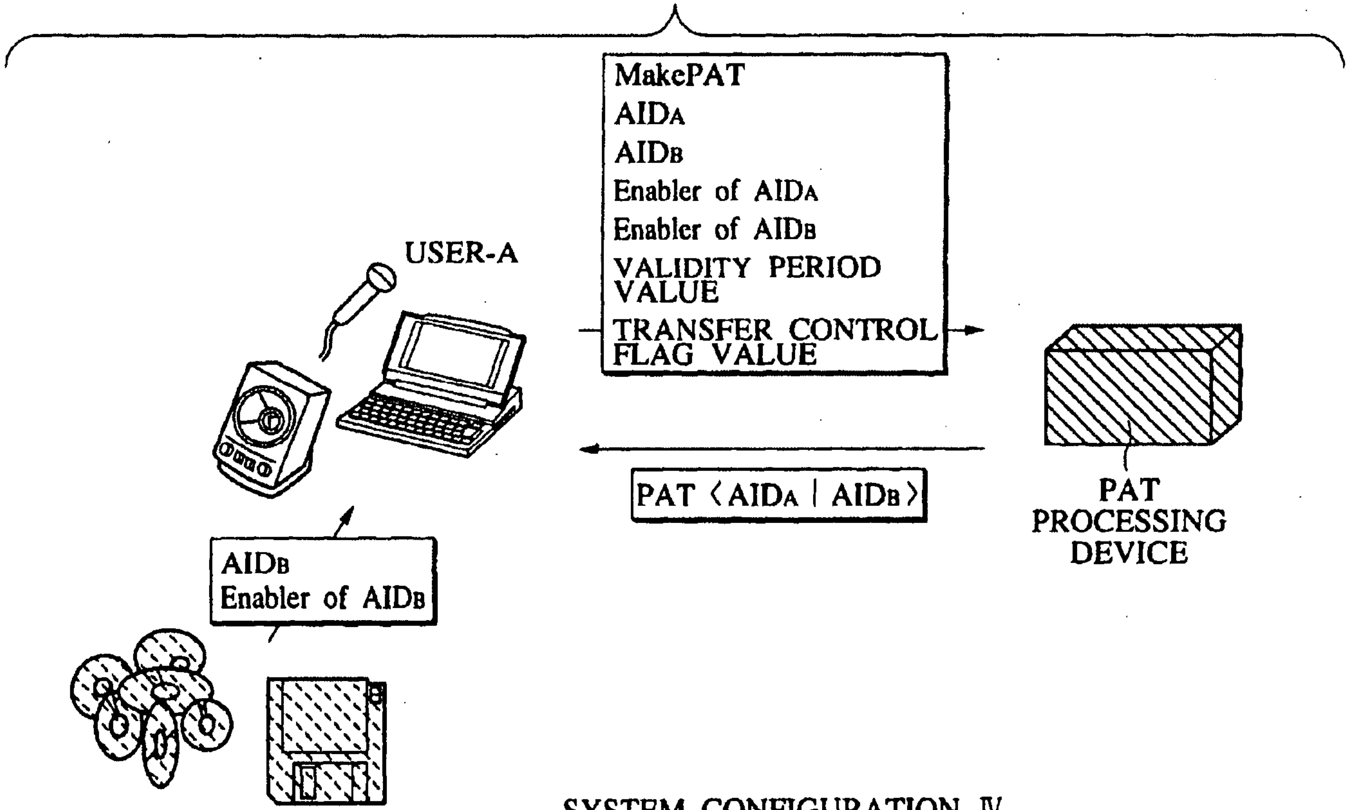

MakePAT
AIDA
AIDB
Enabler of AIDA
Enabler of AIDB
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
PAT PROCESSING DEVICE
PAT ⟨AIDA | AIDB⟩
USER-A
AIDB
Enabler of AIDB
USER-B
SYSTEM CONFIGURATION II

FIG.16


USER-A
MakePAT
AID$_A$
AID$_B$
Enabler of AID$_A$
Enabler of AID$_B$
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
REMOTE HOST
MakePAT
AID$_A$
AID$_B$
Enabler of AID$_A$
Enabler of AID$_B$
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
PAT ⟨AID$_A$ | AID$_B$⟩
PAT ⟨AID$_A$ | AID$_B$⟩
AID$_B$
Enabler of AID$_B$
USER-B
PAT PROCESSING DEVICE

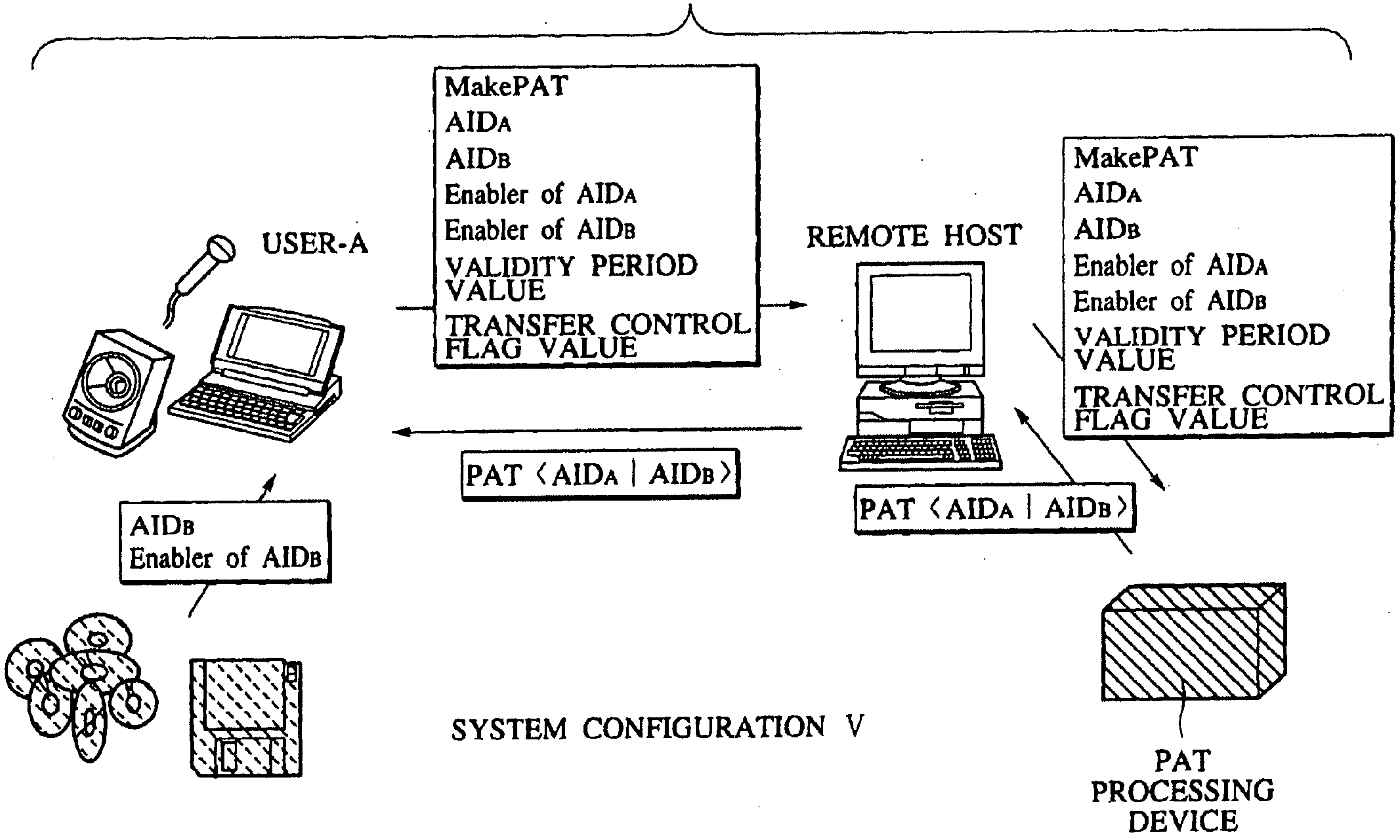
SYSTEM CONFIGURATION Ⅲ

FIG.17


USER-A
MakePAT
AIDA
AIDB
Enabler of AIDA
Enabler of AIDB
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
PAT ⟨AIDA | AIDB⟩
PAT PROCESSING DEVICE
AIDB
Enabler of AIDB

SYSTEM CONFIGURATION Ⅳ

FIG.18


USER-A
MakePAT
AIDA
AIDB
Enabler of AIDA
Enabler of AIDB
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
REMOTE HOST
MakePAT
AIDA
AIDB
Enabler of AIDA
Enabler of AIDB
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
PAT ⟨AIDA | AIDB⟩
PAT ⟨AIDA | AIDB⟩
AIDB
Enabler of AIDB
PAT PROCESSING DEVICE

SYSTEM CONFIGURATION V

FIG.19

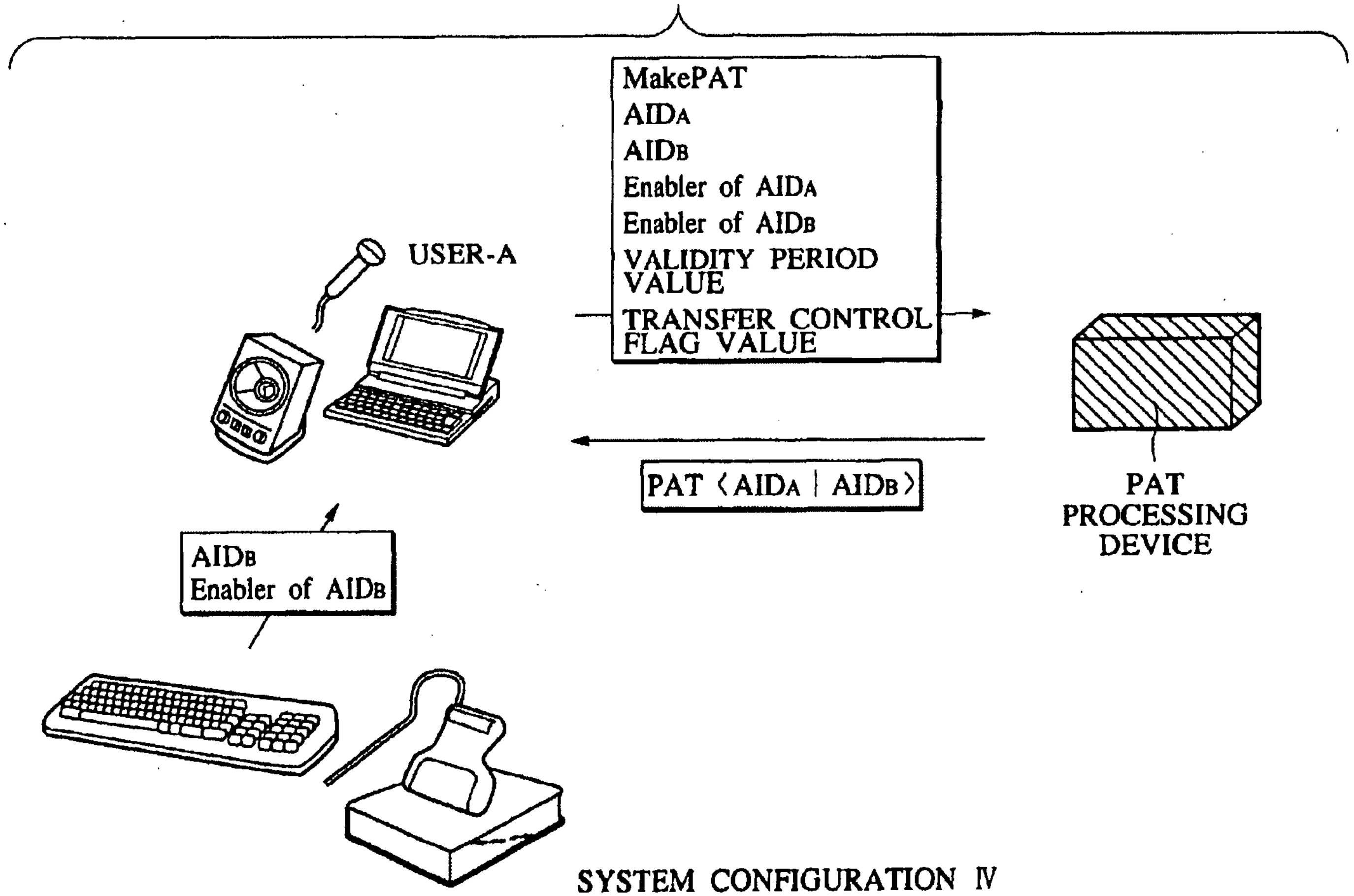
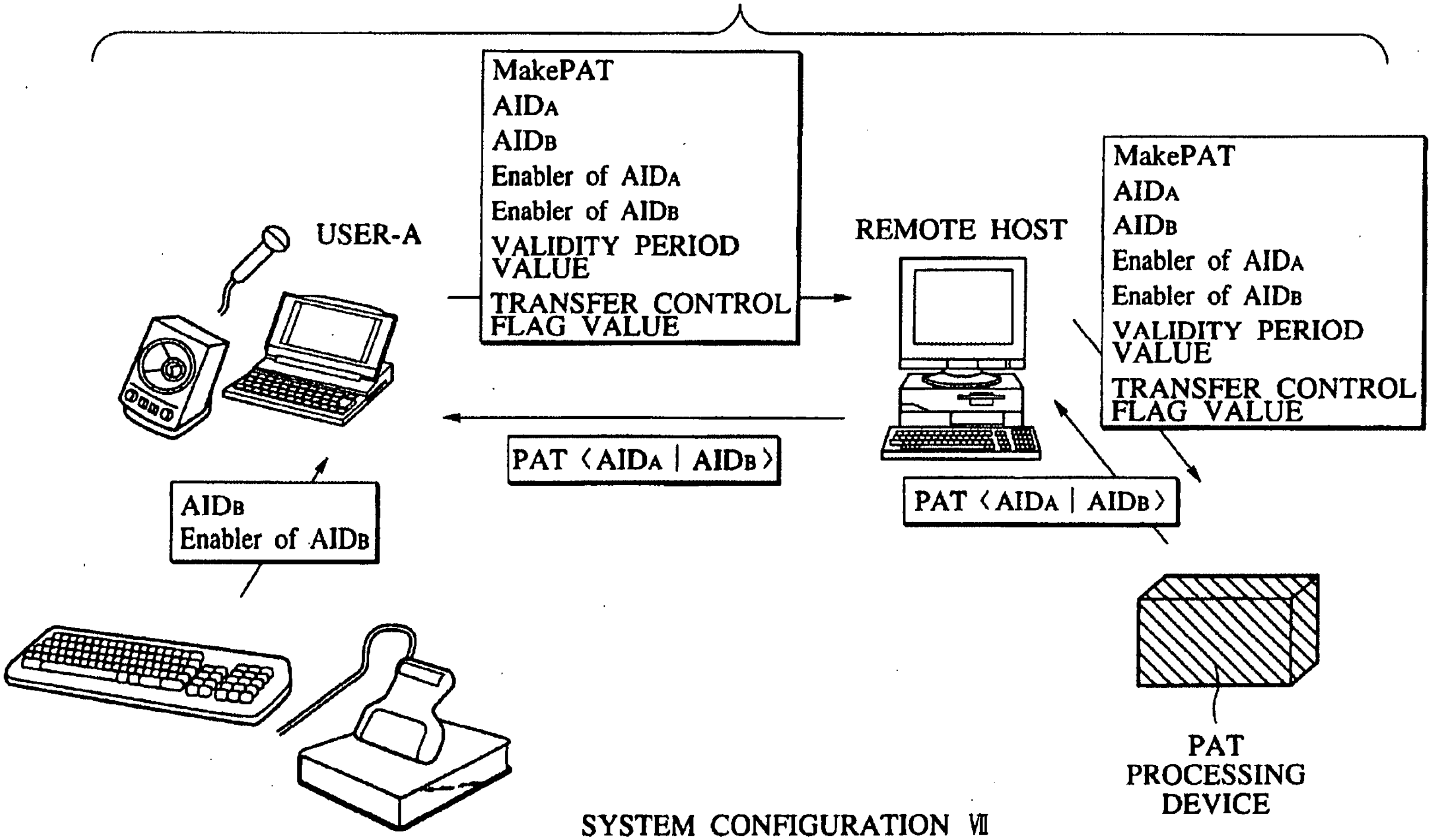

MakePAT
AIDA
AIDB
Enabler of AIDA
Enabler of AIDB
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
USER-A
PAT 〈AIDA | AIDB〉
PAT PROCESSING DEVICE
AIDB
Enabler of AIDB
SYSTEM CONFIGURATION IV

FIG.20


USER-A
MakePAT
AID_A
AID_B
Enabler of AID_A
Enabler of AID_B
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
REMOTE HOST
MakePAT
AID_A
AID_B
Enabler of AID_A
Enabler of AID_B
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
PAT ⟨AID_A | AID_B⟩
PAT ⟨AID_A | AID_B⟩
AID_B
Enabler of AID_B
PAT PROCESSING DEVICE

SYSTEM CONFIGURATION Ⅶ

FIG.21

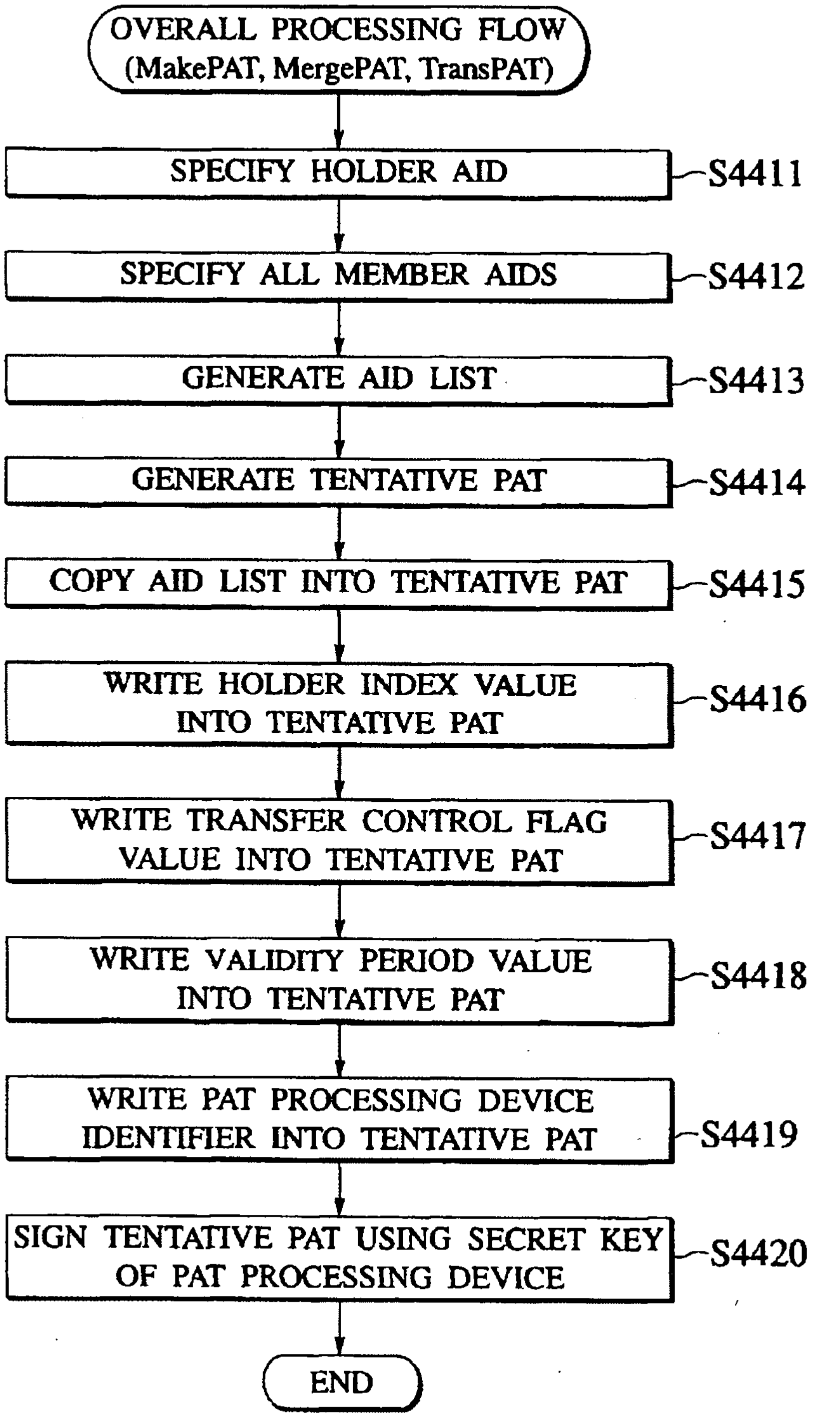

OVERALL PROCESSING FLOW (MakePAT, MergePAT, TransPAT)
SPECIFY HOLDER AID
S4411
SPECIFY ALL MEMBER AIDS
S4412
GENERATE AID LIST
S4413
GENERATE TENTATIVE PAT
S4414
COPY AID LIST INTO TENTATIVE PAT
S4415
WRITE HOLDER INDEX VALUE INTO TENTATIVE PAT
S4416
WRITE TRANSFER CONTROL FLAG VALUE INTO TENTATIVE PAT
S4417
WRITE VALIDITY PERIOD VALUE INTO TENTATIVE PAT
S4418
WRITE PAT PROCESSING DEVICE IDENTIFIER INTO TENTATIVE PAT
S4419
SIGN TENTATIVE PAT USING SECRET KEY OF PAT PROCESSING DEVICE
S4420
END

FIG.22

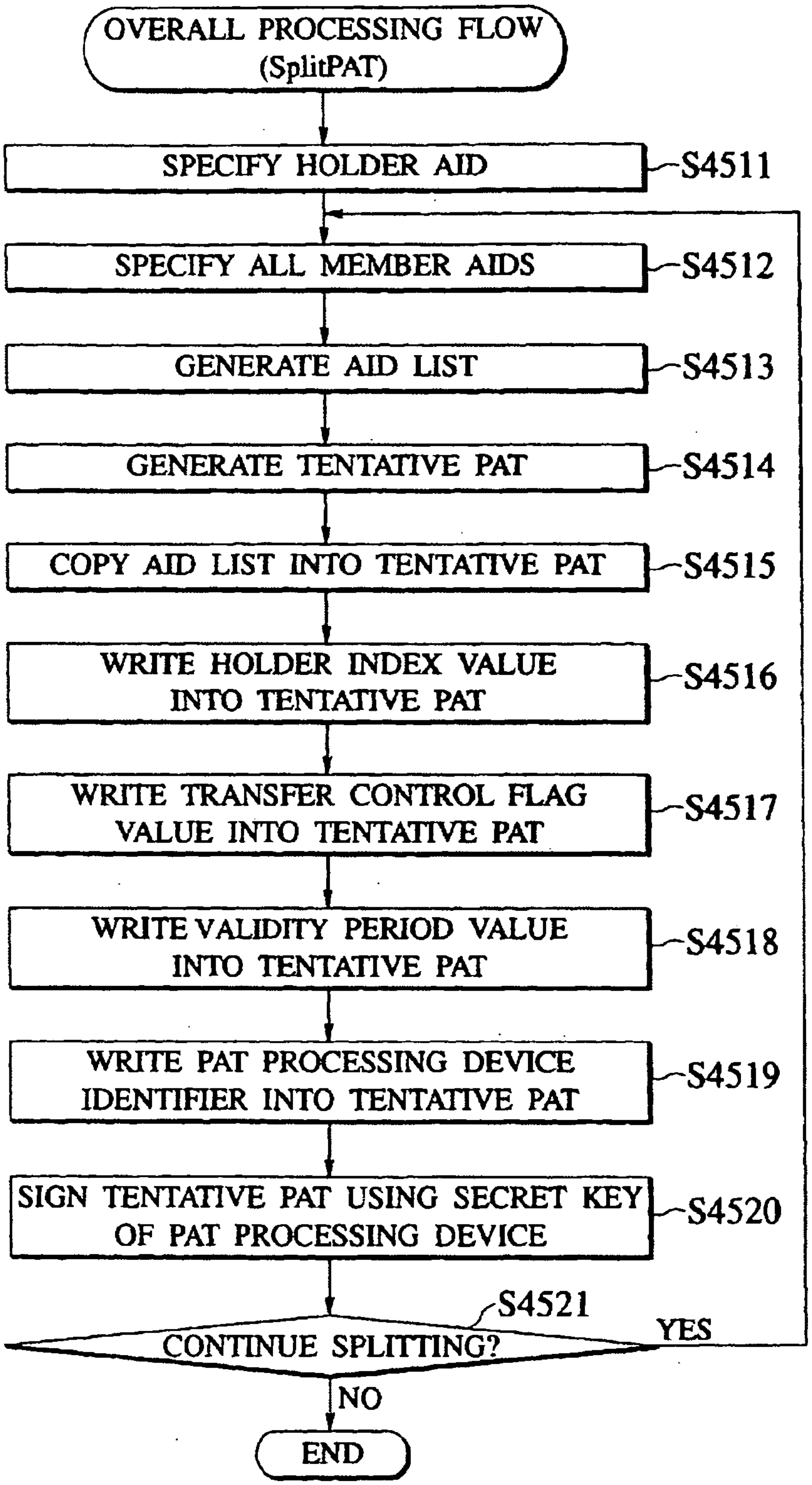

OVERALL PROCESSING FLOW (SplitPAT)
SPECIFY HOLDER AID
S4511
SPECIFY ALL MEMBER AIDS
S4512
GENERATE AID LIST
S4513
GENERATE TENTATIVE PAT
S4514
COPY AID LIST INTO TENTATIVE PAT
S4515
WRITE HOLDER INDEX VALUE INTO TENTATIVE PAT
S4516
WRITE TRANSFER CONTROL FLAG VALUE INTO TENTATIVE PAT
S4517
WRITE VALIDITY PERIOD VALUE INTO TENTATIVE PAT
S4518
WRITE PAT PROCESSING DEVICE IDENTIFIER INTO TENTATIVE PAT
S4519
SIGN TENTATIVE PAT USING SECRET KEY OF PAT PROCESSING DEVICE
S4520
S4521
CONTINUE SPLITTING?
YES
NO
END

FIG.23

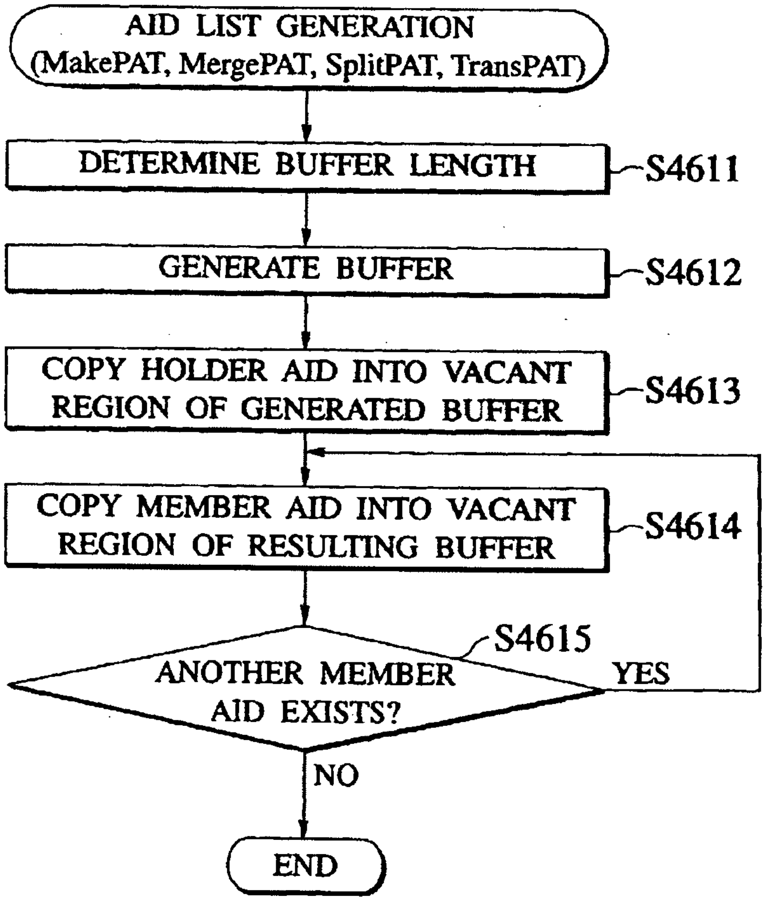

AID LIST GENERATION
(MakePAT, MergePAT, SplitPAT, TransPAT)
DETERMINE BUFFER LENGTH
S4611
GENERATE BUFFER
S4612
COPY HOLDER AID INTO VACANT REGION OF GENERATED BUFFER
S4613
COPY MEMBER AID INTO VACANT REGION OF RESULTING BUFFER
S4614
S4615
ANOTHER MEMBER AID EXISTS?
YES
NO
END

FIG.24

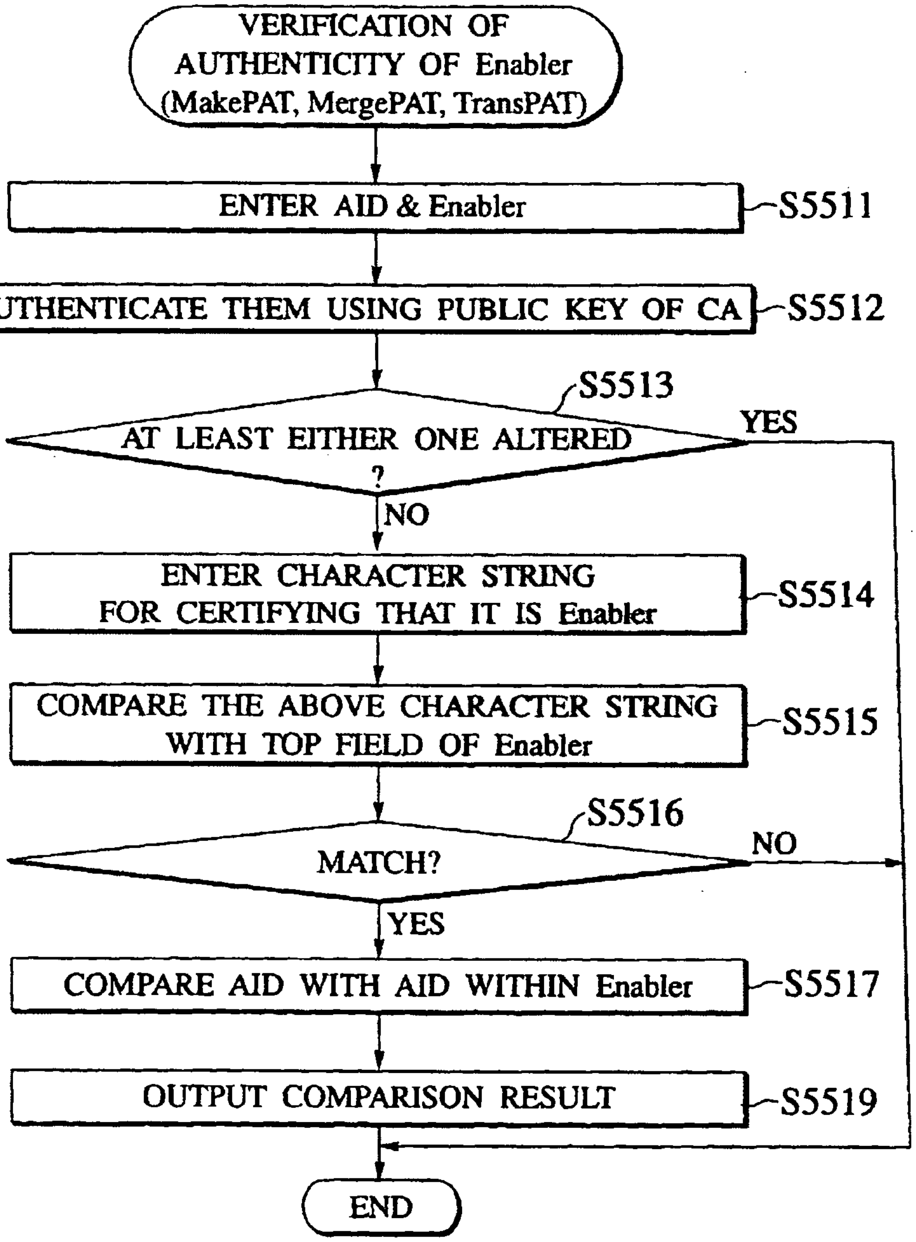

VERIFICATION OF AUTHENTICITY OF Enabler (MakePAT, MergePAT, TransPAT)
ENTER AID & Enabler
S5511
AUTHENTICATE THEM USING PUBLIC KEY OF CA
S5512
S5513
AT LEAST EITHER ONE ALTERED ?
YES
NO
ENTER CHARACTER STRING FOR CERTIFYING THAT IT IS Enabler
S5514
COMPARE THE ABOVE CHARACTER STRING WITH TOP FIELD OF Enabler
S5515
S5516
MATCH?
NO
YES
COMPARE AID WITH AID WITHIN Enabler
S5517
OUTPUT COMPARISON RESULT
S5519
END

FIG.25

DATA STRUCTURE OF Null-AID

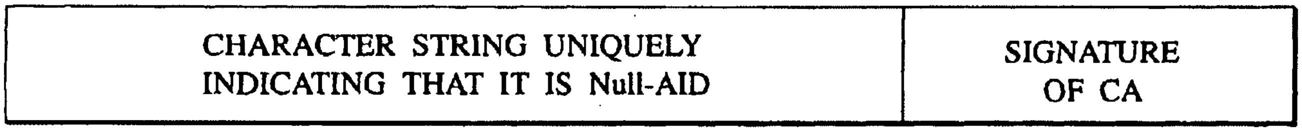

CHARACTER STRING UNIQUELY INDICATING THAT IT IS Null-AID
SIGNATURE OF CA

FIG.26

DATA STRUCTURE OF Enabler of Null-AID

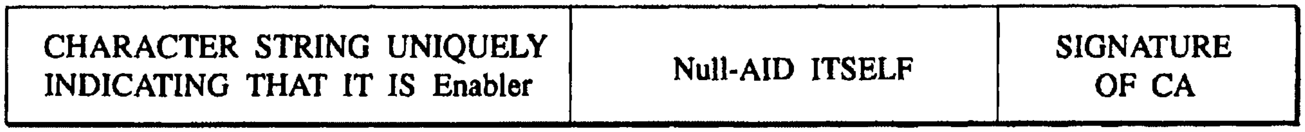

CHARACTER STRING UNIQUELY INDICATING THAT IT IS Enabler
Null-AID ITSELF
SIGNATURE OF CA

FIG.27

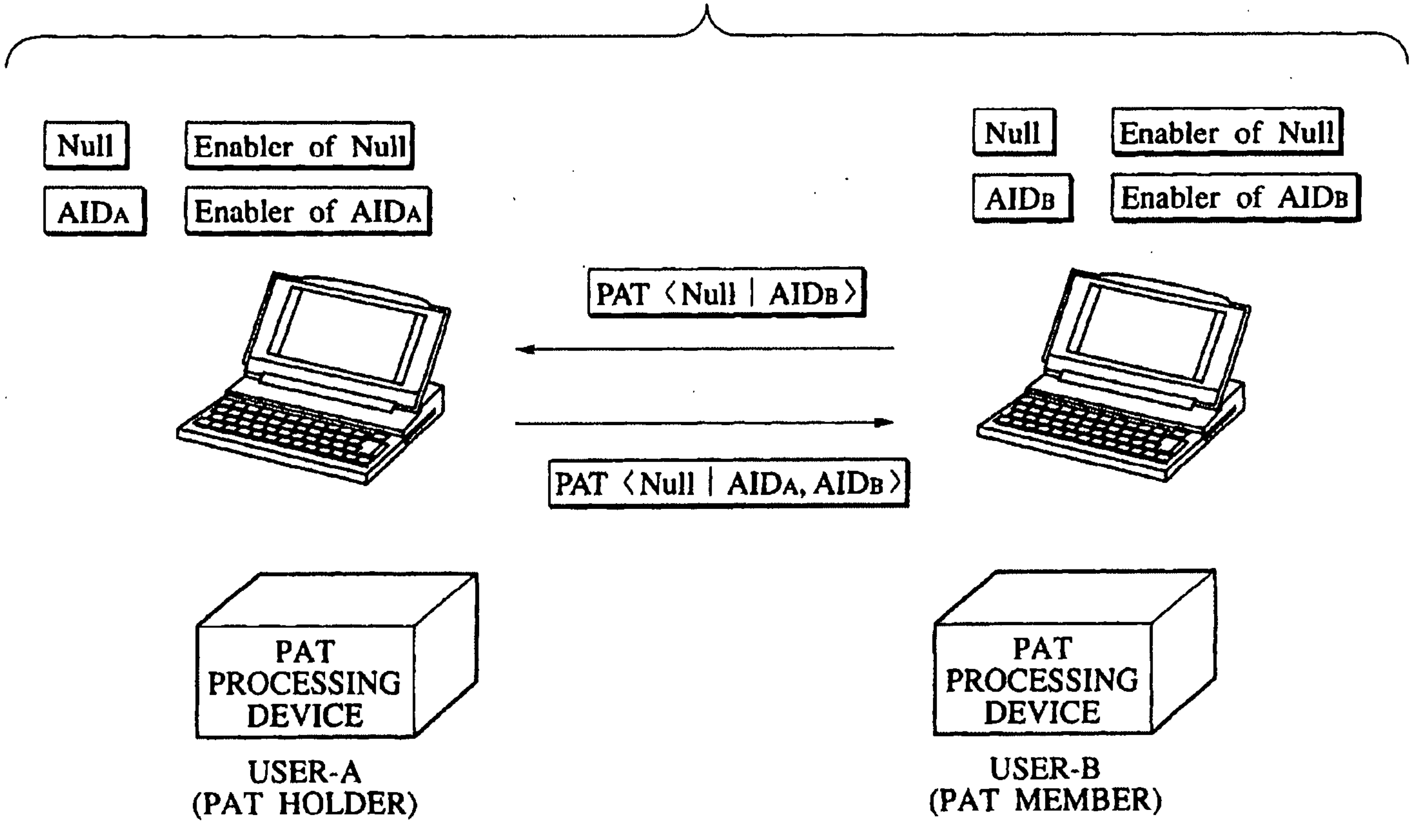

Null
Enabler of Null
AID_A
Enabler of AID_A
Null
Enabler of Null
AID_B
Enabler of AID_B
PAT ⟨Null | AID_B⟩
PAT ⟨Null | AID_A, AID_B⟩
PAT PROCESSING DEVICE
USER-A (PAT HOLDER)
PAT PROCESSING DEVICE
USER-B (PAT MEMBER)

FIG.28

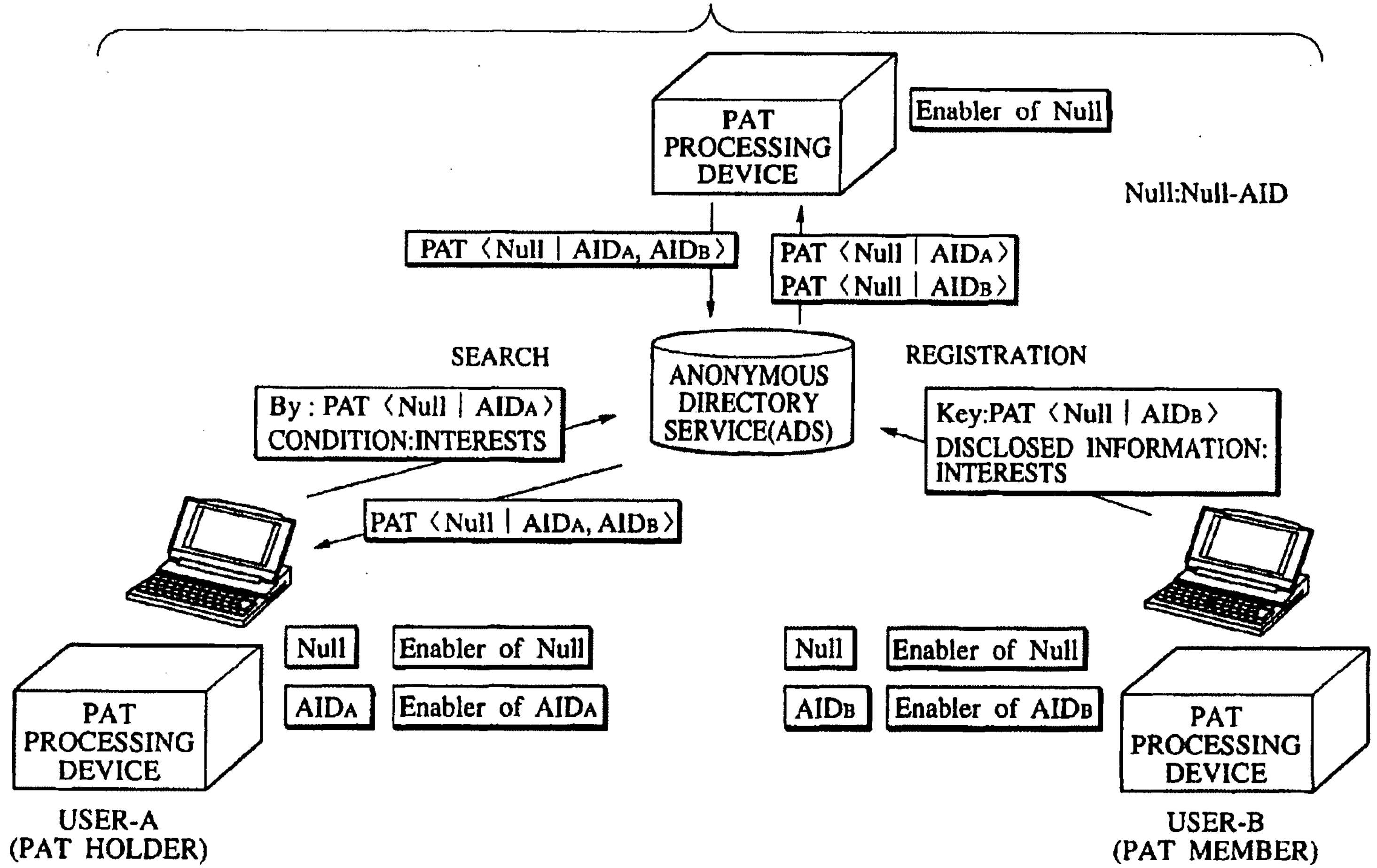

PAT PROCESSING DEVICE
Enabler of Null
Null:Null-AID
PAT ⟨Null | AIDA, AIDB⟩
PAT ⟨Null | AIDA⟩
PAT ⟨Null | AIDB⟩
SEARCH
ANONYMOUS DIRECTORY SERVICE(ADS)
REGISTRATION
By : PAT ⟨Null | AIDA⟩
CONDITION:INTERESTS
Key:PAT ⟨Null | AIDB⟩
DISCLOSED INFORMATION: INTERESTS
PAT ⟨Null | AIDA, AIDB⟩
Null
Enabler of Null
AIDA
Enabler of AIDA
Null
Enabler of Null
AIDB
Enabler of AIDB
PAT PROCESSING DEVICE
USER-A
(PAT HOLDER)
PAT PROCESSING DEVICE
USER-B
(PAT MEMBER)

FIG.29

DATA STRUCTURE OF God-AID

CHARACTER STRING UNIQUELY INDICATING THAT IT IS God-AID	SIGNATURE OF CA

FIG.30

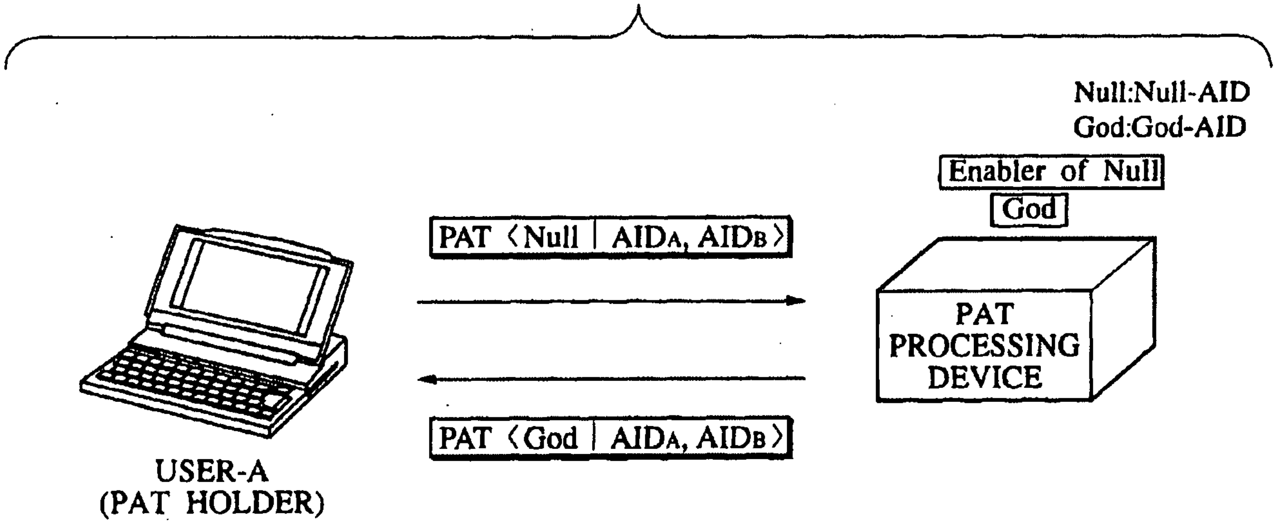

Null:Null-AID
God:God-AID
Enabler of Null
God
PAT PROCESSING DEVICE
PAT ⟨Null | AIDA, AIDB⟩
PAT ⟨God | AIDA, AIDB⟩
USER-A
(PAT HOLDER)

FIG.31

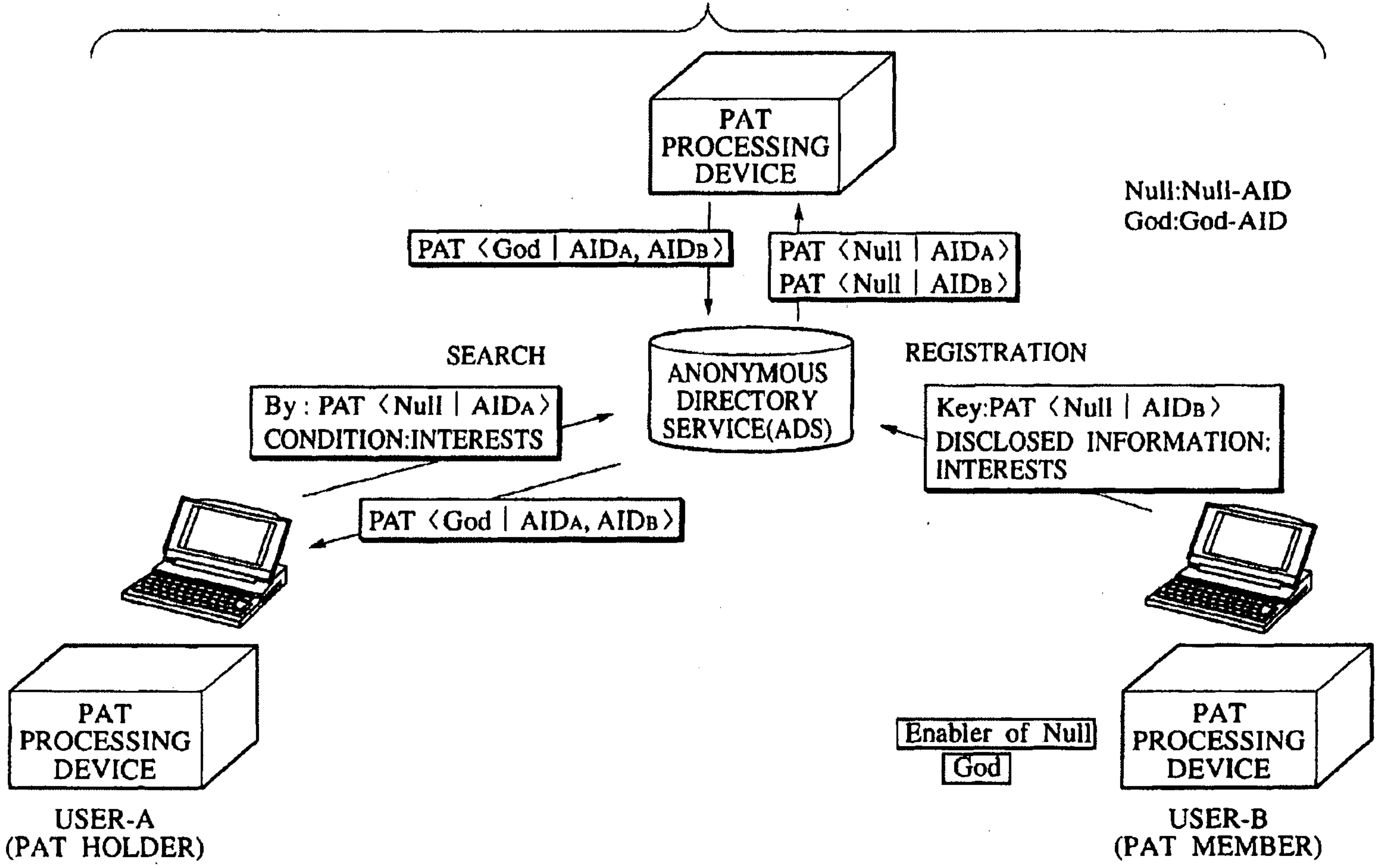

PAT PROCESSING DEVICE
Null:Null-AID
God:God-AID
PAT 〈God | AIDA, AIDB〉
PAT 〈Null | AIDA〉
PAT 〈Null | AIDB〉
SEARCH
ANONYMOUS DIRECTORY SERVICE(ADS)
REGISTRATION
By : PAT 〈Null | AIDA〉
CONDITION:INTERESTS
Key:PAT 〈Null | AIDB〉
DISCLOSED INFORMATION:
INTERESTS
PAT 〈God | AIDA, AIDB〉
PAT PROCESSING DEVICE
USER-A
(PAT HOLDER)
Enabler of Null
God
PAT PROCESSING DEVICE
USER-B
(PAT MEMBER)

FIG.32

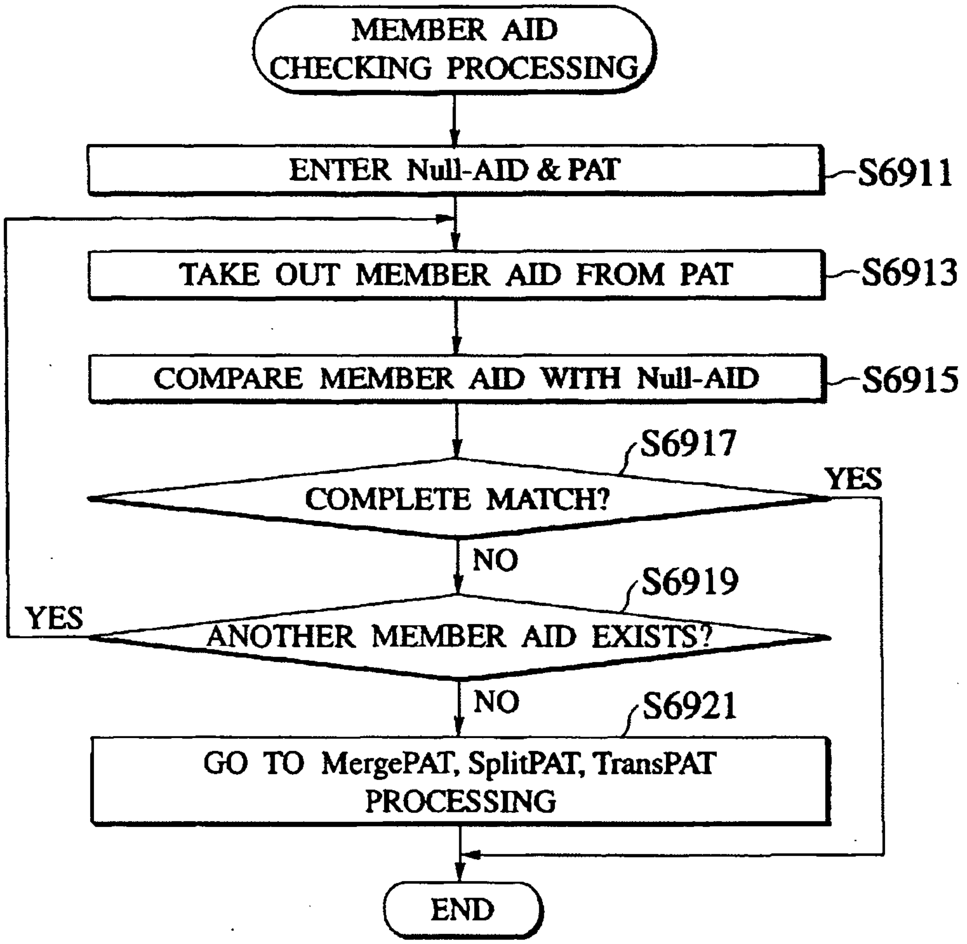

MEMBER AID CHECKING PROCESSING
ENTER Null-AID & PAT
S6911
TAKE OUT MEMBER AID FROM PAT
S6913
COMPARE MEMBER AID WITH Null-AID
S6915
S6917
COMPLETE MATCH?
YES
NO
S6919
YES
ANOTHER MEMBER AID EXISTS?
NO
S6921
GO TO MergePAT, SplitPAT, TransPAT PROCESSING
END

FIG.33

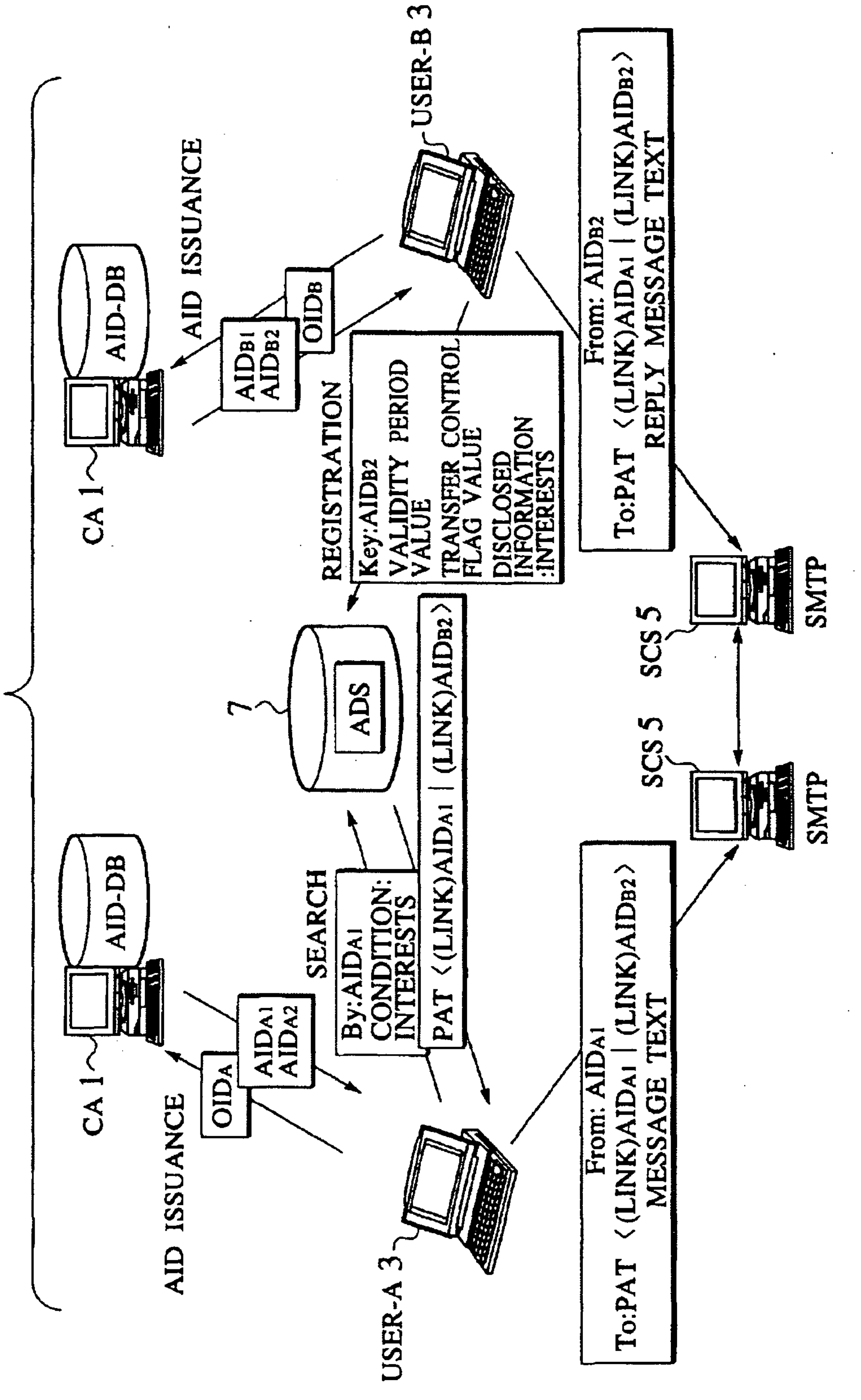

AID ISSUANCE
CA 1
AID-DB
OIDA
AIDA1
AIDA2
SEARCH
By:AIDA1
CONDITION:
INTERESTS
PAT ⟨(LINK)AIDA1 | (LINK)AIDB2⟩
7
ADS
USER-A 3
From: AIDA1
To:PAT ⟨(LINK)AIDA1 | (LINK)AIDB2⟩
MESSAGE TEXT
SCS 5
SMTP
SCS 5
SMTP
CA 1
AID-DB
AID ISSUANCE
AIDB1
AIDB2
OIDB
USER-B 3
REGISTRATION
Key:AIDB2
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
DISCLOSED INFORMATION :INTERESTS
From: AIDB2
To:PAT ⟨(LINK)AIDA1 | (LINK)AIDB2⟩
REPLY MESSAGE TEXT

FIG.34

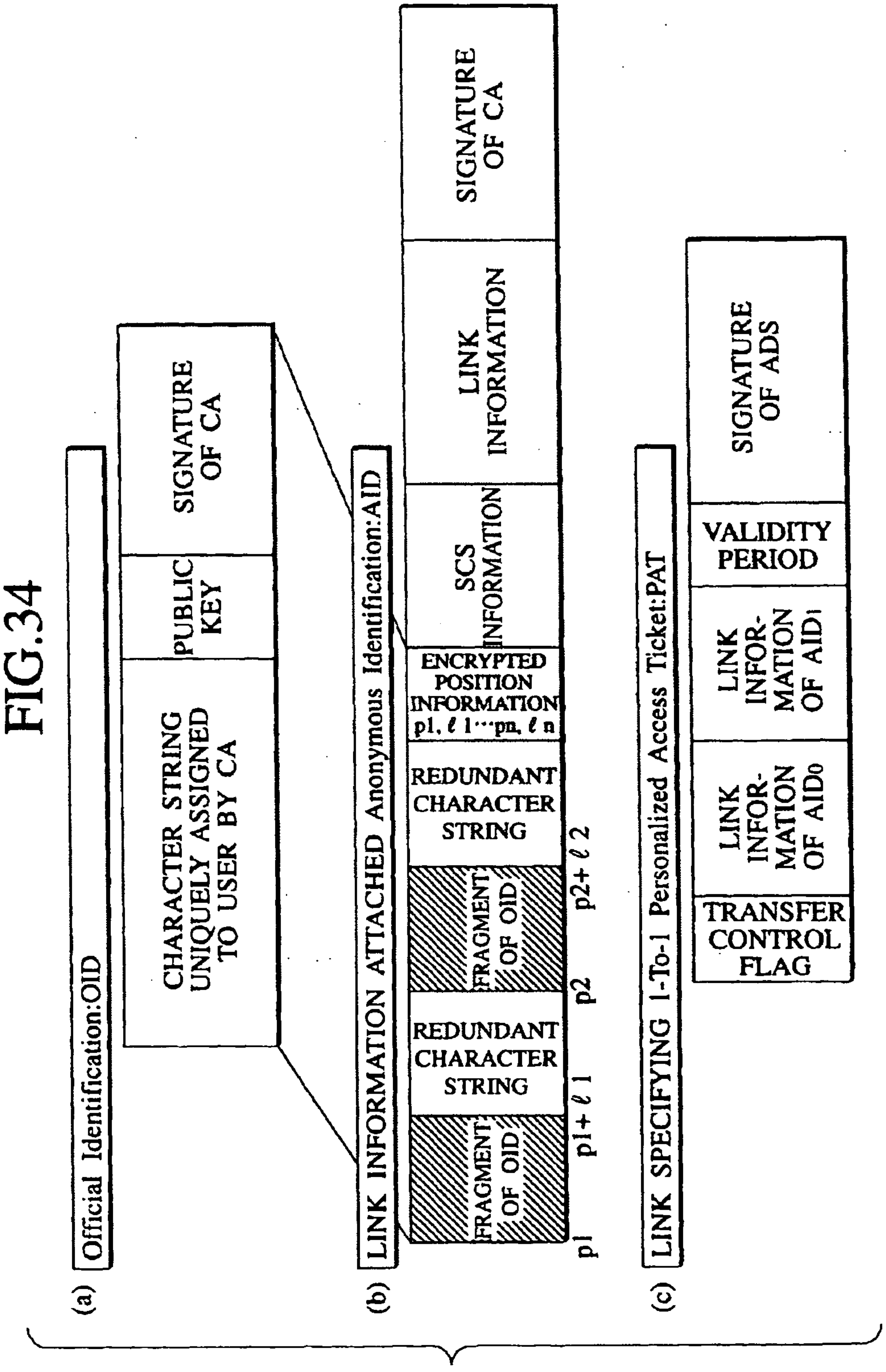

(a)
Official Identification:OID
CHARACTER STRING UNIQUELY ASSIGNED TO USER BY CA
PUBLIC KEY
SIGNATURE OF CA
(b)
LINK INFORMATION ATTACHED Anonymous Identification:AID
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
ENCRYPTED POSITION INFORMATION p1, ℓ 1···pn, ℓ n
SCS INFORMATION
LINK INFORMATION
SIGNATURE OF CA
p1
p1+ ℓ 1
p2
p2+ ℓ 2
(c)
LINK SPECIFYING 1-To-1 Personalized Access Ticket:PAT
TRANSFER CONTROL FLAG
LINK INFOR-MATION OF AID0
LINK INFOR-MATION OF AID1
VALIDITY PERIOD
SIGNATURE OF ADS

FIG.35

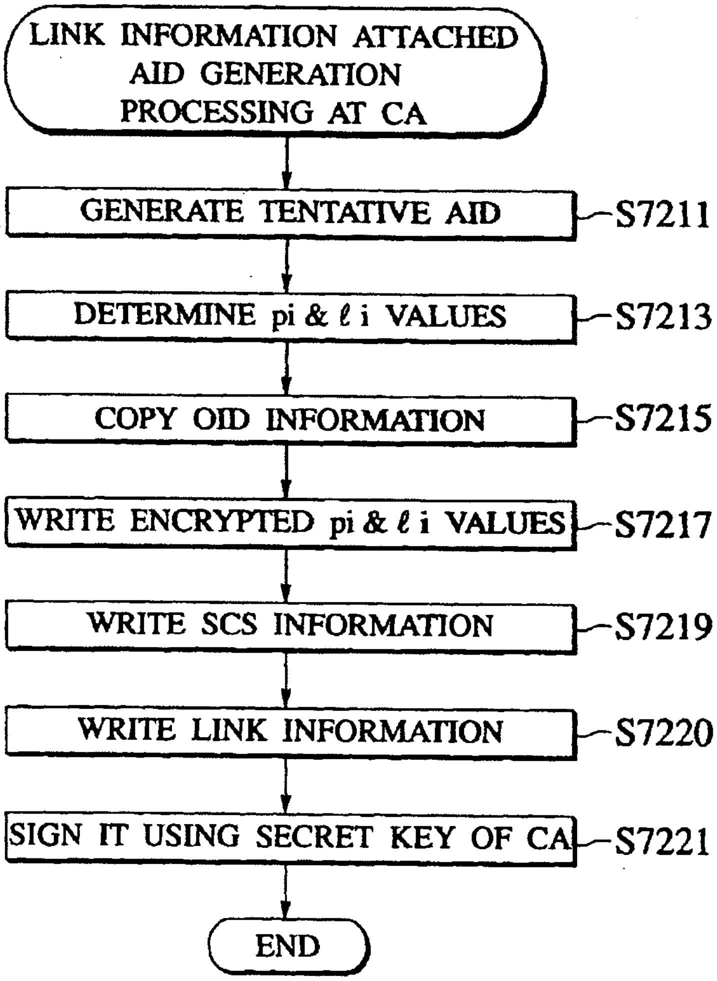

LINK INFORMATION ATTACHED AID GENERATION PROCESSING AT CA
GENERATE TENTATIVE AID
S7211
DETERMINE pi & ℓ i VALUES
S7213
COPY OID INFORMATION
S7215
WRITE ENCRYPTED pi & ℓ i VALUES
S7217
WRITE SCS INFORMATION
S7219
WRITE LINK INFORMATION
S7220
SIGN IT USING SECRET KEY OF CA
S7221
END

FIG.36

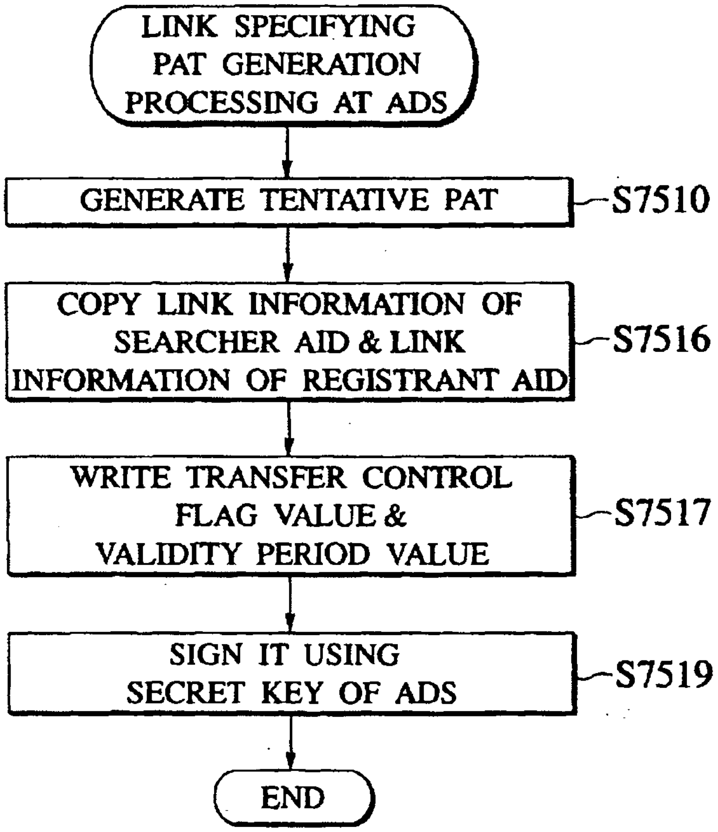

LINK SPECIFYING PAT GENERATION PROCESSING AT ADS
GENERATE TENTATIVE PAT
S7510
COPY LINK INFORMATION OF SEARCHER AID & LINK INFORMATION OF REGISTRANT AID
S7516
WRITE TRANSFER CONTROL FLAG VALUE & VALIDITY PERIOD VALUE
S7517
SIGN IT USING SECRET KEY OF ADS
S7519
END

FIG.37

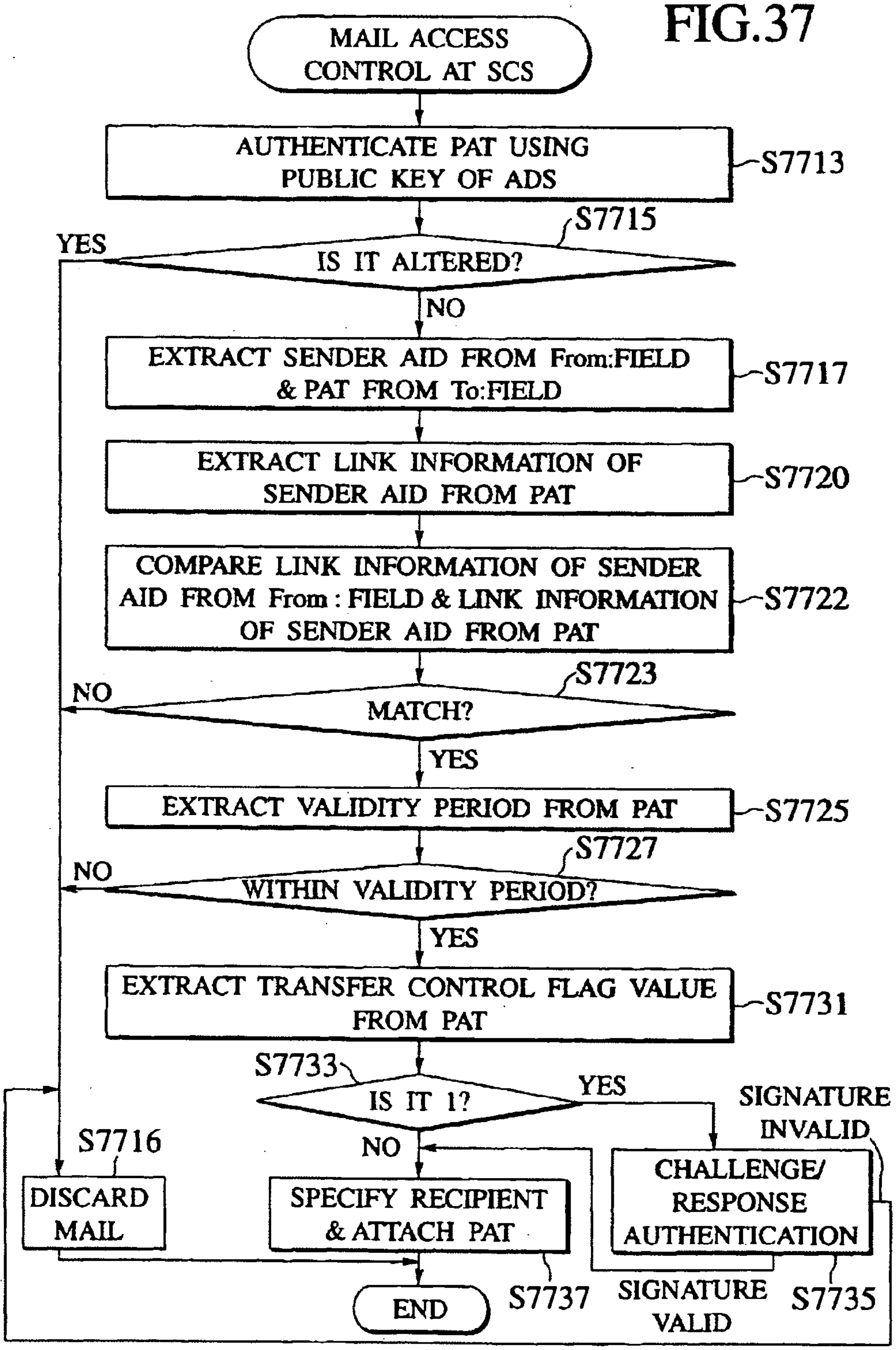

MAIL ACCESS CONTROL AT SCS
AUTHENTICATE PAT USING PUBLIC KEY OF ADS
S7713
S7715
YES
IS IT ALTERED?
NO
EXTRACT SENDER AID FROM From:FIELD & PAT FROM To:FIELD
S7717
EXTRACT LINK INFORMATION OF SENDER AID FROM PAT
S7720
COMPARE LINK INFORMATION OF SENDER AID FROM From : FIELD & LINK INFORMATION OF SENDER AID FROM PAT
S7722
S7723
NO
MATCH?
YES
EXTRACT VALIDITY PERIOD FROM PAT
S7725
S7727
NO
WITHIN VALIDITY PERIOD?
YES
EXTRACT TRANSFER CONTROL FLAG VALUE FROM PAT
S7731
S7733
IS IT 1?
YES
SIGNATURE INVALID
NO
S7716
DISCARD MAIL
SPECIFY RECIPIENT & ATTACH PAT
CHALLENGE/ RESPONSE AUTHENTICATION
END
S7737
SIGNATURE VALID
S7735

FIG.38

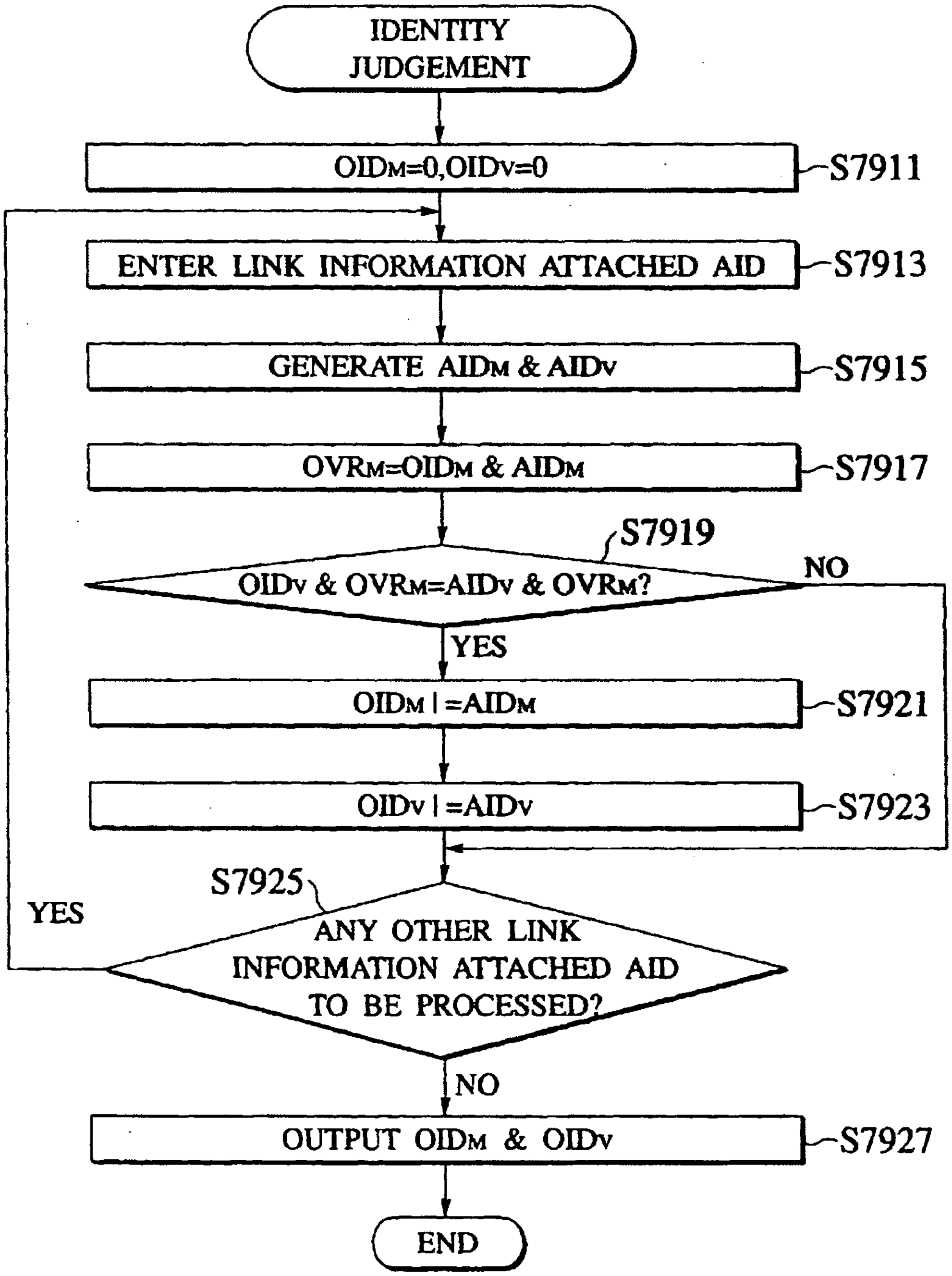

IDENTITY JUDGEMENT
OIDM=0,OIDV=0
S7911
ENTER LINK INFORMATION ATTACHED AID
S7913
GENERATE AIDM & AIDV
S7915
OVRM=OIDM & AIDM
S7917
S7919
OIDV & OVRM=AIDV & OVRM?
NO
YES
OIDM | =AIDM
S7921
OIDV | =AIDV
S7923
S7925
YES
ANY OTHER LINK INFORMATION ATTACHED AID TO BE PROCESSED?
NO
OUTPUT OIDM & OIDV
S7927
END

FIG.39

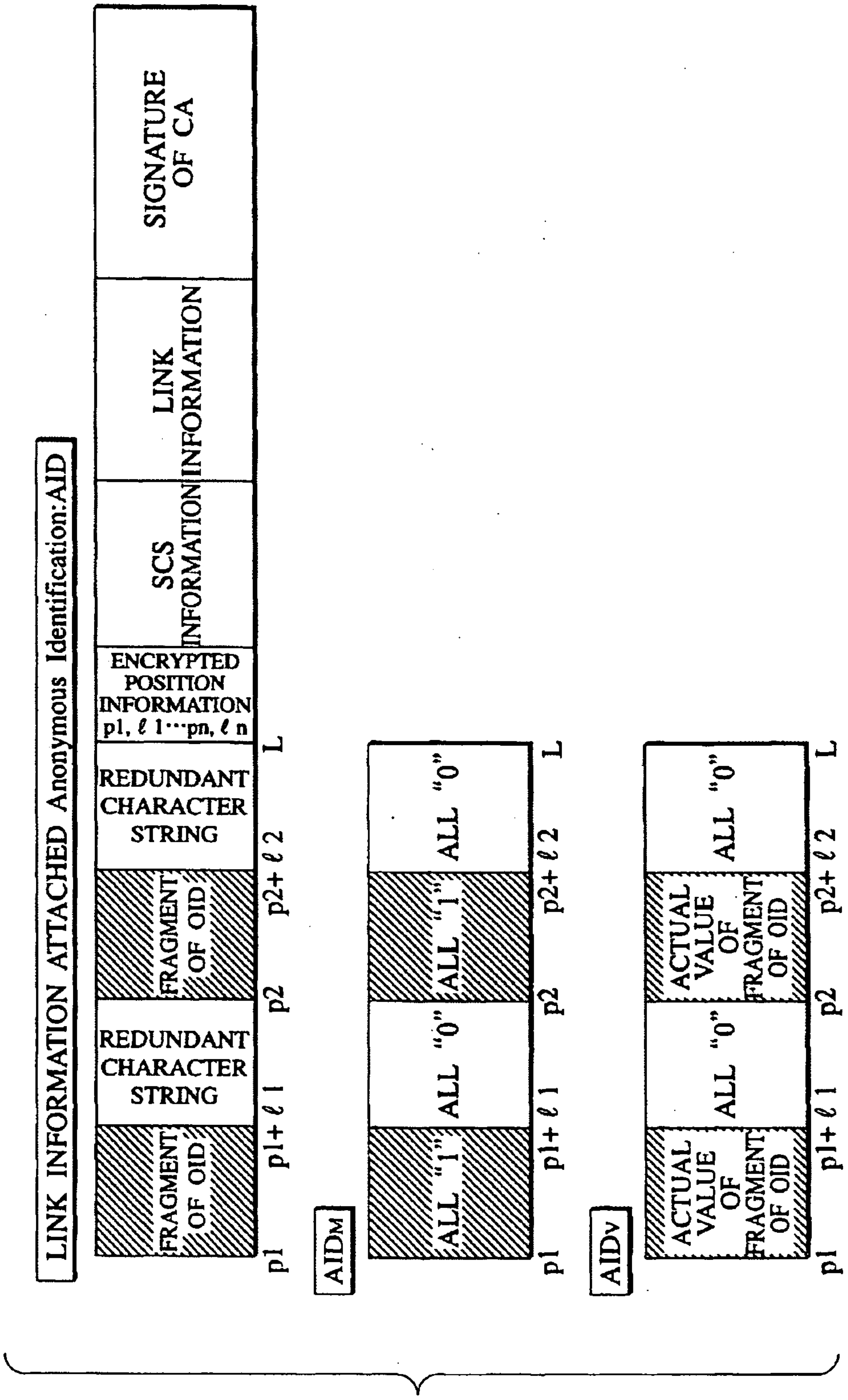

LINK INFORMATION ATTACHED Anonymous Identification:AID
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
ENCRYPTED POSITION INFORMATION p1, ℓ 1···pn, ℓ n
SCS INFORMATION
LINK INFORMATION
SIGNATURE OF CA
p1
p1+ℓ 1
p2
p2+ℓ 2
L
AIDM
ALL "1"
ALL "0"
ALL "1"
ALL "0"
p1
p1+ℓ 1
p2
p2+ℓ 2
L
AIDV
ACTUAL VALUE OF FRAGMENT OF OID
ALL "0"
ACTUAL VALUE OF FRAGMENT OF OID
ALL "0"
p1
p1+ℓ 1
p2
p2+ℓ 2
L

FIG.40

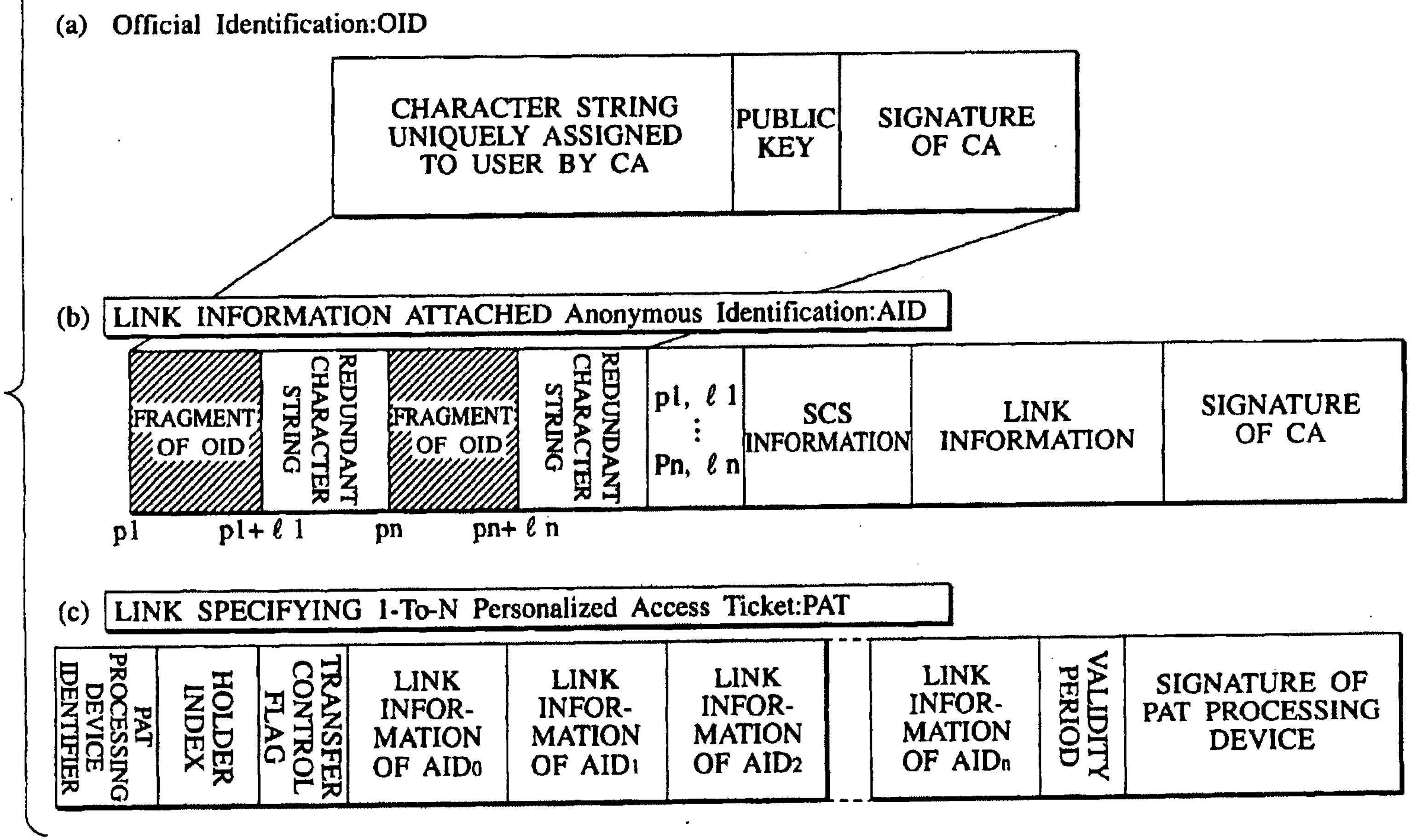

(a) Official Identification:OID
CHARACTER STRING UNIQUELY ASSIGNED TO USER BY CA
PUBLIC KEY
SIGNATURE OF CA
(b) LINK INFORMATION ATTACHED Anonymous Identification:AID
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
p1, ℓ 1
⋮
Pn, ℓ n
SCS INFORMATION
LINK INFORMATION
SIGNATURE OF CA
p1
p1+ ℓ 1
pn
pn+ ℓ n
(c) LINK SPECIFYING 1-To-N Personalized Access Ticket:PAT
PAT PROCESSING DEVICE IDENTIFIER
HOLDER INDEX
TRANSFER CONTROL FLAG
LINK INFOR-MATION OF AID0
LINK INFOR-MATION OF AID1
LINK INFOR-MATION OF AID2
LINK INFOR-MATION OF AIDn
VALIDITY PERIOD
SIGNATURE OF PAT PROCESSING DEVICE

FIG.41

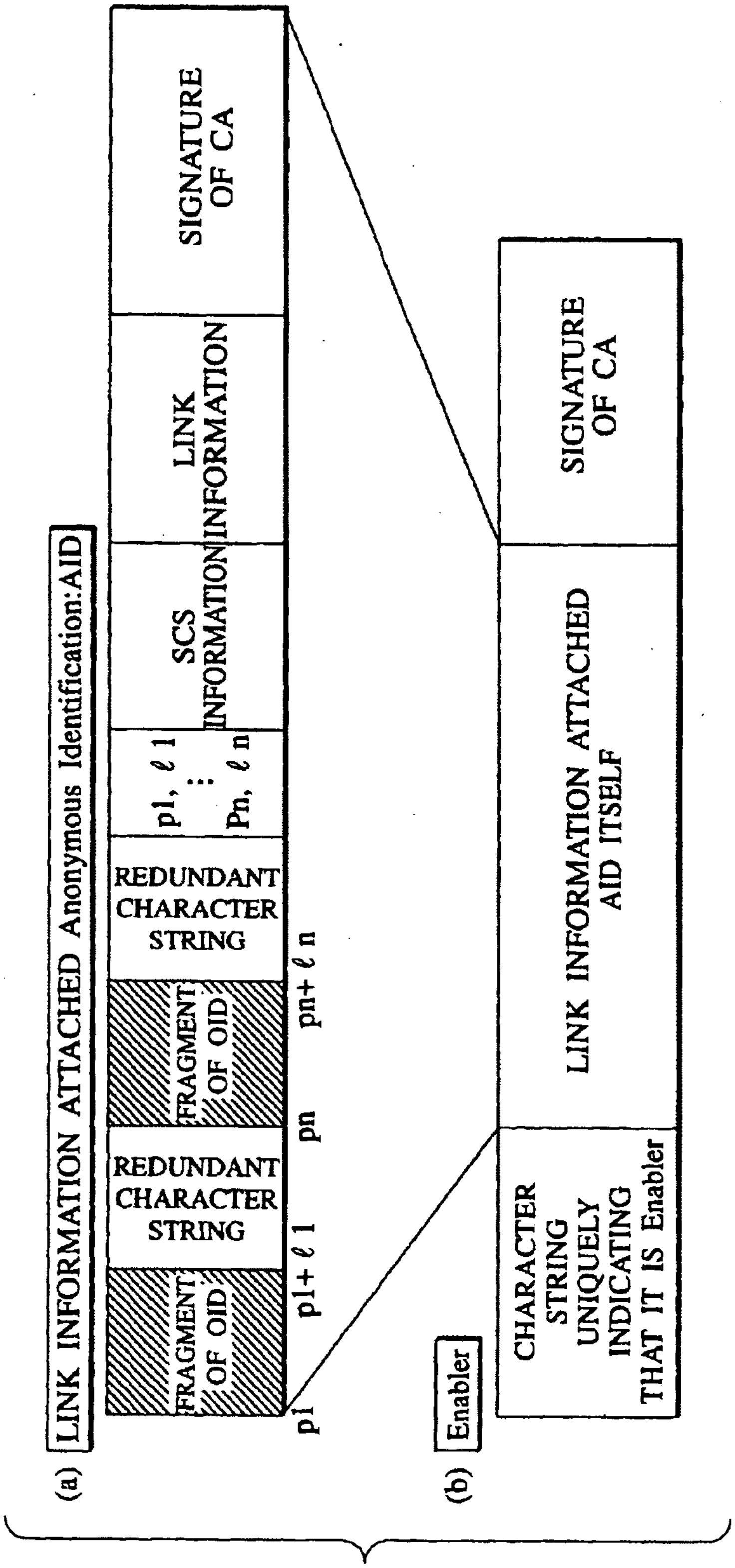

(a)
LINK INFORMATION ATTACHED Anonymous Identification:AID
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
p1, ℓ 1
:
Pn, ℓ n
SCS INFORMATION
LINK INFORMATION
SIGNATURE OF CA
p1
p1+ ℓ 1
pn
pn+ ℓ n
(b)
Enabler
CHARACTER STRING UNIQUELY INDICATING THAT IT IS Enabler
LINK INFORMATION ATTACHED AID ITSELF
SIGNATURE OF CA

FIG.42

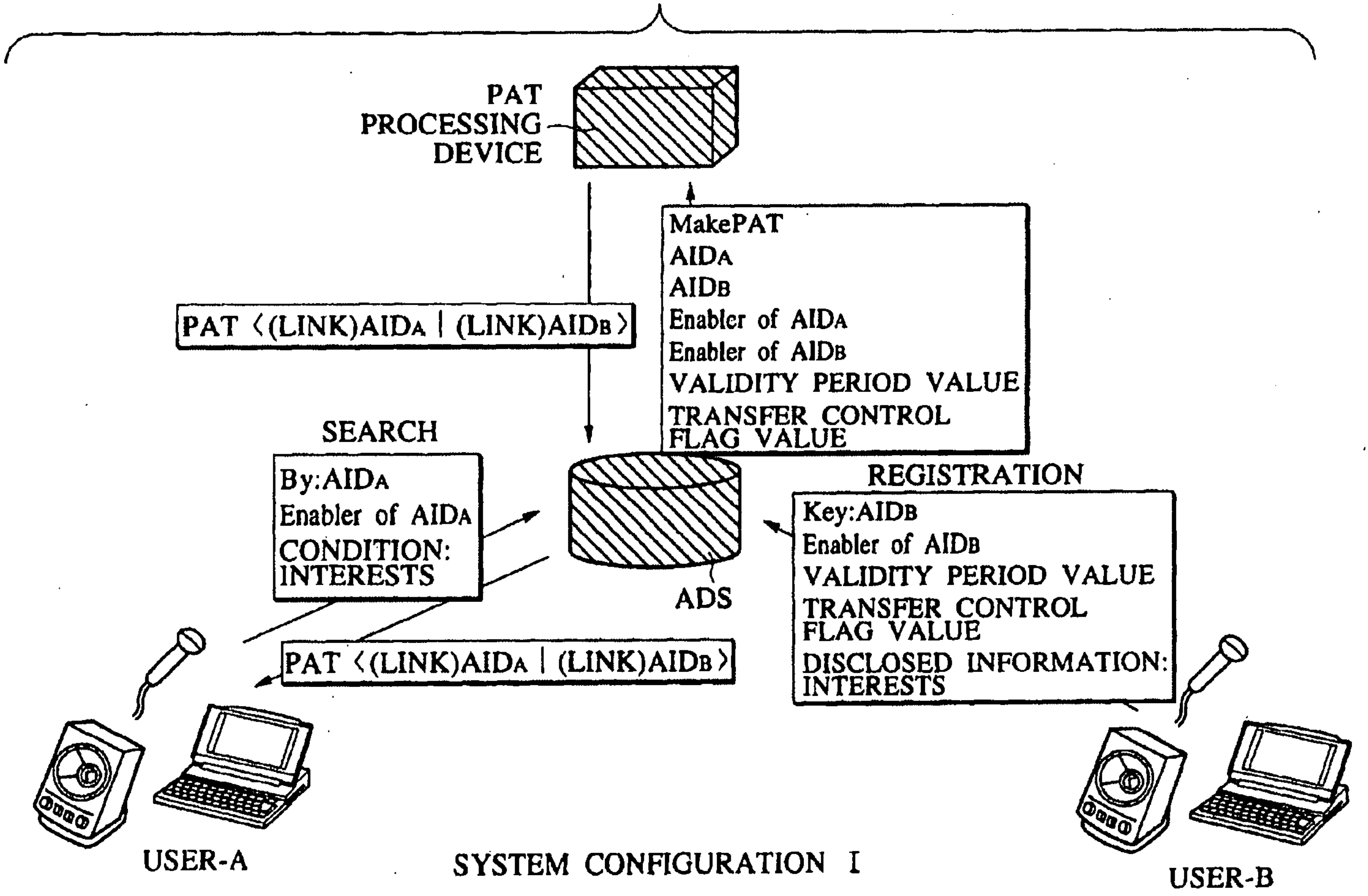
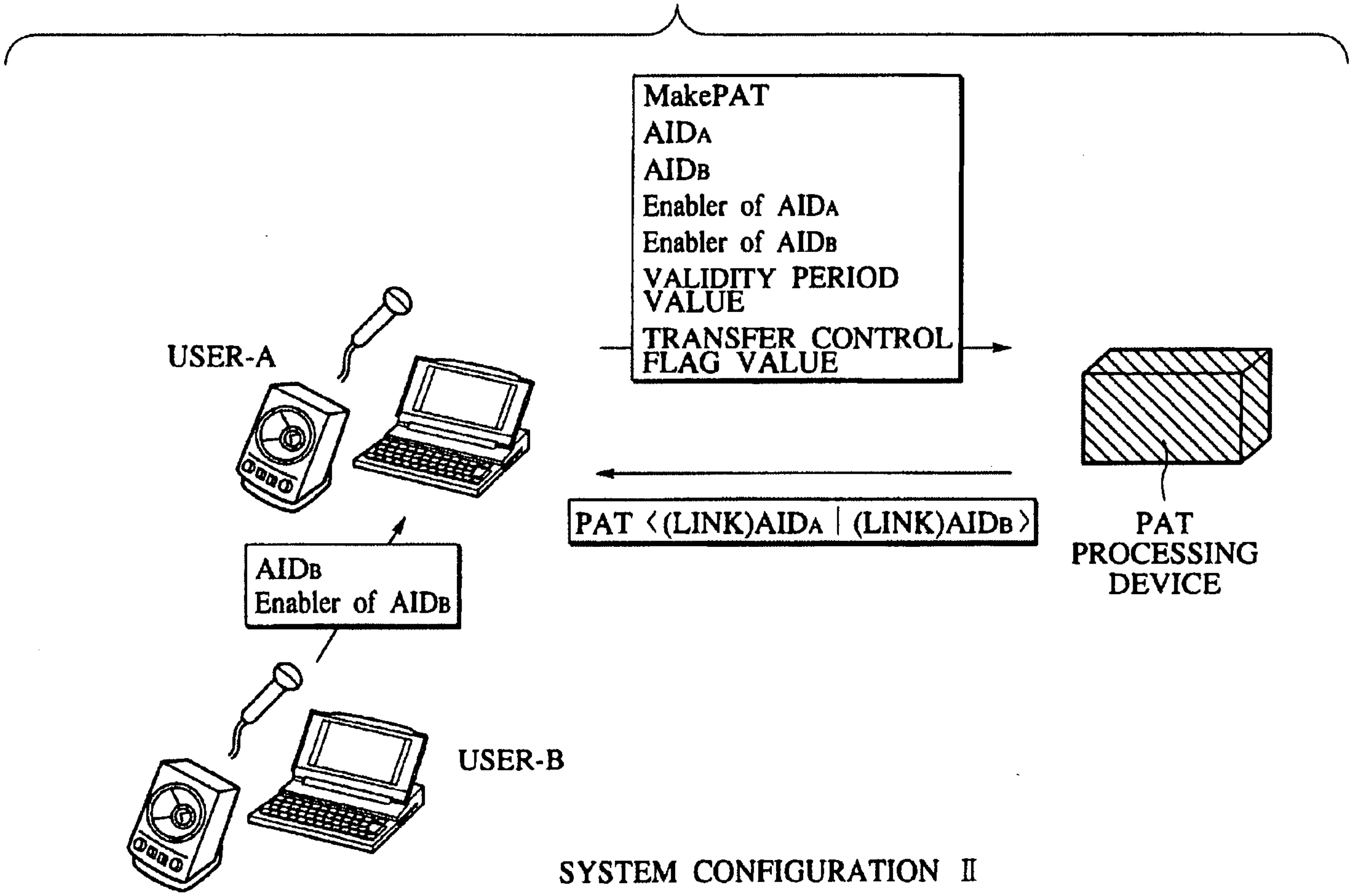
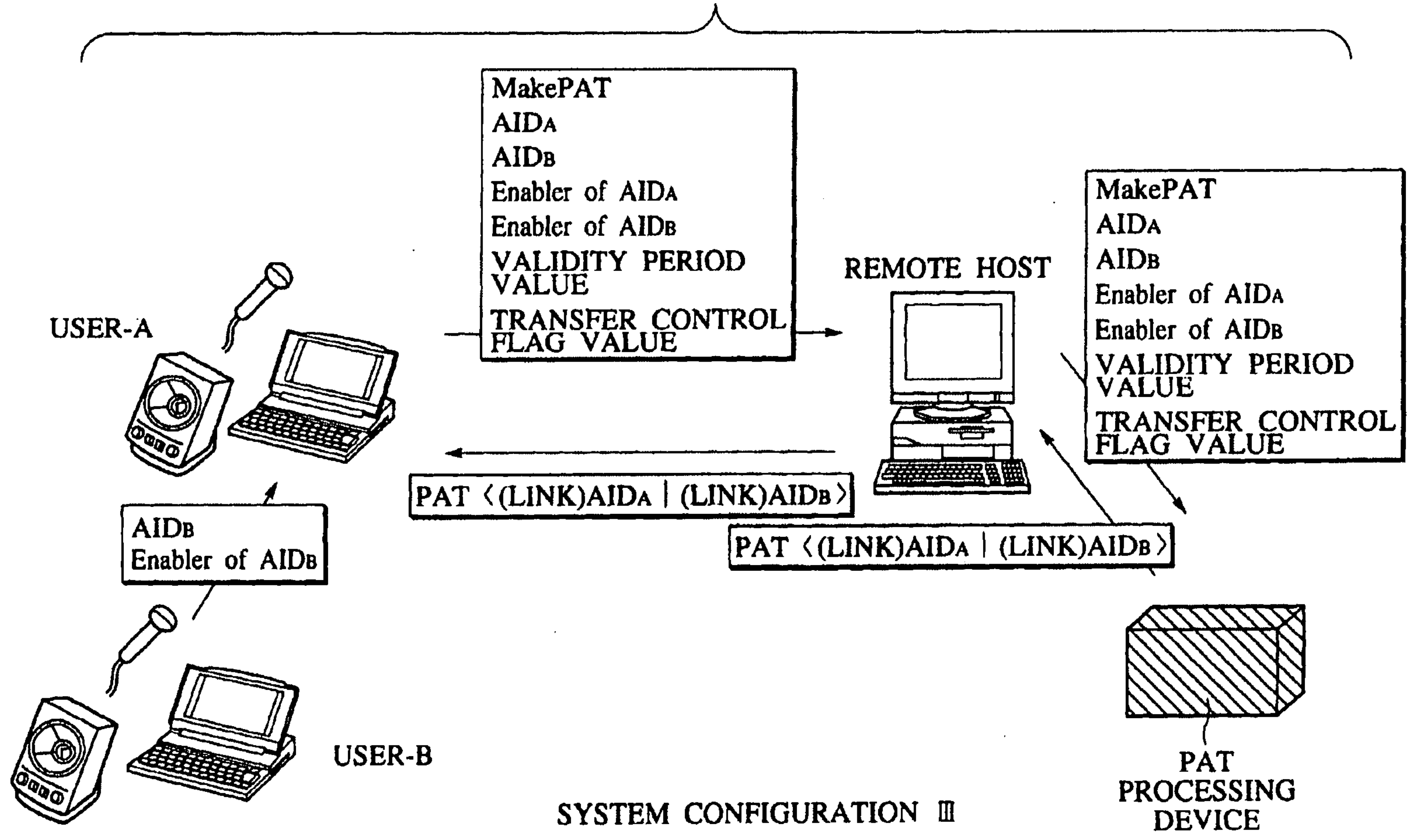

PAT
PROCESSING
DEVICE
MakePAT
AID$_A$
AID$_B$
Enabler of AID$_A$
Enabler of AID$_B$
VALIDITY PERIOD VALUE
TRANSFER CONTROL
FLAG VALUE
PAT 〈(LINK)AID$_A$ | (LINK)AID$_B$〉
SEARCH
By:AID$_A$
Enabler of AID$_A$
CONDITION:
INTERESTS
ADS
REGISTRATION
Key:AID$_B$
Enabler of AID$_B$
VALIDITY PERIOD VALUE
TRANSFER CONTROL
FLAG VALUE
DISCLOSED INFORMATION:
INTERESTS
PAT 〈(LINK)AID$_A$ | (LINK)AID$_B$〉
USER-A
SYSTEM CONFIGURATION I
USER-B

FIG.43


MakePAT
AIDA
AIDB
Enabler of AIDA
Enabler of AIDB
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
USER-A
PAT ⟨(LINK)AIDA | (LINK)AIDB⟩
PAT PROCESSING DEVICE
AIDB
Enabler of AIDB
USER-B

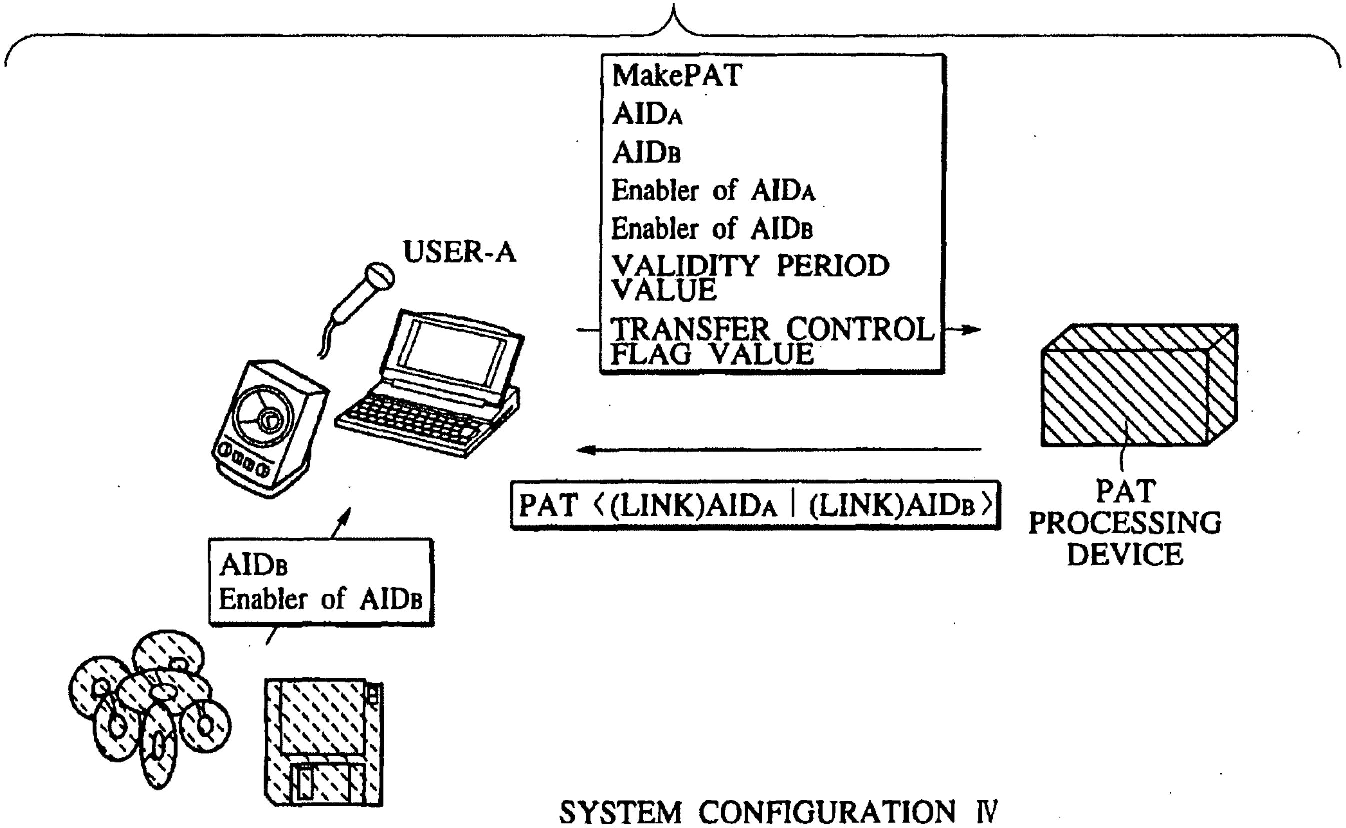
SYSTEM CONFIGURATION II

FIG.44


USER-A
MakePAT
AIDA
AIDB
Enabler of AIDA
Enabler of AIDB
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
REMOTE HOST
MakePAT
AIDA
AIDB
Enabler of AIDA
Enabler of AIDB
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
PAT 〈(LINK)AIDA | (LINK)AIDB〉
PAT 〈(LINK)AIDA | (LINK)AIDB〉
AIDB
Enabler of AIDB
USER-B
PAT PROCESSING DEVICE

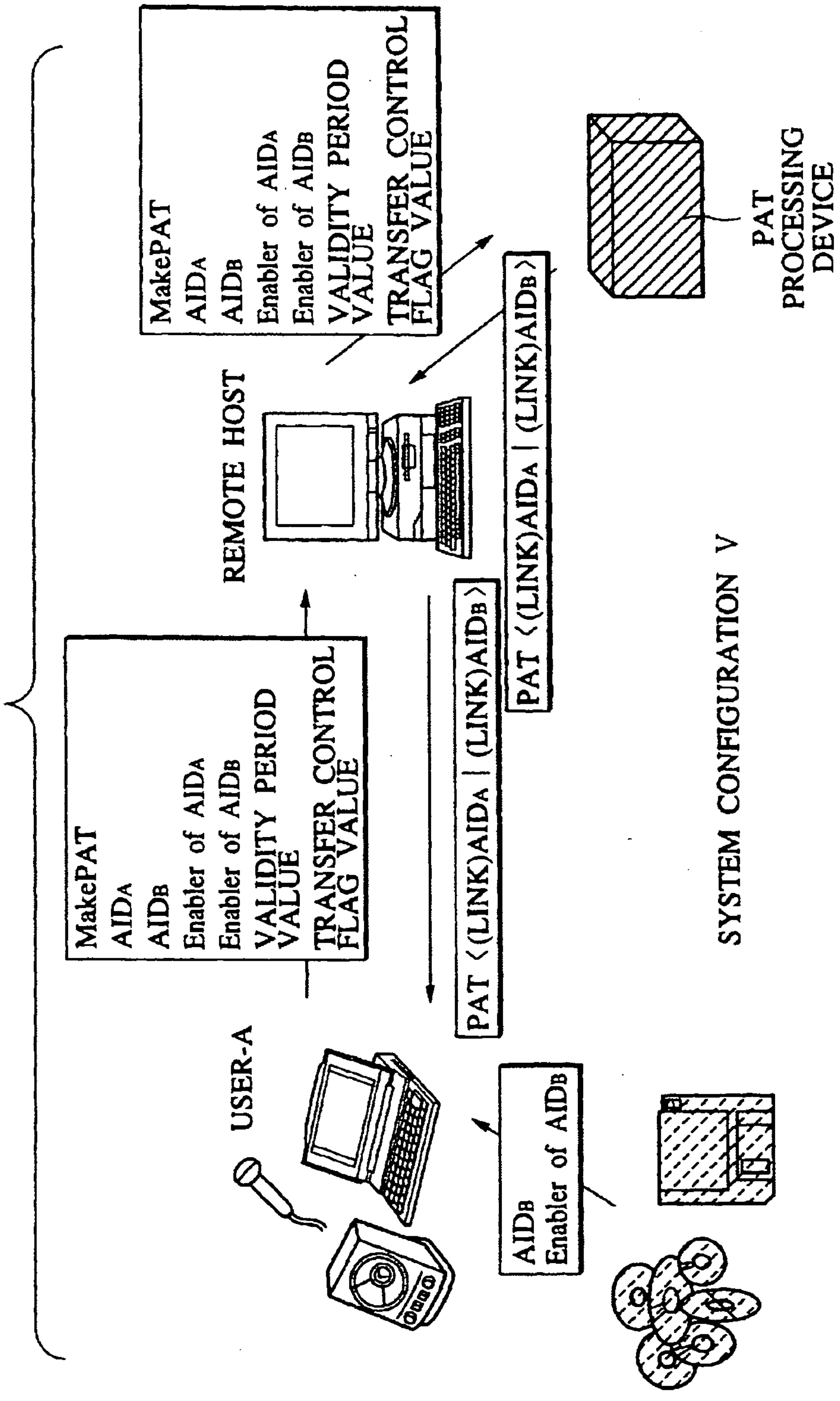
SYSTEM CONFIGURATION Ⅲ

FIG.45


USER-A
MakePAT
AID_A
AID_B
Enabler of AID_A
Enabler of AID_B
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
PAT 〈(LINK)AID_A | (LINK)AID_B〉
PAT PROCESSING DEVICE
AID_B
Enabler of AID_B

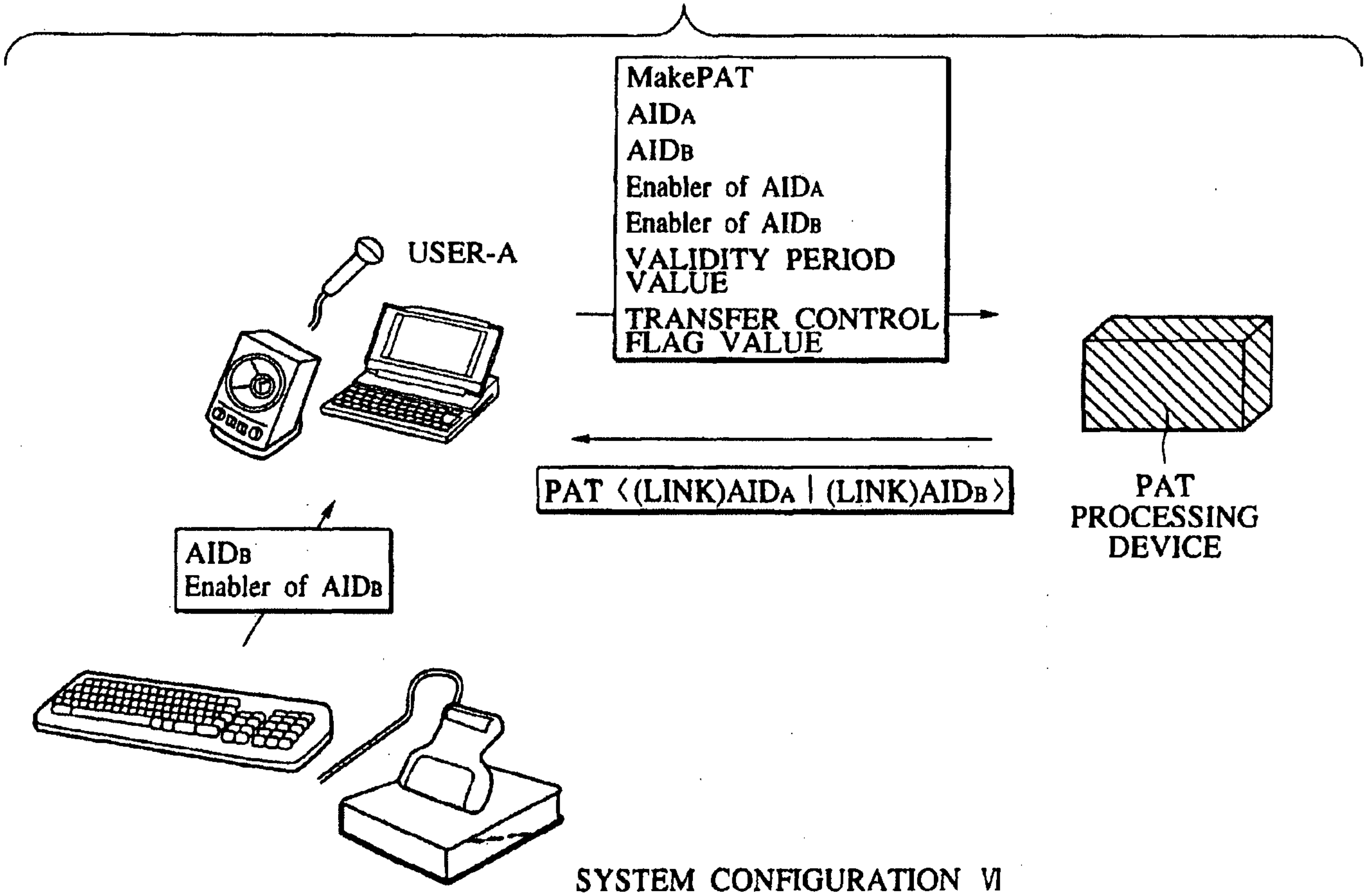
SYSTEM CONFIGURATION Ⅳ

FIG.46


MakePAT
AID$_A$
AID$_B$
Enabler of AID$_A$
Enabler of AID$_B$
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
REMOTE HOST
PAT 〈(LINK)AID$_A$ | (LINK)AID$_B$〉
PAT PROCESSING DEVICE
USER-A
MakePAT
AID$_A$
AID$_B$
Enabler of AID$_A$
Enabler of AID$_B$
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
PAT 〈(LINK)AID$_A$ | (LINK)AID$_B$〉
AID$_B$
Enabler of AID$_B$

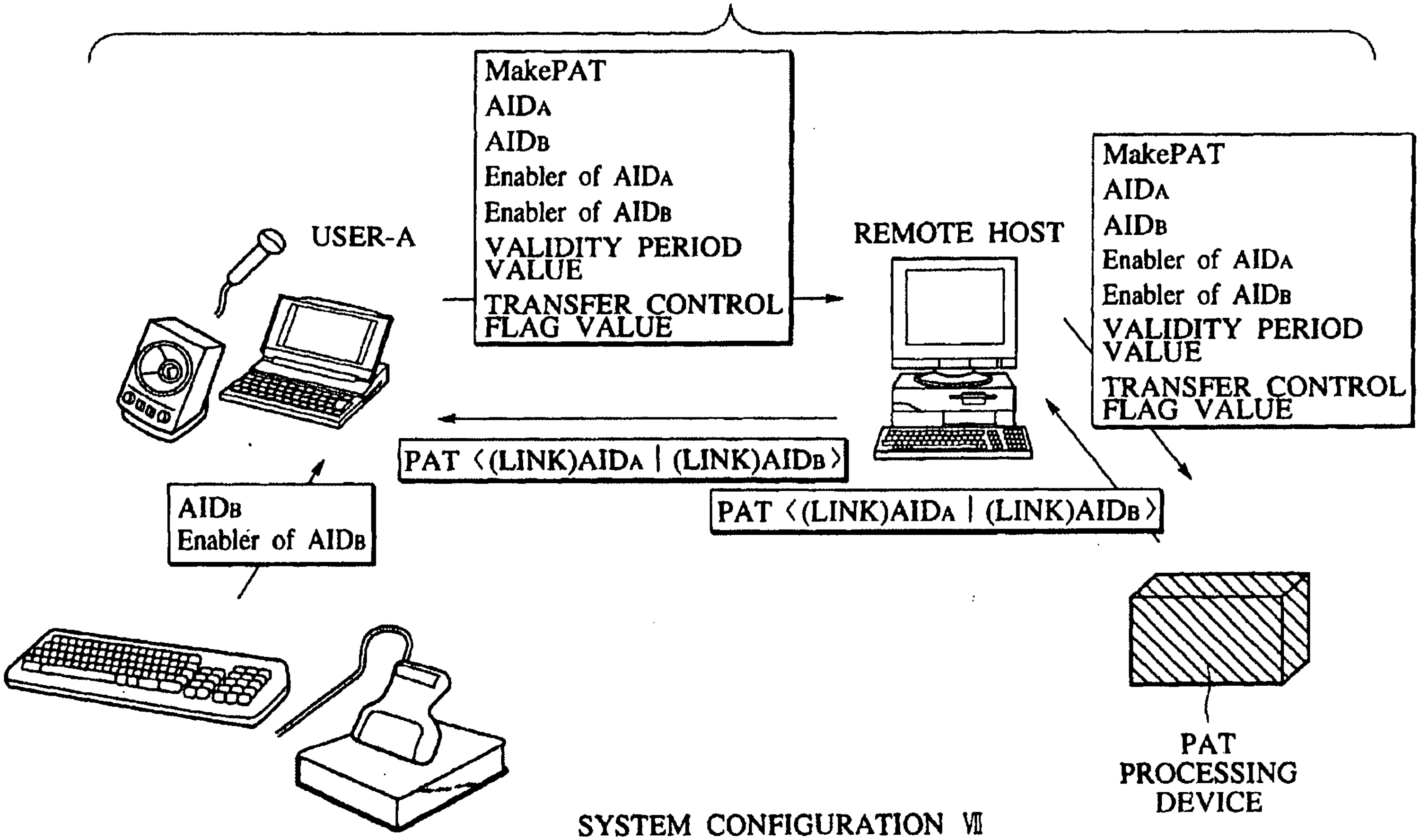
SYSTEM CONFIGURATION V

FIG.47


USER-A
MakePAT
AIDA
AIDB
Enabler of AIDA
Enabler of AIDB
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
PAT ⟨(LINK)AIDA | (LINK)AIDB⟩
PAT PROCESSING DEVICE
AIDB
Enabler of AIDB

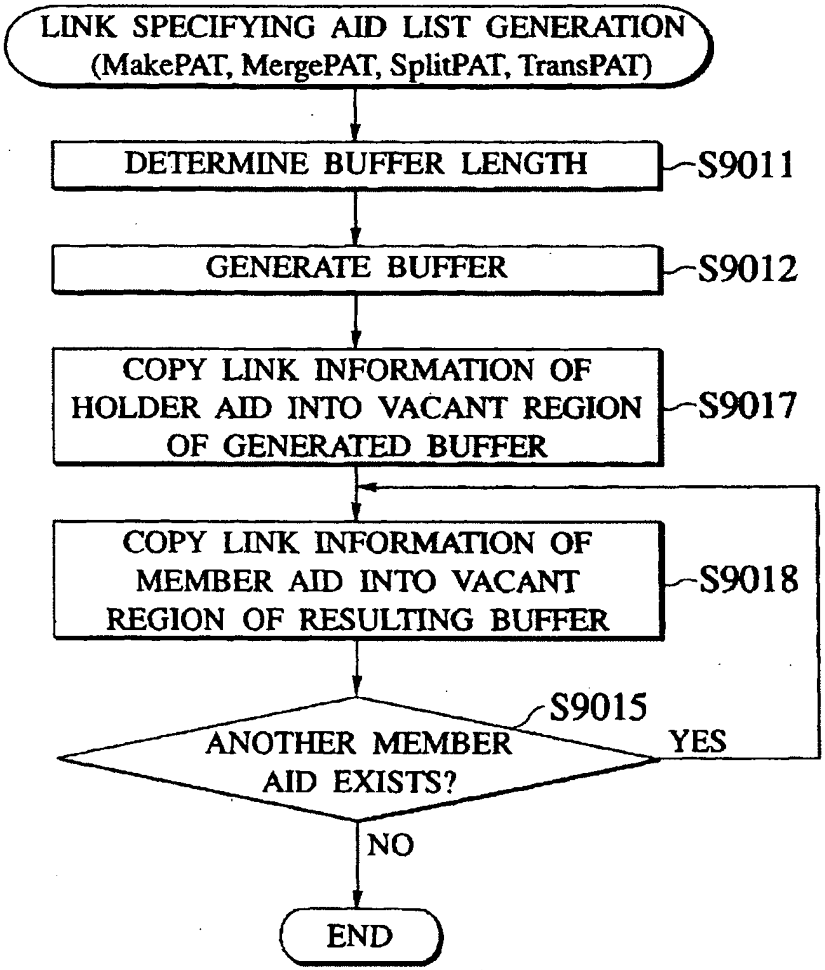
SYSTEM CONFIGURATION VI

FIG.48


MakePAT
AIDA
AIDB
Enabler of AIDA
Enabler of AIDB
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
USER-A
REMOTE HOST
MakePAT
AIDA
AIDB
Enabler of AIDA
Enabler of AIDB
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
PAT 〈(LINK)AIDA | (LINK)AIDB〉
PAT 〈(LINK)AIDA | (LINK)AIDB〉
AIDB
Enabler of AIDB
PAT PROCESSING DEVICE

SYSTEM CONFIGURATION Ⅶ

FIG.49


LINK SPECIFYING AID LIST GENERATION
(MakePAT, MergePAT, SplitPAT, TransPAT)
DETERMINE BUFFER LENGTH
S9011
GENERATE BUFFER
S9012
COPY LINK INFORMATION OF HOLDER AID INTO VACANT REGION OF GENERATED BUFFER
S9017
COPY LINK INFORMATION OF MEMBER AID INTO VACANT REGION OF RESULTING BUFFER
S9018
S9015
ANOTHER MEMBER AID EXISTS?
YES
NO
END

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 964 572 A1

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
15.12.1999 Bulletin 1999/50

(51) Int. Cl.^{6}: **H04N 5/00**

(21) Application number: **98401374.8**

(22) Date of filing: **08.06.1998**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(71) Applicant:
CANAL+ Société Anonyme
75711 Paris Cedex 15 (FR)

(72) Inventor: **Declerck, Christophe**
28210 Senantes (FR)

(74) Representative:
Cozens, Paul Dennis et al
Mathys & Squire
100 Grays Inn Road
London WC1X 8AL (GB)

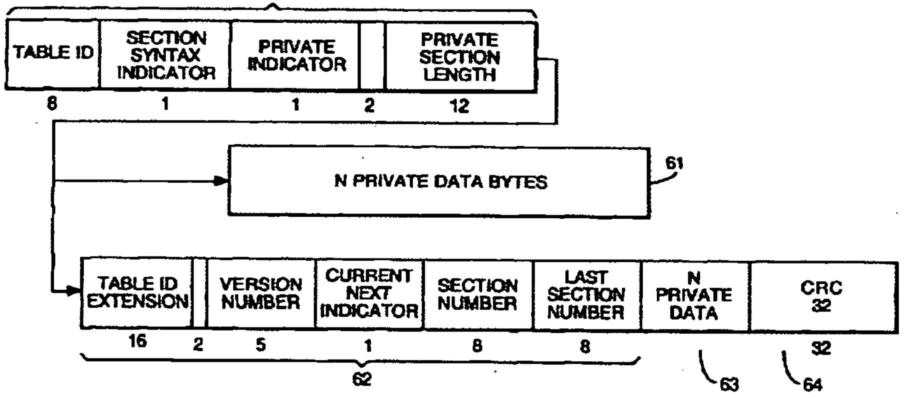
(54) Decoder and security module for a digital transmission system

(57) A decoder 12 in particular for a digital television system and adapted to receive a transport packet stream containing table or section data encapsulated within the packet payloads. The decoder is characterised in comprising a means 80 for filtering table or section data configurable in response to filter data received from a portable security module 30 such as a smart card.

The invention equally extends to a portable security module 30 including a memory holding such data as is necessary to configure the table or section filter 80, and a method for processing a transport packet stream including encapsulated table and section data using such a decoder 12 and security module 30.

In a preferred embodiment, the filter 80 is adapted to filter out conditional access messages in response to the table or section filter data received from the portable security module 30, these messages being thereafter forwarded to the security module for processing.

Fig.4.


60
TABLE ID
SECTION SYNTAX INDICATOR
PRIVATE INDICATOR
PRIVATE SECTION LENGTH
8
1
1
2
12
N PRIVATE DATA BYTES
61
TABLE ID EXTENSION
VERSION NUMBER
CURRENT NEXT INDICATOR
SECTION NUMBER
LAST SECTION NUMBER
N PRIVATE DATA
CRC 32
16
2
5
1
8
8
32
62
63
64

EP 0 964 572 A1

Printed by Xerox (UK) Business Services
2.16.7/3.6

Description

[0001] The present invention relates to a decoder and security module for a digital transmission system and method of operating a decoder and security module, in particular for use in a digital television system.

[0002] Conventional digital television broadcast systems transmit data in the form of discrete transport stream packets or transport packets, each packet being of a predetermined length and containing a header and a payload. The MPEG standard is the currently favoured standard in this domain and sets out, amongst other things, a predetermined format for such packets.

[0003] The packet header comprises general descriptive data regarding the packet, whilst the payload comprises the data to be processed at the receiver. The packet header includes at least a packet ID or PID identifying the packet. The payload of the packet may contain audio, video or other data such as application data or, in particular, conditional access system data.

[0004] Conventionally, the incoming data stream is filtered by a receiver/decoder according to the PID of each packet. Data requiring immediate processing such as audio or visual data is communicated to an appropriate processor in the form of what is conventionally known as a packetised elementary stream or PES. This continuous flux of data, which is formed by assembling the payloads of the transport packets, itself comprises a sequence of packets, each PES packet comprising a packet header and payload.

[0005] Other data not requiring immediate processing may also be encapsulated within the payloads of the transport packets. Unlike PES data, which is treated immediately by a processor to generate a real time output, this sort of data is typically processed in an asynchronous manner by the decoder processor. In this case, data is formatted in a single table or a series of sections or tables, each including a header and a payload, the header of the section or table including a table ID or TID.

[0006] In the case where the access to a transmission is to be restricted, for example, in a pay TV system, conditional access data may be included in a table or section broadcast in the transport stream with the transmission. This conditional access data is filtered by the receiver/decoder and passed to a portable security module, such as smart card, inserted in the decoder. The data is then processed by the smart card in order to generate, for example, a control word subsequently used by the decoder to descramble a transmission.

[0007] One problem with known systems lies in the volume of data that will be received and processed by the receiver/decoder and notably the volume of conditional access messages eventually forwarded to the smart card or security module. In particular, the processing capabilities of a smart card processor and the capacity of the communication channel between the decoder and smart card may be insufficient to handle a given volume of messages. This problem is exacerbated by the increasing tendency for programmes to be transmitted with multiple conditional access messages enabling access by different operators to the same programme (e.g. a football match or a thematic television channel).

[0008] According to the present invention, there is provided a decoder for a digital transmission system adapted to receive a transport packet stream containing table, section or other packetised data encapsulated within the packet payloads and characterised in that the decoder comprises a means for filtering the encapsulated data configurable in response to filter data received from a portable security module.

[0009] Filtering data at the table or section level in response to information from the security module enables a more precise identification and selection of data to be carried out, for example, to extract relevant conditional access messages addressed to the module. In practice, and as will be described below, this filtering at the table or section level may be carried out after and in addition to a filtering carried out at the transport packet level.

[0010] Preferably, the means for filtering encapsulated data is configurable in response to filter data comprising at least a table ID or section ID value transmitted by the portable security module. The means for filtering encapsulated data may equally be configurable in accordance with other data received from the portable security module.

[0011] In a preferred embodiment, the means for filtering encapsulated data is further adapted to forward to the security module conditional access data obtained in accordance with the filter data received from the security module.

[0012] Whilst the present invention is particularly adapted to enable a reduction of the volume of conditional access messages communicated between the decoder and the module, it will be nevertheless appreciated that the encapsulated data may be configured by the security module to extract data other than conditional access data and having a destination other than the security module.

[0013] Conditional access data filtered and forwarded to the security module may comprise entitlement control messages (ECMs) and/or entitlement management messages (EMMs).

[0014] Even within a group of messages associated with a single conditional access system there may be a large number of messages irrelevant to a particular user within that system. For example, within a single conditional access system a number of different groups of users may be defined leading to the generation of a number of EMMs, not all of which may be relevant to a given user.

[0015] Preferably therefore, filter data provided by the security module comprises data used by the filter means to extract group and/or individual entitlement management messages addressed to the security module.

[0016] In one embodiment, the decoder is adapted to receive a control word generated by the security module in response to the conditional access data forwarded thereto, the control word being used by the decoder to descramble a scrambled transmission.

[0017] In addition to a filtering at the table or section level, the decoder may further carry out a transport level filtering in order, for example, to extract only these packets comprising data associated with the particular conditional access system used by the security module. Preferably, therefore the decoder further comprises a means for filtering transport packet data configurable in response to data received from the security module.

[0018] Advantageously, the means for filtering transport packet data may be configurable in response to data representing the identity of the conditional access system received from the security module.

[0019] In one embodiment, the transport packet filtering means is adapted to extract transport packets containing a program map table and a conditional access table, the decoder further comprising selection means adapted to receive the program map table and conditional access table from the transport packet filtering means and conditional access identity data from the security module and thereafter configure the transport packet filtering means to extract transport packet data associated with the conditional access system in question.

[0020] In order to preserve security in the system, some or all communications between the security module and the decoder may be encrypted. In particular, the descrambling control word generated by the security module and eventually transmitted to the decoder may be encrypted.

[0021] The present invention has been described above in relation to a decoder. Other aspects of the invention relate to a method of filtering encapsulated data in a transport packet stream and a security module for use with a decoder or method of the present invention. In one embodiment, the security module may conveniently comprise a smart card.

[0022] Whilst the present invention may apply to any packet transmission system comprising a transport stream layer and a table or section layer, the present invention is particularly applicable to a decoder adapted to receive an MPEG compatible data stream.

[0023] In this regard, the term "table, section or other packetised data" refers in its broadest sense to any data table, alone or in a sequence, and comprising a header and payload and that is itself encapsulated within a transport packet stream. As will be described in the preferred embodiment, the present invention is particularly applicable to filtering of data contained within an MPEG table, notably a single MPEG short form table. Other embodiments are nevertheless conceivable, for example, in which filtering is carried out on PES packets encapsulated within the transport packet payloads.

[0024] In the context of this application, the term MPEG refers to the data transmission standards developed by the International Standards Organisation working group "Motion Pictures Expert Group" and in particular but not exclusively the MPEG-2 standard developed for digital television applications and set out in the documents ISO 13818-1, ISO 13818-2, ISO 13818-3 and ISO 13818-4. In the context of the present patent application, the term MPEG includes all variants, modifications or developments of MPEG formats applicable to the field of digital data transmission.

[0025] As used herein, the term "smart card" includes, but not exclusively so, any chip-based card device, or object of similar function and performance, possessing, for example, microprocessor and/or memory storage. Included in this term are devices having alternative physical forms to a card, for example key-shaped devices such as are often used in TV decoder systems.

[0026] The term "decoder" or "receiver/decoder" used herein may connote a receiver for receiving either encoded or non-encoded signals, for example, television and/or radio signals, which may be broadcast or transmitted by some other means. Embodiments of such receiver/decoders may include a decoder integral with the receiver for decoding the received signals, for example, in a "set-top box", a decoder functioning in combination with a physically separate receiver, as well as a decoder including additional functions, such as a web browser or integrated with a video recorder or a television.

[0027] As used herein, the term "digital transmission system" includes any transmission system for transmitting or broadcasting digital data, for example primarily audiovisual or multimedia digital data. Whilst the present invention is particularly applicable to a broadcast digital television system, the invention may also be applicable to a fixed telecommunications network for multimedia internet applications, to a closed circuit television, and so on.

[0028] As used herein, the term "digital television system" includes for example any satellite, terrestrial, cable and other system.

[0029] There will now be described, by way of example only, a preferred embodiment of the invention, with reference to the following figures, in which:

Figure 1 shows the overall architecture of a digital TV system according to this embodiment;

Figure 2 shows the architecture of the conditional access system of Figure 1;

Figure 3 shows the hierarchy of MPEG-2 packets, in particular those associated with conditional access messages;

Figure 4 shows the structure of long form and short form MPEG-2 private sections;

Figure 5 shows the elements of a receiver/decoder for use in this embodiment;

Figure 6 shows the elements of the receiver/decoder used to process the transport stream, in particular in relation to conditional access messages; and

Figure 7 shows the structure of the PID and section filters of the filter unit of Fig. 6.

[0030] An overview of a digital television broadcast and reception system 1 is shown in Figure 1. The invention includes a mostly conventional digital television system 2 which uses the MPEG-2 compression system to transmit compressed digital signals. In more detail, MPEG-2 compressor 3 in a broadcast centre receives a digital signal stream (for example a stream of audio or video signals). The compressor 3 is connected to a multiplexer and scrambler 4 by linkage 5. The multiplexer 4 receives a plurality of further input signals, assembles one or more transport streams and transmits compressed digital signals to a transmitter 6 of the broadcast centre via linkage 7, which can of course take a wide variety of forms including telecom links.

[0031] The transmitter 6 transmits electromagnetic signals via uplink 8 towards a satellite transponder 9, where they are electronically processed and broadcast via a national downlink 10 to earth receiver 11, conventionally in the form of a dish owned or rented by the end user. The signals received by receiver 11 are transmitted to an integrated receiver/decoder 12 owned or rented by the end user and connected to the end user's television set 13. The receiver/decoder 12 decodes the compressed MPEG-2 signal into a television signal for the television set 13.

[0032] A conditional access system 20 is connected to the multiplexer 4 and the receiver/decoder 12, and is located partly in the broadcast centre and partly in the decoder. It enables the end user to access digital television broadcasts from one or more broadcast suppliers. A smartcard, capable of decrypting messages relating to commercial offers (that is, one or several television programmes sold by the broadcast supplier), can be inserted into the receiver/decoder 12. Using the decoder 12 and smartcard, the end user may purchase events in either a subscription mode or a pay-per-view mode.

[0033] An interactive system 17, also connected to the multiplexer 4 and the receiver/decoder 12 and again located partly in the broadcast centre and partly in the decoder, may be provided to enable the end user to interact with various applications via a modemmed back channel 16.

[0034] The conditional access system 20 will now be described in more detail.

[0035] With reference to Figure 2, in overview the conditional access system 20 includes a Subscriber Authorization System (SAS) 21. The SAS 21 is connected to one or more Subscriber Management Systems (SMS) 22, one SMS for each broadcast supplier, by a respective TCP-IP linkage 23 (although other types of linkage could alternatively be used). Alternatively, one SMS could be shared between two broadcast suppliers, or one supplier could use two SMSs, and so on.

[0036] First encrypting units in the form of ciphering units 24 utilising "mother" smartcards 25 are connected to the SAS by linkage 26. Second encrypting units again in the form of ciphering units 27 utilising mother smartcards 28 are connected to the multiplexer 4 by linkage 29. The receiver/decoder 12 receives a "daughter" smartcard 30. It is connected directly to the SAS 21 by Communications Servers 31 via the modemmed back channel 16. The SAS sends, amongst other things, subscription rights to the daughter smartcard on request.

[0037] The smartcards contain the secrets of one or more commercial operators. The "mother" smartcard encrypts different kinds of messages and the "daughter" smartcards decrypt the messages, if they have the rights to do so.

[0038] The first and second ciphering units 24 and 27 comprise a rack, an electronic VME card with software stored on an EEPROM, up to 20 electronic cards and one smartcard 25 and 28 respectively, for each electronic card, one card 28 for encrypting the ECMs and one card 25 for encrypting the EMMs.

[0039] The operation of the conditional access system 20 of the digital television system will now be described in more detail with reference to the various components of the television system 2 and the conditional access system 20.

Multiplexer and Scrambler

[0040] With reference to Figures 1 and 2, in the broadcast centre, the digital audio or video signal is first compressed (or bit rate reduced), using the MPEG-2 compressor 3. This compressed signal is then transmitted to the multiplexer and scrambler 4 via the linkage 5 in order to be multiplexed with other data, such as other compressed data.

[0041] The scrambler generates a control word used in the scrambling process and included in the MPEG-2 stream in the multiplexer. The control word is generated internally and enables the end user's integrated receiver/decoder 12 to descramble the programme.

[0042] Access criteria, indicating how the programme is commercialised, are also added to the MPEG-2 stream. The programme may be commercialised in either one of a number of "subscription" modes and/or one of a number of "Pay Per View" (PPV) modes or events. In the subscription mode, the end user subscribes to one or more commercial offers, or "bouquets", thus getting the rights to watch every channel inside those bouquets. In the preferred embodiment, up to 960 commercial offers may be selected from a bouquet of channels.

[0043] In the Pay Per View mode, the end user is provided with the capability to purchase events as he wishes. This can be achieved by either pre-booking the event in advance ("pre-book mode"), or by purchasing the event as soon as it is broadcast ("impulse mode"). In the preferred embodiment, all users are subscribers, whether or not they watch in subscription or PPV mode, but of course PPV viewers need not necessarily be subscribers.

Entitlement Control Messages

[0044] Both the control word and the access criteria are used to build an Entitlement Control Message (ECM). This is a message sent in relation with a scrambled program; the message contains a control word (which allows for the descrambling of the program) and the access criteria of the broadcast program. The access criteria and control word are transmitted to the second encrypting unit 27 via the linkage 29. In this unit, an ECM is generated, encrypted and transmitted on to the multiplexer and scrambler 4. During a broadcast transmission, the control word typically changes every few seconds, and so ECMs are also periodically transmitted to enable the changing control word to be descrambled. For redundancy purposes, each ECM typically includes two control words; the present control word and the next control word.

[0045] Each service broadcast by a broadcast supplier in a data stream comprises a number of distinct components; for example a television programme includes a video component an audio component, a sub-title component and so on. Each of these components of a service is individually scrambled and encrypted for subsequent broadcast to the transponder 9. In respect of each scrambled component of the service, a separate ECM is required. Alternatively, a single ECM may be required for all of the scrambled components of a service. Multiple ECMs are also generated in the case where multiple conditional access systems control access to the same transmitted program.

Programme Transmission

[0046] The multiplexer 4 receives electrical signals comprising encrypted EMMs from the SAS 21, encrypted ECMs from the second encrypting unit 27 and compressed programmes from the compressor 3. The multiplexer 4 scrambles the programmes and sends the scrambled programmes, the encrypted EMMs and the encrypted ECMs to a transmitter 6 of the broadcast centre via the linkage 7. The transmitter 6 transmits electromagnetic signals towards the satellite transponder 9 via uplink 8.

Programme Reception

[0047] The satellite transponder 9 receives and processes the electromagnetic signals transmitted by the transmitter 6 and transmits the signals on to the earth receiver 11, conventionally in the form of a dish owned or rented by the end user, via downlink 10. The signals received by receiver 11 are transmitted to the integrated receiver/decoder 12 owned or rented by the end user and connected to the end user's television set 13. The receiver/decoder 12 demultiplexes the signals to obtain scrambled programmes with encrypted EMMs and encrypted ECMs.

[0048] If the programme is not scrambled, that is, no ECM has been transmitted with the MPEG-2 stream, the receiver/decoder 12 decompresses the data and transforms the signal into a video signal for transmission to television set 13.

[0049] If the programme is scrambled, the receiver/decoder 12 extracts the corresponding ECM from the MPEG-2 stream and passes the ECM to the "daughter" smartcard 30 of the end user. This slots into a housing in the receiver/decoder 12. The daughter smartcard 30 controls whether the end user has the right to decrypt the ECM and to access the programme. If not, a negative status is passed to the receiver/decoder 12 to indicate that the programme cannot be descrambled. If the end user does have the rights, the ECM is decrypted and the control word extracted. The decoder 12 can then descramble the programme using this control word. The MPEG-2 stream is decompressed and translated into a video signal for onward transmission to television set 13.

Entitlement Management Messages (EMMs)

[0050] The EMM is a message dedicated to an individual end user (subscriber), or a group of end users. Each group may contain a given number of end users. This organisation as a group aims at optimising the bandwidth; that is, access to one group can permit the reaching of a great number of end users.

[0051] Various specific types of EMM can be used. Individual EMMs are dedicated to individual subscribers, and are typically used in the provision of Pay Per View services; these contain the group identifier and the position of the subscriber in that group.

[0052] Group subscription EMMs are dedicated to groups of, say, 256 individual users, and are typically used in the administration of some subscription services. This EMM has a group identifier and a subscribers' group bitmap.

[0053] Audience EMMs are dedicated to entire audiences, and might for example be used by a particular operator to provide certain free services. An "audience" is the totality of subscribers having smartcards which bear the same conditional access system identifier (CA ID). Finally, a "unique" EMM is addressed to the unique identifier of the smartcard.

Subscriber Management System (SMS)

[0054] A Subscriber Management System (SMS) 22 includes a database 32 which manages, amongst others, all of the end user files, commercial offers, subscriptions, PPV details, and data regarding end user consumption and authorization. The SMS may be physically remote from the SAS.

[0055] Each SMS 22 transmits messages to the SAS 21 via respective linkage 23 which imply modifications to or creations of Entitlement Management Messages (EMMs) to be transmitted to end users.

[0056] The SMS 22 also transmits messages to the SAS 21 which imply no modifications or creations of EMMS but imply only a change in an end users state (relating to the authorization granted to the end user when ordering products or to the amount that the end user will be charged).

[0057] The SAS 21 sends messages (typically requesting information such as call-back information or billing information) to the SMS 22, so that it will be apparent that communication between the two is two-way.

Subscriber Authorization System (SAS)

[0058] The messages generated by the SMS 22 are passed via linkage 23 to the Subscriber Authorization System (SAS) 21, which in turn generates messages acknowledging receipt of the messages generated by the SMS 21 and passes these acknowledgements to the SMS 22.

[0059] In overview the SAS comprises a Subscription Chain area to give rights for subscription mode and to renew the rights automatically each month, a Pay Per View Chain area to give rights for PPV events, and an EMM Injector for passing EMMs created by the Subscription and PPV chain areas to the multiplexer and scrambler 4, and hence to feed the MPEG stream with EMMs. If other rights are to be granted, such as Pay Per File (PPF) rights in the case of downloading computer software to a user's Personal Computer, other similar areas are also provided.

[0060] One function of the SAS 21 is to manage the access rights to television programmes, available as commercial offers in subscription mode or sold as PPV events according to different modes of commercialisation (pre-book mode, impulse mode). The SAS 21, according to those rights and to information received from the SMS 22, generates EMMs for the subscriber.

[0061] The EMMs are passed to the Ciphering Unit (CU) 24 for ciphering with respect to the management and exploitation keys. The CU completes the signature on the EMM and passes the EMM back to a Message Generator (MG) in the SAS 21, where a header is added. The EMMs are passed to a Message Emitter (ME) as complete EMMs. The Message Generator determines the broadcast start and stop time and the rate of emission of the EMMs, and passes these as appropriate directions along with the EMMs to the Message Emitter. The MG only generates a given EMM once; it is the ME which performs cyclic transmission of the EMMs.

[0062] On generation of an EMM, the MG assigns a unique identifier to the EMM. When the MG passes the EMM to the ME, it also passes the EMM ID. This enables identification of a particular EMM at both the MG and the ME.

[0063] In systems such as simulcrypt which are adapted to handle multiple conditional access systems e.g. associated with multiple operators, EMM streams associated with each conditional access system are generated separately and multiplexed together by the multiplexer 4 prior to transmission.

Conditional Access Messages in the Transport Stream

[0064] The different nature of ECM and EMM messages leads to differences vis à vis the mode of transmission of the messages in the MPEG transport stream. ECM messages, which carry the control words needed to descramble a programme are necessarily linked to the video and audio streams of the programme being transmitted, in contrast EMM messages are general messages broadcast asynchronously to transmit rights information to individual or groups of customers. This difference is reflected in the placing of ECM and EMM messages within the MPEG transport stream.

[0065] As is known, MPEG transport packets are of a fixed length of 188 bytes including a header. In a standard packet, the three bytes of the header following the synchronisation data comprise:

TABLE I

Transport error indicator	1 bit
Payload unit indicator	1 bit
Transport priority	1 bit
PID	13 bits
Transport scrambling control	2 bits
Adaptation field control	2 bits
Continuity counter	4 bits

[0066] The characteristics of these fields are largely determined by the MPEG standard.

[0067] Referring to Figure 3, the organisation of data within a transport stream will be described. As shown, the transport stream contains a programme association table 40 ("PAT"), the PID in the header of the packet being fixed by the MPEG-2 standard at a value of 0x00. The programme access table 40 provides the entry point for access to programme data and contains a table referring to the PID values of the programme map tables ("PMT") 41, 42 associated with a number of programmes. Each programme map table 41, 42 contains in turn a reference to the PID values of the packet streams of the audio tables 43 and video tables 44 of that programme.

[0068] As shown, the programme map table 42 also contains references to the PID values of other packets 45, 46 containing additional data relating to the programme in question. In the present case ECM data generated by a number of conditional access systems and associated with the programme in question is contained within the referred packets 45, 46.

[0069] In addition to the programme access table PAT 40, the MPEG transport stream further comprises a conditional access table 47 ("CAT"), the PID value of which is fixed at 0x01. Any packet headers containing this PID value are thus automatically identified as containing access control information. The CAT table 47 refers to the PID values of MPEG packets 48, 49, 50 associated with EMM data associated with one or more conditional access systems. As with the PMT packets, the PID values of the EMM packets referred to in the CAT table are not fixed and may be determined at the choice of the system operator.

Private Section Data

[0070] In conformity with the MPEG-2 standard, information contained with a packet payload is subject to a further level of structure according to the type of data being transported. In the case of audio, visual, teletext, subtitle or other such rapidly evolving and synchronised data, the information is assembled in the form of what is known as a packetised elementary stream or PES. This data stream, which is formed by assembling the payloads of the transmitted packets, itself comprises a sequence of packets, each packet comprising a packet header and payload. Unlike the transmitted packets in the transport stream, the length of PES packets is variable.

[0071] In the case of other data, such as application data or, in this example, ECM and EMM data, a different format from PES packeting is proscribed. In particular, data contained in the transport packet payload is divided into a series of sections or tables, the table or section header including a table ID or TID identifying the table in question. Depending on the size of the data, a section may be contained entirely within a packet payload or may be extended in a series of tables over a number of transport packets. In the MPEG-2 context, the term "table" is often used to refer to a single table of data, whilst "section" refers to one of a plurality of tables with the same TID value.

[0072] As with transport packet data and PES packet data, the data structure of a table or section is additionally defined by the MPEG-2 standard. In particular, two possible syntax forms for private table or section data are proposed; a long form or a short form, as illustrated in Figure 4.

[0073] In both the short and long form, the header includes at least the data 60 comprising:

TABLE II

Table id	8 bits
Section syntax indicator	1 bit

TABLE II (continued)

Private indicator/reserved	1 bit
ISO reserved	2 bits
Section length	12 bits

[0074] The private indicator and private section lengths are comprised of data not fixed by the MPEG-2 standard and which may be used by the system operator for his own purposes.

[0075] In the case of short form, the header 60 is immediately followed by the payload data 61. In the case of the long form, a further header section 62 is provided before the payload 63 and the message equally includes a CRC check value 64. The long form, which is typically used when a message is so long that it must be divided into a number of sections, contains the information necessary to assemble the sections, such as the section number, the number of the last section in the sequence of sections etc.

[0076] For further information regarding the long and short form table data, the reader is directed to the MPEG-2 standard.

[0077] In the case of conditional access ECM and EMM messages, the data may usually be accomodated in a single table and the short form will be the appropriate format. A specific syntax for such short form conditional access messages is proposed in the context of the present invention, namely:

TABLE III

Table id (filter data)	8 bits (1 byte)
Section syntax indicator	1 bit
Private indicator/reserved	1 bit
ISO reserved	2 bits
Section length	12 bits
CA specific header field (filter data)	56 bits (7 bytes)

[0078] For such CA messages, the table id value may be set by the system operator at, for example, 0x80 and 0x81 for ECM messages (for example, odd and even messages) and 0x82 to 0x8F for EMM messages. These values are not MPEG-2 proscribed and may be chosen at the discretion of the system operator.

[0079] Equally, in the case of the CA specific header field, hereby designated as the first 7 bytes of the payload following the header, the parameters may be set by the system operator to reflect, for example, the fact that the CA message is an EMM message carrying individual, group or audience subscription information. In this manner the "header" of such a table or section is extended.

[0080] The advantages of such message syntax will become clear later, with regard to the processing and filtering of messages by the receiver/decoder, notably by using the Table id and CA specific field data.

Receiver/decoder

[0081] Referring to Figure 5, the elements of a receiver/decoder 12 or set-top box for use in a digital broadcast system and adapted to be used in the present invention will now be described. As will be understood, the basic elements of this decoder are largely conventional and their implementation will be within the capabilities of one skilled in the art.

[0082] As shown, the decoder 12 is equipped with several interfaces for receiving and transmitting data, in particular a tuner 70 for receiving broadcast MPEG transmissions, a serial interface 71, a parallel interface 72, and a modem 73 for sending and receiving data via the telephone network. The decoder also includes a first and second smart card reader 74 and 75, the first reader 74 for accepting the subscription smart card and the second reader 75 for accepting bank and/or other smart cards.

[0083] The decoder also includes a receiver 76 for receiving infra-red control signals from a handset remote control 77 and a Peritel ouput for sending audiovisual signals to a television 13 connected to the decoder.

[0084] Processing of digital signals received via the interfaces and generation of output signals is handled by an ensemble of hardware and software elements here grouped together as a central control unit 78. The software architecture of the control unit within the decoder may correspond to that used in a known decoder and will not be described here in any detail. It may be based, for example, on a virtual machine interacting via an interface layer with a lower level operating system implemented in the hardware components of the decoder. In terms of hardware architecture, the control unit 78 will be equipped with a processor, memory elements such as ROM, RAM, FLASH memory etc. as in known decoders.

[0085] Applications processed by the control unit 78 may be resident applications stored in the ROM or FLASH of the decoder or applications broadcast and downloaded via the MPEG interface 2 of the decoder. Applications can include program guide applications, games, interactive services, teleshopping applications, as well as initiating applications to enable the decoder to be immediately operational upon start-up and applications for configuring aspects of the decoder. Applications are stored in memory locations in the decoder and represented as resource files comprising graphic object descriptions files, unit files, variables block files, instruction sequence files, applications files, data files etc.

Filtering of Conditional Access Data

[0086] Figure 6 shows in schematic form the elements necessary for processing packet and table data in accordance with this embodiment of the invention. As will be understood, the elements shown in this figure may be implemented in hardware, software or in combination of the two.

[0087] The broadcast transmission received from the satellite receiver are passed via the conventional tuner 70 and an associated demodulator unit 79. The tuner 70 typically scans a range of frequencies, stopping when a chosen carrier frequency is detected within that range. The signals are then treated by the demodulator unit 79 which extracts and forwards the transport packet stream to a demux and filter unit 80. The filter structure of the demux and filter unit 80 will be described in detail below in relation to Figure 7. As will be understood, the actual choice of components needed to implement such a unit is at the discretion of the manufacturer and the most important aspect of such a unit is the chosen filter configuration.

[0088] In the case of data encrypted in accordance with a conditional access system as per the present embodiment, the filter unit interacts with a smart card 30 (or any other secure device) inserted in the decoder 12 and a channel parameter application 81, typically implemented as a software application in the decoder.

[0089] The filter unit 80 extracts from the transport packet stream the PMT and CAT tables present in the stream. Referring back to Figure 3, this filtering operation is carried out at a PID level, the CAT table being identified by the PID value 0x01 and the appropriate PMT table corresponding to the chosen broadcast channel being extracted via the PAT table (PID value: 0x00) and the PID value of the chosen channel identified in the PAT table.

[0090] The channel parameter application 81 additionally receives from the smart card 30 an identification of the conditional access system associated with that smart card. Again, referring back to Figure 3, a first conditional access system is associated with ECM and EMM data in the packets 45 and 48, respectively. Using the conditional access system ID received from the smart card 30 and the PMT and CAT tables received from the filter unit 80, the application 81 determines the PID values of the conditional access packets associated with the conditional access system in question and returns these values to the filter unit 80.

[0091] In the case of a simplified system, where a relatively small number of ECM and EMMs are emitted, no other filtering may be necessary and these PID values may be used by the filter unit 80 to extract all relevant ECM and EMM private sections from the identified packets and to thereafter forward the data contained within these sections to the smart card 30.

[0092] This conditional access data is then processed by the microprocessor within the smart card 30 and the control word associated with the transmission passed to a descrambling unit 83. The descrambling unit 83 receives scrambled audiovisual or other data information extracted from the transport packet stream by the demux and filter unit 80, descrambles the information using the control word and thereafter passes the data to a convention MPEG-2 chip which prepares the data for subsequent display on the associated television display.

[0093] However, whilst a PID level filter enables an extraction of those ECM and EMM messages associated exclusively with the conditional access system in question, there may nevertheless be a large proportion of messages irrelevant to the user. These messages may include group EMM messages for other user groups, individual EMM messages for other users etc. The throughput of conditional access messages passed to the smart card may therefore be very high. Given the limitations of the processor power and memory of smart cards, this throughput may be in practice more than the card can handle.

[0094] In order to overcome this problem, the smartcard 30 is adapted to pass further filter data to the unit 80 for use in a section or table level filter process.

[0095] Referring to the Table III above, tables containing conditional access data include Table id and CA specific header fields which are chosen to identify, for example, the presence of an EMM or ECM (table id values 0x80 or 0x81 and 0x82 to 0x8F, respectively) and the type of message (CA specific data identifying the group concerned by a group EMM message, the presence of an audience EMM message etc.). Depending on the data that it requires, the smart card 30 will send the necessary table id and CA specific data to configure the filter unit to extract and return only those conditional access messages of interest to the smart card. In this way, the flow of data sent to the smart card may be reduced to conform with the processing capabilities of the smart card microprocessor.

[0096] Referring to Figure 7, the details of the filtering unit 80 will be described. Typically, the unit may be implemented as a hardware resource, driven by a firmware managing application with the receiver/decoder. As shown, a first set of filters 85 carries out a PID filtering process using the CA PID information received from the channel parameter application. The PID filters 85 may equally be configured to extract other relevant packets such as the PMT, CAT tables sent to the channel parameter application. Other PID filters (not shown) may be used to extract the audiovisual PES packet information eventually sent to the descrambler etc.

[0097] Once stripped of the packet header, the private section or table data is then routed to a set of prefilters 86 adapted to filter the 8 bytes in the extended header of a table. As shown in Table III, 1 byte of the extended header is associated with the table id, 7 bytes with the CA specific information. The filtering operation is carried out by comparison of the 8 byte pattern in a table with the filter data received from the smart card. Some bits within the 8 byte, 64 bit pattern may be masked or ignored in the evaluation. In this embodiment, 32 different patterns are proposed, a subset of these patterns being applied by the prefilters in dependence of the information received from the smart card. If one pattern matches, the section is sent to the FIFO buffer element 87. If no pattern matches, the section is ignored. The filters 86 equally act to extract from the appropriate sections the PMT and CAT table information, which is passed to a FIFO buffer 88.

[0098] Due to the characteristics of the transport layer, the arrival of sections is bursty. The buffer capacity of the buffers 87, 88 must be sufficient to handle an average rate of 5Mbits/s, with the insertion of packets being based on a regular allocation with a possible deviation of ± 25%.

[0099] In order to better understand the invention, a proposed example of operating instructions handled by the section filters 86 will now be outlined.

Filter_all_sections (Filter_id, Target, Mask, Trigger_conditions, p/n)
This command retrieves every section matching the target except masked bits after trigger_conditions occured.
Filter_next_section (Filter_id, Target, Mask, Trigger_conditions, p/n)
This command retrieves the next section matching the target except masked bits after trigger_conditions occured. Trigger_conditions are related to other filters previously identified as matching.
Filter_id is an index between 0 and 31, pointing to a filter and an output queue. In addition, it gives the queueing priority, 0 being the highest priority.
Target is an 8 bytes pattern.
Mask is an 8 bytes pattern showing the bits to be masked in the target, value 0 means masked.
Trigger_conditions is a 32 bib bitmap, ORing filter_id triggering that filter. Bit set at 0 means no trigger condition. Self trigger condition is ignored.
p/n is a value, normally set to 1, positive for normal operation as described above. When set to 0 it means negative filtering, i.e., retrieve sections not matching target.

Examples of use:

Example 1:

[0100]

Filter_all_sections(5, 0x8C7C453AA8BBFF00, 0XFF557FFFEEFFFF00, 0, 1) will capture all EMMs corresponding To matching criteria.

Example 2:

[0101]

Filter_next_section(0, 0x8000000000000000, 0xFF00000000000000, 0, 1)
Filter_next_section(1, 0x8100000000000000, 0xFF00000000000000, 5, 1)
Filter_next_section(2, 0x8000000000000000, 0xFF00000000000000, 3, 1)

will start an ECM capture process with odd/even toggle.

Example 3:

[0102]

```
Filter_next_section(8, 0xPMT_TID0000Version_number00000000, 0xFF00001F00000000, 0, 0)
Filter_next_section(1, 0x8100000000000000, 0xFF00000000000000, 0x14, 1)
Filter_next_section(2, 0x8000000000000000, 0xFF00000000000000, 0x12, 1)
```

will start an ECM capture process with odd/even toggle, starting when there is a change in the PMT.

[0103] In terms of communication of CA messages and filter data to and from the smart card 82 and filter unit 80, a standard protocol such as ISO7816 may be used. Since not all of the data in the filtered private section is required by the smart card 82, the section may be modified and a message of the following format sent to the smart card:

Table id	8 bits
Zero	11 bits
Filter id	5 bits
CA specific header field	56 bits
CA message	N*8 bits

[0104] The meaning of each of these terms will be clear from the above description. In terms of the filter data sent from the smart card 82 to the filter 80, the following format may be used:

Number of filters	8 bits
Filtering instruction	5 bits
Filter id	5 bits
Target	64 bits
Mask	64 bits
Trigger conditions	5 bits
p/n	1 bit

Number_of_filters describes the number of filters to be set in this instruction.
Filtering_instruction is describing the type of instruction (filter next section, filter all sections).
Filter_id is an index pointing to a filter and an output queue. In addition, it gives the queueing priority, 0 being the highest priority.
Target is the target pattern.
Mask is a pattern showing the bits to be masked in the target, value 0 means masked.
Trigger_conditions is a bitmap. ORing filter_id triggering that filter. Bit set at 0 means no trigger condition. Self trigger condition is ignored.
p/n is a value, normally set to 1, positive for normal operation as described above. When set to 0 it means negative filtering, i.e., retrieve sections <u>not</u> matching target.

[0105] In practice, communications between the smart card and the receiver/decoder may be subject to a level of encryption or scrambling for security reasons. In particular, communications between the smart card 82 and filter unit 80, as well as the control word stream sent to the descrambler unit 83 may be encoded in this way. Encryption algorithms suitable for this purpose are widely known (RSA, DES etc.).

Claims

1. A decoder adapted to receive a transport packet stream containing table, section or other packetised data encapsulated within the packet payloads and characterised in that the decoder comprises a means for filtering the encapsulated data configurable in response to filter data received from a portable security module.

2. A decoder as claimed in claim 1 in which the means for filtering encapsulated data is configurable in response to filter data comprising at least a table ID or section ID value transmitted by the portable security module.

3. A decoder as claimed in claim 1 or 2 in which the means for filtering encapsulated data is further adapted to forward to the security module conditional access data obtained in accordance with the filter data received from the security module.

4. A decoder as claimed in claim 3 in which conditional access data forwarded to the security module comprises entitlement control messages (ECMs) and/or entitlement management messages (EMMs).

5. A decoder as claimed in claim 3 or 4 in which filter data provided by the security module comprises data used by the filter means to extract group and/or individual entitlement management messages addressed to the security module.

6. A decoder as claimed in any of claims 3 to 5 in which the decoder is adapted to receive a control word generated by the security module in response to the conditional access data forwarded thereto, the control word being used by the decoder to descramble a scrambled transmission.

7. A decoder as claimed in any preceding claim further comprising a means for filtering transport packet data configurable in response to data received from the security module.

8. A decoder as claimed in claim 7, in which the means for filtering transport packet data is configurable in response to data representing the identity of the conditional access system received from the security module.

9. A decoder as claimed in claim 8 in which the transport packet filtering means is adapted to extract transport packets containing a program map table and a conditional access table, the decoder further comprising selection means adapted to receive the program map table and conditional access table from the transport packet filtering means and conditional access identity data from the security module and thereafter configure the transport packet filtering means to extract transport packet data associated with the conditional access system in question.

10. A decoder as claimed in any preceding claim adapted to process encrypt and/or decrypt communications to and from the portable security module.

11. A security module for use with a decoder as claimed in any preceding claim and characterised in comprising a memory means for storing filter data subsequently communicated to the decoder to configure the means for filtering encapsulated data.

12. A security module as claimed in claim 13 comprising a smart card.

13. A method of processing a transport packet stream containing table, section or other packetised data encapsulated within the packet payloads characterised by receiving the transport stream in a decoder and filtering the encapsulated data in response to filter data received from a portable security module.

14. A method of processing a transport packet stream as claimed in claim 13 further comprising generating encapsulated data including conditional access data and filtering at the decoder using the encapsulated data and in response to filter data supplied by the portable security module.

Fig.1.

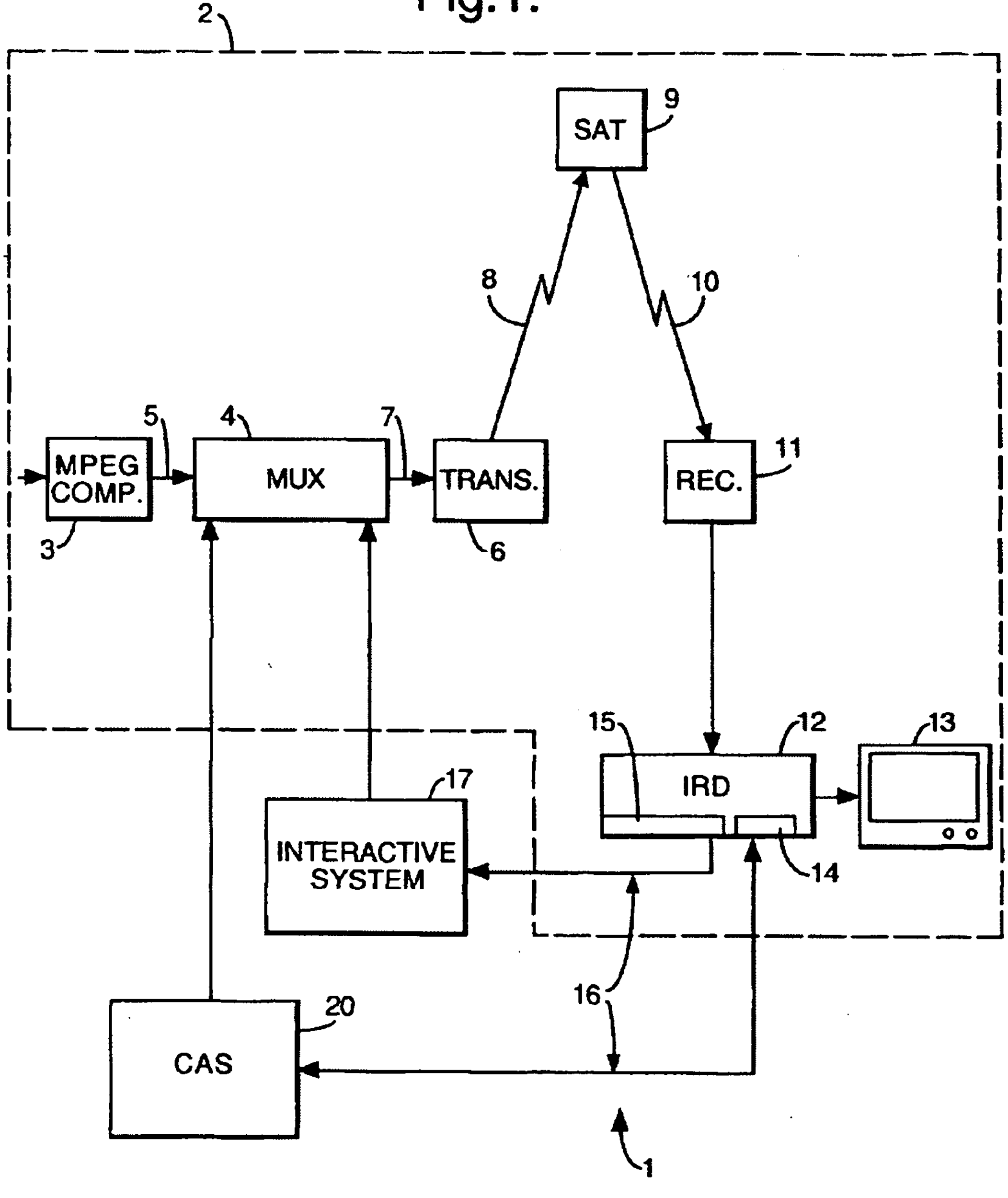

SAT
MPEG COMP.
MUX
TRANS.
REC.
IRD
INTERACTIVE SYSTEM
CAS

Fig.2.

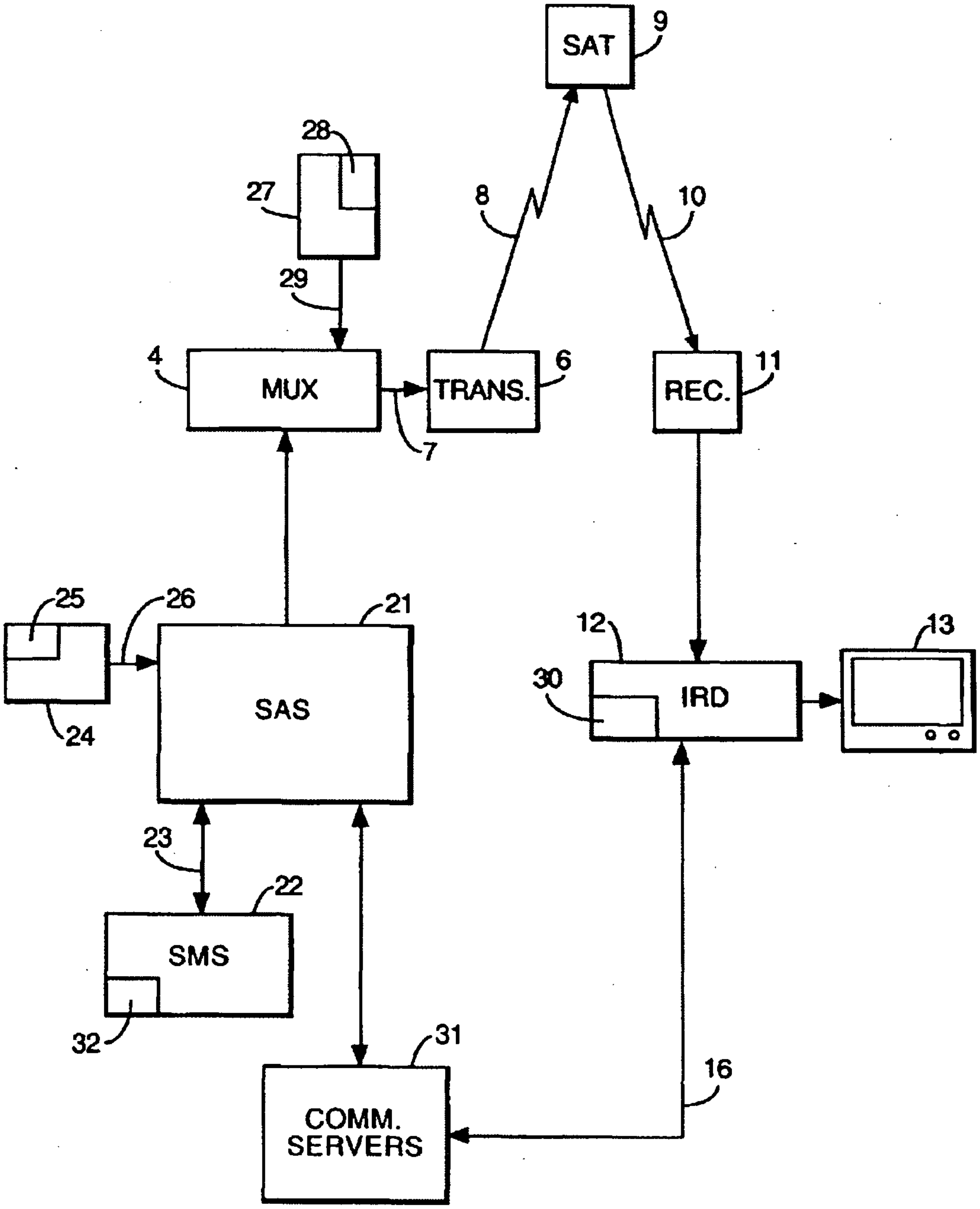

SAT
MUX
TRANS.
REC.
IRD
SAS
SMS
COMM. SERVERS

Fig.3.

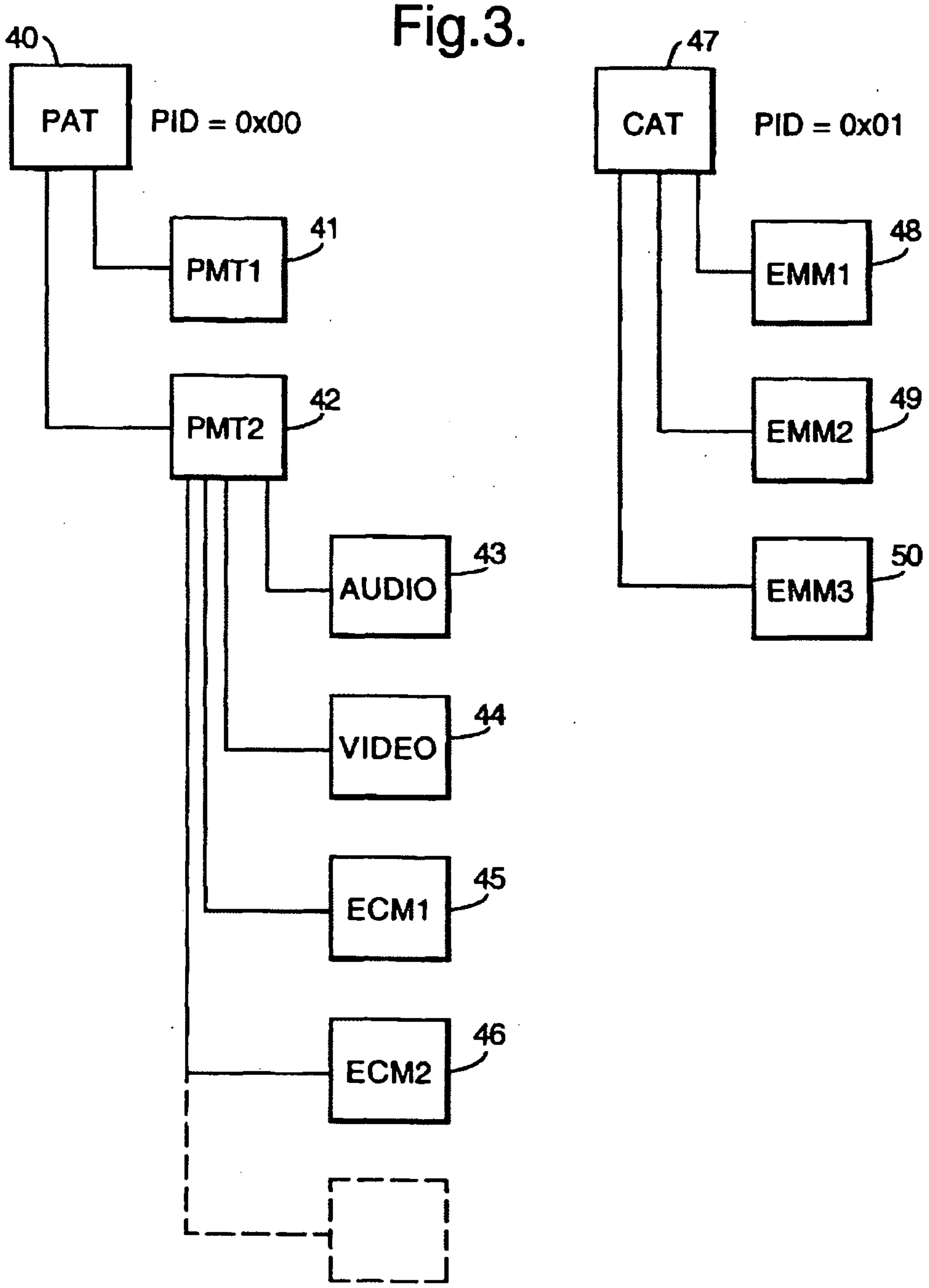

40
PAT
PID = 0x00
PMT1
41
PMT2
42
AUDIO
43
VIDEO
44
ECM1
45
ECM2
46
47
CAT
PID = 0x01
EMM1
48
EMM2
49
EMM3
50

Fig.4.

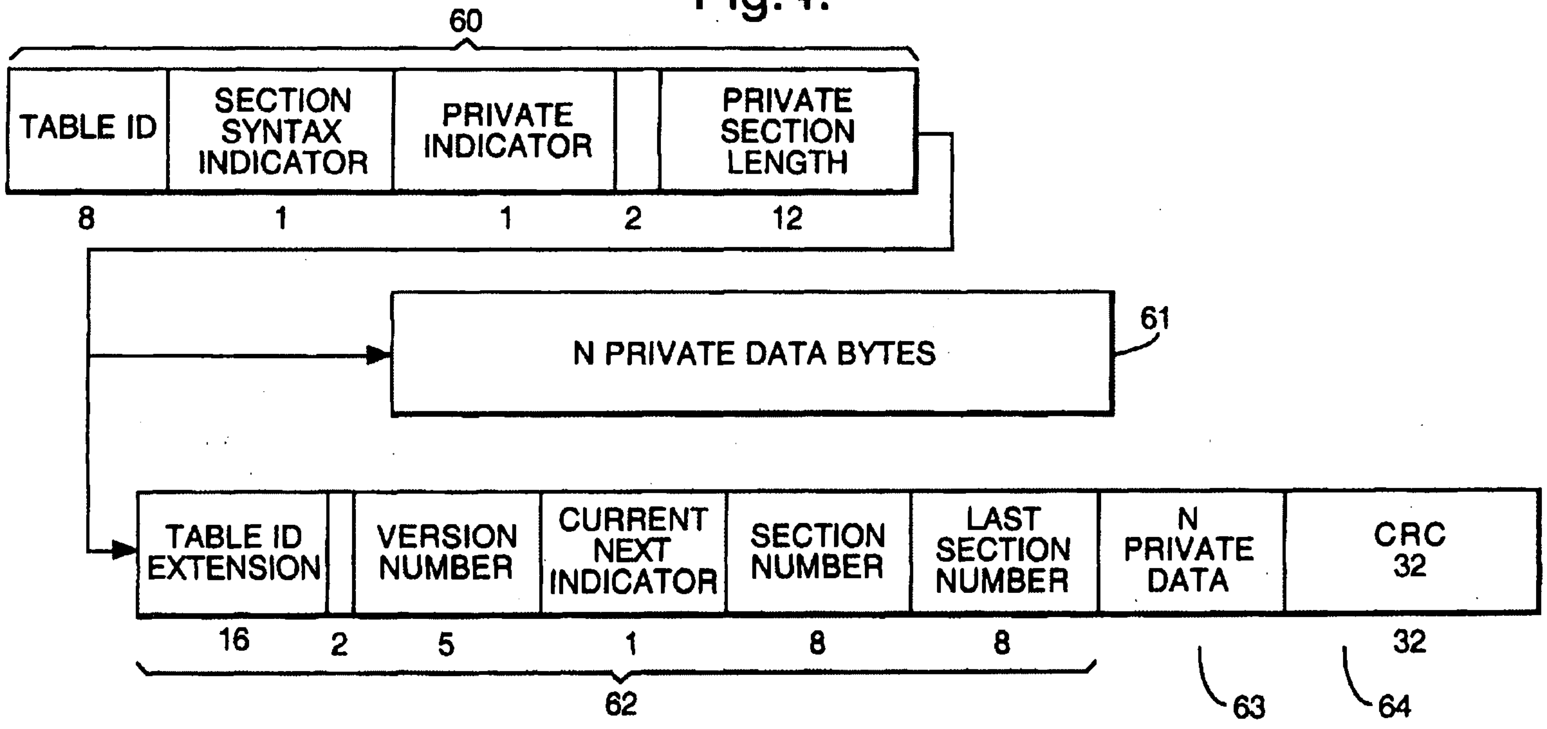
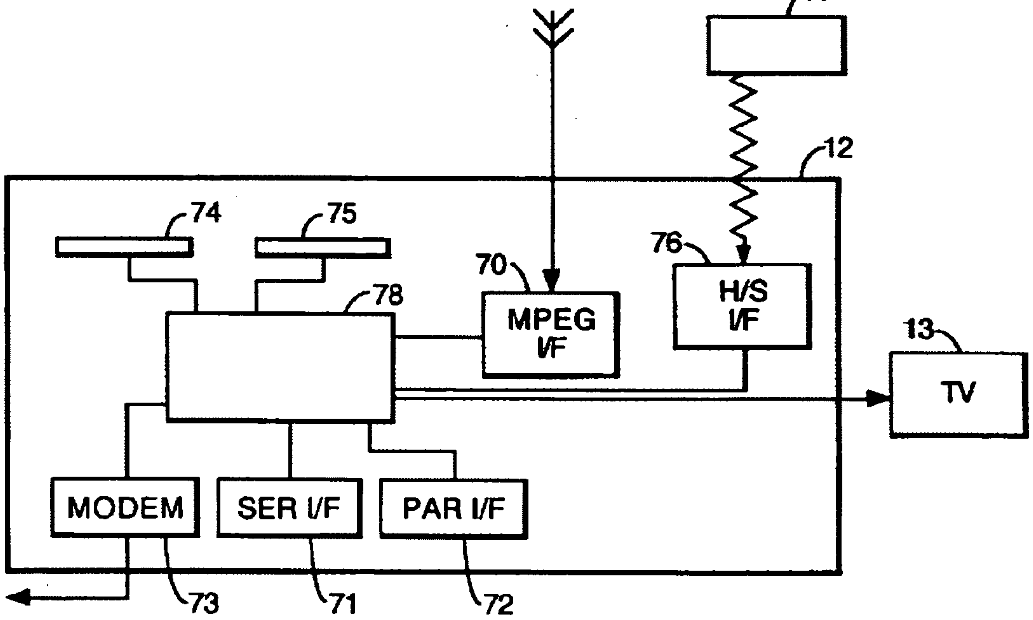

60
TABLE ID
SECTION SYNTAX INDICATOR
PRIVATE INDICATOR
PRIVATE SECTION LENGTH
8
1
1
2
12
N PRIVATE DATA BYTES
61
TABLE ID EXTENSION
VERSION NUMBER
CURRENT NEXT INDICATOR
SECTION NUMBER
LAST SECTION NUMBER
N PRIVATE DATA
CRC 32
16
2
5
1
8
8
32
62
63
64

Fig.5.


77
12
74
75
70
76
78
MPEG I/F
H/S I/F
13
TV
MODEM
SER I/F
PAR I/F
73
71
72

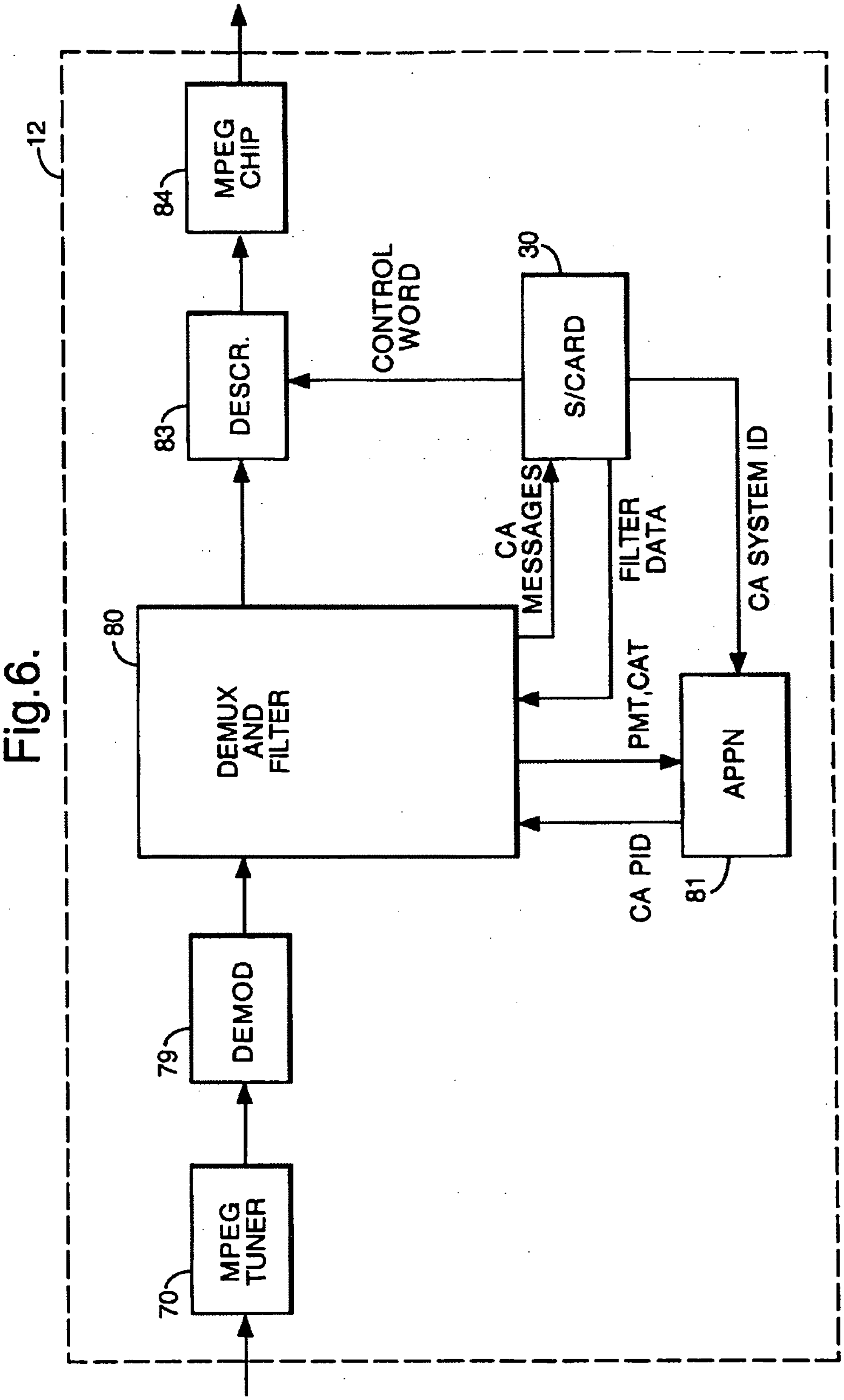
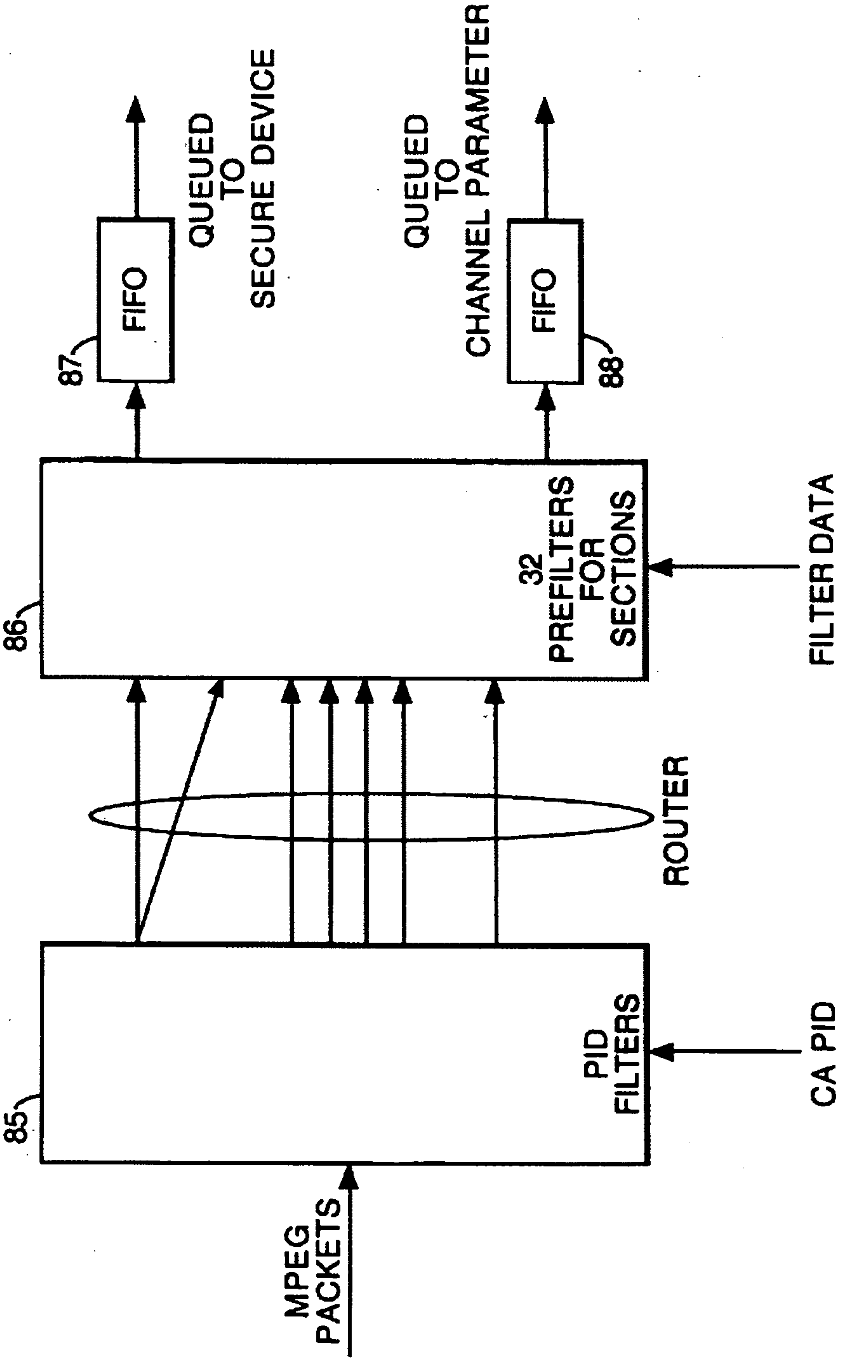

12
MPEG TUNER
70
DEMOD
79
DEMUX AND FILTER
80
DESCR.
83
MPEG CHIP
84
CONTROL WORD
S/CARD
30
CA MESSAGES
FILTER DATA
CA SYSTEM ID
PMT,CAT
APPN
81
CA PID

Fig.6.

Fig.7.


MPEG PACKETS
85
PID FILTERS
CA PID
ROUTER
86
32 PREFILTERS FOR SECTIONS
FILTER DATA
87
FIFO
QUEUED TO SECURE DEVICE
88
FIFO
QUEUED TO CHANNEL PARAMETER

European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 98 40 1374

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	WO 95 29560 A (THOMSON CONSUMER ELECTRONICS) 2 November 1995 * page 1, line 35 - page 2, line 25 * * page 4, line 23 - page 8, line 35 * * figure 3 *	1,3-5,8, 10-14	H04N5/00
A		2,6,7,9	

X	WO 97 46008 A (THOMSON CONSUMER ELECTRONICS) 4 December 1997 * page 3, line 17 - page 10, line 9 *	1-3,6-14	
A		4,5	

A	"FUNCTIONAL MODEL OF A CONDITIONAL ACCESS SYSTEM" 21 December 1995 , EBU REVIEW- TECHNICAL, NR. 266, PAGE(S) 64 - 77 XP000559450 * the whole document *	1-14	

A	SCHOONEVELD VAN D: "STANDARDIZATION OF CONDITIONAL ACCESS SYSTEMS FOR DIGITAL PAY TELEVISION" PHILIPS JOURNAL OF RESEARCH, vol. 50, no. 1/02, July 1996, pages 217-225, XP000627672 * page 218, line 12 - page 220, line 9 *	1-14	TECHNICAL FIELDS SEARCHED (Int.Cl.6)
	-----		H04N

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	3 November 1998	Fassnacht, C

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

Booking by means of a virtual access ticket

Publication number: EP1103922
Publication date: 2001-05-30
Inventor: LAUTENSCHLAGER WOLFGANG (DE); STUERZ HEINZ (DE)
Applicant: CIT ALCATEL (FR)
Classification:
- international: ***G06Q10/00; G07B15/00;*** **G06Q10/00; G07B15/00;** (IPC1-7): G07F7/08; G06F17/60; G07B15/00; G07F17/42
- European:
Application number: EP20000124578 20001110
Priority number(s): DE19991056359 19991124

Also published as:
EP1103922 (A3)
DE19956359 (A1)

Cited documents:
EP0950968
US5598477
NL9301902
EP0713198
GB2317258
more >>

Report a data error here

Abstract of **EP1103922**

The booking method has a reservation request received from a customer by a reservation agent, with the customer charge logged by the agent and an electrical signal containing coded data corresponding to an access authorisation transmitted back to the customer, for storage on an electronic data carrier, acting as a virtual entry ticket. Also included are Independent claims for the following: (a) a central server for a reservation booking method; (b) a computer program for a reservation booking method

---

Data supplied from the ***esp@cenet*** database - Worldwide

(12) UK Patent Application (19) GB (11) 2 022 969 A

(21) Application No 7924218
(22) Date of filing 11 Jul 1979
(23) Claims filed 11 Jul 1979
(30) Priority data
(31) 14400/78
(32) 12 Apr 1978
(33) United Kingdom (GB)
(43) Application published
19 Dec 1979
(51) INT CL²
G06K 15/20
(52) Domestic classification
H4T 4A2 4B1
(56) Documents cited
None
(58) Field of search
H4T
(71) Applicants
Data Recall Limited,
Sondes Place, Dorking,
Surrey RH4 3EF
(72) Inventor
Mark-Eric Jones
(74) Agents
Reddie & Grose

(54) Video display control apparatus

(57) Video display control apparatus for a visual display device (11, Fig. 1, not shown) employing a television-type raster in a word processor has a display memory (13), a column counter 25 and a row counter 28 adapted to address the display memory. Each location of the display memory has an address comprising a column number and a row number. A clock oscillator (18) and a timing chain (19) produce raster timing signals and column and row timing signals. The count in the column counter 25 tracks the line being scanned, and the count in the row counter tracks successive groups of lines in the raster. The display data output of the display memory controls a character matrix memory (15) acting through a parallel-to-serial converter (16) to cause alphanumeric characters to be displayed in rows by the display device. So that the information display by the display device can be varied in a convenient manner, the row counter 28 is coupled to the display memory 13 through a random access memory 30 which stores information from a central processor unit (14). This stored information determines which set of sequential row addresses shall be supplied to the display memory as the row counter 28 carries out its counting sequence, and includes an instruction associated with a selected row address which causes a reset signal 31 to be supplied to the column counter 25 so that for this row the characters displayed start at the character stored in the first column of locations in the display memory, the column addresses generated by the column counter 25 being otherwise selectable as any set formed by a predetermined number of consecutive column addresses for alphanumeric character locations in the display memory.

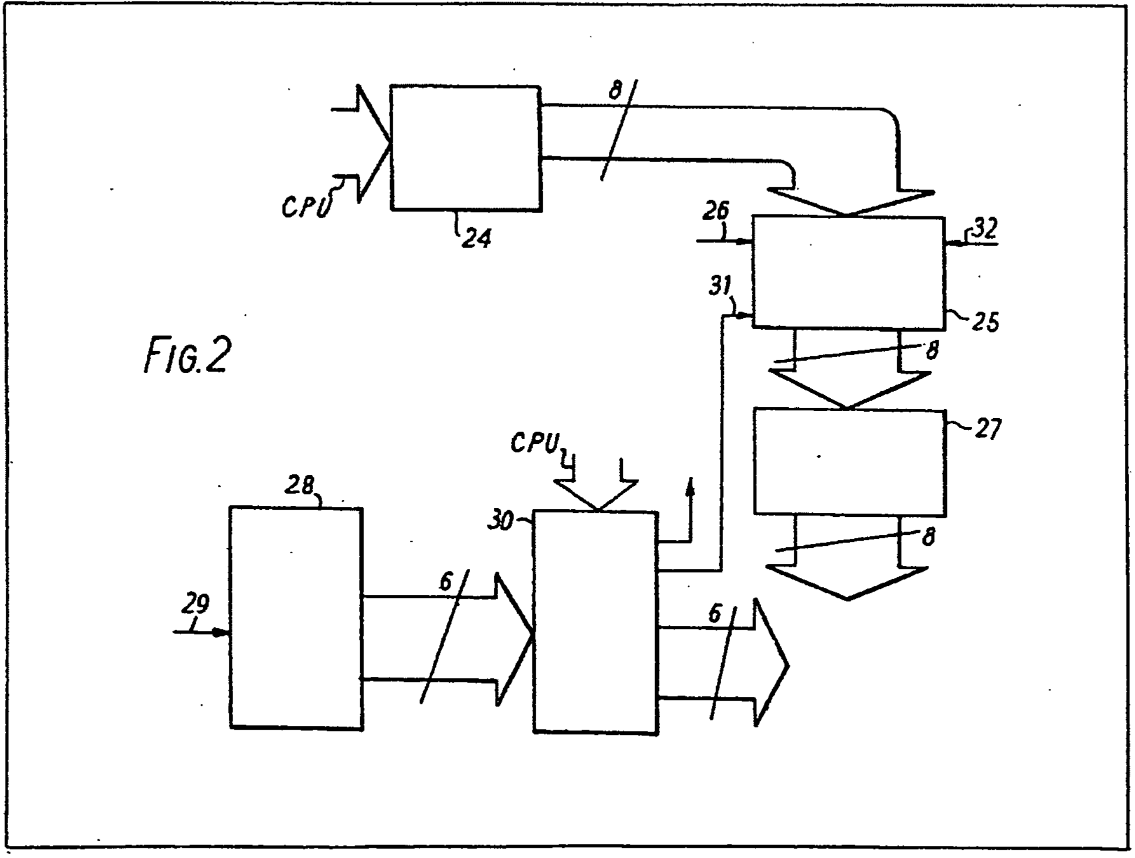
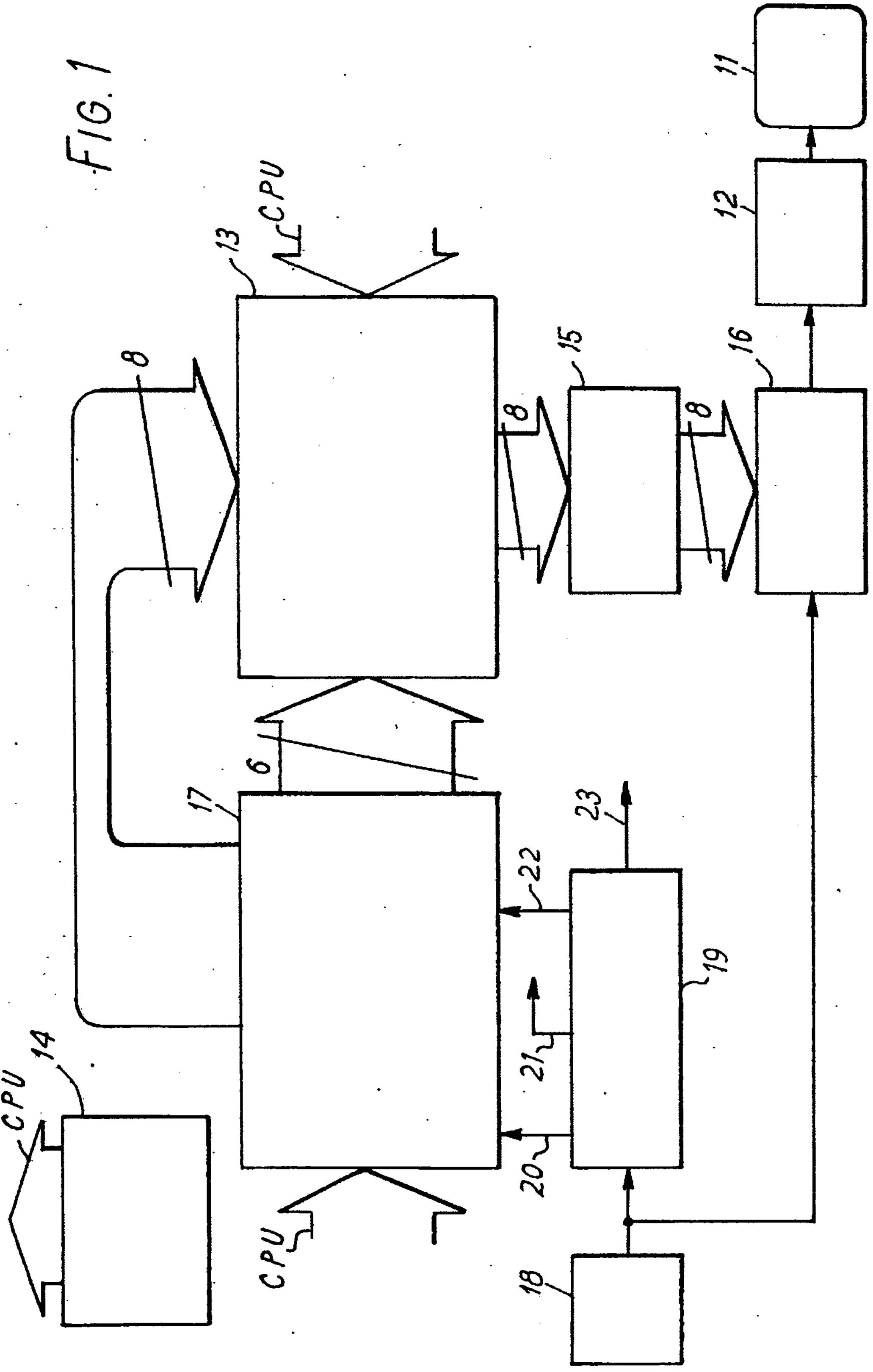


FIG. 2

GB 2 022 969 A

FIG. 1


CPU
13
8
8
15
8
16
12
11
6
17
22
23
19
21
20
CPU
14
CPU
18

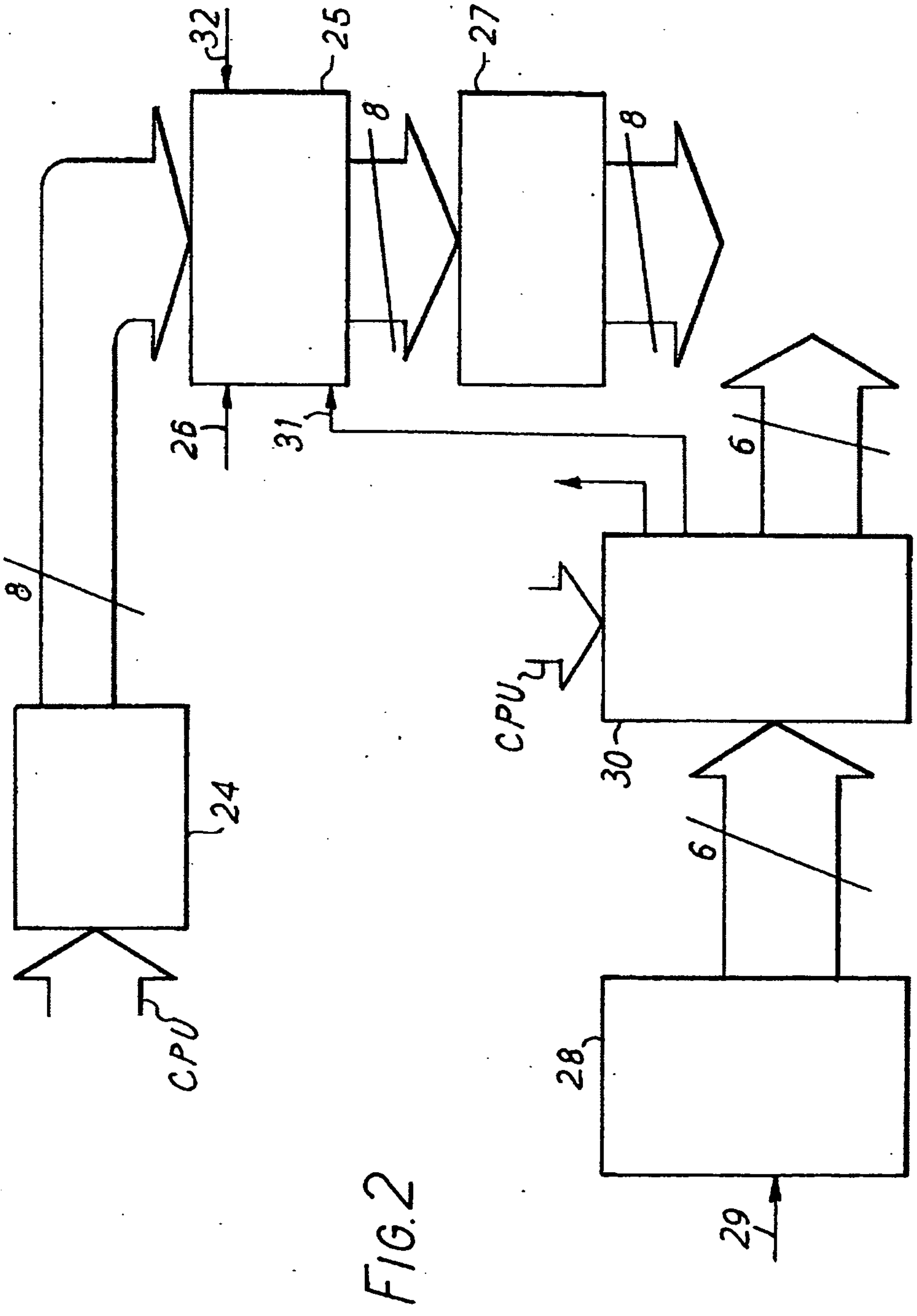

32
25
27
8
8
8
26
31
6
CPU
30
6
24
CPU
28
29

FIG. 2

SPECIFICATION
Video display control apparatus

This invention relates to video display control apparatus for use with a visual display device employing a television-type scanning raster.

Visual display devices are now employed in monitoring or simply displaying information constituting the output of, for example, a computing system, a commercial information disseminating network, or a word processor. At present, such display devices are usually in the form of a cathode ray tube operated with a television-type scanning raster. It is frequently the case that the quantity of data stored in the system supplying the visual display device is greater than the amount that can be displayed simultaneously.

An object of the present invention is to provide control apparatus enabling a visual display device to vary the information display thereby in a convenient manner.

According to the present invention, therefore, there is provided video display control apparatus for use with a visual display device employing a television-type scanning raster, the control apparatus including a display memory, a column counter and a row counter adapted to address the display memory, each of a plurality of locations of the display memory having an address comprising a column number and a row number, timing means for producing raster timing signals and column and row timing signals, the timing means being so coupled to the column counter and the row counter that, in operation, the count in the column counter changes in a manner representative of the scanning of a line of the raster and the count in the row counter changes in a manner representative of the succession of lines in the raster, and means coupled to data output terminals of the display memory for producing display signals representative of display data held in addressed locations of the said plurality of locations, characterised in that the row counter is coupled to the display memory through a random access memory adapted to store a row holding instruction relating to a selected row address and to supply to the column counter a row holding signal such that the column counter in response thereto carries out its column counting or countings for the selected row address through a predetermined series of column numbers, the column counter being adapted to count a predetermined number of column numbers starting from a column number which is selectable except in the presence of the row holding instruction.

Since the count in the column counter changes in a manner representative of the scanning of a line of the raster and the count in the row counter changes in a manner representative of the succession of lines in the raster, and the column and row timing signals are such that the count in the column counter changes faster than the count in the row counter. Although the terms column and row are thus associated with the scanning of a line of the raster and the succession of lines in the raster respectively, the lines of the raster in the display in operation may be so orientated as to run from top to bottom of the display as viewed by a user. Normally, however, the lines will be orientated so as to run from left to right in the display.

Preferred features of the apparatus are defined in the sub-claims appended hereafter.

The invention will now be described in more detail, solely by way of example, with reference to the accompanying drawings, in which:—

Fig. 1 is a block diagram of a word processor embodying the invention; and

Fig. 2 is a block diagram showing in more detail part of the embodiment of Fig. 1.

In the word processor of Fig. 1, a cathode ray display tube 11 receives a video signal from a video output stage 12. Scanning circuitry for the cathode ray display tube 11 is not shown and produces a scanning raster on the screen of the tube 11, the scanning raster being formed by a large number of horizontal lines. The scanning of the raster is similar to that of a television raster except that there is no interlacing of the lines. The lines in the scan making up each frame of the raster are produced in sequence starting at the top of the frame. A display memory 13 stores alphanumeric character codes in a plurality of locations arranged to represent, for example, an array of 128 columns by 64 rows. The character codes are supplied to the display memory 13 by a central processor unit 14 which receives this information from a flexible disc, not shown, or a keyboard, not shown.

Whenever one of the locations containing a character code in the display memory is addressed, the character code is supplied to a character matrix memory 15 which stores a character scan dot code for each possible alphanumeric character. In the present example, each alphanumeric character is formed by a selection of dots from a matrix of 10 by 13 dot positions, each matrix being 13 dots high and 10 dots wide. Consequently, 13 line scans are required to scan each complete character. Thus one row consists of 13 horizontal successive lines of dots, in coded form, supplied by the matrix memory 15 to a parallel-toserial converter 16 in the form of a 10 bit shift register. The serial output of this converter is supplied to the video output stage 12 which correspondingly supplies video dot signals to the cathode ray display tube 11.

The display memory 13 is addressed by an addressing unit 17 which provides the address for each of the alphanumeric character locations of the display memory in the form of a 6 bit row address combined with an 8 bit column address. In effect, a selected succession of 80 column addresses is supplied 13 times to the display memory 13 during the supplying of each row address to the display memory 13. Consequently, each of the 13 horizontal lines of dots in coded form supplied to the converter 16 consists of 80 groups of dots, each group lying in a respective column and being a selection of the dots forming the character at the location defined by the respective column and the current row.

Timing signals, in the form of pulses, are generated as follows.

A clock oscillator 18 generates clock pulses at, for example, 50 megahertz. The clock pulses are supplied directly to the shift register constituting the converter 16 and thus the dot rate is set at the frequency of the clock oscillator 18. The clock pulses are also supplied directly to a timing chain 19 which consists of a chain of frequency dividers (not shown). Four outputs 20, 21, 22 and 23 from the timing chain 19 are shown. Streams of pulses at successively lower rates are supplied at these outputs 20 to 23. The highest pulse rate, which is at the output 20, is supplied to the addressing unit 17 to determine the rate at which column addresses are generated. This rate is accordingly the character clock rate and may be, for example, 5 megahertz. The pulses supplied at the output 21 are generated at a rate which is used as the line frequency for the raster of the cathode ray display tube 11. Each pulse at the output 21 is very short and corresponds substantially to a line sync pulse. The rate of the pulses at the output 22 is 1/13th that of the pulses at the output 21. The pulses at the output 22 are supplied to the addressing unit 17 where they serve to determine the row address rate. The rate of the pulses at the output 23 is 1/68th of the rate of the pulses at the output 22. The pulses at the output 23 are accordingly used as frame sync pulses, i.e. the pulses which determine the instants at which rasters on the cathode ray display tube 11 are completed.

The central processor unit 14 supplies to the addressing unit 17 information which determines which succession of 80 of the 128 columns is to be addressed by the addressing unit, and which one of the 64 rows is to serve as the starting row during addressing by the addressing unit. This facility enables the cathode ray display tube 11 to display the information contained in any array of 80 columns by 64 rows selected from the array of 128 columns by 64 rows representing the stored alphanumeric characters in the display memory 13. For example, if the array represented by the locations in the display memory 13 is considered to consist of columns 1 to 128 numbered from the left and rows 1 to 64 numbered from the top, the addressed array may consist of columns 21 to 100 by rows 10 to 64 followed by rows 1 to 9. Furthermore, the information supplied to the addressing unit 17 by the central processor unit 14 can include an instruction for a selected row of the addressed array to consist of the locations in columns 1 to 80 of that row while the other rows consist of the locations in another succession of 80 columns, for example, columns 21 to 100. The means whereby this latter operation is carried out will now be described with reference to Fig. 2.

In Fig. 2, the addressing unit 17 is shown to consist of a roll left right offset latch 24 which holds the current value of the left hand column to be displayed, this value being supplied to the latch by the central processor unit, a column counter 25 coupled to the latch 24 to receive therefrom an 8 bit output representing the left hand column value held by the latch 24, and receiving at an input 26 the character rate pulses supplied by the output 20 of the timing chain 19, a buffer 27 coupled to the 8 bit output of the counter 25 and having an 8 bit output at which the column addresses supplied to the display memory 13 appear in operation, a row counter 28 which receives at an input 29 the row rate pulses provided at the output 22 of the timing chain 19, and a random access memory 30 coupled to the row counter 28 to receive therefrom a 6 bit output, and having an 8 bit output of which 6 bits are supplied to the display memory 13 as the row addresses, the 7th bit of the output being supplied to a reset input 31 of the column counter 25 and the 8th bit of the output being supplied to the display memory as a blanking signal to force the main memory to provide no alphanumeric character as output during the active time of the signal on the 8th bit of the output of the random access memory 30. The random access memory 30 also receives an input from the central processor unit which determines the prevailing relationship between the 6 bit output of the row counter 28 and the first 6 bits of the output of the random access memory 30 which are supplied as row addresses to the display memory 13. The input to the random access memory 30 from the central processor unit also determines for each row address generated by the random access memory 30 the accompanying values of the 7th and 8th bits of the output of the random access memory. In particular, the value of the 7th bit for each row address is either high or low, and in response to one of these values, the column counter 25 is reset to zero. The column counter 25 is arranged to count a succession of 112 column numbers starting from the number of the left hand column supplied to it by the latch 25 unless the counter 25 is reset to zero in which case the count of 112 successive column numbers is started at zero. Consequently, in the display on the cathode ray display tube 11, rows of alphanumeric characters are presented which start at the left hand end with the character in the left hand column determined by the value supplied to the counter 25 by the latch 24 when for the row address supplied to the display memory 13 by the random access memory 30 the 7th bit of the output of the random access memory 30 is not such as to reset the column counter 25. However, when the 7th bit of the output of the random access memory 30 accompanying the row address supplied to the display memory 13 is such as to reset the column counter 25, the corresponding row of alphanumeric characters displayed by the cathode ray display tube 11 starts at its left hand end with the character occurring in the first column of locations in the display memory 13 for that row. Line fly-back blanking pulses are supplied to another input 32 of the column counter 25 to set the counter 25 to the start of each cycle of counting each blanking pulse occurring during the last 32 counts. In the present example, the column counter 25 is capable of counting from 0 to 255. It will be realized that the selection of the left hand column by means of the left hand column number supplied by the latch 24 to the counter 25 enables that area of the array of locations containing alphanumeric characters in the display memory 13 which is to be displayed by the cathode ray display tube 11 to be shifted to the left and to the right. Such shifting is referred to as rolling. The fixing of a particular row to the first 80 columns by the 7th bit of an output from the random access memory 30 enables rows thus selected to be held in the display on the cathode ray display tube 11 while the other rows are rolled to the left or to the right. This facility is particularly useful in the case of rows constituting headings for information appearing in the display.

The row counter 28 is such as to count from 0 to 67 and supplies its count in coded form as the 6 bit output to the random access memory 30. In a manner determined by the instructions received by the random access memory 30 from the central processor unit, the random access memory 30 translates the count of the row counter 28 into an 8 bit output signal in which the first 6 bits constitutes a row address, the 7th bit constitutes the signal to be supplied to the reset input 31 of the column counter, and the 8th bit constitutes a signal to the display memory 13 instructing that memory 13 to either provide the contents of the addressed locations or to provide a blank output signal.

The counting operation carried out by the row counter 28 is synchronised with the raster of the cathode ray display tube 11 so that the counts 64, 65, 66, and 67 occur during the frame fly-back blanking time. This locking of the counting cycle of the counter 28 to the raster timing ensures that rows of characters are automatically placed in the desired positions in the displayed array.

The random access memory 30 may be a Motorola MCM 6810AL which has a capacity of a 128 times 8 bits. The display memory 13 may be formed of 32 Texas Instruments TMS4044—15, each being a 4K by 1 bit static random access memory. The character matrix memory 15 may be formed of 8 Texas Instruments TMS4044—15. Where the random access memory 30 is a Motorola MCM 6810AL, the 6 bit input from the row counter 28 is multiplexed with the input which the random access unit 30 receives from the central processor unit.

CLAIMS

1. Video display control apparatus for use with a visual display device employing a television-type scanning raster, the control apparatus including a display memory, a column counter and a row counter adapted to address the display memory, each of a plurality of locations of the display memory having an address comprising a column number and a row number, timing means for producing raster timing signals and column and row timing signals, the timing means being so coupled to the column counter and the row counter that, in operation, the count in the column counter changes in a manner representative of the scanning of a line of the raster and the count in the row counter changes in a manner representative of the succession of lines in the raster, and means coupled to data output terminals of the display memory for producing display signal representative of display data held in addressed locations of the said plurality of locations, characterised in that the row counter is coupled to the display memory through a random access memory adapted to store a row holding instruction relating to a selected row address and to supply to the column counter a row holding signal such that the column counter in response thereto carries out its column counting or countings for the selected row address through a predetermined series of column numbers, the column counter being adapted to count a predetermined number of column numbers starting from a column number which is selectable except in the presence of the row holding instruction.

2. Apparatus according to claim 1, wherein a latch for storing a selected column number is coupled to the column counter, and the column counter is adapted to effect counting of a predetermined number of column numbers starting from the column number stored in the latch except in the presence of the row holding instruction.

3. Apparatus according to claim 1 or 2, characterised in that the column counter has a reset input terminal, the random access memory is so coupled to the column counter as to supply row holding instructions to the reset input terminal, and the column counter is such as to reset to the count zero whenever a row holding instruction is present at the reset input terminal.

4. Apparatus according to claim 3, characterised in that the random access memory is adapted to encode the count in the row counter as a different count related thereto by a constant which is selectable.

5. Apparatus according to claim 4, wherein the said locations of the display memory are filled by a central processor unit which is arranged to supply the column number to be stored to the said latch, and to supply the instructions to the random access memory which determine the said constant and determine the said selected row address.

6. Video display control apparatus substantially as described herein before with reference to the accompanying drawings.

Printed for Her Majesty's Stationery Office by the Courier Press, Leamington Spa, 1979. Published by the Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
G06F 17/60

A2

(11) International Publication Number: **WO 99/24928**

(43) International Publication Date: 20 May 1999 (20.05.99)

(21) International Application Number: PCT/US98/23648

(22) International Filing Date: 6 November 1998 (06.11.98)

(30) Priority Data:
08/965,185 6 November 1997 (06.11.97) US

(71) Applicant: INTERTRUST TECHNOLOGIES CORP. [US/US]; 460 Oakmead Parkway, Sunnyvale, CA 94086 (US).

(72) Inventors: SHEAR, Victor, H.; 5203 Battery Lane, Bethesda, MD 20705 (US). VAN WIE, David, M.; Apartment 216, 965 East El Camino Real, Sunnyvale, CA 94087 (US). WEBER, Robert, P.; 215 Waverley Street #4, Menlo Park, CA 94025 (US).

(74) Agent: FARIS, Robert, W.; Nixon & Vanderhye P.C., 8th floor, 1100 N. Glebe Road, Arlington, VA 22201 (US).

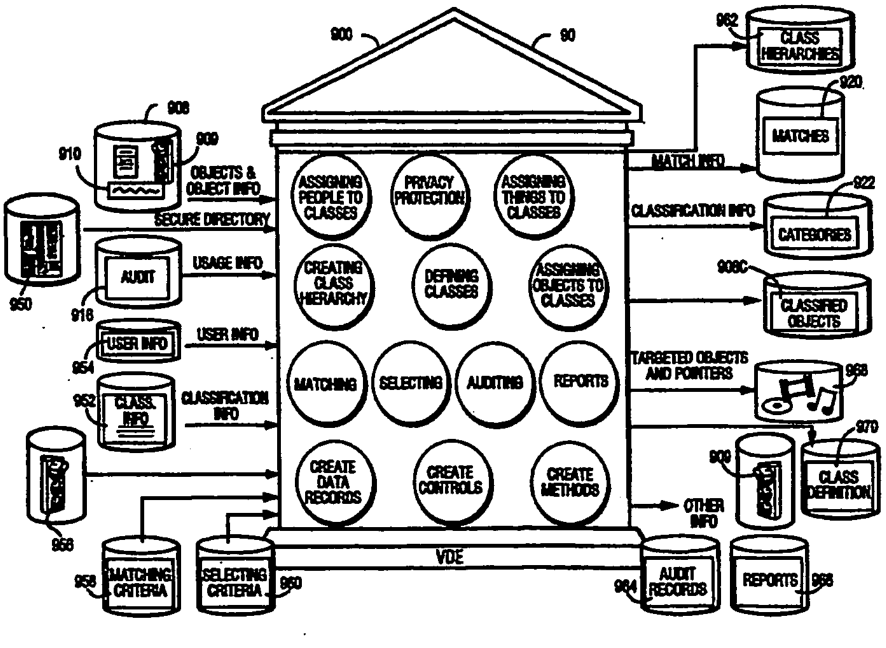
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: SYSTEMS AND METHODS FOR MATCHING, SELECTING, NARROWCASTING, AND/OR CLASSIFYING BASED ON RIGHTS MANAGEMENT AND/OR OTHER INFORMATION

(57) Abstract

Rights management information is used at least in part in a matching, narrowcasting, classifying and/or selecting process. A matching and classification utility system comprising a kind of Commerce Utility System is used to perform the matching, narrowcasting, classifying and/or selecting. The matching and classification utility system may match, narrowcast, classify and/or select people and/or things, non-limiting examples of which include software objects. The Matching and Classification Utility system may use any pre-existing classification schemes, including at least some rights management information and/or other qualitative and/or parameter data indicating and/or defining classes, classification systems, class hierarchies, categroy schemes, class assignments, category assignments, and/or class membership. The Matching and Classification Utility may also use at least some rights management information together with any artificial intelligence, expert system, statistical, computational, manual, or any other means to define new classes, class hierarchies, classification systems, category schemes, and/or assign persons, things, and/or groups of persons and/or things to at least one class.


900
90
962
CLASS HIERARCHIES
920
MATCHES
MATCH INFO
908
909
910
OBJECTS & OBJECT INFO
SECURE DIRECTORY
950
AUDIT
916
USAGE INFO
USER INFO
954
USER INFO
952
CLASS. INFO
CLASSIFICATION INFO
966
958
MATCHING CRITERIA
SELECTING CRITERIA
960
ASSIGNING PEOPLE TO CLASSES
PRIVACY PROTECTION
ASSIGNING THINGS TO CLASSES
CREATING CLASS HIERARCHY
DEFINING CLASSES
ASSIGNING OBJECTS TO CLASSES
MATCHING
SELECTING
AUDITING
REPORTS
CREATE DATA RECORDS
CREATE CONTROLS
CREATE METHODS
VDE
CLASSIFICATION INFO
922
CATEGORIES
908C
CLASSIFIED OBJECTS
TARGETED OBJECTS AND POINTERS
968
970
909
CLASS DEFINITION
OTHER INFO
964
AUDIT RECORDS
REPORTS
968

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

SYSTEMS AND METHODS FOR MATCHING, SELECTING, NARROWCASTING, AND/OR CLASSIFYING BASED ON RIGHTS MANAGEMENT AND/OR OTHER INFORMATION

FIELDS OF THE INVENTIONS

The inventions relate to electronic rights and transaction management. More particularly, the inventions relate to automated systems, methods and techniques for efficiently matching, selecting, narrowcasting, categorizing and/or classifying in a distributed electronic rights and/or other event and/or transaction management environment. For example, the inventions provide electronic computer based systems, methods and techniques for matching, classifying, narrowcasting, and/or selecting digital information describing people and/or other things. This matching, classifying, narrowcasting, and/or selecting can be based, at least in part, on elements of rights management information and/or one or more other categories of information -- wherein such information is used for efficient, trusted event management assuring the execution of one or more controls related to, including, for example, consequences of processing such digital information describing people and/or other things. The present inventions also provide systems and methods for efficiently determining class hierarchies, classification schemes, categories, and/or category schemes and/or the assignment of objects, persons and/or things to said class hierarchies, classification schemes, categories, and/or category schemes using at least some rights management information.

BACKGROUND AND SUMMARY OF THE INVENTIONS

The modern world gives us a tremendous variety and range of options and choices. Cable and satellite television delivers hundreds of different television channels each carrying a different program. The radio dial is crowded with different radio stations offering all kinds of music, news, talk, and anything else one may care to listen to. The corner convenience store carries newspapers from around the country, and a well stocked newsstand allows you to choose between hundreds of magazines and publications about nearly every subject you can think of. Merchandise from all corners of the world is readily available at the shopping mall or by mail order. You can pay by check, in cash, or using any number of different kinds of credit cards and ATM cards.

This tremendous variety is good, but it also presents problems. Sometimes, it is hard or inefficient for us to find what we want and need because there are too many things to evaluate and choose from, and they are often located in too many places. We can waste a lot of time searching for the things we need or want at the right price, with the rights features, and at a particular time.

Sometimes, we never find things that satisfy what we feel we need or want. This happens when we don't know what to look for, how to look for it, or don't have the necessary assistance or tools to search successfully. For example, we may not know the best way of looking for something. Sometimes, we know what we are looking for but can't express or articulate it in ways that help us look. And sometimes, we don't even know what we are looking for. You may know you need something, know its missing, but never really know how to communicate to others what you are looking for. For example, someone who speaks only English may never find resources using Japanese or Spanish. In general, we often don't have the time or resources to look for all the things that would give us the most benefit or make us the most satisfied.

It's Hard To Find Mass Media Things You Want Or Need.

Figure 1A shows, as one example, how frustrating it can be to find anything to watch on the hundreds of television channels that may be available. The man in Figure 1A spends a lot of time "channel surfing," trying to find something he is interested in watching. He may be moderately interested in golf, but may not like the particular golf tournament or golf players being broadcast at 7 o'clock on a particular channel. After flipping through other channels, he might think an action movie looks interesting only to find out after watching it for a while that he isn't really interested in it after all. A documentary on horses also seems interesting at first, but he finds it boring after watching it awhile because it doesn't give him the kind of information he is interested in. The whole process can be frustrating and he may feel he wasted a lot of time. Figure 1B shows the man getting so frustrated at the wasted time and energy that he thinks that maybe watching television is just not worth it . What the man really needs is a powerful yet efficient way to find those things that most satisfy his desires -- that is, match his needs and/or his interests.

Our Mail Overloads Us With Things We Don't Want or Need

The same thing can happen with information sent to us in the mail. It can be fun to receive some kinds of mail, such as personal letters, or magazines and catalogs on topics of personal interest. Certain other mail, such as bills, may not be fun but are usually important. Unfortunately, our mailboxes are typically overflowing with yet another kind of mail commonly referred to as "junk mail." The person in Figure 2 finds his mailbox stuffed to the overflowing point with mail he never asked for and has absolutely no interest in. Most of this junk mail ends up unread and in the trash. However, it can take a long time to sort through all this mail to be sure you are only throwing out only the junk mail and not the good mail you are interested in or need. For example, it's sometimes hard to distinguish credit card bills from offers for new credit cards you don't need or want. Wouldn't it be useful if your mail could be automatically "cleaned" of the mail you had no interest in and you received only the mail you wanted or needed?

Sorting through things to identify things you might want, then selecting what you actually want, can be a frustrating and time consuming experience. For example, it wastes the time of the person who receives the junk mail, and it also wastes the time, money and effort of the people who spend their money to send mail to people hoping that they will buy their products.

As frustrating as finding and selecting may be to consumers, they often create even greater problems for businesses and people who want to locate or provide information, goods and services. It is often said, that in the world of business, "Information is Power" and "efficiency is the key to success." To find or sell the most relevant or useful information and to provide the ability to most efficiently allow business to operate at its best, we need easy-to-use tools that can help us navigate, locate, and select what matches our interests. In the modern world, it is often difficult to find out what different people like, and to supply people with the opportunity to select the best or most satisfying choices.

Past attempts outside the computer world to match up people with information, goods and/or services have had limited success. For example, attempts to "target" mass mailings may increase the chance that they will go to people who are interested in them, but the entire process is still very wasteful and inefficient. It is considered a good success rate to match the interests of only a few percent of the recipients of "junk" mail. Telemarketing campaigns that use the telephone to reach potential consumers can be very expensive, very annoying to consumers who are not interested in the products being marketed, and very costly and inefficient. A much more ideal situation for all concerned is enabling businesses to send information only to individual consumers likely to find the information interesting, desirable, convincing, and/or otherwise useful. That way, businesses save time and money and consumers aren't unproductively hassled by information, phone calls, junk mail, junk e-mail and the like. However, right now it is extremely difficult to accomplish this goal, and so businesses continue to annoy consumers while wasting their own time, money, and effort.

Because of the Vast Amount of Information Available, Even Systems that Provide a High Degree of Organization May Be Difficult to Use or Access

You can find yourself wasting a lot of time finding things -- even in places where finding things is supposed to be easy. For example, a library is a place where you can find all sorts of useful information but can also waste a lot of time trying to find what you are looking for. Modern libraries can be huge, containing tens or even hundreds of thousands or millions of different books, magazines, newspapers, video tapes, audio tapes, disks, and other publications. Most libraries have an electronic or manual card catalog that classifies and indexes all of those books and other materials. This classification system is useful, but it often has significant limitations.

For example, normally a card catalog will classify materials based only on a few characteristics (for example, general subject, author and title). The boy in Figure 3 is looking for information on American League baseball teams during World War II for a high school report. The card catalog led to the general subject of baseball and other sports, but, looking at the catalog, he can't identify any books that seem to provide the specific information he wants to see, so he must rely on books classified as "histories of sports" or "histories of baseball." He can spend lots of time looking through the books on the shelves, going back to the card catalog, and going back to the shelves before he finds a reference that's reasonably helpful. He may need to go ask an expert (the librarian) who is familiar with the books the library has on sports and may know where to look for the information. Even then, the boy may need to flip through many different books and magazines, and look in many different places within the library before he finds the information he is looking for.

Finding Products You Want or Need Can Be Very Difficult and Time Consuming

The same kind of frustrating experience can happen when you shop for a particular kind of item. While some people enjoy shopping, and have fun seeing what is in various stores, many people dislike spending time shopping, searching for the best or most affordable item. And sometimes even people who like to shop don't have the time to shop for a specific item.

For example, the man in Figure 4 goes into a shopping mall looking for a tie to fit very tall people. He didn't wear a tie to work that day, but, at the last minute, an important meeting was scheduled for later that day and he needs to dress up. The shopping mall has a large variety of stores, each selling a range of merchandise. But the man may only have a short time to look. For example, he may be on his lunch break, and needs to get back to work soon. He can't spend a lot of time shopping. He may therefore need to rely on tools to help him identify where he wants to buy the tie. Perhaps he uses a mall directory that classifies the different stores in terms of what kinds of merchandise they sell (for example, clothing, books, housewares, etc.). Perhaps he asks at the malls help desk staffed by "experts" who know what is available in the shopping mall. But even these resources may not tell him where to buy Italian silk ties that are discounted and cost $20. So he does the best he can with the available resources.

These Problems Are Worse in the Digital World

The electronic or digital world offers a rapidly growing, vast array of electronically published products and services. For example, computer superstores have a dizzying array of different software products. Furthermore, music is now published primarily in digital form on optical disks, and video will soon be published that way too. And, of particular interest related to certain of the inventions described by this document, the Internet now has millions of home pages with an overwhelmingly variety and quantity of digital information, and, these millions of home pages, in turn, point or "link" to millions of other web pages as well.

Today, for example, you can use the Internet to:

- read electronic newspapers, books and magazines and see them on your computer screen;
- get music in electronic form and play it using your computer;
- send and receive electronic mail all over the world;
- download reports and other information compiled by governments, companies, industries, universities, and individuals;
- watch videos and animations;
- play games with "cyber-friends" located around the world;
- chat with individuals and groups who share at least some interests in common;
- participate in "virtual reality" worlds, games, and/or experiences;
- (offer to) buy, and/or (offer to) sell nearly anything; and
- conduct electronic transactions and commerce.

Today on the Internet and you can also find nearly anything and everything you can possibly imagine, although finding exactly what you really want may be time consuming and frustrating. This is because the Internet and World Wide Web provide perhaps the best example of an environment that is particularly hard to navigate. There are an overwhelming number of choices -- too many to easily relate to or understand -- and many of which are terribly hard to find, even using the various Web searching "engines." The Internet is particularly exciting because it has the potential to provide to nearly everyone access to nearly every kind of information. Information can also come from an almost limitless variety of sources. But today, so much information on the Internet is superficial or useless, and too many choices can be more a curse than a blessing if you don't have meaningful, easy ways to eliminate all but a relatively few choices. And the situation will only become much worse as more Web sites appear, and as digital information is distributed in "objects" or "containers" providing enhanced security and privacy but possibly more difficult access and identifiability.

As time passes, more and more valuable and desirable information will be available in digital containers. However, unless tools are developed to solve the problem, there will be no efficient or satisfying means to sort through the potentially trillions of digital containers available on tens of millions of Web pages, to find containers satisfying a search or fulfilling an information need. Furthermore, existing information searching mechanisms typically provide no way to readily perform a search that matches against underlying commercial requirements of providers and users.

It Will Be Difficult to Find Rights Management Scenarios Matching Your Requirements

If, for example, you have an auto repair newsletter and you want to create an article containing information on auto repair of Ford Bronco vehicles, you may wish to look for detailed, three dimensional, step-by-step "blow-up" mechanical images of Ford Bronco internal components. Perhaps these are available from hundreds of sources (including from private individuals using new, sophisticated rendering graphics programs, as well as from engineering graphics firms). Given the nature of your newsletter, you have decided that your use of such images should cost you no more than one penny to redistribute per copy in quantities of several thousand -- this low cost being particularly important since you will have numerous other costs per issue for acquiring rights to other useful digital information products which you reuse and, for example, enhance in preparing a particular issue. You therefore wish to search and match against rights management rules associated with such products -- non-limiting examples of which include:

- cost ceilings,
- redistribution rights (e.g., limits on the quantity that may be redistributed),
- modification rights,
- class related usage rights,
- category related usage rights,

- sovereignty based licensing and taxation fees,
- import and export regulations, and
- reporting and/or privacy rights (you don't want to report back to the product provider the actual identity of your end users and/or customers.

If you can't match against your commercial requirements, you may be forced to waste enormous amounts of time sifting through all of the available products matching Ford Bronco internal components -- or you may settle for a product that is far less than the best available (settling on the first adequate product that you review).

Computers Don't Necessarily Make It Easier to Find Things

Anyone who has ever used the Internet or the World Wide Web knows that networks, computers and electronics, when used together, do not necessarily make the overall task of finding information easier. In fact, computers can make the process seem much worse. Most Internet users will probably agree that trying to find things you are interested on the Internet can be a huge time drain. And the results can be very unsatisfactory. The rapid growth rate of information available on the Web is continually making this process of finding desired information even harder. You can spend many hours looking for information on a subject that interests you. In most cases, you will eventually find some information of value -- but even using today's advanced computer search tools and on-line directories, it can take hours or days. With the advent of the technology advances developed by InterTrust Technologies Corp. and others, publishers will find it far more appealing to make their valuable digital information assets available on-line and to allow extractions and modifications of copyrighted materials that will vastly expand the total number of information objects. This will enormously worsen the problem, as the availability of valuable information products greatly expands.

It Is Usually Hard to Find Things On the Internet

There are many reasons why it is difficult to find what you want on the Internet. One key reason is that, unlike a public library, for example, there is no universal system to classify or organize electronic information to provide information for matching with what's important to the person who is searching. Unlike a library, it is difficult on the Internet to efficiently browse over many items since the number of possible choices may be much larger than the number of books on a library shelves and since electronic classification systems typically do not provide much in the way of physical cues. For example, when browsing library shelves, the size of a book, the number of pictures in the book, or pictures on magazine covers may also help you find what you are interested in. Such physical cue information may be key to identifying desired selections from library resources. Unfortunately, most digital experiences typically do not provide such cues without actually loading and viewing the work in digital form.

Thus, another reason why the electronic or digital world can make it even harder to find information than ever before has to do with the physical format of the information. The digital information may provide few or no outward cues or other physical characteristics that could help you to even find out what it is – let alone determine whether or not you are interested in it, unless such cues are provided through special purpose informational (for example, graphical) displays. On the Internet, everyone can be an electronic publisher, and everyone can organize their offerings differently -- using visual cues of their own distinctive design (e.g., location on a web page, organization by their own system for guiding choices). As one example, one publisher might use a special purpose graphical representation such as the video kiosk to support an electronic video store. Other publishers may use different graphical representations altogether.

Historically, there has been no particular need for consistent selection standards in conventional, non-electronic store based businesses. Indeed, it is often the unique display and choice selection support for customers' decision processes that make the difference between a successful store and a failure. But in the electronic world--where your choice is not among a few stores but rather is a choice among potentially thousands or even millions of possibly useful web sites and truly vast numbers of digital containers -- the lack of a consistent system for describing commercially significant variables that in the "real" world may normally be provided by the display context and/or customized information guidance resource (catalog book, location of goods by size, etc.) seriously undermines the ability of digital information consumers to identify their most desirable choices.

Adding to this absence of conventional cues, the enormity of available choices made available in cyberspace means that the digital information revolution, in order to be practical, must provide profoundly more powerful tools to filter potentially desirable opportunities from the over abundance of choices. In sum, the absence of the ability to efficiently filter from a dimensionally growing array of choices, can completely undermine the value of having such a great array of choices.

In the "real" world, commercial choices are based on going to the right "store" and using the overall arrays of available information to identify one's selection. However, as information in digital and electronic form becomes more and more important, the problem of relating to the vast stores of information will become a nightmare. For example, picture yourself in a store where each shopping aisle is miles long, and each item on the shelf is packaged in the same size and color container. In an actual store, the product manufacturers put their products into brightly colored and distinctively shaped packages to make sure the consumer can readily find and select their product. These visual cues distinguish, for example, between a house brand

and a specific name brand, between low fat and regular foods, and between family size and small size containers.

On the Internet, a digital "store" is likely to be many stores with vast resources integrating products from many parties. If you were limited to conventional classification and matching mechanisms, you would be unable to sift through all the material to identify the commercially acceptable, i.e., an item representing the right information, at the right price, providing license rights that match your interests. Certainly, if each digital package looks the same, you are at a loss in making reasonable decisions. You can't tell one from another just by looking at it.

While information written on the "outside" of a digital package may be useful, you simply don't have the time to read all the packages, and anyway, each packager may use different words to describe the same thing and the descriptions may be difficult to understand. Some people may write a lot of information on the outside of their package, and others may write little or nothing on the outside of the package. If there is no universal system agreed upon by everyone for defining what information should be written on the outside of the package and how it should be formatted, using such a store would be painfully difficult even if you could limit the number of choices you were evaluating.

There is a Need For Efficient and Effective Selection Based, at Least in Part, on Rights Management Information

Unlike a real store where all breakfast cereals are shelved together and all soft drinks are in the same aisle, there may be no single, universal way to display the organization of all of the information in a "digital store" since, by its nature, digital information frequently has many implications and associated rules. For example, there now exist highly developed rights management systems such as described in U.S. Patent application Serial No. 08/388,107 of Ginter et al., filed 13 February 1995, for "Systems And Methods For Secure Transaction Management And Electronic Rights Protection (hereafter "Ginter et al") – the entire disclosure (including the drawings) of which is expressly incorporated into this application as if expressly set forth herein. Many rules associated with any given piece of digital information may, combinatorially, given rise to many, very different, commercial contexts that will influence the use decisions of different potential users in many different ways (e.g., cost, auditing, re-use, redistribution, regulatory requirements, etc.).

No readily available systems developed for the digital information arena provide similarly satisfying means that describe the many commercial rules and parameters found in individual custom catalogs, merchandise displays, product specifications, and license agreements. Further, no readily available mechanisms allow "surfing" across vast choice opportunities where electronic matching can single out those few preferred items.

As one example, picking an appropriate image may involve any or all of the following:

- price,
- republishing (redistribution) rights,
- rights to extract portions,
- certified usable in certain sovereignties (e.g., pornographic content not allowed in Saudi Arabia),
- size,
- format, etc.,
- use and reuse administrative requirements (e.g., which clearinghouses are acceptable to rightsholders, what is the requirement for reporting usage information – is the name of your customer required, or only the use class(es) or none -- is advertising embedded), and
- other features.

No previously readily available technology allows one to efficiently make selections based on such criteria.

By their nature, and using the present inventions in combination with, amongst other things, "Ginter et al", the packages in a digital store may be "virtual" in nature -- that is, they may be all mixed up to create many, differing products that can be displayed to a prospective customer organized in many different ways. This display may be a "narrowcasting" to a customer based upon his matching priorities, available digital information resources (e.g., repository, property, etc.) and associated, available classification information. In the absence of an effective classification and matching system designed to handle such information, digital information of a particular kind might be just about anywhere in the store, and very difficult to find since the organization of the stores digital information resources have not been "dynamically" shaped to the matching interests of the potential customer.

These Inventions Solve These Problems

The present inventions can help to solve these problems. It can give you or help you to find the things you like, need or want. For example, it can deliver to you, (including narrowcasting to you), or help you to find:

- things that match your interests;
- things that match your lifestyle;
- things that match your habits;
- things that match your personality;
- things you can afford and/or accept your preferred payment method;
- things that help you in your work;
- things that help you in your play;

- things that help you to help others;
- things that other people who are similar to you have found helpful,
- things that fulfill the commercial objective or requirements of your business activities; and
- things that will make you happy and fulfilled.

The present inventions can expand your horizons by helping you to find interesting or important things, things that you enjoy, things that optimize your business efficiency, and things that help you make the best digital products or services you can -- even if you didn't know precisely what or how to look for what you may need. It can also help you by allowing things you didn't know existed or know enough to look for – but that you may be interested in, want or need – to find you.

The Present Inventions Can Use "Metaclasses" to Take Multiple Classifications Into Account

In some areas, multiple classifications may already exist and thus it is important for a consumer to be able to find what he or she is looking for while taking into account not only that there may be multiple classifications, but also that some classifications may be more authoritative than others. For example, Consumer Reports may be more authoritative on certain topics than more casual reviews published, for example, in the local weekly newspapers.

As another example, consider a book that rates restaurants according several factors, including, for example, quality, price, type of food, atmosphere, and location. In some locations there may be many guides, but they may review different sets of restaurants. One guide may rate a particular restaurant highly while one or more others may consider it average or even poor. Guides or other sources of ratings, opinions, evaluations, recommendations, and/or value may not be equally authoritative, accurate, and/or useful in differing circumstances. One consumer may consider a guide written by a particular renowned expert to be more authoritative, accurate, and/or useful than a guide reflecting consumer polls or ballots. However, another consumer may prefer the latter because the second consumer may perceive the tastes of those contributing opinions to be closer to his or her own tastes than those of the experts.

In accordance with the present inventions, a person may be able to find a restaurant that meets specified criteria – for example, the highest quality, moderately priced Cantonese and/or Hunan Chinese food located in Boston or Atlanta – while weighting the results of the search in favor of reviews from travel books rather than from the local newspapers. As this example indicates, the searching may be according to class of authoritative source (and/or classes sources considered authoritative by the consumer) instead of weighting individual reviewers or sources. Thus in accordance with the present inventions, search may be performed at least in part based on classes of classes, or "metaclasses."

The Present Inventions Can Make Choices Easier

One simple way to look at some examples of the present inventions is as a highly sensitive electronic "matchmaker" that matches people or organizations with their best choices, or even selects choices automatically. The present inventions can match people and/or organizations with things and/or services, things with other things and/or services, and/or even people with other people. For example, the matching can be based on profiles that are a composite of preference profiles of one or more specific users, one or more user groups, and/or organizations -- where the contribution of any given specific profile to the composite profile may be weighted according to the specific match circumstances such as the type and/or purpose of a given match activity.

Figure 5 shows a simplified example of an electronic matchmaker that can match up two people with like interests. Sarah loves hiking, country and western music, gardening, movies and jogging. Mark loves movies, hiking, fast cars, country and western music, and baseball. The electronic matchmaker can look at the interests, personalities and/or other characteristics of these two people and determine that they are compatible and should be together -- while maintaining, if desired, the confidentiality of personal information. That is, unlike conventional matchmaking services, the present inventions can keep personal information hidden from the service provider and all other parties and perform matching within a protected processing environment through the use of encryption and protected processing environment-based matching analysis.

For example, certain matching of facts that are maintained for authenticity may be first performed to narrow the search universe. Then, certain other matching of facts that are maintained for secrecy can be performed. For example, matching might be based on shared concerns such as where two parties who have a given disability (such as cancer or HIV infection) that is certified by an authority such as a physician who is certified to perform such certification; or the same income level and/or bank account (as certified by an employer and/or financial authority such as a bank). Some or all of such secret information may or may not be released to matched parties, as they may have authorized and/or as may have been required by law when a match is achieved (which itself may be automatically managed within a protected processing environment through the use of controls contributed by a governmental authority).

Figure 5A shows an electronic matchmaker that matches an electronic publisher with mystery stories for his quarterly electronic mystery anthology, where the matching is based on price, redistribution rights, editing rights, attribution requirements (attributing authorship to the author), third party rating of the writers quality, length of story, and/or the topical focus of the story (for example). Here, rule managed business requirements of publisher and writers are matched allowing for great efficiency in matching, coordination of interests, and automation of electronic business processes and value chain activities.

The convenience of the "electronic matchmaker" provided in accordance with the present inventions extends to commerce in physical goods as well -- as illustrated in Figure 5b. In this non-limiting example, the electronic matchmaker is communicating to the consumer via the Internet and World Wide Web. The matchmaker has found the lowest quoted price for a Jeep sports utility model given, in this one example, a multitude of factors including:

- model,
- color,
- options package,
- availability, and
- discounts resulting from the consumer's membership in certain classes (such as membership in the American Association of Retired Persons, membership in the American Automobile Association, and being a graduate of Stanford University).

Membership in these associations and alumni status may be conveyed or indicated by possession of a special electronic document called a "digital certificate," "membership card," and/or other digital credential that warrants or attests to some fact or facts.

Thus, the electronic matchmaker provided in accordance with these inventions can also match people with things. Figure 6 shows two people, Harry and Tim. Harry loves sports most of all, but also wants to know a little about what is going on in the business world. The business world is most important to Tim, but he likes to keep up with the baseball scores. The electronic matchmaker in accordance with these inventions can learn about what Harry and Tim each like, and can provide information to a publisher so the publisher can narrowcast a newspaper or other publication customized for each of them. A newspaper company can narrowcast to Harry lots of sports information in his newspaper, and it can narrowcast to Tim mostly business information in his newspaper. In another example, Harry's newspaper may be uniquely created for him, differing from all other customized newspapers that emphasize sports over business information. But information that Harry and Tim respectively want to maintain as authentic or secret can be managed as such.

The electronic matchmaker can also match things with other things. Figure 7 shows how the electronic matchmaker can help a student put together a school project about big cats. The electronic matchmaker can help the student locate and select articles and other material about various kinds of big cats. The electronic matchmaker can, for example, determine that different articles about tigers, lions and cheetahs are all about big cats – but that articles about elephants and giraffes are not about big cats. If there is a charge for certain items, the electronic matchmaker can find only those items that the student can afford, and can make sure the student has the right to print pictures of the big cats. The electronic matchmaker can help the student to collect this information together so the student can make a colorful poster about big cats.

The electronic matchmaker can match up all sorts of different kinds of things. Figure 8 shows the electronic matchmaker looking at three different objects. The matchmaker can determine that even though objects A and C are not identical, they are sufficiently similar that they should be grouped together for a certain purpose. The electronic matchmaker can determine that for this purpose, object B is too different and should not be grouped with objects A and C. For a different purpose, the electronic matchmaker may determine that objects A, B and C ought to be grouped together.

The Present Inventions Can Make Use of Rights Management Information

How does the electronic matchmaker find out the information it needs to match or classify people and things? In accordance with a feature provided by these inventions, the electronic matchmaker gets information about people and things by using automated, computerized processes. Those processes can use a special kind of information sometimes known as rights management information. Rights management information may include electronic rules and/or their consequences. The electronic matchmaker can also use information other than rights management information.

An example of rights management information includes certain records about what a computer does and how it does it. In one simple example, records may give permission to read a particular news article if that the customer is willing to pay a nickel to purchase the article and that the nickel may be paid using a budget provided by a credit card company or with electronic cash. A customer might, for example, seek only news articles from providers that take electronic cash and/or process information with a certain information clearinghouse as described in U.S. Patent application Serial No. 08/699,712 to Shear et al., filed 12 August 1996, for "Trusted Infrastructure Support Systems, Methods And Techniques For Secure Electronic Commerce Electronic Transactions And Rights Management" (hereafter "Shear et al") – the entire disclosure (including the drawings) of which is expressly incorporated into this application as if expressly set forth herein.

The Present Inventions Can Maintain Privacy

Figure 9 shows one way in which the electronic matchmaker can get information about a person. In this example, the electronic matchmaker asks Jill to fill out a computer questionnaire about what she likes. The questionnaire can also ask Jill what information she wishes to be maintained as authentic, and what information (e.g., encrypted by the system) may be used for secure matching only within a protected processing environment and can not be released to another party, or only to certain specified parties. The questionnaire answering process may be directly managed by a protected processing environment to ensure integrity and secrecy, as appropriate.

For example, the questionnaire may ask Jill whether she likes baseball and whether she is interested in volcanoes. The electronic matchmaker can also ask Jill if it is okay to look at records her computer maintains about what she has used her computer for in the past. These computer records (which the computer can maintain securely so that no one can get to them without Jill's permission) can keep a history of everything Jill has looked at using her computer over the past month and/or other time period – this process being managed, for example, through the use of a system such as described in the "Ginter et al."

Looking at Figure 10, Jill may have used her computer last week to look at information about baseball, volcanoes and Jeeps. With Jill's permission, the electronic matchmaker can employ a protected processing environment 154 (schematically shown here as a tamper-resistant "chip" within the computer – but it can be hardware-based, software-based, or a combination of hardware and software) to look at the computer's history records and use them to help match Jill up with other kinds of things she is or may be interested in. For example, the electronic matchmaker can let an electronic publisher or other provider or information gatherer (e.g., market survey conductor, etc.) know that Jill is interested in team sports, geology and sports utility vehicles with or without more revealing detail -- as managed by Jill's choices and/or rights management rules and controls executing in her computer's protected processing environment 154. The provider can send information to Jill – either automatically or at Jill's request – about other, related things that Jill may be interested in.

Figure 11 shows an example of how rights management and other information Jill's computer maintains about her past usage can be useful in matching Jill up with things she may need or want. The computer history records can, for example, show that Jill looked at hockey information for three hours and football information for five hours during the past week. They can indicate that Jill uses a Discover credit card to pay for things, usually spends less that $10 per item, averages $40 per month in such expenses, and almost never buys new programs for her computer.

The electronic matchmaker can, with and subject to Jill's permission, look at and analyze this information. As one example, the electronic matchmaker can analyze relevant rules and controls provided by third parties who have rights in such information -- where such rules are controlled, for example, by Jill's computer's protected processing environment 154. It can also look at and analyze Jill's response to computer questionnaires indicating that she likes baseball and football. The electronic matchmaker can, based on all of this information, automatically select and obtain videos and/or other publications for Jill about team sports and that cost less than

$10 and that accept payment using a Discover card, so that Jill can preview and select those in which she may have a particular interest and desire to acquire.

Figure 12 shows that the electronic matchmaker can take into account computer history records for lots of different people. The electronic matchmaker can work with other rights management related computer systems such as "usage clearinghouses" (non-limiting examples of which are described in each of "Ginter et al" and "Shear et al") to efficiently collect rights management related information. The ability to collect history records from many different people can be very useful. For example, this can allow the electronic matchmaker to distinguish between things that are very popular and things that are not so popular.

The present inventions provide great increases in efficiency and convenience. It can save you a lot of time and effort. It can allow computers to do a lot of the work so you don't have to. It can allow you to compete with larger businesses -- and allow large business to function more efficiently -- by allowing the location of resources particularly appropriate for certain business activities. You can delegate certain complex tasks to a computer, freeing you to be more productive and satisfied with electronic activities. These automated processes can be "smart" without being intrusive. For example, they can learn about your behavior, preferences, changing interests, and even your personality, and can then predict your future interests based on your past behavior and interest expressions. These processes can ensure confidentiality and privacy – so that no one can find out detailed information about you without your consent. Across the full range of personal and business activities, the present inventions allow a degree of basic efficiency, including automation and optimization of previously very time consuming activities, so that interests and possible resources are truly best matched.

The present inventions handle many kinds of important issues and addresses the widest range of information and rights and automation possibilities. For example, the present inventions are capable of handling (but are not limited to):

- consumer information;
- computer information;
- business information;
- entertainment information;
- other content information;
- information about physical products;
- all other kinds of information.

It can reflect and employ all kinds of rights to optimize matching processes, including:

- content rights;
- privacy rights;
- governmental and societal rights;
- provider rights;

- distributor rights;
- consumer rights;
- workflow rights;
- other value chain participant rights;
- work flow rights;
- business and personal rights and processes of all kinds.

It can employ all kinds of parameter information, including:

- budget,
- pricing
- redistribution
- location (of party, item, etc.)
- privacy
- identity authenticity and/or specificity
- any other parameter information.

Pricing (for example the price of a specific item) can be used in matching based upon price per unit and/ or total price for a volume purchase, price for renting, right to redistribute, cost for redistributing items, etc.

Privacy can be used for establishing matching contingent upon usage reporting requirements for viewing, printing, extracting, dedistributing, listening, payment, and/or requiring the reporting of other information such as personal demographics such as credit worthiness, stored value information, age, sex, marital status, race, religion, and/or usage based generated profiling information based materially upon, for example, a users history of usage of electronic content and/or commercial transactions, etc.

Identity can be used for matching based upon, for example, such as the presence of one or more specific, class, and/or classes of certificates, including, for example, specific participant and/or group of participant, including value chain certificates as described in "Shear et al".

With the inventions described herein, commercial requirement attributes embodied in rules (controls and control parameter data) are employed in classification structures that are referenced by search mechanisms, either, for example, directly through reading rule information maintained in readable (not encrypted) but authentic (protected for integrity) form, through reading rule information maintained securely, through processes employing a protected processing environment 154 of a VDE node, and/or through the creation of one or more indexes and/or like purpose structures, that, directly, and/or through processes employing a protected processing environment 154, automatically compile commercial and other relevant (e.g., societal regulatory information such as a given jurisdiction's copyright, content access and/or taxation regulations) for classification/matching purposes.

The present inventions can employ computer and communication capabilities to identify information, including:

- topical classification such as described by conventional library classification systems,
- commercial characterizations -- including commercial parameter data such as pricing, size, quality, specific redistribution rights, etc.,
- creator (e.g., a publisher or manufacturer), distributor, societal, user, and other participant interests information,
- information generated by automated profiling of any and all of such parties or collections of parties,
- matching (including electronically negotiating a match) between the interests of any of such parties,
- where appropriate, the use of statistical procedures, expert systems, and artificial intelligence tools for profiling creation and/or analysis, matching, and/or negotiation.

The present inventions thus provide for optimal user, provider, and societal use of electronic cyberspace resources (for example, digital information objects available across the Internet, sent by direct broadcast satellite, transmitted over a cable TV system, and/or distributed on optical disk).

Of particular importance is the notion of classes of content, classes of users, and classes of providers. For example, the present inventions can make use of any/all of the following:

- topical identification, for example, such as information represented in typical library subject and/or author and/or catalog and/or keyword search and retrieval information systems;
- any commercial requirements, associated with the use of electronic information (and/or to products, including non-electronic products, and/or to any service), including information embodied in encrypted rules (controls and/or parameter data) governing rights in electronic value chain and electronic interaction contexts, and further including information guaranteed for integrity;
- any information descriptive of an available resource (which may include any information, product, and/or service, whether available in electronic and/or physical forms) such as: the quality of a digital product as evaluated and ranked and/or otherwise specified by one or more third parties and/or independent third parties (e.g., Consumer Reports, a trusted friend, and/or a professional advisor), the size of a product, length in time in business of a service or in the market of a product, a product's or service's

market share, and/or subject governmentally and/or other societally imposed rules and/or integrity guaranteed descriptions, including any associated regulatory requirements, such as societal requirements granting and/or reporting access to information, for example, information on how to create a nuclear bomb to a confidential government auditing agency (this allowing free access to information while protecting societal rights);

- any information descriptive of a user and/or department and/or organization and/or class of users and/or departments and/or organizations (including, for example, such descriptive information encrypted and/or guaranteed for integrity) wherein such information may include, for example, name, physical and/or network and/or cyber-wide logical network location, organizational and/or departmental memberships, demographic information, credit and/or trustworthiness information, and profile preference and usage history information, including any generated profile information reflecting underlying preferences, and/or classes based on said descriptive information and/or profiles.

Some Of The Advantageous Features And Characteristics Provided By The Present Inventions

The classification, matching, narrowcasting, analysis, profiling, negotiation, and selection capabilities of the present inventions include the following capabilities (listed items are not mutually exclusive of each other but exemplary samples):

- Enables highly efficient provision of classes of information, entertainment, and/or services to classes of individuals and/or entities that have (and/or may obtain) the right(s) to such information and are likely to find identified information interesting, useful, and/or entertaining.
- The present inventions also provide systems and methods for efficiently determining class hierarchies, classification schemes, categories, and/or category schemes and/or the assignment of objects, persons and/or things to said class hierarchies, classification schemes, categories, and/or category schemes using at least some rights management information.
- Helps systems, groups, and/or individuals classify, locate, and/or obtain specific information and/or classes of information made available through so-called "publish and subscribe" systems and methods using, among other things, subject-based addressing and/or messaging-based protocol layers.

- Provides fundamentally important commercial and societal rules based filtering to identify desired electronic information and/or electronic information containers through the use of classification structures, profiling technology, and matching mechanisms that harness the vast information opportunities in cyberspace by matching the information needs of users against commercial and/or societal rules related to the use of available information resources, including, for example, commercial and/or societal consequences of digital information use imposed as provider requirements and specified through the use of, and enforced by the use of, a trusted rights management system such as described in "Ginter et al".
- Enables content creators and/or distributors to efficiently "stock the shelves" of retail electronic content outlets and similar merchanisers (both electronic and hard goods) with products and/or services most likely to be purchased and/or used by the customers of such merchanisers. This includes both identifying and "stocking" the most desirable products and/or other user desired resources and optimally presenting such products and/or other

resources in a manner optimized for specific users and/or user classes.

- Matching may be based on history of matching, that is, matching derived at least in part from previous matching, one non-exhaustive example of which includes learned matching for increasing efficiency.
- Enables matching for value chains where the matching is against a plurality of co-participating value chain parties requirements and/or profiles against match opportunities, and/or matching by matches comprised of match input and/or aggregation of match rule sets of providers used to "dock" with one or more user needs, interests, requirements match sets.
- Helps match persons and/or things using fuzzy matching, artificial intelligence (e.g., expert systems), and other methods that that match using plural match sets from providers and/or receivers.
- Makes search easier by using smart agents that match at least in part using at least one class.
- Helps bring buyers and sellers together through cross matching, where both parties offer to provide and/or receive content and/or physical goods for consideration, including barter matching and negotiated barter and other kinds of matching.

- Helps potential customers find those members (e.g., objects such as digital information containers) of any one or more classes of content most useful, entertaining, and/or interesting to them.
- Facilitates organizations securely and efficiently acquiring and distributing for internal use certain classes of content available from external providers and/or more securely and/or efficiently managing classes of their own content, including being able to authorize certain classes of employees to use specified classes of internal and/or external content.
- Efficiently supporting matching between users and digital information where participants in a chain of handling and control have specified rules and usage consequences for such digital information that may depend on class membership, for example, on class(es) of content and/or class(es) of value chain participants and/or classes of electronic events, wherein such participants include, for example, users and/or participants contributing rules and consequences.
- Enables first individuals and/or organzations to locate efficiently other individuals, organizations, products, and/or services who have certain characteristics that corresponds to such first individuals' and/or

organizations' interests, including interests generated by profiling information locally gathered through local event auditing at a VDE installation.

- Facilitates businesses informing a customer about things of special interest to her or him, such as classes of goods, services, and/or content, including directing such information to a customer at least in part based on profiling information locally gathered at a VDE installation through local event auditing at a VDE installation.
- Allows trading companies to match suppliers of certain classes of goods and/or services with those who desire to purchase and/or use those classes of goods and/or services, wherein such matches may include fulling a commercial business interaction and may further include one or more sequences of matches and/or nested matches (a sequence and/or grouping of matches within a given organization or group, wherein such matches may be required to occur in a certain order and/or participate along with other matches in a group of matches before a given match is fulfilled).
- Enhances equity portfolio management by making easier for traders to identify those equities having certain desired characteristics, such as belonging to

the class of equities that will have the greatest positive effect on the value of the trader's portfolio given certain classes of information and assumptions. Such matches may take into account information external to the fulfilment of a given trade, for example, one or more certain other market or specific variable thresholds must be met before an equity is traded, such as a certain rise in the an index stock value of, and/or revenue of, certain one or more network hardware suppliers before a certain quantity of equity is purchased at a certain price for stock of a certain network hardware supplier raw network component manufacturer, and wherein, for example, such determinations can be performed highly efficiently at a user VDE installation as the point of control, where such node receives such trusted information in, for example, VDE containers, as is necessary for a control decision to occur to purchase such equity of such network hardware supplier raw component manufacturer.

- Makes easier automated foreign currency exchange by enabling currency traders to identify members of the class of possible trades and/or conversions that are likely to produce the best returns and/or minimize losses.

- Helps consumers and organizations manage their affairs more efficiently and effectively and helps providers of services by automatically matching users with services that meet certain specified criteria, such as, for example, U. S. and Swiss banks offering the highest interest rates on certain time based classes of bank deposit instruments.
- Enables distributers of software and other content to identify one or more classes of users who are most likely to be interested in purchasing or otherwise using certain classes of software.
- Enables rightsholders to employ rules and/or usage consequences dependent on membership in one or more classes where class membership may be indicated by posession of a special digital document called a "certificate."
- Enables rightsholders to employ rules and/or usage consequences at least partially dependent on roles and responsibilities within an organization, where those roles and responsibilities may be indicated by posession of a digital certificate, digital membeship card, and/or other digital credential.
- Facilitates more efficient automation of manufacturing and other workflow processes by, for example, matching certain manufacturing steps and/or

processes with performance parameter data associated with available classes of equipment capable of performing those steps and/or processes.

- Makes easier the administration and enforcement of government and/or societal rights by, for example, providing matching means for automatically applying certain classes of tax rules to appropriate classes of sales and other transactions.
- Enables altering the presentation of information and/or other content depending on the matching between preferences of the user and one or more classes of content being presented.
- Enables processing or altering (narrowcasting) of an event (e.g., the presentation of information and/or other content), for example, dynamically adjusting the content of an event, in response to a matching among the preferences and/or reactions of a user and/or user group, one or more classes of content being processed through one or more events, one or more classes of one or more users participating in and/or otherwise employing the one or more events, and/or event controls (i.e., rules and/or parameter data).
- Allows the rules and usage consequences and the presentation of information to vary according to the difficulty of the information, including, for example,

adjusting the difficulty of an electronic game so that it is neither too frustratingly difficult nor too easy to use.

- Enables a user to efficiently locate content in one or more particular classes, where class is defined at least in part by weighted topical classification, where, for example, a document or other object is classified in one or more categories where at least one category reflects the absolute or relative attention given to that class in the object being classified.
- Facilitates users' creation of a new document from parts of two or more documents, where at least one of such parts is identified and/or retrieved based upon matching the part's membership in one or more classes identified by trusted, commercial controls employed through the use of a rights management system.
- Enables users to search for, locate, and use only those parts of a document that belong to one or more specified classes, including those parts having certain commercial controls, for example, reflecting acceptable usage restrictions and/or pricing.
- Enhances search and retrieval by creating new classes of content discriptors that incorporate various

dispirate standards for content description and/or location.

- Allows consumers to easily locate services having certain specified characteristics, for example, gambling services offering the most favorable odds and/or specified rules for a particular game or games.
- Helps consumers obtain certain classes of tickets to certain classes of events.

The above capabilities, and others described in this application, are often ideally managed by distributed commerce nodes of a distributed, rights management environment embedded in or otherwise connected to the operating system clients of a distributed computing environment such as described in "Ginter et al" and further described in "Shear et al", and employing, for example, rules, integrity management, container, negotiation, clearinghouse services, and trusted processing capabilities described in "Ginter et al" and "Shear et al".

The Present Inventions Make Use Of Many Kinds Of Information And/Or Data

As discussed above, these inventions provide, among other things, matching, classification, narrowcasting, and/or selection based on rights management and other information. In particular preferred examples, these matching, classification, narrowcasting, and/or selection processes and/or techniques may be based at least in part on rights management information. The rights management information may be an input to the process, it may be an output from the process, and/or the process can be controlled at least in part by rights management information. Information in addition to, or other than, rights management information may also be an input, an output, and/or a basis for controlling, the process and/or techniques.

Rights management information may be directly or indirectly inputted to the matching, classification and/or selection process. For example, rights management controls, rules and/or their consequences may be an input. Examples of such controls and/or rules include object registration related control set data, user related control set data and/or computer related control set data. In addition or alternatively, information provided based on control sets or rules and their consequences may be inputted. The following are examples of such information that may be provided based, for example, on rules and consequences:

- information exhaust;
- user questionnaires,
- audit trail related information;
- aggregated usage data;
- information measuring or otherwise related to user behavior;
- information measuring or otherwise related to user preferences;

- information measuring or otherwise related to user personality;
- information measuring or otherwise related to group behavior;
- information measuring or otherwise related to group preferences;
- information measuring or otherwise related to group culture
- information measuring or otherwise related to organizational behavior;
- information measuring or otherwise related to organizational preferences;
- information measuring or otherwise related to organizational culture;
- information measuring or otherwise related to institutional behavior;
- information measuring or otherwise related to institutional preferences;
- information measuring or otherwise related to institutional culture;
- information measuring or otherwise related to governmental behavior;
- information measuring or otherwise related to governmental preferences;

- information measuring or otherwise related to governmental culture;
- information measuring or otherwise related to societal behavior;
- information measuring or otherwise related to societal preferences;
- information measuring or otherwise related to societal culture;
- object history related information;
- other types of information;
- any combinations of information including, some, all or none of the information set forth above.

The processes, techniques and/or systems provided in accordance with these inventions may output rights management related information such as, for example:

- one or more control sets;
- various rules and/or consequences;
- information used by control sets;
- certificates;
- other rights management information.

In accordance with various preferred embodiments provided by these inventions, information other than rights management information may also be used, at least in part, as an input, output and/or to control the matching, classification, narrowcasting, and/or selection processes, systems and/or techniques. Examples of such information include:

- content object information;
    - full text
    - portions of objects
    - portions of sub-objects
    - abstracts
    - metadata
    - other content object related information
- user information
    - census information
    - purchasing habits
    - credit and financial transaction related information
    - governmental records
    - responses to questionnaires
    - survey results
    - other user information
- computer related information
    - identification information
    - configuration information
    - other computer related information
- combinations of information.

Matching/Classifying/Selection

Systems, methods and techniques provided in accordance with these inventions can classify a variety of types of things including, for example:

- people
- computers
- content
- events
- transactions
- objects of all types
- combinations of things;
- combinations of people and things.

The matching, classifying and/or selecting processes provided in accordance with these inventions are very flexible and useful. For example, they may be used to associate people with information, information with other information, people with other people, appliances with people, appliances with information, and appliances with other appliances. The present inventions in their preferred examples can associate any kind of information, object or thing with any other kind of information, object or thing.

Different Associations Between Classes and Rights

The processes, systems and/or techniques provided in accordance with these inventions can provide and/or take into account many different kinds of associations between classes and rights. For example, they can look at what rights are available to a user, computer, data structure or any other object. They can also look to rights selected by an object (for example, the subset of rights a user has chosen or otherwise identified). Alternatively or in addition, they can look to rights that have been exercised by a user or in conjunction with an object or other thing, and they can look to the consequences of exercising such a right(s).

Embodiments in Accordance With the Present Inventions Can Be Used to Define Classes Based on Uni-Dimensional and/or Multi-Dimensional Attributes and/or Characteristics

Example processes, systems and/or techniques provided in accordance with these inventions can be used to define classes based on uni-dimensional and/or multi-dimensional attributes and/or characteristics. Any one or more attributes can be used. The attributes and/or characteristics can be flexibly defined. They may define groups or classes containing elements sharing certain attributes in common. There can, for example, be a spectrum of classification that takes into account gray areas as to whether a particular person or thing possesses a certain one or a number of particular attributes and/or characteristics. Or classification may have a higher degree of certainty or definition. For example, a process can test to determine whether particular people or things are inside or outside of particular classes or groups based on one or a number of attributes or characteristics (for example, whether you live in Denver, are under the age of 25 and are single). In accordance with additional specific features provided by these inventions, there may be a minimum number of different classes set up to "cover" a particular situation – with every person or thing either being within or outside of a given, disjoint class or group.

Preferred Examples In Accordance With The Present Inventions Are Extensible to Accommodate Changing Conditions

The systems, methods and/or techniques provided by these inventions are extensible to accommodate changing conditions. For example, they can be made to readily adapt to changes in rules, consequences, topics, areas and/or subjects pertaining to groups such as, for example categories, and any other variable. Furthermore, partially and/or entirely new variables may be introduced to one or more existing sets of variables -- for example, to extend or otherwise modify a model to account for additional variables, to apply a new strategy, to adapt to new network and/or installation circumstances, to adapt to new user factors, to change analysis and/or other processing characteristics, and so on.

Preferred Examples In Accordance With The Present Inventions Are Compatible With Pre-Existing or Any New Classification Techniques or Arrangements

The example systems, methods and/or techniques provided by these inventions can be made fully compatible with any classification and/or categorization means, method, process, system, technique, algorithm, program, and/or procedure, presently known or unknown, for determining class and/or category structures, definitions, and/or hierarchies, and/or the assignment of at least one object, person, thing, and/or member to at least one class and/or category, that without limitation may be:

- implemented by computer and/or other means; and/or
- based upon discrete and/or continous mathematics; and/or
- using nominal, ordinal, interval, ratio and/or any other measurement scale and/or measurement mode; and/or
- including parameter data; and/or
- entail linear and/or non-linear estimation methods; and/or
- any other methods.

For example, classification can be performed using any or all of the following example classification techniques:

- Statistical techniques that identify one or more clusters of cases sharing similar profiles and/or features, including any of the family of cluster analysis methods, for example, those described in Hartigan (Hartigan, J. A., Clustering Algorithms, New York: Wiley, 1975);
- Methods for numerical taxonomy, for example, as described, for example, by Sneath and Sokal (Sneath, Peter H.A. and Robert R. Sokal, Numerical

Taxonomy: The Principals and Practice of Numerical Classification, San Francisco: W.H. Freeman, 1973);

- Any of the methods for cluster analysis, factor analysis, components analysis, and other similar data reduction/classifiction methods, for example, those implemented in popular statistical and data analysis systems known to those skilled in the arts, for example, SAS and/or SPSS;
- Pattern classification techniques, including components analysis and neural approaches, for example, those described by, for example, Schurmann (Schurmann, Jurgen, Pattern Classification: A Unified View of Statistical and Neural Approaches, New York: John Wiley & Sons, 1966);
- Statistical techniques that identify one or more underlying dimensions of qualities, traits, features, characteristics, etc., and assign parameter data indicating the extent to which a given case has, possesses, and/or may be characterized by the underlying dimension, factor, class, etc. and/or result in the definition of at least one class and/or the assignment of at least one case to at least one class, for example, as described by Harman (Harman, Harry H., Modern Factor Analysis, 3rd ed. rev., Chicago: University of Chicago Press), and/or as implemented

by SAS and/or SPSS and/or other statistical analysis programs.

- Statistical methods that employ fuzzy logic and/or fuzzy measurement and/or whose assignment to at least one class entails probabilities different from 1 or zero.
- Baysian statistical classification techniques that use estimates of prior probabilities in determining class definitions and/or the assignment of at least one case to at least one class;
- Any statistical and/or graphical classification and/or data reduction method that uses rotation of reference axes, regardless of whether orthogonal or oblique rotations are used, for example, as described in Harman, and as implemented in SAS and/or SPSS and/or other statistical programs;
- Statistical methods for two and three way multidimensional scaling, for example, the methods described by Kruskal and Wish (Krusgal Joseph B. and Myron Wish, Multidimensional Scaling, Beverly Hills, CA: Sage Publications, 1978), and/or by Shepard, et al. (Shepard, Roger N., A. Kimball Romney, and Sara Beth Nerlove, Multidimensional Scaling: Theory and Applications in the Behavioral Sciences, New York: Seminar Press, 1972);

- Knowedge based approaches to classification, for example, as described by, for example, Stefik (Stefik, Mark, "Introduction to Knowledge Systems," San Francisco: Morgan Kauffman, 1995); and
- any other classification techniques or arrangements pre-existing or yet to be developed.

Preferred Examples In Accordance With The Present Inventions Are Fully Compatible With A Wide Array of Technologies Including the Distributed Commerce Utility System and the Virtual Distribution Environment

Systems, methods and/or techniques provided in accordance with these inventions build upon and can work with the arrangements disclosed in “Ginter et al”; “Shear et al”; and other technology related to transaction and/or rights management, security, privacy and/or electronic commerce.

For example, the present inventions can make particular use of the security, efficiency, privacy, and other features and advantages provided by the Virtual Distribution Environment described in “Ginter et al”.

As another example, a matching and classification arrangement can be constructed as a distributed commerce utility system as described in “Shear et al”. The present inventions can work with other distributed commerce utility systems, and can enhance or be a part of other commerce utility systems.

By way of non-exhaustive, more specific examples, the present inventions can be used in combination with (and/or make use of) any or all of the following broad array of electronic commerce technologies that enable secure, distributed, peer-to-peer electronic rights, event, and/or transaction management capabilities:

- a "VDE" ("virtual distribution environment") providing, for example, a family of technologies by which applications can be created, modified, and/or reused;
- a standardized control and container environment which facilitates interoperability of electronic appliances and efficient creation of electronic commerce applications and models;
- a programmable, secure electronic transaction management foundation having reusable and extensible executable components;
- seamless integration into host operating environments of electronic appliances or direct employment of such technologies in electronic commerce applications;
- cyberspace digital content rights and transaction management control systems that may operate in whole or in part over Internets, Intranets, optical media and/or over other digital communications media;
- support of an electronic "world" within which most forms of electronic transaction such as content usage,

distribution, auditing, reporting, and payment activities can be managed;

- Transaction Operating Systems (operating systems that have integrated secure, distributed, and programmable transaction and/or event management capabilities);
- Rights Operating Systems (operating systems that have integrated, distributed, and programmable rights management capabilities);
- secure content container management;
- clearinghouse functions related to content usage;
- overall electronic commerce architectures that provide electronic commerce automation through the use of secure, distributed digital events management;
- the general enablement of traditional commerce behavior in the digital commerce world;
- enhanced inherent, distributed efficiencies of conventional commerce practices with powerful, reliable electronic security, and with the programmability and electronic automation efficiencies made possible by modern computing;
- trusted operation of a freely configurable, highly efficient, general purpose digital marketplace in which

parties "come together" to establish commercial relationships;

- support of "real" commerce in an electronic form (that is, the progressive creation of commercial relationships that form, over time, a network of interrelated agreements representing a value chain business model);
- enabling content control information to develop through the interaction of (and/or negotiation between) securely created and independently submitted sets of content and/or appliance control information;
- interconnection of appliances providing a foundation for much greater electronic interaction and the evolution of electronic commerce;
- a variety of capabilities for implementing an electronic commerce environment;
- a neutral, general purpose platform for commerce;
- an architecture that avoids reflecting specific distribution biases, administrative and control perspectives, and content types;
- a broad-spectrum, fundamentally configurable and portable, electronic transaction control, distributing, usage, auditing, reporting, and payment operating environment;

- systems and methods that uniquely enable electronic commerce participants to protect their interests during the sequence of activities comprising an electronic commerce model;
- ability of commerce participants to assure protection by specifying rules and controls that monitor and enforce their interests during the processing of remote commerce events;
- permitting commerce participants to efficiently participate in, and manage, the distributed electronic activities of a digital value chain;
- allowing commerce model participants to, for example, securely and cooperatively govern and automate the distributed electronic activities comprising their collective electronic business models;
- allowing commerce model participants to securely contribute electronic rules and controls that represent their "electronic" interests;
- rules and controls that extend a "Virtual Presence™" through which the commerce participants govern remote value chain activities according to their respective, mutually agreed to rights;

- a Virtual Presence taking the form of participant specified electronic conditions (rules and controls) that must be satisfied before an electronic event may occur;
- rules and controls that enforce the party's rights during "downstream" electronic commerce activities;
- control information delivered by, and/or otherwise available for use with, the VDE content containers constituting one or more "proposed" electronic agreements which manage the use and/or consequences of the use of such content and which can enact the terms and conditions of agreements involving multiple parties and their various rights and obligations;
- rules and controls from multiple parties forming aggregate control sets ("Cooperative Virtual Presence™") that ensure that electronic commerce activities will be consistent with the agreements amongst value chain participants;
- control sets defining the conditions which govern interaction with protected digital content (disseminated digital content, appliance control information, etc.);
- conditions used to control not only digital information use itself, but also the consequences of such use to protect the individual interests of commerce participants

and form cooperative, efficient, and flexible electronic commerce business models;

- true, efficient electronic cooperative governance of value chain activities;
- empowering each commerce model participant to securely deliver, and persistently maintain control over, the rules and controls they contributed specifying constraints on, and consequences of, electronic conduct;
- extending Cooperative Virtual Presence over time and involving the execution of controls, and the use of content, at physically dispersed locations, such as Internet user sites;
- a chain of handling and control in which dispersed locations are bound together through the use of secure communication techniques and unique, secure digital container technology;
- ability to preserve the rights of parties through a series of transactions which may occur at different times and different locations;
- extending the ability of electronic content providers to control the use of proprietary information;
- allowing content providers to limit use to authorized activities and amounts;

- allowing participants (e.g., actors, directors, script and other writers, musicians, studios, publishers, distributors, retailers, advertisers, credit card services, content end-users, and others) involved in a business model to have the ability to embody their range of agreements and requirements, including use limitations, into an "extended" agreement comprising an overall electronic business model;
- representing such an extended agreement by electronic content control information which can automatically enforce agreed upon rights and obligations;
- a competitive, general purpose electronic commerce architecture supporting the distributed, secure "unmanned" electronic interaction;
- distributing such capabilities across networks and involving the sequence (or web) of distributed activities underlying electronic value chains;
- cooperative electronic governance of distributed electronic commerce processes that optimizes electronic commerce value propositions;
- the capability of electronically, remotely representing the interests of commerce participants to support efficient, flexible, commerce model automation;

- enabling rules and controls that are independently contributed by multiple parties to securely merge together and form the collective rules and controls sets that reflect the electronic commerce agreements between parties;
- using rules and controls sets to collectively, automatically, govern remote electronic conduct;
- securely managing the integration of control information provided by two or more parties;
- constructing electronic agreements between VDE participants that represents a "negotiation" between the control requirements of two or more parties and enacts the terms and conditions of a resulting agreement;
- ensuring and/or enforcing the rights of each party to an electronic agreement regarding a wide range of electronic activities related to electronic information and/or appliance usage;
- the ability to broadly support electronic commerce by securely managing independently delivered VDE component objects containing control information (normally in the form of method, data, or load module VDE objects);
- using independently delivered control information to negotiate with senior and other pre-existing content

control information to securely form derived control information;

- ensuring that all requirements specified by derived control information are satisfied before VDE controlled content is accessed or otherwise used;
- ensuring that all load modules and any mediating data which are listed by the derived control information as required are available and perform their required function;
- use of independently delivered control components to allow electronic commerce participants to freely stipulate their business requirements and trade offs;
- allowing electronic commerce, through the various control requirements stipulated by VDE participants, to evolve into forms of business which are the most efficient, competitive and useful -- much as with traditional, non-electronic commerce;
- providing commerce participants with the ability to freely fashion the chains of handling and control pathways that protect data and processes and the freedom to shape the models within which their Virtual Presence operates -- allowing commerce participants to optimally formulate their electronic commerce value propositions;

- VDEs configured to support the various underlying agreements between parties that define important electronic commerce pathways of handling for electronic content, content and/or appliance control information, content and/or appliance usage information and payment and/or credit;
- allowing content creators and other providers to specify the pathways that, partially or fully, must be used to disseminate commercially distributed property content, content control information, payment administrative content, and/or associated usage reporting information;
- empowering commerce participants, subject to the rules and controls previously set in a value chain, to freely fashion control models implementing their Virtual Presence by using GUI templates or rights programming languages employing commerce/rights management components;
- component based control methods that allow the present inventions to efficiently operate as a highly configurable content control system;
- content control models that can be iteratively and asynchronously shaped, modified, and otherwise updated to accommodate the needs of VDE participants;

- iterative and/or concurrent multiple participant processes through the submission and use of secure, control information components (e.g., executable code such as load modules and/or methods, and/or associated data);
- control information for Virtual Presence employed in protected processing environment nodes located at user sites to ensure that digital events are governed in accordance with the collective rights of commerce model participants;
- digital events that launch or require other digital events;
- digital events that may include, for example, content use consequences such as collection of audit information, secure communication of such information, payment for content use, or satisfaction of any other electronically stated condition;
- events that occur within either the secure setting of a local node, or more widely within the secure environment of a distributed system of nodes;
- the association of Virtual Presence rules and controls with protected information enclosed within one or more electronic content containers to achieve a high order of configurability for Virtual Presence chains of handling and control;

- distribution using VDE that may package both the electronic content and control information into the same VDE container, and/or may involve the delivery to an end-user site of different pieces of the same VDE managed property from plural separate remote locations and/or in plural separate VDE content containers and/or employing plural different delivery means;
- content control information that is partially or fully delivered separately from its associated content to a user VDE installation in one or more VDE administrative objects;
- delivery of portions of said control information from one or more sources;
- making control information available for use by access from a user's VDE installation secure sub-system to one or more remote VDE secure sub-systems and/or VDE compatible, certified secure remote locations;
- use of delivery means that may include electronic data storage means such as optical disks for delivering one portion of said information and broadcasting and/or telecommunicating means for other portions of said information;

- allowing a content provider to deliver different business rules to a large corporate customer, compared with rules delivered to "retail" customers;
- supporting separation of content and Virtual Presence controls to allow a provider to associate different control sets with the same content – and not requiring the provider to create one set of content controls that apply to all types of customers;
- allowing content provider modification over time of rules and controls to reflect sales, new pricing, special discounts, etc. – while limiting this right by rules and controls provided by other parties having more senior rights;
- employing secure object container technology to efficiently implement Virtual Presence chains of handling and control;
- use of software container technology to significantly facilitate the organized dissemination of digital content, including the specialized form of digital content constituting rights control information;
- employing object software technology and using object technology to form containers for delivery of at least in part encrypted or otherwise secured information;

- using containers that contain electronic content products or other electronic information and some or all of their associated permissions (control) information;
- distributing container objects along pathways involving content providers and/or content users;
- securely moving containers between nodes of a VDE arrangement, which nodes operate VDE foundation software and execute control methods to enact electronic information usage control and/or administration models;
- employing delivered containers both for distributing VDE control instructions (information) and/or to encapsulate and electronically distribute content which has been at least partially secured;
- supporting the essential needs of electronic commerce value propositions by uniting fundamental configurability with secure Virtual Presence;
- virtual presence across virtual networks in accordance with the underlying agreement amongst commerce model participants to allow each participant to enjoy secure, reliable electronic automation of commerce models;
- allowing each rights holder's Virtual Presence at a remote site to possess the sole authority to administer or delegate the participant's electronic rights;

- capabilities that contribute to establishing an environment of trusted cooperative governance;
- practical enhancements relating to the establishment of secure event management and the maintenance of secure audit, encryption, budget, and other relevant information;
- control structures for an overall, distributed, secure rights/event administration environment;
- processes for interaction between independently delivered rules and controls, including electronic negotiation;
- creating distributed rights operating systems;
- integrating control processes into host operating environments;
- secure semiconductors to support protected processing environments;
- a secure, programmable, digital event management component architecture in which components are fully assembleable and reusable;
- differing assemblages of components formed to reflect an exhaustive array of commerce model functional capabilities, overall model implementations, and ad hoc event management scenarios;

- support for the full range of digital content types, delivery modes, and reporting and other administrative activities;
- traveling objects;
- smart agents;
- "atomic" load module operation to support "sparse space," cost-effective, secure processing semiconductors;
- smart card and other traveling client nodes;
- creating rights management software container technologies, including extraction, embedding, and other secure container content management processes;
- Chain of Handling and Control generation of secure objects (containers) and associated control information;
- audit reconciliation and usage pattern evaluation processes;
- specialized cryptographic implementations;
- use of a specialized electronic rights and commerce language, unique applications for fingerprinting and/or watermarking technologies, secure control structures, the formulation of new types of metering technologies, reciprocal event management (employing dispersed user sites) for automating web-like commerce models, and many other designs and capabilities;

- mechanisms to persistently maintain trusted content usage and reporting control information through both a sufficiently secure chain of handling of content and content control information and through various forms of usage of electronic information;
- rights management technology supporting persistent, distributed controls;
- means enabling continuing Virtual Presence through Chains of Handling and Control;
- persistency of control as a unique and fundamentally important attribute underlying Virtual Presence and Chain of Handling and Control for enabling true commerce behavior in cyberspace including ad hoc relationships and activities, distributed processes, and reliable enforcement of agreements between parties;
- Persistent Virtual Presence controls that continue to be enforced -- to the extent required by the controls themselves -- as protected digital content is, for example, used and reused, copied and further distributed, extracted and embedded, audited and reported;
- persistency responsive to rules and controls associated with electronic events, that causes new secure content containers to be created automatically by systems and methods supplying the procession of secure transport

vehicles required by Chain of Handling and Control for conveying disseminated content, associated rules and controls, and audit information and payment;

- container creation to carry extracted content, payment tokens, control information, audit information, and the like;
- securely generated containers carrying with them rules and controls stipulated by rules and controls associated with one or more triggered electronic events;
- capabilities for persistency and independent secure delivery and merging of rules and controls that provide technical means for ensuring that dynamic user behavior can be encouraged, rather than discouraged;
- dynamic user behavior encouraged as a critical link in building ad hoc relationships and cost-effectively distributing content, while simultaneously ensuring that rights holders are protected and retain control over their business models;
- enabling ad hoc behavior that frees users from constraints on their conduct resulting from inflexible, first generation technologies;
- support for enterprising behavior that is characteristic of traditional commerce resulting in more efficient and more satisfying electronic commerce experiences;

- general purpose character electronic commerce technologies provided by a combination of important capabilities including component, object oriented, programmable control language; secure specialized container technology; independent delivery of secure control information mechanisms; Chain of Handling and Control persistency of control mechanisms; event driven operating system functions; and the advanced security architecture – allowing multiple simultaneous models to evolve, and practically and efficiently operate;
- general purpose rights and event management architecture that is intrinsically reusable for many simultaneous models -- providing enormous competitive economic advantages over technologies that are essentially single model by design;
- commerce architecture client nodes that are basic pieces of reusable cyberspace infrastructure;
- generalized configurability resulting, in part, from decomposition of generalized requirements for supporting electronic commerce and data security into a broad range of constituent "atomic" and higher level components (such as load modules, data elements, and methods) that may be variously aggregated together to form control methods for commercial electronic agreements and data security arrangements;

- a secure operating environment employing VDE foundation elements along with securely deliverable VDE components that enable electronic commerce models and relationships to develop;
- the unfolding of distribution models in which content providers, over time, can expressly agree to, or allow, subsequent content providers and/or users to participate in shaping the controls for, and consequences of, use of electronic content and/or appliances;
- a very broad range of the functional attributes important for supporting simple to very complex electronic commerce and data security activities;
- electronic information and/or appliance usage control (including distribution), security, usage auditing, reporting, other administration, and payment arrangements;
- capabilities that rationalize the support of electronic commerce and electronic transaction management stemming from the reusability of control structures and user interfaces for a wide variety of transaction management related activities;
- content usage control, data security, information auditing, and electronic financial activities that can be

supported with tools that are reusable, convenient, consistent, and familiar;

- a general purpose Rights Operating System employing a reusable kernel and rights language components that provides the capabilities and integration needed for the advanced commerce operating systems of the future;
- a general purpose, reusable electronic commerce capabilities that all participants can rely on will become as important as any other capability of operating systems;
- such a rights operating system providing rights and auditing operating system functions and other operating system functions -- the rights and auditing operating system functions securely handling tasks that relate to virtual distribution environment;
- secure processing units and/or protected processing environments that provide and/or support many of the security functions of the rights and auditing operating system functions;
- an overall operating system designed from the beginning to include the rights and auditing operating system functions plus the other operating system functions -- or incorporation of the rights and auditing operating system

functions as an add-on to a preexisting operating system providing the other operating system functions;

- operating system integration and the distributed operating systems; and
- a rational approach - a transaction/distribution control standard - allowing all participants in VDE the same foundation set of hardware control and security, authoring, administration, and management tools, for widely varying types of information, business market model and/or personal objectives;

Any or all of these features may be used in combination with the inventions disclosed herein.

Brief Description of the Drawings

These and other features and advantages will be better and completely understood by referring to the following detailed description of presently preferred example embodiments in accordance with the drawings, of which:

Figures 1A-4 show "prior art" examples of how it is hard to find things you need or want;

Figures 5-12 are simplified examples of what example systems, methods and techniques in accordance with these inventions can do;

Figures 13, 14 and 14A show an example matching and classification utility system architecture;

Figures 15-15G show examples of how a matching and classification utility system can interact with other commerce utility systems;

Figures 16A-16C show examples of distributed matching and classification utility system organizations;

Figure 17 shows example matching and classification utility system functionality definitions;

Figures 18-46(B) show example steps that may be performed by the example matching and classification utility system; and

Figures 47-70 show some example matching and classification utility system applications.

Detailed Description Of Presently Preferred Example Embodiments

Figures 5-12 and the discussion above provide an introduction to the following detailed description of presently preferred embodiments in accordance with these inventions. The "electronic matchmaker" shown in Figures 5-12 is implemented in these more detailed embodiments by a matching and classification utility system 900.

Example Matching And Classification Utility

Figure 13 shows an example matching and classification utility system 900 as including:

- an object classifier 902;
- a user (people) classifier 904; and
- a matching engine 906.

Object classifier 902 classifies things. User classifier 904 classes people. Matching engine 906 matches things with other things, things with people, and/or people with other people.

In more detail, object classifier 902 receives information about objects and uses that information to classify those objects into groups based on the qualities or characteristics of the objects. For example, the object classifier 902 may classify objects of the type described in in "Ginter et al". Such objects may comprise information and/or associated rules for using the information. For example, object classifier 902 may receive as inputs:

- rights management information 909 such as rules and/or associated consequences;
- things 908 controlled or affected by such rights management information including, for example content objects or other information subject to such rules;
- items 910 such as metadata, abstracts or the like that describe the things 908; and/or

- other information of any type.

Object classifier 902 classifies and/or selects things based at least in part on these inputs.

In this example, user classifier 904 is a type of object classifier that is specially adapted to classify people. User classifier 904 can classify people based, for example, on:

- audit trails 912 indicating how people have used their computers and other electronic appliances;
- profiles 914 developed by asking users questions about their preferences;
- controls 909' that are associated, at least in part, with the user or things the user uses;
- object descriptors 910' that describe objects used by the user; and/or
- other information about and/or relating to the user.

User classifier 904 classifies and/or selects people based at least in part on these inputs.

Matching engine 906 receives, as inputs, the classifications and/or selections made by the object classifier 902 and/or the user classifier 904. Matching engine 906 matches things with things, things with people and/or people with people (or any combination of these) based on these selection and/or classification inputs.

Example More Detailed Architecture

Figure 14 shows a more detailed architectural diagram of matching and classification utility 900. In this example, matching and classification utility 900 receives a variety of inputs including, for example, some or all of the following:

- objects 908 and/or information about objects including controls 909 and/or object descriptors 910;
- content 950;
- audit trail information 916;
- user information such as profiles 914;
- class information 952;
- user information 954;
- other rights management information 956;
- matching criteria 958;
- selection criteria 960; and/or
- other information.

Matching and classification utility 900 in this example can provide a variety of different outputs including, for example, some or all of the following:

- matching information 920;
- class hierarchies 962;
- category definitions 922 and class definitions 970;
- classified objects 908C;
- audit records 964 indicating the results of classification, matching, and or selecting processes;

- reports 966 indicating the results of classification, matching, and/or selecting processes;
- targeted objects and/or pointers 968;
- controls 909;
- other rights management information; and
- other classification, matching and/or selection related information.

A Preferred Embodiment Matching and Classification Utility 900 is a VDE-Aware Commerce Utility System

In the preferred embodiment, matching and classification utility 900 is constructed as a commerce utility system 90 as described in “Shear et al”, and may comprise one or more processes securely distributed over one or more secure electronic appliances within a "Virtual Distribution Environment" as described in “Ginter et al”. Furthermore, the present inventions can be used in combination with and/or make use of a wide array of distributed electronic administrative and support services that may be referred to as the "Distributed Commerce Utility." Such a Distributed Commerce Utility may be, among other things, an integrated, modular array of administrative and support services for electronic commerce and electronic rights and transaction management. The Distributed Commerce Utility provides, among other advantages, comprehensive, integrated administrative and support services for secure electronic commerce and other forms of electronic interaction. These administrative and support services can be used to supply a secure foundation for conducting financial management, rights management, certificate authority, rules clearing, usage clearing, secure directory services, and other transaction related capabilities functioning over a vast electronic network such as the Internet and/or over organization internal Intranets, or even in-home networks of electronic appliances. Such electronic interactions supported by the Distributed Commerce Utility may, for example, entail the broadest range of appliances and distribution media, non-limiting examples of which include networks and other communications channels, consumer appliances, computers, convergent devices such as WebTV, and optical media such as CD-ROM and DVD in all their current and future forms.

These administrative and support services can, for example, be adapted to the specific needs of electronic commerce value chains in any number of vertical markets, including a wide variety of entertainment applications. Electronic commerce participants can, for example, use these administrative and support services to support their interests, and/or they can shape and reuse these services in response to competitive business realities. Non-exhaustive examples of electronic commerce participants include individual creators, film and music studios, distributors, program aggregators, broadcasters, and cable and satellite operators.

The Distributed Commerce Utility can, for example, make optimally efficient use of commerce administration resources, and can, in at least some embodiments, scale in a practical fashion to optimally accommodate the demands of electronic commerce growth. The Distributed Commerce Utility may, for example, comprise a number of Commerce Utility Systems. These Commerce Utility Systems can provide a web of infrastructure support available to, and reusable by, the entire electronic community and/or many or all of its participants. Different support functions can, for example, be collected together in hierarchical and/or in networked relationships to suit various business models and/or other objectives. Modular support functions can, for example, be combined in different arrays to form different Commerce Utility Systems for different design implementations and purposes. These Commerce Utility Systems can, for example, be distributed across a large number of electronic appliances with varying degrees of distribution.

Such a "Distributed Commerce Utility" provides numerous additional capabilities and benefits that can be used in conjunction with the particular embodiments shown in the drawings of this application, non-exhaustive examples of which include:

- Enables practical and efficient electronic commerce and rights management.

- Provides services that securely administer and support electronic interactions and consequences.

· Provides infrastructure for electronic commerce and other forms of human electronic interaction and relationships.

· Optimally applies the efficiencies of modern distributed computing and networking.

· Provides electronic automation and distributed processing.

· Supports electronic commerce and communications infrastructure that is modular, programmable, distributed and optimally computerized.

· Provides a comprehensive array of capabilities that can be combined to support services that perform various administrative and support roles.

· Maximizes benefits from electronic automation and distributed processing to produce optimal allocation and use of resources across a system or network.

· Is efficient, flexible, cost effective, configurable, reusable, modifiable, and generalizable.

· Can economically reflect users' business and privacy requirements.

· Can optimally distribute processes -- allowing commerce models to be flexible, scaled to demand and to match user requirements.

· Can efficiently handle a full range of activities and service volumes.

· Can be fashioned and operated for each business model, as a mixture of distributed and centralized processes.

· Provides a blend of local, centralized and networked capabilities that can be uniquely shaped and reshaped to meet changing conditions.

· Supports general purpose resources and is reusable for many different models; in place infrastructure can be reused by different value chains having different requirements.

· Can support any number of commerce and communications models.

· Efficiently applies local, centralized and networked resources to match each value chain's requirements.

· Sharing of common resources spreads out costs and maximizes efficiency.

· Supports mixed, distributed, peer-to-peer and centralized networked capabilities.

· Can operate locally, remotely and/or centrally.

· Can operate synchronously, asynchronously, or support both modes of operation.

· Adapts easily and flexibly to the rapidly changing sea of commercial opportunities, relationships and constraints of "Cyberspace."

Any or all of these features may be used in combination with the inventions disclosed herein.

In more detail, as shown in Figure 14A, matching and classification utility 900 may include one or more rights operating system layers 90-1; one or more commerce utility support service layers 90-4; one or more service application connect layers 90-3; and one or more service functions 90-B. One or more protected processing environments 154 may be used to support secure functions 90-D. Matching and classification utility 900 may be controlled, at least in part, by rights management information such as for example:

- VDE-compatible controls 909;
- rules and/or their consequences; and/or
- other rights management information.

Matching and Classification Utility Can Interact With Other Commerce Utility Systems

Figure 15 shows that matching and classification utility 900 can interact and interrelate with other commerce utility systems described in "Shear et al" including for example:

- financial clearinghouses 200,
- usage clearinghouses 300,
- rights and permissions clearinghouses 400,

- certifying authorities 500,
- secure directory services 600,
- transaction authorities 700,
- VDE administrators 800, and/or
- other commerce utility systems 90.

Figures 15A-15G show example detailed interactions between matching and classification utility 900 and these various other commerce utility systems 90.

Figure 15A shows interactions between matching and classification utility 900 and a financial clearinghouse 200. For example, matching and classification utility 900 may send the financial clearinghouse 200:

- requests for information,
- class information such as classes and/or class assignments,
- bills and charges, and/or
- other information.

Financial clearinghouse 200 may send matching and classification utility 900:

- money,
- audit records,
- payment data,
- user data, and/or
- other information.

Figure 15B shows example interactions between matching and classification utility 900 and usage clearinghouse 300. Matching and classification utility 900 may send the usage clearinghouse 300:

- requests for information,
- class information such as classes and/o class assignments,
- audit information, and/or
- other information.

Matching and classification utility 900 may receive from usage clearinghouse 300:

- requests for class information,
- usage and/or rights management information,
- audit records, and/or
- other information.

Figure 15C shows example interaction between matching and classification utility 900 and rights and permissions clearinghouse 400. In this example, rights and permissions clearinghouse 400 sends matching and classification authority 900:

- controls sets and/or object information;
- requests for class information;
- clearinghouse usage data; and/or
- other information.

In this example, matching and classification utility 900 sends the rights and permissions clearinghouse 400:

- rights management information such as control sets,
- requests for information,
- class related information such as classes and/or class assignments, and/or
- other information.

Figure 15D shows example interaction between matching and classification utility 900 and certifying authority 500. In this example, certifying authority 500 sends matching and classification utility 900:

- revocation lists,
- certificates,
- certifying authority usage information,
- requests for classification information, and/or
- other information.

In this example, the matching and classification utility 900 sends the certifying authority 500:

- revocation list checks,
- requests for certificates,
- requests for usage information,
- classification related information such as classes and/or class assignments, and/or
- other information.

Figure 15E shows an example interaction between the matching and classification utility 900 and a secure directory services 600. In this example, the matching and classification utility 900 sends the secure directory services 600:

- directory lookup information,
- class related information such as classes and/or class assignments,
- requests for information, and/or
- other information.

In this example, the secure directory services 600 sends the matching and classification utility 900:

- directory services usage information,
- directory information,
- requests for classification information, and/or
- other information.

Figure 15F shows an example interaction between the matching and classification utility 900 and a transaction authority 700. In this example, the matching and classification utility 900 sends the transaction authority 700:

- class related information such as classes and/or class assignments,
- requests for transaction usage information,
- requests for control sets, and/or
- other information.

In this example, the transaction authority 700 sends the matching and classification utility 900:

- transaction usage information,
- transaction control sets,
- requests for classification information, and/or
- other information.

Figure 15G shows an example interaction between the matching and classification utility 900 and a VDE administrator 800. In this example, the matching and classification utility 900 sends the VDE administrator 800:

- requests for administration,
- class related information such as classes and/or class assignments,
- requests for node and/or web information, and/or
- other information.

In this example, the VDE administrator 600 sends the matching and classification utility 900:

- requests for classification information,
- administrative information,
- node and/or user data, and/or
- other information.

Matching and Classification Utility System Can Be In a Hierarchy of Commerce Utility Systems

Figure 16A shows an example of an administrative and support service hierarchy including matching and classification utility system(s) 900. In this example, a number of centralized overall matching and classification utility systems 900 and/or other Commerce Utility Systems 90 delegate some or all of their work responsibilities to other Commerce Utility Systems 90. In the particular example shown, Commerce Utility Systems 154 may provide services to one or more members of one or more classes, for example, to members of the class "manufacturing companies in the Pacific rim." Organizations, such as companies, non-profit groups or the like may have their own Commerce Utility Systems 156. Certain electronic commerce or other activities (the entertainment industry, for example) might have their own vertically-specialized Commerce Utility Systems 158. Certain geographical, territorial or jurisdictional groups (e.g., Commerce Utility Systems services provided with a particular nation or state within nation, one example of which might be all purchasers of particular products within the state of Wisconsin) may have their own territorial/jurisdictional specialized Commerce Utility Systems 160. Commerce Utility Systems 154, 156, 158, 160 lower in the hierarchy may, in turn, further delegate authorities or responsibilities to particular consumers, organizations or other entities.

In one example arrangement, the Commerce Utility Systems 90 to which authority has been delegated may perform substantially all of the actual support work, but may keep the delegating Commerce Utility Systems 90 informed through reporting or other means. In another arrangement, the delegating Commerce Utility Systems 90 have no involvement whatsoever with day to day activities of the

Commerce Utility Systems to whom they have delegated work. In still another example arrangement, the more specialized Commerce Utility Systems do some of the work and the more overarching Commerce Utility Systems do other parts of the work. The particular division of work and authority used in a particular scenario may largely depend on factors such as efficiency, trustedness, resource availability, the kinds of transactions being managed, and a variety of other factors. Delegation of clearing authority may be partial (e.g., delegate usage aggregation but not financial or rights management responsibilities), and may be consistent with peer-to-peer processing (e.g., by placing some functions within consumers' electronic appliances while keeping some other functions centralized).

Matching and Classification Utilities Can Provide Services to Classes of Nodes, Users, Content Services and/or Transaction Services

Figure 16B shows an example of how Matching and Classification Utilities 900 can provide services to classes of nodes, users, content services and/or transaction services. In this example, matching and classification utility systems 900(1), ... 900(N) provide horizontally specialized matching and/or classification services for different purposes. For example, matching and classification utility 900(1) serves VDE administrative type functions by classifying VDE deployment related information and associated objects. Matching and classification utility 900(2) specializes in higher education classification tasks. Matching and classification utility 900(3) specializes in business information related tasks, and matching and classification authority 900(N) specializes in trading transactions. Any of these specialties can be combined together, so that a single utility system 900 can perform multiple functions or portions of functions.

Multi-Function Commerce Utility Systems Can be Organized Hierarchically or Peer-to-Peer

Figure 16C shows a still different, more complex Matching and Classification Commerce Utility System 900 environment including elements of both a hierarchical chain of command and a high degree of cooperation in the horizontal direction between different multi-function matching and classification utility systems 900. In this example, there are five different levels of responsibility with a master or overarching matching and classification utility system 900(1) on level 1 having the most authority and with additional matching and classification utility systems on levels 2, 3, 4, and 5 having successively less power, authority, control, scope and/or responsibility. Figure 16C also shows that different matching and classification utility systems 900 on the same level may have different functions, scopes and/or areas of responsibility. For example:

- a Matching and classification utility system 900(2)(1) may be a "type A" Matching and classification utility system,
- Matching and classification utility system 900(2)(2) might be a "type B" Matching and classification utility system, and

- Matching and classification utility system 900(2)(3) might be a "type C" Matching and classification utility system.

On the next level down, Matching and classification utility systems might be type A Matching and classification utility system (such as, 900(3)(1) and 900(3)(2)), they might be type B Matching and classification utility systems (such as, 900(3)(4)), they might be type C Matching and classification utility systems (such as, 900(3)(5), 900(3)(6)), or they might be hybrids -- such as, Matching and classification utility system 900(3)(3) which is a hybrid having type A and type B functions. Figure 16C also shows that additional clearinghouses on levels 4 and 5 might have sub-types as well as types.

A matching and classification utility 900 might break out along content classes (e.g., movies; scientific, technical and medical; and software). Subtype A might include first run movies, oldies, and art films; subtype B might handle journals and textbooks; and type C might be responsible for games, office, educational content. Peer-to-peer communications between clearinghouses could involve differing classes of consumers, differing jurisdictional classes, differing payment methods classes, and/or any other class distinction.

Matching and Classification Utility System Can Be Constructed From Object-Oriented Service Functions

Figure 14A shows Matching and Classification Utility 900 can be constructed from service functions. Figure 17 shows in more detail how a matching and classification utility system 900 can be constructed based on service functions such as for example:

automatic class generation,

automatic matching,

automatic class assignment,

class based searching,

class based directory,

audit by class,

market research,

rights management language processing,

other service functions.

Example Detailed Steps Carried Out By Matching and Classification Utility System 900

The next section of the specification describes some example steps performed by the matching and classification utility 900.

Example Steps to Categorize Objects and/or Users and/or Appliances

Figure 18 shows example steps to categorize objects, and Figure 19 shows example steps to categorize users 95 and/or appliances 100. The overall categorization steps in these examples are -- at this level -- similar to one another. The processes begin by getting input data (Figure 18, block 1840, Figure 19, block 1840′). Next, a classification and/or categorization method is selected (Figures 18, block 1842; Figure 19, block 1842′). The process then assembles a data matrix and applies the selected classification method to the data matrix (Figure 18, blocks 1844, 1846; Figure 19, blocks 1844′, 1846′). In addition or alternatively, other data reduction methods may be used (Figure 18, block 1848; Figure 19, block 1848′). Next, the process assigns objects and/or users and/or appliances to the categories developed by the classification method that has been applied (Figure 18, block 1849; Figure 19, block 1849′). Finally, the process stores the results in electronic and/or non-electronic storage in the "write output data" step (Figure 18, block 1850; Figure 19, block 1850′).

The "get input data" step 1840, 1840′ may involve obtaining attribute and/or parameter data from various sources including, for example:

- electronic appliance related attribute data;
- user demographic data;
- user psychographic data;
- available rights management rules and/or consequences (e.g., permissions records);

- exercised rights management rules and/or consequences (e.g., permissions records);
- rights management and/or other audit and/or usage records;
- any third party source of any information, including rights management, usage, audit, statistical, personal, organizational, political, economic, social, religious, business, government, medical, research, academic, literary, military, and/or information and/or data in any format known or unknown concerning any and all other topics that may contribute to the definition of at least one class and/or the assignment of at least one object to a class.

Detailed example steps for harvesting this data are detailed below in connection with Figures 24-46B. This resulting attribute data may be accumulated and aggregated together to form a composite record used as the input to the classification process.

Figure 20 shows an example composite record 1852. This composite classification record may contain attributes derived from any or all of a variety of rights management and/or other data "harvesting" processes. For example, composite record 1852 may include demographic and/or psychographic data obtained by querying the user 95. It may contain usage data obtained by monitoring audit information produced by various usage transactions. It may contain information reflecting user choices concerning rights management information, the rights management information available to particular users and/or objects, and rights management processes actually performed with respect to particular users and/or particular objects. The information may be analyzed first to provide statistical and/or other summary information, or individual, more granular information may be provided. The composite record 1852 may also contain attributes of particular electronic appliance 100 installations. The particular example composite record 1852 shown in Figure 20 is one non-limiting example composite attribute record containing attributes obtained through a number of different "harvesting" processes. The composite record 1852 may be organized in a way to allow easy and efficient selection of desired attributes in the course of a database lookup, for example, and to allow easy and efficient selection and/or coding as input to any aspect of a classification and/or the assignment of one or more objects to at least one or more classes.

The Figure 21 example cluster analysis process is one example of steps that may be performed as part of the "apply classification method(s)" block 1846, 1846′ of Figures 18, 19. (A classification method, or any other method described in these processes, may be utilized as part of a "knowbot", "agent", "traveling agent", and/or "smart agent", a non-limiting example of which is described in “Ginter et al”, for example, Figure 73.) In this particular example, the process selects variables and cases (blocks 1860, 1862, Figure 21), and then assembles an appropriate data matrix (block 1864). A conventional cluster analysis is then applied (block 1866, Figure 21). The clusters may be interpreted to determine what they mean (Figure 21, block 1868), or they may be compared with previous results and if sufficiently similar, they may be assumed to reflect the same classes as the earlier classification procedure thus minimizing the need for additional interpretation of the clustering results. Step 1868 may be performed automatically or manually, or a combination of automatic and manual processing may be used. Finally, individual cases may be assigned to individual clusters to complete the classification process (Figure 21, block 1870).

Figures 22, 23 show two examples of classification outputs produced by the Figure 21 process. In the Figure 22 example, information from several individuals has been used to create two example categories that reflect differing use profiles. More classes may have been defined than those example classes shown here. Users assigned to the same class have many more features, behavior, and/or other attributes in common than each of them does with members assigned to other classes.

In example Figure 22, members of class 1 tend to spend more per content item purchased, travel abroad more frequently, are more interested in national and international news, business and travel information, and generally do not participate in "pay per view" events and/or content consumption. Members of class 1 also tend to add new rights and/or modify existing rights management controls for content, for instance, to add a markup and redistribute the content in one example, may be less likely to express a religious preference and/or affiliation, and tend to use the Internet as an area for "surfing" and exploration.

Members of class 2 tend to pay less for content purchased, seldom travel abroad, tend to be interested in sports, religious content and events, and are more often consumers of movies than are members of class 1. Members of class 2 are more likely to "pay per view" than are members of class 1, and are much less likely to add new controls to content and/or modify rights acquired. Members of class 2 are more likely to express a religious preference and among those that do, Protestant denominations are more frequently mentioned. Members of class 2 may use the Internet, but tend to do so as part of their work role and responsibilities rather than as entertainment, hobbies, and other leisure-time pursuits.

Some methods of classification produce parameter data rather than assignment of objects to more discrete (or fuzzy or other kinds of) classes. Instead, this parameter data may indicate the extent to which an object possesses one or more traits, attributes, or class characteristics. For instance, a person may have been assigned to class 1 (call it "the cosmopolitan class") or class 2 (call it "the parochial class") as shown in Figure 22; however, using other procedures the same example persons may be assigned parameter data reflecting the extent or degree to which they are "cosmopolitan" or "parochial" or some of each.

In the example process that generates the information shown in Figure 23A, data for several individuals has been arranged in a case (row) by variable (column) matrix and using means known to those skilled in the arts, subjected to principal components analysis with subsequent Varimax axis rotation. Components with eigenvalues >1.0 were retained for subsequent rotation and use. After rotation, each case was assigned a score on each retained (and rotated) component. Each score indicates the extent to which the case has the characteristic represented by the component.

The hypothetical data in Figure 23A shows how strongly each variable (the column of the input matrix) is correlated with the underlying characteristic or component. For example, "region of the US" and "Family income" are highly correlated while "owns a sports utility vehicle" is not.

Using results such as these plus the input data matrix, a score is assigned to each case indicating the extent to which they posses the trait, attribute, characteristic indicated by each factor or component. The hypothetical data in Figure 23B shows how strongly each case -- a person or thing – is a member of the class, and/or possesses the underlying variable represented by each component. A higher score shows that example case 1 has more of the underlying component 1 than does example case 3, whose score is close to zero. Components (factors) may be bipolar with a zero point and cases whose scores may be positive, negative or zero. Hypothetical example case 5 has a negative score on this component.

This component score information may be used by the matching and classification utility 900 to define certain other classes, such as "the class consisting of the top 5% of those who are cosmopolitan," that is, the 5% with the highest scores on example component 1. The original scores and/or derivative class assignments may be included on attribute records with attribute and/or class information harvested from other sources and/or through other processes.

DATA HARVESTING

Example Steps For Collecting Appliance Related Data

Figure 24 shows example steps performed by the matching and classification utility 900 to collect appliance attribute data. In this example, an electronic appliance 100 may have certain information associated with it. For example, a VDE administrator 800 may initialize appliance 100 with certain information upon appliance installation. In this example, the matching and classification utility 900 can collect this appliance attribute data and use it as part of a matching and/or classification and/or selection process. As shown in Figure 24, the matching and classification utility 900 may initially specify desired appliance attribute fields or other information characteristics the utility is going to collect (Figure 24, block 1502).

The information to be collected depends upon the purpose and use to which the matching and classification utility 900 is to put the information to. The matching and classification utility 900 may use a data dictionary or other mechanism for specifying the desired types of appliance information it is going to collect.

The matching and classification utility 900 next determines whether it already possesses the desired information for this particular appliance 100 (Figure 24, block 1504). For example, the information may have been previously gathered as part of a prior process. If the information is already available, the matching and classification utility 900 sends one or more events to a "create appliance attribute record" method to process the previously gathered data (Figure 24, block 1506). (In all these processes, if the appropriate method is has been sent previously to a VDE installation, only the associated administrative events necessary to activate the method need to be sent in the VDE container.) Alternatively, if the desired data is not already available ("no" exit to decision block 1504, Figure 24), the matching and classification utility 900 performs the other steps shown in Figure 24 to collect the appliance attribute data.

These collecting steps shown in Figure 24 may include sending a VDE container 152 with a "create appliance attribute record" method, and one or more associated administrative events to activate the method, to the VDE administrator 800 (Figure 24, block 1508). The next step (Figure 24, block 1510) may be performed by the VDE administrator 800 processing the administrative event(s) using the "create appliance attribute record" method to determine whether the administrator already has the desired information for the particular electronic appliance 100. If the operation is successful ("yes" exit to decision block 1512, Figure 24), the VDE administrator 800 may send, to the matching and classification utility 900, a VDE container 152 containing one or more administrative events and the appliance attribute record (Figure 24, block 1514). If the operation is not successful ("no" exit to decision block 1512, Figure 24), the "create appliance attribute record" method operating at VDE administrator 800 may, in this example, collect the data directly from the electronic appliance 100 by sending a VDE container to the appliance, the container containing a "create appliance attribute record" method and one or more associated administrative events (Figure 24, block 1516). The appliance 100 may itself process the administrative event(s) using the "create appliance attribute record" method (Figure 24, block 1518) to produce the required appliance attribute record. Appliance 100 may then send a VDE container 152 containing the appropriate administrative event(s) and the appliance attribute record to the matching and classification utility 900 (Figure 24, block 1520).

In another example, blocks 1508-1514 may be bypassed entirely, and the matching and classification utility 900 may (assuming appropriate authorizations are in place) perform block 1516 to send a container 152 with one or more administrative events and the "create appliance attribute record" method directly to the electronic appliance 100.

Figures 25(A) and 25(B) together show example steps performed by the "create appliance attribute data" method shown in Figure 24, blocks 1506, 1510 and 1518. As disclosed in “Ginter et al”, the actual processing steps are performed by one or more load modules associated with the method. This example method (which, as explained above, may be performed by the matching and classification utility 900, the VDE administrator 800, the electronic appliance 100, any other electronic appliance, or a combination of any or all of these) first locates the site configuration record(s) corresponding to the electronic appliance for which appliance attribute data is to be collected (Figure 24A, block 1522). This site configuration record(s) may, for example, be stored in the electronic appliance secure database. The method next locates the permissions record for the site configuration record(s) (Figure 24A, block 1523). The SPE next determines, based upon the permission record(s), whether the method has permission to access and/or use the site configuration record(s) (Figure 25A, block 1524). If the method does not have the appropriate permission ("no" exit to decision block 1524, Figure 25A), the protected processing environment 154 reports the failure and reason for the failure, and the method writes an associated audit record (Figure 25A, block 1525, 1526) and goes on to process a user configuration record(s). On the other hand, if the method does have permission to use the site configuration record(s) ("yes" exit to decision block 1524, Figure 25A), the method copies the required fields from the site configuration record(s) to create an appliance attribute record, and may then write an appropriate audit record (Figure 25A, block 1527).

After completing processing of site configuration records, the method then locates the user configuration record(s) corresponding to the electronic appliance for which appliance attribute data is to be collected (Figure 25B, block 1528). This user configuration record(s) may, for example, be stored in the electronic appliance secure database. The protected processing environment 154 next locates the permissions record for the user configuration record(s) (Figure 25B, block 1529). The protected processing environment 154 determines next, based upon the permission record(s), whether it has permission to access and/or use the user configuration record(s) (Figure 25B, block 1530). If the method does not have the appropriate permission ("no" exit to decision block 1530, Figure 25B), the protected processing environment 154 reports the failure and reason for the failure, and the method writes an associated audit record (Figure 25B, block 1531, 1532) and exits the process. On the other hand, if the method does have permission to use the user configuration record(s) ("yes" exit to decision block 1530, Figure 25B), the method copies the required fields from the user configuration record(s) to create an appliance attribute record, and may then write an appropriate audit record (Figure 25B, block 1533). The method may then, if desired, create a new permissions record corresponding to the appliance attribute record (Figure 25B, block 1534). If a new permissions record is desired, the method may include appropriate "shared secrets," expiration interval(s), and/or other data in an associated MDE to, for example, provide a basis for controlling access, use, and modification of the permissions record.

Figures 26A-26C show examples of appliance attribute records created by Figure 25B, block 1532. Figure 26A shows an example appliance attribute record that may include, for example, an appliance identification field 1536(1) and any number of attribute fields 1538(1)...1538(n). Figure 26B shows a more specific appliance attribute record example including an appliance ID field 1536(1), an operating system field 1538(A), a country field 1538(B), a state field 1538(C), a VDE administrator organization field 1538(D), a VDE version field 1538(E), and a VDE maintenance level field 1538(F). Figure 26C shows that different encodings may be used for any/all of the various attribute fields 1538.

Example Steps for Collecting Demographic Data

Figures 27A, 27B show example steps for collecting demographic data. In this example, the matching and classification utility 900 initially specifies demographic data fields it is interested in (Figure 27A, block 1540). The matching and classification utility 900 next determines whether the required data is already available to it (e.g., based on previous inquiries responded to by the user 95) (block 1542, Figure 27A). If the required data is already available ("yes" exit to decision block 1542, Figure 27A), the matching and classification utility 900 may send one or more events to a "create demographic attribute record" method to process the data (block 1544, Figure 27A).

On the other hand, if the required data is not available to the matching and classification utility ("no" exit to decision block 1542, Figure 27A), the matching and classification utility may send a container 152 to another commerce utility system 90, the container including one or more administrative events associated with a "demographic data query" method and a "create demographic attribute record" method (Figure 27A, block 1546). The other commerce utility system 90 may then process the one or more events using the "demographic data query" method, and write an associated audit record (Figure 27A, block 1548). It may determine whether the required demographic data is available (Figure 27A, block 1550). If the information is available ("yes" exit to decision block 1550, Figure 27A), the commerce utility system 90 may process one or more events using a "create demographic attribute record" method in order to analyze the available demographic data, and write a corresponding UDE audit record (Figure 27A, block 1552). The other commerce utility system 90 may then send appropriate one or more administrative events and the demographic data attribute record within a container 152 to the matching and classification utility 900 (Figure 27A, block 1554)).

If the required demographic data is not available ("no" exit to decision block 1550, Figure 27A), the commerce utility system 90 may send an administrative event to the matching and classification utility system 900 within a container 152 informing the matching and classification utility that the required data is not available (Figure 27B, block 1556). The matching and classification utility 900 may then send a "demographic data query" method and a "create demographic attribute record" method within a container 152 (along with appropriate administrative events to activate such methods) directly to the user 95 about which demographic information is to be collected (Figure 27B, block 1558). The user's electronic appliance 100 may, in response, process the one or more events using the "demographic data query" method, which may write an associated audit record (Figure 27B, block 1560). If the required data is not collected ("no" exit to decision block 1562, Figure 27B, the user's appliance 100 may send a "failure" message associated with the appropriate administrative event to the matching and classification utility 900, and write an associated audit record (Figure 27B, block 1564, 1566). If the required demographic data is successfully collected ("yes" exit to decision block 1562, Figure 27B), the user's electronic appliance may process one or more events using the "create demographic record" method supplied by step 1558, which may write an associated audit record (Figure 27B, block 1568). The electronic appliance may then send appropriate administrative events and the demographic attribute record to the matching and classification utility within one or more containers 152 (Figure 27B, block 1570).

Figure 28 shows an example questionnaire "pop-up" screen that may be displayed by the user's appliance 100 as a result of processing events using the "demographic data query" method of block 1548, Figure 27A, and/or block 1560, Figure 27B. In this example, information is collected directly from a user 95 by displaying a questionnaire on a display device that is part of the user's appliance 100. The questionnaire may ask for various demographic information such as:

- name
- address
- city
- state
- zip code
- gender
- date of birth
- education level
- marital status
- number of children

- age of first child
- gender of first child
- other information

The user is requested to provide the information by filling in the various fields within the questionnaire. The questionnaire may assure the user that all information the user provides will be treated as confidential, by, for example, disclosing the rules that will be associated with access to and use of the information.

Steps similar to those shown in Figure 25A, 25B may be performed to create a demographic attribute record based on the results of a demographic data query. Figure 29A-29C show examples of different user demographic attribute information records resulting from this process. Figure 29A shows an example demographic attribute record 1572 including a user ID field 1574 and any number of attribute fields 1576(1), ... 1576(n). Figure 29B shows a more specific example of a demographic attribute record including, for example, a user ID number 1574, a gender attribute field 1576(A), an age field 1576(B), a highest educational level field 1576(C), a citizenship field 1576(D), a country of residence field 1576(E), a district field 1576(F), a city field 1576(G), and a street address field 1576(H). Figure 29C shows a different detailed encoding example for demographic attribute record 1572-1.

Example Steps for Collecting Psychographic Data

Figure 20 shows example steps that may be performed to collect user psychographic data. In this particular example, the matching and classification utility 900 initially specifies desired psychographic data it requires in order to perform a particular classification/matching process (Figure 30, block 1580). The matching and classification utility 900 determines if the required data is already available to it (Figure 30, block 1582). If the required data is already available ("yes" exit to decision block 1582, Figure 30), the matching and classification utility 900 sends one or more events to a "create psychographic attribute record" method in order to analyze the available data and provide appropriate psychographic attributes (Figure 30, block 1584). If, on the other hand, the required data is not available to the matching and classification utility 900 ("no" exit to decision block 1582, Figure 30), appropriate steps are performed to collect the required data. In this example, the matching and classification utility 900 may send a "psychographic data query" method and a "create psychographic attribute record" method within one or more containers 152 (along with appropriate administrative events to activate such methods), to appropriate repositories that may contain the required data (Figure 30, block 1586). If the required data is available from the repositories ("yes" exit to decision block 1588, Figure 30), then an electronic appliance at the repository (in this example) processes one or more events using the "create psychographic attribute record" method supplied by block 1586 in order to generate an appropriate attribute record(s) containing the attribute information the matching and classification utility 900 is interested in (Figure 30, block 1590). This information, and associated one or more events, may be sent to the matching and classification utility 900 within one or more containers 152 (Figure 30, block 1592).

If the required data is not available from the repository ("no" exit to decision block 1588, Figure 30), then the repository may send a "failure" message associated with one or more administrative events to the matching and classification utility 900 within a container 152 (Figure 30, block 1594). The matching and classification utility 900 may, in response, send one or more administrative events, a "collect psychographic data" and "create psychographic attribute record" method directly to the user's electronic appliance 100 within one or more containers 152 (Figure 30, block 1596). The user's electronic appliance 100 may, in turn, process the events using the "collect psychographic data" and "create psychographic attribute record" methods (Figure 30, block 1598, 1600), and send the resulting attribute data record(s) to the matching and classification utility (Figure 30, block 1592).

Figure 31 shows an example psychographic questionnaire "pop-up" screen that may be displayed to the user 95 upon performance of Figure 30, block 1598. This questionnaire may collect various psychographic information from the user, including for example:

- mood information
- emotion information
- habit information
- behavioral information
- cognitive information
- medical information
- physical information
- patient information
- counseling information
- aptitude information
- testing information
- other information
- combinations of types of information.

The questionnaire may inform the user that all information collected will be treated as "confidential," and may also, if desired, indicate that the user will be compensated for providing the information.

Figures 32A-32C show some example user psychographic attribute information records 1602 that may be created by Figure 30, block 1584, 1590 and/or 1600. Figure 32A shows that a psychographic attribute record 1602 may include a user ID field 1604 and any number of attribute fields 1606(1), ... 1606(n). Figure 32B shows a more detailed user psychographic attribute record 1602 example including a user ID field 1604, a field 1606a indicating whether the user is introverted or extroverted, a field 1606b indicating whether the user is a sensing or intuitive person, a field 1606c indicating whether the user is primarily a thinking person or a feeling person, a field 1606(d) indicating whether the user is primarily a judging person or a perceiving person, and a field 1606(e) indicating an overall psychographic / behavioral profile such as, for example, the iVALS standard provided by SRI. Figure 32C shows a different kind of encoding (in this case, binary) for the various attributes 1606.

Example Method for Determining Attributes Based on Available Rules and Consequences

Figure 33 shows an example method for determining attributes based on available rules and consequences. The matching and classification utility 900 may first send one or more administrative events and a "send permission records" method request to an electronic appliance 100 within one or more containers 152 (Figure 33, block 1610). In response, the appliance may process the events using the method, which may write an associated audit record (Figure 33, block 1612). If this step is performed successfully ("yes" exit to

Figure 33, decision block 1614), the appliance sends appropriate administrative events and the requested permission records to the matching and classification utility 900 within one or more containers 152, and the method writes an associated audit record indicating it has performed this transaction (Figure 33, block 1616). The matching and classification utility may process events using a corresponding "create attribute record from permission records" method to obtain attributes from these provided permission records (Figure 33, block 1618). If the step of block 1612 failed (as indicated by the "no" exit to decision block 1614, Figure 33), the method may send a "failure" message to the matching and classification utility 900, and write an associated audit record (Figure 33, block 1620).

Figure 34 shows a variation on the Figure 33 example in which the appliance 100 rather than the matching and classification utility 900 creates the rules attribute record based on a "create rules attribute record from permissions records" method supplied by the matching and classification utility, and then sends the rules attribute record to the matching and classification utility (see Figure 34, blocks 1622, 1624).

Example Method to Construct Attribute Records from Permissions Records

Figures 35A, 35B show example steps for constructing attribute records from permissions records. The steps shown in Figure 35A, 35B may, for example, be performed as part of the method shown in block 1618 of Figure 33.

In this example method 1618, the matching and classification utility 900 may first check relevant permissions to ensure that it has the authority to perform the desired transactions (Figure 35A, block 1630). For example, the matching and classification utility 900 may examine a permissions record about the permissions records it has collected, this permissions record it is examining indicating what entities have authority to perform operations with respect to the permissions record to be analyzed. Presuming the matching and classification utility 900 has the appropriate permission, it opens a permissions to be analyzed (Figure 35A, block 1632), and performs a sequence of steps 1634-1650 to extract relevant information from the permissions record. For example, information from the permissions record header can be copied into the attribute record (Figure 35A, block 1634), and then the method may locate the rights record header (block 1636, Figure 35A). Information from the rights record header may be copied into the attribute record (block 1638, Figure 35A), along with the identifier for the corresponding right(s) (blocks 1640, 1642, Figure 35A). The process may then recursively locate and harvest data from each method header contained within the rights record (blocks 1644, 1646, 1648, Figure 35B). The process may recursively repeat steps 1638-1648 for each rights record within the permissions record (as tested for by decision block 1650, Figure 35B). Finally, the entire process of steps 1632-1652 may be performed recursively for multiple permissions records to harvest the appropriate rules and consequences information from each of a number of permissions records (see decision block 1652, Figure 35B).

Figure 36 shows example steps to perform the "check permissions" operation shown in Figure 35A, block 1630. In this example, the process locates the permissions record from which information is desired to be harvested (Figure 36, block 1660), and then determines whether there is a permissions record for that permissions record (Figure 36, decision block 1662). If there is no permissions record that controls that permissions record (and assuming that authorization or additional permission is required to access the permissions record from which information is to be harvested) (Figure 36, "no" exit to decision block 1662), the process reports failure, writes an audit record, and ends (Figure 36, blocks 1664, 1666, 1668). On the other hand, if there is a permissions record that controls access to the permissions record from which information is to be harvested ("yes" exit to decision block 1662, Figure 36), the process determines whether that permissions record for the permissions record enables usage by the matching and classification utility 900 (Figure 36, decision block 1670). If the matching and classification utility 900 does not have permission ("no" exit to decision block 1670, Figure 36), the process reports failure, writes an audit record to that effect, and ends (blocks 1672, 1674, 1676, Figure 36)). On the other hand, if the matching and classification utility 900 is granted permission ("yes" exit to decision block 1670, Figure 36), the process accesses and uses the permissions record for the permissions record from which information is to be harvested (Figure 36, block 1678).

Figures 37A-37C show examples of attribute records containing information harvested from permissions records. Attribute record 1680-1 shown in Figure 37A includes a user identification field 1682, an object identification field 1684, and any number of attribute fields 1686(1), ..., 1686(n). The attribute record 1680-2 shown in Figure 37B includes, as a more detailed example, a user ID number field 1682, an object ID field 1684, a right ID field 1686a, a method identifier field 1686b, another right ID field 1686c, and corresponding method type fields 1686(d), a further right ID field 1686e and two corresponding method attribute fields 1686f, 1686g, a further right ID field 1686h and corresponding method attribute fields 1686i, 1686j.

Figure 37C shows a different example in coding for the Figure 37B example attribute record.

Example Steps for Assembling Rules and Consequences

Figure 38 shows example steps for assembling rules and consequences. In this example, the matching and classification utility 900 may send one or more administrative events and a "get user rights table" method within a container 152 to an electronic appliance (Figure 38, block 1690). The electronic appliance 100 processes the one or more events using the "get URT" method, which may writes an associated audit record (Figure 38, block 1692). The method then determines whether the associated URT records are available (Figure 38, decision block 1694). If the records are not available ("no" exit to decision block 1694, Figure 38), the method sends a failure notice to the matching and classification utility 900, and writes an associated audit record (block 1696, Figure 38). If, on the other hand, the URT records are available ("yes" exit to decision block 1694, Figure 38), the method packages the URT records and associated one or more administrative events into a container 152, and sends the container to the matching and classification utility 900 (Figure 38, block 1698). The matching and classification utility 900 may then process the administrative events using a "create attribute record from URT" method in order to extract or harvest the information from the URT(s) (Figure 38, block 1700).

Figure 39 shows another example sequence of steps for assembling rules and consequences. In this example, the matching and classification utility 900 sends one or more administrative events and a "create attribute record from URT" method to the electronic appliance 100 that stores or has access to the user rights table information (Figure 39, block 1702). The appliance then processes the events using the method sent to it, and the method writes associated audit information as it processes (Figure 39, block 1704). If the URT records are available and the step completes successfully ("yes" exit to decision block 1706, Figure 39), the method sends the resulting URT attribute record(s) and one or more administrative events to the matching and classification utility within a container 152, and writes corresponding audit information to an audit trail (Figure 39, block 1710). On the other hand, if an error condition arises either because the URT records are not available or because the method for some other reason cannot complete successfully, the method sends a failure notice within a container 152, and writes an associated audit record ("no" exit to decision block 1706, Figure 39, block 1708).

Figures 40A, 40B show example steps performed by blocks 1700, 1704 to "create attribute record from user rights table." The method begins by checking associated permissions for the user rights table records (Figure 40A, block 1720). Assuming that appropriate user and/or group permission is available, the method next locates the user rights table (Figure 40A, block 1722), and then begins recursively analyzing the user rights table information to harvest desired attribute information from it (Figure 40A, blocks 1724 and following). In this particular example, the method locates the user rights table record (block 1724, Figure 40A, and then locates the first rights record header within the first user choice record within the URT record (blocks 1726, 1728, Figure 40A). The method copies rights record header information to the attribute record (block 1730), and then locates the right identifier and copies that to the attribute record (blocks 1732, 1734). The method then recursively locates each method header within the user rights table right record, and copies corresponding attribute information to the attribute record (blocks 1736, 1738, 1740, Figure 40B). Steps 1728-1740 are performed recursively for each rights record within the user choice record (see Figure 40B), decision block 1742), and the above steps are performed recursively for each user choice record within the user rights table (see decision block 1744, Figure 40B). Additionally, steps 1724-1744 are performed recursively for each user rights table record within the user rights table (see Figure 40B, decision block 1746). As a last example step, the method creates a permissions record that controls access and use of the attribute record it has created (Figure 40B, block 1748).

Figure 41 shows example steps performed by the check permissions block 1720 shown in Figure 40A. For example, the sequence of steps may begin by locating a corresponding permissions record (Figure 41, block 1750) and then determining whether there is a permission record corresponding to the corresponding user rights table entry (Figure 41, decision block 1752). If there is no such entry ("no" exit to decision block 1752), the method may report failure, write an audit record, and end (blocks 1754, 1756, 1758, Figure 41). If there is a corresponding permissions record ("yes" exit to decision block 1752, Figure 41), then the permissions record may be examined whether it enables usage for the matching and classification utility 900 (decision block 1760, Figure 41). If the permissions record does not enable usage by the matching and classification utility 900 ("no" exit to decision block 1760, Figure 41), the method may report a failure, write an audit record, and end (blocks 1762, 1764, 1766, Figure 41). On the other hand, if the matching and classification utility 900 does have the required permissions to enable usage ("yes" exit to decision block 1760, Figure 41), the method may access the permissions record (if any) for the user rights table for use in controlling access to the user rights table itself (block 1768, Figure 41).

Figures 42A-42C show example rights attributes records 1770 that may be obtained from the processes above. The Figure 42A example rights attribute record 1770-1 includes a user or group ID field 1772, an object ID field 1774, and any number of attribute fields 1776(1), ... , 1776(n). The more detailed example rights attribute record 1770-2 shown in Figure 42B includes a user ID number field 1772, an object ID field 1774, a right ID field 1776a and corresponding method attribute field 1776b, another right ID field 1776c and corresponding method attribute field 1776d, a right ID field 1776e and corresponding method attribute fields 1776f, 1776g, and another right ID field 1776h and corresponding method attribute field 1776i.

Figure 42C shows how the rights attribute record 1770 can be encoded numerically as opposed to using characters, as one example.

Example Steps for Assembling Usage Audit Records

Figure 43 shows example steps for assembling usage audit records for purposes of matching and/or classification. In this example, the matching and classification utility 900 may send one or more administrative events and a "get audit records" method to a VDE appliance 100 within a container 152 (Figure 43, block 1780). The appliance 100 may process the one or more events using the "get audit records" method, which may write an associated audit record (block 1782, Figure 43). If the audit records are not available ("no" exit to decision block 1784, Figure 43), the method may send a failure notice within a container to the matching and classification utility 900, and may then write an associated audit record (Figure 43, block 1786). On the other hand, if the audit records are available ("yes" exit to decision block 1784), the method may send one or more administrative events and the audit records within a container 152 to the matching and classification utility 900, and write an associated audit record (block 1788, Figure 43). The matching and classification utility 900 may then process the one or more administrative events using a "create attribute record from audit record" method in order to extract or harvest the information from the audit record it will use to perform matching and/or classification (block 1790, Figure 43).

Figure 44 shows another sequence of example steps that may be used to assemble usage audit records for purposes of matching and/or classification. In the Figure 44 example, the matching and classification utility 900 sends one or more administrative events and a "create attribute record from audit record" method to an electronic appliance 100 within one or more containers 152 (Figure 44, block 1792). The appliance 100 may then process the one or more administrative events using the "create attribute record from audit record" method, which may write an associated audit record (block 1794, Figure 44). The method may determine, in this process, whether audit records are available (Figure 44, decision block 1796). If no audit records are available ("no" exit to decision block 1796), the method may send a failure notice to the matching and classification utility 900 (Figure 44, block 1798). On the other hand, if audit records are available, the method may create the corresponding usage attribute records and associated administrative event(s), package them into a container 152, send the container to the matching and classification utility 900, and write corresponding audit records (Figure 44, block 1799).

Figures 45A, 45B show example steps for performing the method (shown in Figure 44, block 1794, for example) of creating attribute record(s) from audit records. In this example, the method first locates the audit records in a secure database or other storage facility (Figure 45(A), block 1800). The method next selects an appropriate UDE audit record to analyze (Figure 45(A), block 1802), and determines whether a permission record is available that applies to this particular audit record (Figure 45(A), decision block 1804). If a permissions record is required and is not available, the process reports failure, writes an associated audit record, and ends (Figure 45 blocks 1806, 1808, 1810). If, on the other hand, a required permissions record is available ("yes" exit to decision block 1804, Figure 45), the process determines whether the permissions record grants the device or process permission to use the audit record(s) for this particular purpose (decision block 1812, Figure 45). If such permission is not available ("no" exit to decision block 1812, Figure 45A), the process reports failure, writes an associated audit record, and terminates (Figure 45A, blocks 1814, 1816, 1818).

If any applicable permissions record is available and grants permission to the matching and classification utility 900 ("yes" exit to decision block 1812), the process determines multiple audit records need to be analyzed together as an overall event (Figure 45A, decision block 1820). For example, an "atomic transaction" in which multiple steps are performed to achieve an overall result may have multiple audit records (e.g., from multiple appliances 100) that may need to be analyzed together in order to make sense out of the overall transaction. As another example, an object may have subparts (e.g., sub-objects) on which operations can be performed – but it may be important for matching and/or classification purposes to analyze the results of such multiple operations together in order to determine appropriate attribute(s) for matching and/or classification. If it is necessary to aggregate multiple audit records together for analysis (decision blocks 1820, 1822, Figure 45A), then the process proceeds to analyze those audit records together and create corresponding summary transaction information (Figure 45A, block 1824).

The process next determines whether it needs to produce aggregated audit statistics in order to perform the associated matching and/or classification operation (Figures 45A, 45B, decision block

1826). For example, multiple operations may be performed on a certain object. It may be important to know statistics about such operations (e.g., the number of times the object was opened on a certain day, the number of users who opened the object in a certain time period, etc.). If such aggregated statistics are required ("yes" exit to decision block 1826, Figure 45B), the process proceeds to create such aggregated statistics (block 1828, Figure 45B).

The process next copies selected audit record information to an audit attribute record (Figure 45B, block 1830). The process then determines whether it needs to process more audit records (decision block 1832, Figure 45B). If more audit records are required to be processed ("yes" exit to decision block 1832, Figure 45B), control returns to Figure 45A, block 1802 to select the next audit record. Otherwise ("no" exit to decision block 1832, Figure 45B), the process creates a permissions record associated with the newly created attribute record(s) (Figure 45B, block 1834), and completes.

Figures 46A, 46B show example usage attributes/statistic records that the Figure 45A-B process may create. The Figure 46A attribute record 1830-1 may include, for example, a user ID 1832, an object ID 1834, and any number of attribute fields 1836(1), ... , 1836(n). The more detailed Figure 46B example attribute record 1830-2 includes a user ID number 1832, an object ID 1834, a right ID 1836a and associated method characteristic 1836b, another right ID 1836c and associated method 1836d and associated statistic 1836e, a further right ID 1836f and associated method attribute 1836g, another right ID 1836h and associated methods 1836i, 1836j, and associated additional attributes 1836k-1836o. The characteristics shown in fields 1836k-1836o could, for example, be derived from an aggregate of any number of individual audit records recording individual transactions associated with the object identified in field 1834.

EXAMPLES

The following are some non-limiting examples of how Matching and Classification Utility 900 may be useful in certain applications.

Example: Matching and Classification Utility 900 Can Support Narrowcasting or "Push" Distribution Models Based On Classes

Interactions with content, transactions, and other events on the World Wide Web are mainly driven today by following chains of hypertext links, using various search engines, and/or indexes, to say nothing of just plain luck and persistence, to find interesting and/or useful content and/or services. Time consuming and generally inefficient, these search activities share in common the feature that each consumer must intentionally "pull" desired content from a Web site to their computer after successfully identifying specific content or services of interest at that time. The present inventions also support "pull" models—a topic to be addressed shortly. However, the present inventions also support narrowcasting or "push" models of content distribution as well.

In one example, the matching and classification utility 900 can facilitate much more automated and therefore more efficient and effective content creation, access and/or distribution services that "push" information and/or services to users. Example Figure 47 shows an example "information push" model 2000 in which an arbitrary number of users 2001(1)-2001(n) each have a VDE node (e.g., a protected processing environment 154) installed on their appliances. These example appliances may be of any kind, including computers, so-called Web television or Web-TV, DVD appliances with some form of backchannel, a settop box with a "back channel", and so on.

Perhaps with the permission of the user or other authority, such as an administrator within an organization, the VDE node collects various usage information or "info exhaust" according to the rules and usage consequences provided by one or more value chain participants. At times specified by default and/or by the associated rules and consequences, audit records are sent, in this example, in VDE containers 2006(1)-2006(n) to a usage clearinghouse 300, which in turn, may send all or a portion of these audit records in a VDE container 2008 to the matching and classification utility 900. The audit records may contain rights management information, including, but not limited to the amount of usage, the amount paid, if any, the payment method used, if any, VDE control sets, and/or data that identify various attributes of the node, user, and/or known and/or used object(s). The audit records may also contain information about objects known to the VDE node (objects with PERC records - see Figures 35A, 35B and associated discussions) and/or objects that have been used (objects with URT entries - see Figures 40A-40B and associated discussions) on the node.

The matching and classification utility 900 may also receive from one or more providers 2010 content objects 2003 themselves, for example, information in text format and/or metadata 2005 associated with content objects. Using at least one classification method, the matching and classification utility 900 may create at least one object class hierarchy, object class, object classification scheme, object category and/or object category scheme using at least some rights management information and assign at least one object to at least one category and/or class.

The matching and classification utility 900 takes the usage information and other rights management information received from the VDE nodes and/or other information sources and may create at least one category and may assign at least one node and/or user to a cateogry and/or class. In Figure 47, the matching and classification utility 900 sends a VDE container 2002 to content provider 2010 with information showing the classes of content used by one or more nodes and/or users along with a request that the provider 2010 send similar content back to one or more users 2001. At least one content provider 2010 then sends at least one VDE container 2004 to user A with content and/or information about available content that may be of interest to user A given the history of content usage as reflected in VDE audit records and/or other rights management information. In this "push" example, classes of content or information about available content may be pushed automatically from (a class of) content providers to one or more members of class of users and/or nodes. Consequently, users do not have to search as intensely, if at all, for content of interest to them.

In this example, user A receives content that may be most like content the user has already used, perhaps like content used most frequently in the recent past. The present inventions also support the matching and classification utility 900 and/or content provider sending content that is in a class or classes more distant from topics of prior and current interest to a particular user and/or group of users. Certain classification methods familiar to those skilled in the arts may provide quantitative indicators of distance that, in turn, may be used as at least one criterion for selection.

In another example, matching content to users and/or nodes may be based in part on class assignments that are in turn based in part on information concerning user preferences solicited by the matching and classification utility 900 or other value chain participant, such as a market research firm, advertising agency, provider, distributor, VDE administrator 800, or other Commerce Utility System.

Although the matching and classification utility 900 and/or content provider may send "more of the same," in another example the present inventions support providers at least occasionally sending content more distantly related to the user's apparent interests to determine if the user's circle of interest might be a little larger than that indicated by past usage and other, related rights management information alone.

In another example, providers may from time to time send content unrelated to the user's apparent interests that may nevertheless reflect the interests of persons and/or groups sharing at least one attribute with the user. For instance, the matching and classification utility 900 may, by sending a VDE container with appropriate user and content class information, suggest to a provider that user A receive content similar to content used by another member or members in the same group or class as user A. In one example, the matching and classification utility 900 may suggest sending business information related to a particular vertical market segment because others in the same class as user A have paid attention to that market.

In support of various content narrowcasting or "push" models, the matching and classification utility 900 may provide content class related information to a "subject switch" or "subject mapper," which in turn, matches participants desiring information in one or more specified classes with one or more sources of content in the requested class or classes.

The non-limiting subject switching example 2050, Figure 47A, shows a number of customers 2053(1)-2053(n) each with an appliance 2052(1) -2052(n) such as a personal computer. Other arrangements may include appliances such as a WebTV interface and/or an intelligent "settop box" connected to an interface device that uses one or more (digital) TVs for display. Still other arrangements may include an NC computer without a local hard disk logically connected to at least one server, a personal digital assistant with a network connection, and/or any other appliances with suitable processing, storage, and communications capabilities.

Referring again to Figure 47A, each customer appliance 2052 may have a VDE secure node installation 2054 incorporating a protected processing environment 154, as described in “Ginter et al”, and messaging services software 2058 that manages communications with other appliances. (In an alternative example, some appliances may lack secure nodes or sufficiently secure nodes, and receive appropriate one or more protected processing environment 154 based services from one or more servers and/or peers.) These appliances may be located in the same physical and/or logical environment, such as on the same local area network, and/or may be distributed across wide area networks such as multi-location corporate Intranets and/or the Internet itself. Among other tasks, messaging services 2058 "listens" for messages destined for that particular appliance or for broadcast messages intended for at least one appliance in the set of appliances that receive the broadcast. In certain instances no appliance may actually be "listening." In other examples, the messaging services 2058 may incorporate delivery assurance capabilities that assure delivery through use of explicit or implicit acknowledgments of receipt combined with the ability to retransmit information that has not been acknowledged. Messaging services 2058 may be designed such that an operator may select from one or more delivery assurance levels, for example "no receipt acknowledgment," "retry n times, then notify operator if not received," "retry until a certain date/time, then notify operator if not received," "retry n times and/or until a certain date/time, no operator notification necessary," et cetera.

Messaging services 2058 may use the secure node 2054 to package one or more messages in a VDE secure container that may also include one or more sets of rules and usage consequences that may be associated with one or more messages in the container as described in “Ginter et al”. In this example, messaging services 2058 then sends the secure container to one or more destinations using, for instance, TCP/IP and/or some other network protocol(s). Also, messaging services 2058 may broadcast a VDE container to one or more other customers 2053.

In this example, a customer 2053 uses application 2060 to persistently request or "subscribe" to one or more particular classes of content. For example, a highly detailed class might include "business information concerning the US market share of PC vendors, information in text format, costing less than a dollar per item, and for which the subscriber receives the right to excerpt at least one whole paragraph, provided that the excerpted amount constitutes less than 25% of the entire item based on word count." This same and/or another application may also be used to interact with instances of content in the desired class, for example, by displaying information on a computer screen and/or another output device in accordance with the rules and usage consequences associated with that item. If a user no longer has an interest in one or more classes, they may also use the same (or similar) application 2060 to "unsubscribe" from a particular subject, or specify further narrowing or broadening criteria to adjust the flow of content from one or more classes.

Items in the desired class or classes may be available from more than one content source 2074(1)-2074(n). To enhance the efficiency of locating content of interest to the subscriber or other participant, the matching and classification 900 may have created such a class definition and assigned one or more content items to that class. In one example, the matching and classification 900 may have sent one or more methods, and administrative events necessary to invoke the method(s), in a VDE secure container to one or more content sources 2074 where the classification methods are executed.

Such methods may, for example, assign content items to one or more classes. One or more object and/or item identifiers may have been transmitted to the matching and classification utility 900 along with class assignments for each item. If the matching and classification utility 900 has not previously created the desired class and assigned items to it, in response to a request from the subject switch 2051, the matching and classification utility 900 may do so using any appropriate combination of one or more such classification methods and procedures. The matching and classification utility 900 may may create at least one object class hierarchy, object class, object classification scheme, object category and/or object category scheme using at least some rights management information and assign at least one object, item, and/or subscriber to at least one category and/or class.

Subsequent to receipt of the request and/or "subscribe" message from the customer 2053, the subject switch 2051 may query the matching and classification 900 for content sources 2074 that have items in the desired class or classes. The matching and classification utility 900 may respond with information indicating known sources of information in the desired class(es), if any. The subject switch 2051 may then send a VDE container to the appropriate content source(s) 2074 indicating that certain customers 2053 are interested in items in the desired class and that the content source 2074 should send items in this class to this customer 2053 and/or groups of customers, and/or include such content in broadcasts which may be received by such subscribers.

The content sources 2074 may have already received class definitions and class assignment information from the matching and classification utility 900 and/or may have received from the matching and classification utility 900 or another party to the transaction one or more classification methods and associated events to invoke one or more of these methods to perform classification and/or class assignment processes.

In one arrangement, the content source 2074 may send the desired items directly to the subscribing customers 2053 by using the messaging services 2058 and subject switch 2051 to publish each item as it becomes available for distribution. In another example, the content source 2074 may broadcast the information such that subscribers' messaging services 2058 will have the opportunity to access the such items from a broadcast. The content source 2074 may call on messaging services 2058 to use the VDE secure node to package the item in a VDE container along with associated rules and usage consequences and then send that container such that one or more listening messaging services 2058 on other appliances 2052(1)-2052(n) will receive it. Based on subject information contained in the message header and/or in unencrypted (but optionally protected for integrity) areas of the VDE container, the listening messaging services 2058 may identify the message as belonging to a subject class it is listening for, then use the VDE node to open the container and view or otherwise use the item in accordance with that item's associated rules and usage consequences.

In another arrangement, the subject switch 2051 may be located on each customer appliance 2052(1)-2052(n). Using messaging services 2058, each subject switch 2051 may communicate with the matching and classification utility 900 to locate sources of content matching the subscribed classes. In this example, the subject switch 2051 on the local appliance then uses the messaging services 2058 to communicate with one or more content sources 2074 indicating classes of content to which it wishes to subscribe. Using the messaging services 2058, one or more content sources 2074 may directly send and/or broadcast items in the desired classes to subscribing customers 2053 in VDE secure containers along with associated rules and consequences. In another arrangement, the content source 2074 may send one set of rules and usage consequences that apply to members of one or more item classes, thus potentially improving the efficiency of distribution and of rights management. In another example, the rules and content items may be sent in separate VDE containers. In this example, the messaging services 2058 and subject switch 2051 listen for messages that are addressed to those customers who subscribe to a particular content item class and makes those items available to customers using an application 2060.

In another arrangement, messaging services 2058 and/or subject switch 2051 may be installed and run on network routers, network switches, one non-limiting example being ATM switches, and other packet and/or cell switches.

Example: Digital Broadcasting Based On Matching And Classification

"Shear et al" discloses a Digital Broadcasting Network ("DBN") that may function as a cooperative of Web sites and, for example, service providers, with a central and perhaps regional and logical (e.g., market based) headquarters groups, or it may function as a for profit, shareholder corporation in a business model reminiscent of television broadcast companies (e.g., NBC), or it may function as a cooperative or virtual corporation that has some mix or combination of mixes of the above attributes and employ distributed peer to peer, hierarchical, and centralized administrative business relationships and activities.

In one example, plural corporations may join together to provide the advantages of size and coordination with individual participants providing some degree of specialty expertise and the body of entities coordinating together in some fashion in a "higher" level cooperative or corporation.

Figure 48 shows one non-limiting example 2100 of a DBN that includes one or more DBN Web servers 2104(1)-2104(n) and one or more Web users each with VDE nodes. Users are attracted to a specific DBN server (or servers) because it provides access to specialized content and/or services 2108. Based at least in part on rights management information 2110 collected from DBN servers, for example, controls associated with the most frequently requested information, the matching and classification utility 900 creates categories of content (and/or services) and assigns DBN servers to one or more classes according to their specialization(s). The matching and classification utility 900 may may create at least one class hierarchy, class, classification scheme, category and/or category scheme using at least some rights management information and assign at least DBN server and/or at least some information to at least one category and/or class.

For example, one DBN server may specialize in consumer sports information while another may specialize in legal information. DBN servers may specialize in plural content (and/or service) areas. This class and class assignment information is provided to DBN servers, to content (and/or service) providers, or both.

The matching and classification utility 900 in one example sends VDE containers 2112 to content sources 2102 indicating specific classes of content that should be sent to one or more DBN servers 2104. Using this information, content providers 2102(1)-2012(n) then send content in these categories in VDE containers 2106 that match the categories of most frequently hit and/or consumed content on a DBN server 2104(1)-2104(n). (In another example, other information may be used as the basis of classification, matching, and selection.) For instance, the matching and classification utility 900 sends a VDE container 2112(2) to content source 2102(1) with instructions to send content in categories 1,11, and 15 to DBN server 1 (2104(1)). This content may, in turn, be sent to one or more consumers in VDE containers 2108(1), 2108(3).

In one aspect, this example process is analogous to hard goods manufacturers and distributors keeping Wal-Mart shelves stocked with those items in greatest demand based on point of sales and inventory data. One difference, of course, is that in this example, the DBN server is stocked with intangibles in the same or similar class as the intangibles sold rather than providing replacements for hard goods that have been sold off the shelf. In another example, a DBN server may send its classification data to content providers along with a request that they send more of the same. The request may be sent independently of the class information.

In another example, the matching and classification utility 900 may receive content and/or rights management information from providers and go on to create classes of content and/or content providers in which the classes may be partly defined using rights management data. Content on one class may, among other things, be distinguished from content in another class by price, payment methods, usage opportunities (e.g., available for printing, available for viewing pay-per-use), usage consequences, and/or specific permissions. The matching and classification utility 900 may subsequently send a communication, perhaps in a VDE container, to providers indicating that they send content in one or more specified classes to at least one DBN server.

Non-limiting example Figure 48 shows that the DBN 2100 may consist of video 2202 and/or audio 2203 content providers who send certain categories of video and/or audio content 2206 to DBN servers 2204(1)-2204(n) based on the categories of content each server may specialize in, which, in turn, may be determined at least in part on frequency of usage and/or other rights management information sent in VDE containers 2213 to the matching and classification utility 900, or to a usage clearinghouse 300 and then to a matching and classification utility 900. (In another example, other information may be used as the basis of classification, matching, and selection.) The matching and classification utility 900 sends VDE containers 2212 to content sources indicating that they should send content in specific categories 2206 to specific DBN servers 2204. In turn, each DBN server 2204(1)-2204(n) delivers video 2208 and/or audio 2209 in VDE containers to parties interested in such content. In another example, a VDE container may hold both video and audio and/or any other content type.

Example: Matching and Classification Utility 900 Can Also Support "Pull" Distribution Models Based On Classes

Notwithstanding the noted trend toward "push" content delivery models, the present inventions also enhance the efficiency, focus, specificity, and convenience of content "pull" models. In one example 2300 (Figure 49), the matching and classification utility 900 sends in VDE containers 2306(1)-2306(n) at least one administrative event and/or associated method that performs classification and/or class assignments to a VDE-aware appliance. The administrative events and method(s) are processed under the control of the VDE node. In one example, the results of processing the classification method may indicate at least one class of content and/or services of interest to a user and/or node. The classification method may also create at least one class hierarchy, class, classification scheme, category and/or category scheme using at least some rights management information and assign at least one service and/or at least some content to at least one category and/or class.

Subsequently, a VDE container 2308 may be sent to a provider 2302 with information indicating at least one class of content, services, transactions, rules and/or usage consequences, such as the ability to modify, excerpt and/or reformat, and/or events and a request that that the provider send content and/or pointers to services that meets the stated criteria and/or descriptive information about such content, services, transactions, and/or events to the requesting user and/or node. The request may, for example, be initiated explicitly by the user and/or node or may be initiated by the node according to one or more administrative events and associated methods and/or control sets. In turn, the content provider 2302 sends a VDE container 2304 to the requesting user 2306(1) with content that matches the desired selection criteria and/or profile.

The user may elect to use, consume, purchase, and/or rent one or more content objects (or use one or more services). As this one example shows, the user pulls in content and/or interacts with services by matching at least one class indicating user preferences with at least one class of content objects and/or services and/or transaction types.

Example: The Enterprise Distributed Matching And Classification Utility

Businesses and other organizations may be concerned with privacy and confidentiality regarding information and/or services used within the company. This concern may be manifest regardless of whether the information and/or services originated inside and/or outside the organization. Thus some organizations may have strong incentives to take advantage of the present inventions by operating a distributed matching and classification utility 900 to provide matching and classification services within the enterprise while at the same time maintaining a higher degree of confidentiality and privacy by selecting and/or limiting the nature, range, and detail of information sent outside the organization.

Figure 50 shows an example 2400 of an entity 2406 that has one or more VDE enabled appliances and users 2420(1)-2420(5) on a corporate Intranet 2418. These appliances may be, for example, computers, workstations, mainframes, or more specialized devices, such as supercomputers and/or graphics workstations for animation and special effects. The company may also operate internally one or more Commerce Utility Systems, perhaps including a financial clearinghouse 200, a usage clearinghouse 300, and a matching and classification utility 900. The company may also operate at least one content server 2414. These commerce utility systems and servers are also connected to the company Intranet 2418. The company 2406 also maintains one or more connects to the Internet 2402. (In another example the company may maintain connections to at least one private network operated by themselves and/or another party in addition to, or instead of one or more connections to the public Internet.) The content server(s) may provide access to internal, proprietary company information and/or to external, often commercial information. The internal content server may act as a gateway to external providers 2404(A)-2404(C) and/or may host commercial content locally on a content server 2408.

In one example, VDE audit records and/or other rights management information are sent in VDE containers 2412 from one or more VDE nodes 2420 to the enterprise usage clearinghouse 300 which may forward at least some of this usage information in VDE containers 2410 to the enterprise matching and classification utility

900. The enterprise matching and classification utility 900 may also collect from internal information sources 2414 information in addition to audit and rights management information, such as information in a human resources, accounting, and/or budgeting database containing data about company employees. These data may indicate, in one example, titles and responsibilities within the company, budgets allocated for external information and/or services, authority to spend, and budget remaining. The budget and financial information may have come in part from the financial clearinghouse 200. The matching and classification utility 900 may also create at least one class hierarchy, class, classification scheme, category and/or category scheme using at least some rights management information and assign at least service and/or at least some content to at least one category and/or class.

In one example, using at least some VDE rights management data, for example, whether certain information can be viewed by anyone, by any employee, or only by employees in certain job classes, such as "manager," the enterprise matching and classification utility 900 creates one or more categories and assigns one or more employees and/or VDE nodes to one or more topic categories. These categories may, for example, indicate content and/or service topics, subjects, and/or content areas of potential interest to each employee and/or groups of employees sharing at least one attribute in common, for example, similar interests and/or responsibilities.

In turn, the enterprise matching and classification utility 900 sends to at least one external content and/or service provider 2404 on Internet 2402 one or more VDE containers 2424 with information that indicates categories of interest. The content providers 2404 may themselves be specialized; in one example, a content provider may specialize in general business and financial news while another may specialize in scientific, medical, and/or technical information. In another example, a single content and/or service provider may provide an extremely broad range of content and/or services.

The external provider may send at least one VDE container 2422(1) with content and/or rules and consequences and/or metadata about content and/or services to a content server internal to the enterprise. In another example, such VDE container(s) 2422(2) may be sent directly to an employee and/or one or more groups of employees. The information sent by the external provider is tailored to, or in some way responsive to the content and/or service categories requested by the enterprise matching and classification utility 900.

In another example, the enterprise matching and classification utility 900 itself may be a distributed commerce utility implemented on more than one computer and/or other appliance within the enterprise. These several matching and classification utility 900s may serve different geographic areas and/or may themselves specialize in particular content and/or service areas.

In another example, the enterprise matching and classification utility 900 send class and/or class assignment information to a matching and classification utility 900 in another organization that, in turn, may be part of a common value chain.

Example: Chain of Handling and Control Entails Class-based Rules and Usage Consequences

VDE-based value chain management or "chain of handling and control" disclosed in “Ginter et al” enables, amongst other things, plural parties to independently contribute rules and usage consequences under the authority and/or control of more senior or prior participants in the value or distribution chain. Class-based rules may play a role in the efficiency and effectiveness of creating, operating, and/or extending value chain processes.

Figure 51A shows an example 2500 of a publisher ABC 2502 using a VDE packaging application 2510 to put into a VDE secure container 2512 sets of rules and usage consequences that may vary according to class. In this non-limiting example, the class is "content type." The publisher may have rights in a wide variety of content and content types. Consequently, the publisher may create rules for text objects that may differ from rules for audio objects.

The publisher 2502 sends the class-based rules and usage consequences to a first creator 2504 who also has installed VDE on her or his appliance 2516 and who has also been given one or more certificates and/or other digital credentials by the publisher (and/or trusted third party) indicating that he is indeed a creator authorized by the publisher 2502. The publisher has included rules that allow only authorized value chain participants to package content using publisher provided rules and/or to modify, enhance, extent, and/or change some or all of the publisher's rules.

The first creator 2504 then uses a VDE packaging application 2510 to package an image he has created in a VDE container 2514 according to the rules provided by the publisher and with the addition of the creator's own rules. In one example, the first creator contributes rules that implement a one-time 50 cent charge to the consumer for opening and viewing the creator's image. The creator may also contribute rules reflecting his wish to receive audit records with information concerning the consumer and/or context in which the image was used. These creator rules and usage consequences are contributed generally independently of the rules and usage consequences contributed by the publisher. Note that the VDE container 2514 now holds at least the publisher's 2502 rules for each object class, the first creator's image and his associated rules and usage consequences.

A second creator 2506 receives the VDE container from the first creator and using a VDE packaging application 2516 adds a text file to the container 2520 along with her rules and usage consequences. As before, she also has a certificate and/or other digital credential(s) identifying her as authorized by publisher ABC to add and/or modify content and rules and usage consequences. As in the case of the first creator 2504, she adds her text and rules and usage consequences generally independently of controls contributed by prior participants. She may, in one example, prevent printing of the text and charge $1.00 the first time a consumer opens and views the text.

The VDE container 2508 now holds text and rules and usage consequences contributed by creator 2 (2506), an image and rules and usage consequences contributed by creator 1 (2504), and the class based rules (and perhaps other rules as well) contributed by example publisher ABC 2502.

Creator 2 (2506 sends the VDE container 2508 to publisher ABC 2502 who then sends the container 2522 directly and/or indirectly to consumers. When the consumer uses the content, the rules and usage consequences of all three value chain participants (and of other possible participants as well, distributors and repackagers, for example) are applied.

Example 2600, Figure 51B shows that the publisher 2602 may have sent a VDE container 2612 with various rules and usage consequences to a matching and classification authority 900 who may classify the rules and send the rules and their class assignments to a rights and permissions clearinghouse 400. The matching and classification utility 900 may also create at least one class hierarchy, class, classification scheme, category and/or category scheme using at least some rights management information and assign at least one rule to at least one category and/or class.

An authorized first creator 2604 may send a VDE container 2617 to the rights and permissions clearinghouse 400 asking for rules in the class "rules for authorized creators, for image objects, from publisher ABC." The rights and permissions clearinghouse 400 returns a VDE container 2614 with rules in the requested class. The first creator 2604 uses a packaging application 2616 to package his image using these rules plus rules and usage consequences reflecting his rights and wishes and sends the VDE container 2614 to the second creator 2606.

The second creator 2606 also sends a VDE container 2619 to the rights and permissions clearinghouse 400 asking for rules and consequences in the class "rules for authorized creators, for text objects, from publisher ABC." The rights and permissions clearinghouse 400 returns a VDE container 2621 with rules and consequences in the desired class. The second creator 2606 uses a packaging application 2618 that determines that she is a creator authorized by publisher ABC 2602 and goes ahead and adds her text object and her rules and consequences to the VDE container 2608, which is then sent to the publisher ABC 2602 for further augmentation, vending, and/or distribution to other value chain participants.

Example: Secure Directory Services May Provide Class And Class Assignment Information

Whole industries have arisen to target communications to individuals, organizations, groups, and/or other classes sharing at least one common attribute, and/or to provide directories from which others can locate individuals, organizations, groups, and/or other classes. Examples of these industries include direct marketing, advertising, yellow and white pages directories, directories of directories, and various electronic and paper membership lists and professional directories.

In addition to identifying information such as names, e-mail addresses, physical mailing addresses, phone numbers, fax numbers, and/or similar attributes, the secure directory services 600 may also provide information about class membership(s) for individuals, devices, services, groups, and/or organizations. The non-limiting example 2700 shown in Figure 52 includes a secure directory service 600 that has received class and class assignment information for one or more individuals 2716(1)-2716(n). The class assignment information is shown in the bottom four rows of the directory record 2718(1) for one individual.

In this example, a content provider 2702 sends a VDE container 2704 to a secure directory services 600 asking whether the service can provide a list of individuals in class "AF." The requested class could be any class defined by one or more attributes and may be based on usage profiles that include rights management information, non-exhaustive examples of which include price, payment methods accepted, permitted operations, meters, and privacy controls.

The secure directory services 600 returns to the content provider in a VDE container 2706 an indication that there are presently 57 individuals known to that service in class "AF." In turn, the content provider 2702 sends a VDE container 2708 with at least one piece of content and/or rules and usage consequences back to the secure directory services 600 along with instructions requesting that the secure directory services 600 forward the content and/or control sets to each of the 57 members of class "AF" who might be interested in this piece of content. The secure directory services 600, in turn, forwards the content and/or controls (in VDE containers 2714(1)-2714(n)) to members of class "AF," who may elect to interact with the content in accordance with their associated rules and consequences.

In another example, the secure directory service 600 may send identifying information 2710 directly to the content provider 2702 who may then send content 2712 in one or more classes directly to one or more members 2716(1)-2716(n) of the class. The secure directory services 600 may, for example, include permissions for the class information that have expiration dates and/or limits on the number of times the information can be used.

Example: Matching And Classification Utility 900 Supports Class-Based Micro-Merchandising And Micro-Segmented Sales Processes

The present inventions may be used in support of services as well as content distribution based business. Example 2800 (Figure 53) shows a travel company 2801 sending a VDE container 2810 to a matching and classification utility 900 requesting information on those individuals who may be interested in certain combinations of leisure-time activities. These classes might have been defined at least in part on the basis of usage and other rights management information 2816, for example, the kind of leisure-time information recently looked at, for how long, and/or its cost, and/or the kind of Web sites recently frequented, sent from consumer VDE nodes 2802(1)-2802(n) to the matching and classification utility 900, and/or to a usage clearinghouse 300 who, in turn, sends at least some of the usage information (or a summary form of such information) to the matching and classification authority 900. Classes may also be defined using information gathered directly from the consumer 2818, perhaps under the control of VDE. The matching and classification utility 900 may also create at least one class hierarchy, class, classification scheme, category and/or category scheme using at least some rights management information and assign at least one consumer, service, and/or at least some information to at least one category and/or class.

Example Figure 53 shows that a consumer 2802(1) has recently indicated a preference and/or interest in skiing, music, and flying to

Colorado. Another consumer 2802(n) has indicated a preference for and/or interest in surfing Hawaii. These preferences may be determined at least in part on the basis of rights management information. In response queries sent in one or more VDE containers 2810 from the travel company asking for interest and preference information, the matching and classification utility 900 returns one or more VDE containers 2812 with identifying and class information. The travel company may send information about already existing vacation packages and/or packages specially created to meet the specific interests of one or more individuals, for example, information about skiing in Colorado, and rock concerts 2604 to consumer 2802(1) and information 2614 about surfing Hawaii to consumer 2802(n). The recipients may send VDE containers 2806 to the travel company 2801 indicating agreement to buy the package offered or may request additional information or may negotiate terms and conditions such as price, departure date, insurance, and the like. These negotiations may be conducted using the inventions described in "Ginter et al", Figures 75A-76B using VDE negotiations.

Both services and/or hard goods may be offered to particular persons, nodes, groups, and/or entities based on the class membership of the potential purchaser and the class membership of the goods and/or services to be purchased. Thus in another example, the travel company could have included the purchase and/or rental of the skis or of the surf board.

Example: Matching And Classification Utility 900 Supports Trading in Hard Goods

Business to business trading in goods and/or services may be substantially facilitated through services provided by the matching and classification utility 900. Information on certain classes of goods and services may be delivered to certain people, groups, or entities based on the class membership of the recipient. In one example, these various class memberships may be determined using control set and audit information regarding trading preferences and/or transaction patterns. In another example class membership may be determined by actions and/or information provided by at least one value chain participant.

Example 2900 (Figure 54) shows a buyer A 2904 sending a VDE container 2908 to a trading company 2902 with a request asking if trading company will sell company A one or more desired items. Trading company 2902 may then send a VDE container 2910 to a matching and classification utility 900 with a query asking who can supply the desired items under terms and conditions that are also included in the container. Since these terms and conditions may be the subject of negotiations, they may be in a format conducive to VDE-based negotiations as described in "Ginter et al" Figures 75A-76B.

The matching and classification utility 900 may send inquiries 2910 to one or more suppliers 2906(A)-2906(N) and/or may have already received information and/or associated control sets from suppliers in VDE containers 2912. Based on the request from trading company 2902 and supplier 2906 information obtained 2912, the matching and classification authority 900 returns a VDE container 2916 indicating that in this one example, suppliers A 2906(A) and Z 2906 (N) can provide goods in the class(es) defined by trading company's 2902 request(s) 2910. In turn, trading company 2902 sends at least one VDE container 2918 to buyer A 2904 indicating that they will sell buyer A the previously requested items under the enclosed terms and conditions. In another example, there may be some VDE-based (see "Ginter et al", Figures 75A-76B) negotiations between the various parties in this value chain, including between trading company 2902 and buyer A 2904.

In another example, buyer A 2904 may consult the matching and classification authority 900 directly and may then purchase directly from one or more suppliers 2906.

Example: Matching And Classification Utility 900 Supports Securities Trading/Brokering

In addition to hard goods, the matching and classification authority 900 may also support securities trading. Example 3000, Figure 55, shows the matching and classification authority 900 sending to a VDE-aware appliance with one or more stock trading related applications 3004 a VDE container 3010 with an administrative event and method (as described in "Ginter et al") for classifying equities related information, including, as non-limiting examples, current and historical price, volume, and index information, financial performance data for publicly held companies, forecasts, risk management information, options and futures, and the like. The classification method may also utilize rights and permissions, including access control information, permitted operations, and/or expiration times and/or dates for rights management information. The classification method may also create at least one class hierarchy, class, classification scheme, category and/or category scheme using at least some rights management information and assign at least one element to at least one category and/or class.

In turn, using the VDE aware appliance 3004, the stock trader 3006 sends a smart object 3012 to at least one information source 3002 asking for information in at least one class identified by the classification method. In one example, the class may be information concerning "publicly traded companies with annual revenue greater than $500M in the healthcare sector in which the CEO has been in place less than 5 and greater than 1 year and with access restricted to customers (rather than available to anyone) with access and use expiring in 90 days." The information provider(s) 3002 returns a VDE container 3014 with information meeting and/or more closely meeting the stated class criteria. Based upon this and other information, the trader 3006 may go ahead and enter an order for at least one trade in at least one stock 3008. In another example, the trader may create or obtain methods that trade automatically in certain classes of securities.

Example: Matching And Classification Utility 900 Supports Trading in Currency and Debt Instruments

Among the classes of great value to traders are the classes of items whose trading maximize profits and/or minimize losses. Example 3100, Figure 56, shows a trader in currency and/or debt instruments 3102 sending a VDE container with market and other financial and economic information and VDE control set information 3108 to a matching and classification authority 900 with a query 3114 asking the matching and classification authority 900 to identify the class of currency trades and/or debt instrument trades that maximizes profit and/or minimizes losses. The matching and classification authority 900 applies one or more methods to the data and returns at least one class definition 3112, the assignment of possible trades to that class 3110, and relevant control set information, such as controls indicating who may see the information, and those that prevent unauthorized modification of the information. The matching and classification authority 900 may also return methods for executing the trade. The matching and classification utility 900 may also create at least one class hierarchy, class, classification scheme, category and/or category scheme using at least some rights management information and assign at least some trading information to at least one category and/or class.

The example trader 3102 examines the recommendation and sends VDE containers 3118 (A, B) with trade methods and control sets to a foreign exchange market 3104 and/or to a debt instrument market 3106 where the trades are consummated. The markets send back VDE containers 3116(A, B) with audit information indicating the results of the trading order. In another example, the matching and classification authority 900 may be instructed to send trading orders directly to the market(s) for execution. In another example the trader may send a VDE container to at least one source of relevant information asking that source to send certain information to the matching and classification authority 900. In another example, having established the desired trade(s) using the matching and classification authority 900, the trader may place the trade by phone and/or computer and/or other communications device without using VDE.

Example: Matching And Classification Utility 900 Supports Consumers Locating Services That Are Members Of A Specified Class

The services of the matching and classification authority 900 may also benefit consumers by locating certain classes of services. Example 3200, Figure 57, shows a consumer sending a VDE container 3206 to a matching and classification authority 900 asking, "which banks are in class A?," where class A are "those banks that offer the highest savings interest, no ATM fees, online/Web banking using VDE, insured accounts, free checking with balances larger than $2,500, "image" statements (where check images rather than the actual checks are returned), and complete privacy protection (except where legally required to disclose) for VDE based banking transactions.

The example matching and classification authority 900 sends a query in a VDE container 3208 to one (or more) information sources 3202 and receives one or more VDE containers 3210 with the requested information. The matching and classification authority 900 then determines which bank or banks meet the stated criteria of the consumer 3204 and then sends a VDE container 3212 with the answer to the consumer, in this example, banks A, B, and C. The consumer 3204 may then go ahead and execute a financial transaction, for example, transferring funds from one bank to a bank identified by the matching and classification utility 900 as offering higher interest rates, while being assured of maximal privacy for this (and perhaps other) transactions.

In another example, after determining which banks are in the desired class, the matching and classification authority 900 may send a VDE container to one or more banks saying that the consumer wishes to know about their services and requesting the bank to contact the consumer directly. The bank may send controls ensuring the privacy of future interactions with the customer. For example, controls that apply to audit records such that only the bank and the consumer will have permission to access these records.

Example: Matching And Classification Authority 900 Supports Class-Based Software Distribution

VDE and the inventions disclosed in "Ginter et al" at last provide a way of ensuring that the efforts expended on creating software will be rewarded since the software can now be persistently protected, usage information can be collected, and payment ensured. These inventions also support micropayments and microtransactions, thus creating a world in which the price of software objects—any kind of objects actually—may become very small. Pay per use, rental, rent to own, and other pay as you go pricing models together with VDE may create a new explosion of creativity in software design and creation, since use prices will be low and providers can be assured of receiving payment.

The present inventions provide opportunities for software providers to more efficiently market their wares. Example 3300, Figure 58, shows a number of users with VDE installed on their appliances 3304(A-F). These people are using software (and other content). VDE meters usage of various objects and sends audit records in VDE containers 3306 (A-F) to a usage clearinghouse 300, which then sends audit records 3308 to the matching and classification authority 900. A software distributor 3302 sends a VDE container 3310 to the matching and classification authority 900 with a query asking who is in the class, "buys Java applets, with pay per use pricing, and for which the cost per use is between $.0001 and $.001?"

The matching and classification authority 900 returns a VDE container 3312 with a list of names and (network) addresses of those matching, or most nearly matching the desired characteristic(s). The software distributor 3302 then sends at least one VDE container 3314 with at least one software object, and/or a pointer to a software object, in this case a Java applet, and perhaps other relevant information, such as VDE control sets and/or various metadata describing some aspect of the object, for example, what it does, what it costs, etc. The user may then elect to use the object or not. In another example, instead of individuals or VDE nodes, the users might be groups of nodes, users, organizations, parts of an organization, and others that can be identified as belonging to at least one class. In this case, the software may be offered to some or all members of class, group and/or organization.

Example: Matching & Classification Utilities Provide Services To Authenticated Classes of Nodes, Users, Content Services and/or Transaction Services

Among the ways in VDE nodes, users, content services, and/or transaction services can be authenticated is through the use of certificates and/or other digital credentials issued by an appropriate trusted third party, a certifying authority 500, for instance, that warrants and/or attests to some fact or facts, which may include membership in one or more classes, including the identity class. Figure 59 shows a non-limiting example 3400 in which a number of matching and classification authority 900(1-N)s, each of which may provide its services to different classes, where class membership is authenticated using certificates and/or other digital credentials. In other examples, additional authentication mechanisms may be used in combination with, or instead of certificates, such as information known only to the user, VDE node, and/or appliance, including passwords, cryptographic keys, information stored in hardware, and/or software.

In example 3400, Figure 59, commerce participants including, the matching and classification authority 900, may make rules and consequences conditional on class definitions and/or the assignment of members to a class. Class membership may be authenticated by a certificate and/or other digital credential issued by one or more commerce participants in addition to, and/or instead of a trusted third party such as a certifying authority 500. For example, a certificate and/or other digital credential may attest to user identity, that is, that a user is the user he or she claims to be. Nodes, devices, networks, servers, clients, and services, are other non-limiting examples of other commerce elements that may be authenticated with certificates and/or other digital credentials. Any commerce participant may issue a certificate, but other participants are not required to accept a given certificate as an authenticator.

Figure 59 shows multiple matching and classification authorities 900(1)-900(N), each of which may provide services to members of a particular class, in these non-limiting examples, to nodes in a particular deployment (matching and classification authority 900(1)), in a particular vertical segment and/or institution of society, such as Higher Education (matching and classification authority 900(2)), one or more value chains, such as business information content providers (matching and classification authority 900(3)), and/or a particular transaction and/or service arena, such as hard goods trading (matching and classification authority 900(n)). Other commerce utility systems, a certifying authority 500 shown in Figure 59, for instance, may also provide services to a class. In each of these instances, the services of the matching and classification authority 900 may depend upon finding certain authenticating certificate(s) and/or other digital credentials on the appropriate VDE nodes.

For example, matching and classification utility 900(1) provides services to nodes 3410(1-n) in the deployment 3402 administered by VDE administrator 800. Each node may have a certificate 3412 issued by certifying authority 500(1) that provides services to this deployment.

In another example, certifying authority 500(2) provides certificates and/or other digital credentials to participants in a higher education value chain 3404 consisting of an arbitrary number of colleges and universities 3416(1)-3416(n), providers 3418(1) and students 3418(n), and a matching and classification utility 900(2) that provides classification, matching, and selection services to higher education 3404. In one example, the matching and classification utility 900(2) only provides services to value chain participants who have a certificate 3420 issued by certifying authority 500(2).

Matching and classification utility 900(3) services can be provided only to members of one or more classes based on certificates issued by a certifying authority 500(3). In one example, the class is participants in a business information value chain 3406, comprising an arbitrary number of content providers 3424(1)-3424(n), an arbitrary number of users and/or consumers of business information 3422(1)-3422(n), and a certifying authority 500(3) that issues certificates and/or other digital credentials to members of the value chain 3406.

In addition to membership in certain deployment, institutional, and/or content usage classes, the matching and classification authority 900(4) may provide services to members of a certain transactional value chain, in one example, traditional transactions 3408. In this example, a certifying authority 500(4) issues certificates 3432 to one or more companies 3428(1)-3428(n) and one or more trading companies 3430(1)-3430(n). In another example, other participants may receive certificates and/or other digital credentials, including banks and financial institutions, government authorities, for example, tax and/or customs authorities, consumers, suppliers, and/or transportation companies. The matching and classification utility 900(4) provides services only to those entities and/or individuals in possession of the appropriate certificate 3432 indicating that the holder of the certificate is an authenticated participant in one or another trading value chains.

In other examples, a commerce utility system may provide services to more than one class where class membership is indicated by at least one certificate and/or other digital credential issued by a certifying authority 500 and/or value chain participant. In one example, matching and classification authority 900 might provide services to the class "Higher Education" and to the class "K-12 Education."

Possession of a certificate and/or other digital credential may be among the information used to classify a node, user, appliance, device, entity, and/or other commerce participant, and rules and consequences can be made conditional on membership in one or more authenticated classes and/or on the degree of confidence the rule provider has in the trustedness of the certificate and/or other digital credential issuer. In one example, a discount to higher education may be larger if the root for chain of trust for a given certificate is a well-known, highly respected and trusted third party, such as an authoritative accrediting organization, and smaller if the root belongs to the MIS department of a small college. In this example, the provider is willing to grant a higher discount when there is higher certainty that the recipient is in fact a member of a specific class or classes.

Example: Matching And Classification Authority 900 Supports Control Sets Based In Part On Employee Classes, Content Classes, And/Or Certificates And/or Other Digital Credentials

Chain of handling and control enables, amongst other things, multiple organizations to work together in secure, trusted, efficient, cooperative commerce processes. One way in which the present inventions extend these ideas is through control sets with rules and usage consequences that may be based in part on classes and the assignment of persons, entities, devices, content, services, or other process elements to classes of one kind or another by the matching and classification authority 900.

One example technique to classify employees is at least in part according to their roles and responsibilities within an organization. The matching and classification utility 900 supports classification, matching, creation and/or modification of VDE control set(s) based at least in part the class assignment of individual and/or groups of employees. In part by virtue of their employee classification, at least one employee may receive certain rights management information, for example, permission to access certain classes of information or permission to perform one or more permitted operations, transactions and/or events.

Example 3500, Figures 60A-60C shows a nurse 3504(1), physician 3504(2) , and billing clerk 3504(3) all work directly for an example hospital. The present inventions are in no way limited to hospitals, but apply to any organization, group, entity, and/or institution with at least some defined roles and responsibilities and/or other class definitions that apply to employees, members, and/or others associated, affiliated, and/or employed by the organization, group, entity and/or institution. Rights management information may be part of the claim definition, for example, permissions to view, modify, excerpt, and so on.

Control sets may provide permissions conditional on employee class, for example, certain classes of employees may modify certain information and/or classes of information in a database while others may not. Class membership may be indicated by digital credentials, non-limiting examples of which include digital certificates and digital membership cards. Controls may be conditional on other information as well, for example, some computers and/or display devices may not show certain classes of data or updates to certain data elements may not be performed from certain computers or display devices.

Another example role is a representative 3504(4) of an insurance company 3508, who may have access to certain classes of hospital information by virtue of her or his class membership(s), some of which may derive from her or his role in the insurance company 3508 and/or from the insurance company's relationship with the hospital and/or with some of the hospital's patients and/or staff. The present inventions are not limited in application to an insurance company, but may be applied to any individual, group, organization, entity, and/or institution with whom the example hospital and/or other entity has some form of relationship.

An example insurance company 3508 have received a certificate in a VDE container 3534 issued by certifying authority 500(1) attesting to the identity of the insurance company. In another example, this certificate and/or one or more additional certificates may attest to the fact that the insurance company has the appropriate charter, licenses, and other grants of authority to be in the health insurance business. The certifying authority 500(1) may also send a certificate in a VDE container 3532 attesting to hospital's identity. In another example, this certificate and/or one or more additional certificates may attest to the fact that the hospital has the appropriate charter, licenses, and other grants of authority to provide hospital and related services.

The insurance company 3508 may have sent one or more control sets to the hospital in a VDE container 3542. These controls may be based in part on one or more certificates 3530 and/or on the classification output of an example matching and classification utility 900(2) operating within and/or on behalf of the insurance company 3508. The controls in container 3542 may indicate which individuals are actually employees of the insurance company, employee membership in one or more classes, permissions associated with that individual and/or class, and/or permissions associated with specific devices, communications channels (devices, ports, etc.), and/or processes. In this one example, the hospital matching and classification utility 900(1) may create controls using the same and/or additional classes and controls received from the insurance company 3508.

The insurance company 3508 may also provide one or more certificates to the hospital attesting to the fact that one or more information sources within the insurance company are to be take by the hospital as trusted sources. Lastly, in this regard, the insurance company may issue one or more certificates on behalf of each employee attesting that each is in fact an employee of the company and may have certain authorizations.

In example 3500, Figures 60A-60C, a matching and classification utility 900(1) has identified various classes of hospital employees using information from at least one hospital information system 3502 and/or VDE node. The matching and classification utility 900(1) may also make use of certificates issued by a certifying authority 500(1)outside (a trusted third party) and/or a certifying authority 500(2) inside the hospital. Using data dictionaries 3522, patient records 3520, various employee information 3524, automated procedures, and/or other means, the matching and classification utility 900(1) creates classes 3526 of patient record information and associates one or more control sets 3528 with each class of information and/or with a patient record as a whole. These control sets may specify who has permission to use and/or modify the record and/or an element(s) of the record that has been assigned to one or more classes on which the control set(s) may in part depend. In one example, the class based controls 3528 may be combined with other hospital and/or other party controls, controls from the insurance company 3508, to create new controls 3510(1)-3510(n) associated with patient records 3512(1)-3512(n).

The example nurse 3504(1) and physician 3504(2), for example, may be able to view, modify, print, and/or copy patient's name, address, and other similar descriptive information, next of kin, insurance, and medical information in accordance with controls 3510(1) and 3510(2), respectively . In another example, some members of the class "nurse" and/or the class "physician" may have different permissions by virtue of membership in one or more additional classes. A physician who is in the class "hospital administration" may have different permissions, for example, to billing records.

A billing clerk 3504(3) in the hospital may not have permission in control set 3510(3) to view medical information and/or next of kin, and in this example may be restricted to name and other patient descriptive information, insurance information, and billing information from the patient record. A representative 3504(n)of the insurance company may have permission by virtue of control set 3510(n) to view, but no permission to modify, print, or copy patient record 3512(n). In each of these examples, the VDE control sets are at least partially conditional on the presence and/or absence of certain certificates indicating membership in one or more classes.

The present inventions may be applied to any information, person, group, device, network, service, database that pertains to any commerce activity whatsoever, and regardless of whether the parties to the commerce activity are individuals, groups, entities, organizations, institutions, nations, and/or societies.

Example: Matching And Classification Authority 900 Supports Classes And Matching Based In Part On Workflow And Work Process Automation

Not only do the present inventions enhance commerce processes that principally entail information, but the present inventions enhance workflow and work process automation as well. Example 3600, Figure 61, shows PCs 3608(a-c) functioning as station controllers connected to various manufacturing devices 3610 (a-c). These station controllers that exchange data and instructions with the equipment they control and/or manage. The station controllers are VDE-enabled. In another example, the manufacturing equipment may also have VDE nodes installed.

An example work in progress (WIP) and/or manufacturing control application 3606 keeps track of the overall manufacturing processes and exchanges information with other applications not shown, such as materials management, materials ordering, order databases, logistics, inventory, accounts payable, accounts receivable, general ledger, human resources, time cards, and the like.

An example employee 3602 of the company sends a query 3612 in a VDE container 3604 to an enterprise matching and classification utility 900 within the company asking, "which VDE-controlled equipment will be available 3rd shift today, for 2 hours, capable of performing operations xyz with a nominal error rate of less than .0001 per cent?" The enterprise matching and classification utility 900 may request data 3616 from the WIP/manufacturing process control application 3606 and/or may already have access to the required data, indicating equipment availability, security level, capabilities, and statistical error rates. The WIP/manufacturing process control application 3606 may return a VDE container 3618 with the requested information. Based upon the query and available information, the matching and classification utility 900 responds by sending a VDE container 3620 to the employee 3602 with the answer, "equipment B and equipment C." In turn, the employee 3602 sends another VDE container 3622 to the WIP/manufacturing process control application 3606 with VDE a control set(s) indicating B and C should be scheduled for 2 hours on 3rd shift to do xyz operations. As part of this particular chain of handling and control, the WIP/manufacturing process control application 3606 sends VDE container 3624 to the VDE-enabled station controllers for equipment B or C with control sets that schedule work and specify the manufacturing processes and/or "recipes" for those specific equipment 3610(b) or 3610(c). In turn, the respective station controllers carry out their instructions and report progress and completion in VDE containers 3626 sent back to the WIP/manufacturing process control application 3606, which may in one example, provide results to other applications and/or to the employee who originally requested the work to be scheduled and performed.

Example: Matching And Classification Authority 900 Supports Classes And Matching Based In Part On Government/Societal Commerce Administration

Among the rightsholders in commerce processes of all kinds are societies and governments. Governments may foster rules indicating that certain classes of individuals may have not have access to certain classes of content. Some classes of information may be treated as members of classes that define permissions, such as "confidential," "secret," "top secret," and so on. Other non-limiting example governmental rights may address permissions for import, use, and/or export of certain classes of hard goods, services, currency and financial instruments, and content. Travelers entering the United States, for example, are usually asked about currency (and currency equivalents) being brought into the country by the traveler. Children, for example, may be prohibited as a matter of law by governments from viewing content in the class "sexually explicit."

Another example of government rights is that different tax rules may be applied to different classes of electronic commerce transactions using VDE. Example 3700, Figure 62A-62B, shows a certifying authority 500 operated by and/or on behalf of a government issuing a certificate and/or other digital credential indicating jurisdiction, namely, country. The certificate is sent in a VDE container 3710(a) to a VDE administrator 800. The government certifying authority 500 also sends certificates in VDE containers 3710(b)-3710(n) to the government matching and classification authority 900 attesting to the "country," in one example, the United States, and another certificate 3716 attesting to the fact that the matching and classification authority 900 is indeed an authorized service of the United States government.

In one example, the government matching and classification authority 900 has created tax class definitions 3712 and tax control sets 3714 that apply those definitions in various classes of circumstances, including the presence of certain control-related information, such as an appropriate country certificate from an authorized issuer of such jurisdictional certificates. The tax class definitions 3712, tax control sets 3714, and government authority certificates 3716' are sent in at least one VDE container to a rights and permissions clearinghouse 400, who, in one example, redistributes the tax class definitions 3712(1), tax class control sets 3714(1), and/or government authorization certificate 3716(1) to content providers 3702, service providers 3704, and other value chain participants. The certifying authority 500 also sends country certificates to one or more VDE administrators 800 who, in turn, send country certificates 3710' to VDE nodes 3706(A)-3706(n) in their deployment. When content provider 3702 distributes content of any kind, the appropriate tax control sets 3714(A) are also included in the VDE container. A tax control set is applied whenever content is used in accordance with a tax class and provided that the appropriate jurisdictional certificate 3710' is present on the VDE node 3706(a). For instance, a VDE node may have a tax control set to be applied to sales of a class of content, specifically, to the class of "software." Whenever a software vend occurs, the appropriate tax is applied according to these rules.

In another example, the various country and government authority certificates may be sent directly from the certifying authority 500 to one or more VDE nodes 3706. The VDE controls that implement tax policy for one or more classes may also be sent directly to VDE nodes 3706 and/or to VDE administrators 800.

Example: Classification May Be Used In Automatically Selecting The Proper Display Context Based On Classes Of Information

Content objects may be displayed using one or another formats according to class membership of that object. In example 3800, shown in Figure 63A, a matching and classification utility 900 provides content class information 3810 to information providers 3802. A consumer 3807(1) previously has sent a VDE container to a provider of sports information 3802(1) indicating interest in "class b" stories, and perhaps other classes as well. The sports information provider 3802(1) sends back a VDE container 3808(1) with one or more stories in "class b," perhaps "all stories about baseball, New York, Yankees, history, heroes with permission to print" an example of which is 3814(1), along with, in this example, one or more VDE control sets. The VDE container 3808(1) is received by a customer 3807(1) who then displays the content 3814(1) using one or another page formatting technologies based on macros, scripts, administrative events, methods, and/or other techniques. Also included in the VDE container is an image 3812(1) that was selected by the information provider as especially appropriate to the class of story being sent. In this example, perhaps the image 3812(1) is a faint image of Joe DiMaggio. This image also meets the criteria of "permission to print."

Example 3800, Figure 63A, also shows another instance in which a different consumer 3807(n) previously has informed a nature information provider 3802(n) of interest in class A stories. Here the information provider sends a VDE container 3808(n) that holds a class of stories different from the class of interest in the previous example. This VDE container 3808 holds a "class A" story, an example of which is 3814(n), that is displayed with a different image 3812(n), one that is appropriate to the story class, in this case, an image of a dog.

The class assigned to each story may be carried in the container as metadata for one or more story objects in another example. An example Web browser may request of the information provider an image appropriate to that class, which if available, would be sent in another VDE container.

Class may affect display rules in other example ways as well. For instance, several team sports news stories may be displayed in a Web browser window in which a scene from a football or basketball game is faintly discernible in the background. Which image is displayed may be determined by the user's preferences given the classes of stories being presented on the page. The user, may have looked most at stories about the New England Patriots and a Patriots-related image may be displayed as background even stories about teams in addition to (or even instead of) the Patriots were being displayed.

In (another) example 3850, shown in Figure 63B, a matching and classification utility 900 provides class information to a provider 3852(1). Previously, one user 3857(1) has indicated to the provider 3852(1) that she prefers information in topic class A more than information in topic class C and information that costs less than $.50 per article while the other user 3857(n) has the opposite preferences and is not price sensitive. A matching and classification utility 900 may provide classification information, class assignments for objects, administrative events, and/or methods for these and related purposes. Regardless, the information provider 3852(1) sends the identical VDE container 3858 to each of the users 3857. However, their browser and page formatting software 3856 produces different pages in accordance with each user's topic class preferences. In the example first case, the user 3857(1) sees three columns of topic A and one column of topic C while the second example user 3857(n) sees three columns of topic C and one column of topic A. As this example illustrates, the class preferences of users may affect the way in which the user interacts with content in various classes.

In another example, the matching and classification utility 900 may have sent one or more administrative events and/or methods 3859 to at least one user 3857 where the method performs the topic classification on documents and/or establishes topic classes and/or topic classes of greatest interest to the user.

Example: Information May Be Classified With Respect To Difficulty -- And This May Pre-Determine An Appropriate Interface

The class of content and/or the class of user may determine at least one display characteristic. One interesting example way of classifying content is with respect to its difficulty. One example measure of difficulty is reading level, which may reflect such aspects as vocabulary and/or complexity. It is well known that children (and adults) of the same approximate age read at different levels. In the example 3900, shown in Figure 64, a provider sends a VDE container 3902(1) with text at a 4th grade reading level and controls indicating that when used by a person reading at that level, the charge is 50 cents. However, if a person reads at less than the 4th grade level, the charge is only 40 cents. "Reading level" may be indicated by a certificate and/or other digital credential.

A matching and classification utility 900 may send administrative events and/or classification methods 3910 to information providers, one or more other value chain participants, or to the students appliances directly. These methods may, for example, classify documents according to the degree of difficulty and create or modify controls for the whole document and/or subparts of the document, controls that may indicate the different prices for users at different reading levels. The matching and classification utility 900 may also send administrative events and methods to users that know how to make the document appear in the example browser at a lower reading level.

The example VDE container 3902(1) is sent from the provider to a child 3906(1) in the 4th grade who is reading that at that level. When the child opens the container to view (or otherwise use) the text, she or he is charged 40 cents (which might be paid by a third party such as a school and/or parent. The child sees the text as written 3904(1)

Example 3900, Figure 64, also shows the exact same document being read by a student 3906(3) in the class of 2nd grade readers. Now the browser displays the document 3904(3) modified by methods that may make the syntax less complex and may substitute simpler words and/or phrases for harder ones. A similar example document and controls in a VDE container 3902(n) involving a 12th 3906(2)and 9th grader 3906(n) is also shown.

In other examples, the prices may be higher when users are reading text below their capabilities, they may be offered discounts for reading at a higher level, and/or they may be charged more for reading on different levels since modifying the text is a value added process, and providers of that value may wish to be compensated for their efforts.

Example: Classification May Describe Degree Of Focus Of The Content Unit Or Portion On A Topic, Or Characteristics Related To Conventional Formatting, Such As File Type

Sometimes the most interesting and/or useful content is at the intersection of various topics. Also, user often want content in a form or format that will be most useful, and most practical, to them. In the example 4000, shown in Figure 65, a matching and classification utility 900 receives from user 4002 a VDE container 4004 holding a request for documents in the class, "on economics and politics, costing less than $5.00, and in MS Word format." The matching and classification utility 900 responds in this example by providing in a VDE container 4006 at least one Uniform Resource Locator (URL) that points to the location of the document(s) on the World Wide Web.

The user 4002 in this example sends a message in a VDE container 4008 asking for the document identified in the URL. A provider sends back a VDE container 4012 with the desired document 4010 that has been classified by the matching and classification utility 900. In this example, parameter data is provided in the form of scores indicating the relative emphasis on various topic classes, including Economics (score=15), Politics (score=7), and Religion (score=2). Also indicated is the format of the content, which in this example is the desired MS Word. Also conveyed in the VDE container 4012 are a control set indicating, among other things, that the price is $2.98 and no modifications are allowed.

In other examples, the classes might have been much more narrow, for example, "Clinton," "Greenspan", Federal Reserve Policy, Interest Rates. Also, the customer might have requested only those documents for which controls could be obtained that permitted modifications and/or excerpting and/or derivative works. In another example, the matching and classification utility 900 may send one or more administrative events and/or classification and/or matching methods to the customer so that these methods could be applied by the customer. Alternatively, the customer may have send one or more methods as part of a smart object to one or more information providers in search of information meeting the desired criteria.

Example: The Atomic Aspects Can Support Automated Extraction Of Portions Of A Content Unit For Aggregation With Topically Consistent Portions And/Or Units From Other Sources

Not only may people desire specific information, but that information may come from different parts of the same object or parts of two or more objects. The matching and classification utility 900 can support the use of smart, classification based extraction and aggregation methods. as shown in example 4100, Figure 66, where two documents 4102(1,2) have been classified by the matching and classification utility 900 into "chunks" or subobjects reflecting topic classes and VDE controls have been provided for each chunk. The "chunking", classification, and control set creation may be performed and stored in a database and/or may be performed "on the fly" or as needed.

To satisfy a request for information concerning travel to and in the United Kingdom plus background information, an information provider extracts parts of each document in the desired classes and creates a new, recombinant document comprised of the subobjects and packages the new document with appropriate controls in a VDE container 4102(n). VDE controls for the subobjects may also be carried along and may be modified by the provider and/or other participants in a chain of handling and control.

The request for information may have been generated using any query and/or search method, including semantic, Boolean, heuristic, concept-based, and other approaches, and may have been generated explicitly and intentionally by a user and/or other value chain participant, or may have resulted more automatically from the analysis by a matching and classification utility 900 of usage, audit, and/or other rights management information and/or of "info exhaust," and/or of preference, demographic, and/or psychographic data and/or classes of data.

In another example, the matching and classification utility 900 may have sent administrative events and/or classification, search, and/or subobject combining methods 4106 to a provider and/or to a user for execution under the control of a local VDE node.

Example: Matching And Classification Utility 900 Supports Classification For Subsets Of Content Within A Content Unit (Nested Virtual Classifications)

Not only may the matching and classification utility 900 assist in locating whole objects, it may also assist in identifying and/or classifying any number of subobjects for a given whole. New control sets may be associated with each of these subobjects. These new control sets may differ from the control set that applies to the object as a whole. This capability allows matching and classification utility 900 and others value chain participants to locate desired classes of content that may be part of a larger object and possibly to retrieve, pay for, manage, use, or combine these parts in addition to, and/or instead of the whole object.

In example 4200, Figure 67, a VDE container 4202 created by the matching and classification utility 900 holds a text document that in this non-limiting example is the US "State of the Union Address." The matching and classification utility 900 has first classified the entire document in the class "politics." The matching and classification utility 900 has also identified various subparts or subobjects and has classified each them into different classes or categories. In this example, the different classes represent different topic categories.

A user and/or other value chain participant may request only subobjects that have been categorized in one or more desired class(es). The desired subobjects may be packaged in a VDE container 4204 along with appropriate VDE controls for both the overall, new composite object and/or for each of the desired subobjects. (The VDE controls can also be sent separately from the content subobjects.) These controls may pertain to the new whole object created from subparts selected on the basis of their membership in one or more specified class(es) and/or to the whole, new object comprised of these selected subobjects. In another example, the subobjects may be drawn from different documents sharing the same overall topic, for example, from State of the Union addresses given in different years.

In one example, any value chain participant may send distribute one or more subparts of the original object.

In another example, the matching and classification utility 900 may send one or more administrative events and/or methods 4206 to value chain participants who may execute the methods to perform the operations to identify subobjects and/or to subset the whole object in to parts based on class assignments.

Search engines can also use the subobject classifications to provide more precise results. For example, a search engine may have retrieved the State of the Union Address because the search criteria were "US politics speeches," but the whole or part of the object may also have been retrieved searching for "US politics speeches welfare" or "speeches US president defense."

Example: Matching And Classification Utility 900 Supports Classes Of Classes Based On Object Identifier Standards And/Or Other Object Metadata

Among the numerous advantages of the present inventions is the ability to create classes of classes based in part on rights management information. The feature may enhance search efficiency by enabling search engines to locate members of classes provided by any of numerous schemes for object naming and object metadata that have been proposed. For example, the IETF Uniform Resource Locator (URL), the International Standard Book Number (ISBN), International Standard Serial Number (ISSN), MARC library catalog records, and the recent proposed "Dublin Core"(Weibel, Stuart, Jean Godby, Eric Miller, and Ron Daniel, "OCLC/NCSA Metadata Workshop Report", URL http://www.oclc.org:5047

/oclc/research/conferences/metadata/ dublin_core_report.html) are non-limiting examples of prior classifications that can themselves be classified using the present inventions.

Example 4300, Figure 68A-68B, shows several objects 4304(1)-4304(n) each of which may have associated with it various metadata 4302(1)-4302(n) that locates the object in one or more classes, non-limiting examples of which may include network address (URL), price, control set information, permission strings, subject category, title, and publisher.

In example step "1," object metadata 4302 is sent to a matching and classification utility 900 which (example step "2") may create new "classes of classes" 4306. These new classes 4306 are then made available on a Web page 4308 (example step "3") to interested parties who may then search for objects according to their membership in one (or more) of these new classes of classes. In example step "4" an interested party 4320 sends a VDE container with a request to retrieve the Web page 4308 with the classes of metadata information. The Web server (in example step "5") returns a copy of the page 4312 to the interested user 4320, who (in example step "6") sends a VDE container with a query to the matching and classification utility 900 asking, in this example, for objects in new class 3 that cost less than $1.98, and that grant a "modify" permission. In example step "7," the matching and classification utility 900 returns a VDE container 4316 with list of objects that match the criteria. The matching and classification utility 900 may, in turn, provide URLs or other location information for at least one member of the desired class(es) in the list in container 4316.

Example: Matching and Classification Utility 900 Supports Electronic Gambling

Electronic gambling may be among the services that will drive Internet growth in coming years. Such services raise many questions for both providers and for users or players of the service. For example, providers want to be able create attractive, compelling entertainment experiences and in doing so, capture an important share of their intended markets. Users of these services will of course want to locate the most stimulating, entertaining, and perhaps most of all, rewarding gambling experiences.

Gambling providers may, in one example, differing classes of games, rules, payoffs, odds, and/or interfaces. The present inventions can assist players in identifying the nature of various classes and locating specific instances of one or more classes. Within a particular class of games, for example, players may be particularly interested in the odds at the game of blackjack. In one example, a player may prefer playing with a single digital deck of 52 cards and a particular number of (emulated) shuffles rather than with say four decks and more shuffles, the affect of the latter being to create a more random distribution. Smaller decks and fewer shuffles may make it easier to count cards and/or to otherwise increase the odds in favor of the player, or at least in favor of the experienced, knowledgeable player.

In example 4400, shown in Figure 69, an arbitrary number of gamblers 4402(1)-4402(n) whose usage information flows in VDE containers 4404(1)-4404(n) to a usage clearinghouse 300. The usage clearinghouse 300 sends in VDE containers 4406 at least some of this usage information to a matching and classification utility 900. In another example, the usage information may be sent directly from at least one user to the matching and classification utility 900. In this example, an arbitrary number of gambling providers 4406(1)-4406(n) may also send in VDE containers 4408(1)-4408(n) descriptive and/or usage information to the matching and classification utility 900. Based on available information from relevant sources, the matching and classification utility 900 may create one or more classes and assign one or more providers, services, and/or users to a class. These class definitions may at least in part be based on privacy-related control information.

In this one example, a gambler 4402(1) sends a VDE container 4410 with a query concerning best odds for blackjack to a matching and classification utility 900, who, in turn, sends back a VDE container 4412 with content indicating that gambling provider 2 gives the best odds in blackjack, "best" here meaning those most favorable to the player. In another example, the gambler may then contact gambling provider 2 to play, and the play may consist of a series of communications in VDE containers between the gambling provider and the gambler.

Example: Matching and classification utility 900 Supports Electronic Ticket Sales and Distribution

The performing arts, exhibitions, theaters, and conferences are some non-limiting examples of events that may require tickets for admission. Electronic ticket agencies on the Internet and other electronic arenas provide a connection between the consumer and producers of the event. Consumers may want to know such information as the nature of the event, what classes of tickets exist for a given event and/or class of events, the price for different classes of tickets to an event, the availability of different classes of tickets to different classes of events, and similar information.

In the example 4500, shown in Figure 70, an arbitrary number of users 4504(1)-4504(n) whose usage information is sent in VDE containers 4508 to a usage clearinghouse 300 who, in turn, may send at least some of this usage information in at least one VDE container 4526 to a matching and classification utility 900. The usage information may reflect past ticket purchases, prices, seating preferences, preferred payment methods, preferred theaters and other venues, and other user preference and historical information.

Various ticket agencies 4506(1)-4506(n) may send information about specific events 4512 (1)-4512(n) and/or information about agency services 4514(1)-4514(n) to the matching and classification utility 900. In another example, an event promoter may send event information directly to the matching and classification utility 900.

In one example, a user wishes to find four seats for a particular concert or class of concerts and/or other events whose cost is not more than $25.00. The user sends a VDE container with a request for information on who can supply the desired tickets to the desired events at the requested price. In turn, the matching and classification utility 900 returns a VDE container indicating that tick agency 2 can provide the tickets.

In this example, user 2 sends a VDE container with a purchase request to ticket agency 2. The purchase request may specify not only the specific event, desired pricing, and class of tickets, seat location, for example, but payment method as well, MasterCard for example. The ticket agency, in turn, may return a VDE container with confirmation of the ticket purchase at a given price, location, date, event, and/or using a particular payment method.

In another example, the tickets may be digital and may have associated with them one or more "seals", digital signatures, and/or certificates indicating the authenticity and/or integrity of the digital tickets.

* * * *

While the inventions have been described in connection with what is presently considered to be the most practical and preferred embodiments, the inventions are not to be limited to the disclosed embodiments but, on the contrary, is intended to cover various modifications and equivalent arrangements included within the spirit and scope of the appended claims.

WE CLAIM:

1. A method including:

(a) determining at least one class, class hierarchy, classification scheme, category or category scheme;

(b) assigning cases, persons, and/or things to said determined class, class hierarchy, classification scheme, category or category scheme; and

(c) selecting and/or matching cases, persons, and/or things based at least in part on said class, class hierarchy, classification scheme, category or category scheme and/or said assignment,

wherein at least one of said steps (a)-(c) includes the step of using at least some rights management information.

2. A method as in claim 1 wherein said using step includes using at least one control set.

3. A method as in claim 1 wherein said using step includes using at least some information for controlling use of digital information.

4. A method as in claim 1 wherein said using step includes using at least some information for controlling at least one transaction.

5. A method as in claim 1 wherein said using step includes using at least some information for controlling at least one event.

6. A method as in claim 1 wherein said using step includes using at least some information for controlling at least one consequence of digital information use.

7. A method as in claim 1 wherein said using step includes using at least some information for controlling at least one consequence of at least one event.

8. A method as in claim 1 wherein said using step includes the step of using at least some information for controlling at least one consequence of at least one transaction.

9. A method as in claim 1 wherein said using step includes using at least some information outputted by a rights management process.

10. A method as in claim 1 further including the step of outputting at least some rights management information.

11. A method as in claim 1 wherein at least one of steps (a)-(c) includes using at least one secure container.

12. A method as in claim 1 wherein at least one of steps (a)-(c) includes using at least one protected processing environment.

13. A method as in claim 1 further including the step of using at least one of the techniques set forth at pages 60-82 of this specification.

14. A method as in claim 1 wherein said using step includes using at least one or more rules and/or their consequences.

15. A method as in claim 1 wherein at least one of steps (a) and (b) includes at least one of the following steps:

(a) using at least one statistical technique identifying at least one cluster of cases sharing similar profiles and/or features;

(b) using numerical taxonomy;

(c) using at least one of cluster analysis, factor analysis, components analysis, and other similar data reduction/classification technique;

(d) using at least one pattern classification technique, including components analysis and neural approaches;

(e) using at least one statistical technique that identifies at least one underlying dimension of qualities, traits, features, and/or characteristics, and assigning parameter data indicating the extent to which a given case has, possesses, and/or may be characterized by the underlying dimension, factor, class, and/or result in the definition of at least one class and/or the assignment of at least one case to at least one class;

(f) using at least one statistical method employing fuzzy logic and/or fuzzy measurement and/or whose assignment to at least one class entails probabilities different from 1 or zero;

(g) using a Baysian statistical classification techniques that uses an estimate of prior probabilities in determining class definitions and/or the assignment of at least one case to at least one class;

(h) using at least one statistical and/or graphical classification and/or data reduction method that uses rotation of reference axes, regardless of whether orthogonal or oblique rotations are used;

(i) using at least one statistical method for two and three way multidimensional scaling; and

(j) using at least one knowledge based approach to classification.

16. A system including:

an automatic class generator that generates at least one class, class hierarchy, classification scheme, category or category scheme;

an automatic class assigner that assigns cases, persons and/or things to said determined class, class hierarchy, classification scheme, category or category scheme; and

at least one further component for automatically searching, selecting and/or matching cases, persons, and/or things based at least in part on said class, class hierarchy, classification scheme, category or category scheme and/or said assignment,

wherein said system uses at least some rights management information.

17. A system including:

first means for determining at least one class, class hierarchy, classification scheme, category or category scheme;

second means for assigning cases, persons, and/or things to said determined class, class hierarchy, classification scheme, category or category scheme; and

third means for selecting and/or matching cases, persons, and/or things based at least in part on said class, class hierarchy, classification scheme, category or category scheme and/or said assignment,

wherein at least one of said first, second and third means uses at least some rights management information.

18. A Commerce Utility System providing a secure execution space, the Commerce Utility System performing at least one component based service function including at least one secure component for execution within the secure execution space, the Commerce Utility System including a communications facility permitting communication of secure control information with at least one electronic community participant,

wherein said component based service function uses at least one class based at least in part on rights management information.

19. A Commerce Utility System as in claim 18 wherein the component based service function assigns at least one member to at

least one class based at least in part on some rights management information.

20. A Commerce Utility System as in claim 18 wherein the component based service function matches persons and/or things based at least in part on at least some rights management information.

21. A Commerce Utility System as in claim 18 wherein the component based service function selects persons and/or things based at least in part on at least some rights management information.

22. A Commerce Utility System as in claim 18 wherein the component based service function narrowcasts information to recipients based at least in part on at least some rights management information.

23. A system or method including:

a computer network and

a control arrangement within the network that determines and/or uses at least one of the following through use of rights management information:

(a) class hierarchy,

(b) class structure,

(c) classification scheme,

(d) category, and

(e) category scheme.

24. A class-based system including at least one computer that processes digital information, said system including at least one element that uses at least some rights management information.

25. A method of operating a class-based system including at least one computer that processes digital information, said method including the step of using at least some rights management information.

26. A system for assigning at least one thing or person to at least one class including at least one computer that processes digital information, said system including at least one element that uses at least some rights management data in making said assignment.

27. A system for making and/or using at least one class-based assignment including at least one computer that processes digital information, said system including at least one element that uses at least some rights management information.

28. A system for clearing at least one transaction including at least one computer that processes digital information, said system including at least one element that uses at least one class defined, assigned, selected, and/or matched based at least in part on rights management information.

29. A method for authorizing at least one computer and/or computer user including the step of using at least one class defined,

assigned, selected, and/or matched based at least in part on rights management information.

30. A method for authorizing at least one electronic transaction including the step of using at least one class defined, assigned, selected, and/or matched based at least in part on rights management information.

31. A method for initiating and/or performing at least one at least in part secure electronic transaction including the step of using class related information defined, assigned, selected, and/or matched based at least in part on rights management information.

32. An information processing method including the steps of:

securely charging a fee; and

conditioning said charging step at least in part on at least one class defined, assigned, selected, and/or matched based at least in part on rights management information.

33. A method for securely exchanging digital information including the step of at least in part defining, assigning, selecting, and/or matching at least one class based at least in part on rights management information.

34. A method for performing at least one rights operating system based transaction including the step of defining, assigning,

selecting, and/or matching at least one class based at least in part on rights management information.

35. A method for performing at least one protected processing environment operation including the step of defining, assigning, selecting, and/or matching at least one class based at least in part on rights management information.

36. A method of pushing information including the steps of classifying recipients and/or information to be sent to said recipients based at least in part on rights management information, and selecting said information to distribute to said recipients based at least in part on said classifying.

37. A method of pushing information including the steps of classifying recipients and/or information to be sent to said recipients based at least in part on rights management information, and matching at least a portion of said information with at least one class of said recipients based at least in part on said classifying.

38. A method of pushing information as in claim 37 further including the step of creating a classification scheme and/or hierarchy using at least some rights information.

39. A method of pushing information as in claim 37 further including the step of assigning at least some information and/or at least one recipient to a class or category, said assignment based at least in part on rights management information.

40. A subject switch for matching subscribers and/or recipients desiring information in one or more classes with one or more sources of information, wherein the subject switch matches at least one subscriber and/or participant with at least one information source on a mapping based at least in part on rights management information.

41. A subject switch as in claim 40 wherein said information source:

selects at least some information, said selection based on at least one class, and wherein said assignment of said at least some information to said at least one class is based at least in part on rights management information; and

sends at least some said selected information to said subscriber in accordance with said subscriber's subscribing to said class of information.

42. A subject switch as in claim 40 wherein at least one of said subject switch, said subscriber and/or participant and said information source includes at least one computer providing a protected processing environment.

43. A subject switch as in claim 40 wherein at least one subscriber and/or participant uses rights management information at least in part to persistently subscribe to at least some information provided by at least one information source.

44. A subject switch as in claim 40 wherein the subject switch includes means for using at least one class definition for said mapping.

45. A subject switch as in claim 40 wherein the subject switch includes means for responding to a subscriber and/or participant request by providing information indicating information sources in at least one specified or desired class.

46. A subject switch as in claim 40 further including a messaging service for use by at least two of said subject switch, said subscriber and/or participant and said information source and/or participant to communicate electronically.

47. A subject switch as in claim 46 wherein said electronic communications uses at least one secure container.

48. A subject switch as in claim 40 wherein at least one of said subject switch, subscriber, or information source uses at least one control set associated with at least some information received by at least one subscriber.

49. A digital narrowcasting arrangement comprising:

a computer; and

at least one classifying element used to select content to narrowcast to recipients based at least in part on rights management information.

50. A digital narrowcasting arrangement as in claim 49 wherein the classifying element classifies at least one of (a) a recipient, and (b) content, based at least in part on rights management information.

51. A digital narrowcasting arrangement as in claim 49 wherein said classifying element defines at least one class using at least some rights management information.

52. A digital narrowcasting arrangement as in claim 49 wherein the classifying element assigns at least some content to at least one class, said assignment based on at least some rights management information.

53. A digital narrowcasting arrangement as in claim 49 wherein the classifying element defines at least one class based at least in part on content selections previously made by the recipients and/or profiles generated based at least in part on recipient input.

54. A digital narrowcasting arrangement as in claim 49 wherein the classifying element sends a content request including classification data and destination information to at least one provider.

55. An information distribution system including: a computer network; and a selection arrangement that selects information for use by individual recipients using classes based at least in part on rights management information.

56. An information distribution system as in claim 55 wherein the system further includes a classifying element that determines at least one class of content and/or service of interest to at least one recipient.

57. An information distribution system as in claim 56 wherein said classifying element defines at least one class using at least some rights management information.

58. An information distribution system as in claim 56 wherein said classifying element assigns at least some content to at least one class, said assignment based on at least some rights management information.

59. An information distribution system as in claim 55 wherein the system includes means for allowing the user to choose to receive the selected information.

60. An enterprise information system including a computer system for classifying employees, said system including at least one rights management component that distributes information to the employees based at least in part on employee classification.

61. An enterprise information system as in claim 60 wherein the computer matches the information to employees based at least in part on the employee classification.

62. An enterprise information system as in claim 60 wherein the employee classification is used to gather information for

employees without revealing substantial information concerning individual employees.

63. A method for conducting a chain of handling and/or control including the steps of allowing plural parties to contribute rules and/or consequences, and performing at least one classification based at least in part on said rules and/or consequences.

64. A method as in claim 63 wherein at least some of said contributed rules and/or consequences are class based.

65. A method as in claim 63 wherein at least one of said parties modifies at least one of said rules and/or consequences based at least in part on class.

66. A method as in claim 63 including the step of generating class assignments based at least in part on said rules and/or consequences, and sending said class assignments to at least one clearinghouse.

67. A method as in claim 63 including the step of classifying said rules and/or consequences to provide at least one class, and fulfilling at least one request by selecting based on said class.

68. A directory services system for classifying confidential information, the system including:

a communications component that receives directory requests; and

a response component that uses said classification to respond to directory requests while preserving confidentiality of said confidential information.

69. A directory services system as in claim 68 wherein said response component uses at least one classification process to classify items in a directory, and uses results of the classification process, at least in part, to respond to directory requests.

70. A directory services system as in claim 68 wherein said response component sends information to destinations revealed by the results of the classification process without revealing at least some information concerning said destinations to the information source.

71. A microsegmented merchandising technique including the steps of performing classification based at least in part on usage data and/or lifestyle profiles, and distributing offers for products and/or services based at least in part on the classification.

72. A microsegmented merchandising technique as in claim 71 wherein the performing step includes defining at least one class hierarchy based at least in part on rights management information.

73. A microsegmented merchandising technique as in claim 71 further including the step of combining plural offers for different products and/or services based at least in part on said classification.

74. A trading network including:

a communications element for communicating digital signals; and means for matching value chain participants through a classification based at least in part on rights management information.

75. A trading network as in claim 74 further including means for defining at least one class hierarchy based at least in part on rights management information.

76. A trading network as in claim 74 further including means for determining class membership based at least in part on action and/or information provided by at least one value chain participant.

77. A trading network as in claim 74 wherein said matching means includes means for at least in part performing at least one electronic negotiation.

78. A securities trading method including the step of performing a classification process at least in part using at least one rights management element, and using the classification process to select securities for trade.

79. A securities trading method as in claim 78 wherein said classification process includes defining at least one class hierarchy based at least in part on rights management information.

80. A currency/debt trading system including:

a currency or debt trading computer; and

an arrangement coupled to said computer that performs at least one classification process based at least in part on rights management information.

81. A currency/debt trading system as in claim 80 wherein said arrangement includes means for defining at least one class hierarchy based at least in part on rights management information.

82. A currency/debt trading system as in claim 80 wherein the arrangement uses classification to maximize return or minimize loss.

83. A financial institution selection system including a computer that classifies financial institutions based at least in part on rights management information.

84. A software distribution method including the steps of generating class information based at least in part on rights management information, and selecting software to be distributed and/or recipients who are to receive distributed software based at least in part on class information.

85. A software distribution method as in claim 84 wherein said generating step includes defining a class hierarchy using at least some rights management information.

86. A software distribution method as in claim 84 wherein the selecting step includes selecting software to be distributed by classifying the software based at least in part on rights management information associated with the software.

87. A software distribution mehtod as in claim 80 wherein the selecting step includes selecting recipients to receive software based at least in part on usage information provided by a rights management process.

88. A classification technique including the step of authenticating class membership based at least in part on digital credentials and/or certificates.

89. A classification technique as in claim 88 wherein said digital credentials are digital certificates.

90. A classification technique as in claim 88 wherein said digital credentials are digital membership cards.

91. A classification technique as in claim 88 further including the step of deciding class membership based at least in part on rights management information.

92. A classification technique as in claim 88 further including the step of classifying at least one of users, nodes, devices, networks, servers, clients and services based at least in part on rights management information.

93. A classification technique as in claim 88 further including the step of conditioning at least one rights management process at least in part on authenticated class membership.

94. A computer system including:

a first arrangement that generates class-based controls to participants based at least in part on class and/or class-based assignments; and

a second arrangement that allows participants to interact with information and/or one another at least in part using said class-based controls.

95. A computer system as in claim 94 further including means for using said class-based controls to limit participants' access to information and/or services based on participants' classes.

96. A health care computer system including an arrangement for issuing health care workers, administrators and insurers class-based digital credentials and/or certificates, wherein the digital information sent to said health care workers and administrators includes class-based controls that condition use and/or access to information based at least in part on said class-based digital credentials and/or certificates.

97. A health care computer system as in claim 96 further including means for allowing said health care workers, administrators and insurers sharing a common object subject to class-based controls to have access to different portions of the object based at least in part on said class-based controls.

98. A work process automation system including a matching and/or classification computer that matches tasks to resources based at least in part on assigning classifying the tasks and/or the resources to at least one class.

99. A work process automation system as in claim 98 wherein said matching and/or classification computer includes means for defining at least one class hierarchy based at least in part on rights management information.

100. A work process automation system as in claim 98 wherein said matching and/or classification computer includes means for matching based at least in part on rights management information.

101. An automatic governmental and/or societal rights supporting system including a matching and/or classification computing element that assigns and/or classifies entities to at least one class based at least in part on rights management information.

102. An automatic governmental and/or societal rights supporting system as in claim 101 wherein the matching and/or classification computing element includes means for defining a class hierarchy based at least in part on rights management information.

103. An automatic governmental and/or societal rights supporting system as in claim 101 wherein the matching and/or classification computing element includes means for classifying entities based on at least one of the following:

tax status;
right to receive certain information;
right to engage in certain transactions; and
jurisdiction.

104. An automatic taxing authority computer including means for issuing tax class control sets based at least in part on tax-based class definitions, and means for using said tax control sets at least in part to collect and/or enforce taxation.

105. A method for adaptively presenting information differently to different participants, including associating said participants with classes, and controlling presentation based at least in part on class-based control sets included within the information.

106. A method as in claim 105 further including using said class-based control sets to match participants with different portions of said information.

107. A method as in claim 105 further including using said class-based control sets to change the form in which information is presented based at least in part on said classes.

108. A method as in claim 105 further including the step of operating said class-based control sets based at least in part on metadata associated with different portions of said information.

109. A method as in claim 105 further including selecting said class-based control sets between different images for

presentation based at least in part on one or more classes associated with a participant.

110. A method as in claim 105 further including using said class-based control sets to emphasize certain portions of said information over other portions in said presentation based at least in part on one or more classes associated with a participant.

111. A method as in claim 105 further including using at least one computer having a protected processing environment.

112. A method for adaptively presenting information differently to different participants including:

classifying the different participants based on capability; and

using class-based control sets associated with said information to change the difficulty of the presentation based at least in part on said classification.

113. A method as in claim 112 wherein the different recipients are classified based on grade level.

114. A method as in claim 112 including the step of changing the vocabulary and/or syntactical complexity of the presentation based at least in part on said classification.

115. A method as in claim 112 further including the step of using said class-based control sets to ensure that in at least some cases, recipients in different classes pay different levels of compensation for said presentation.

116. A method for adaptively presenting information differently to different participants including:

classifying different participants based on capability, and

using class-based control sets associated with said information to change the language of the presentation based at least in part on said classification.

117. An information searching mechanism including a matching computer element that classifies information based at least in part on rights management information, said computing element including means responsive to user requests to search for information based at least in part on said classification.

118. An information searching mechanism as in claim 117 wherein said matching computer element further includes means for assigning information to classes based at least in part on rights management information.

119. An information searching mechanism as in claim 117 wherein said matching computer element includes means for scoring information based at least in part on user indicated parameters.

120. An information searching mechanism as in claim 117 wherein said matching computer element includes means for responding to at least some user requests by providing Universal Resource Locator designations of where information can be found.

121. An information handling method including the step of using class-based controls to control support extraction and/or aggregation of information.

122. An information handling method as in claim 121further including using a computing element to extract information from plural objects based at least in part on class-based criteria.

123. An information handling method as in claim 121 further including using a computing element to aggregate information based at least in part on class-based criteria.

124. An information handling method as in claim 121 further including using said class-based controls to represent nested or multi-level classifications.

125. An information classification method including the step of generating at least one class hierarchy from other plural classification hierarchies based at least in part on rights management information and/or class-based rights management information based at least in part on classification metadata.

126. An information classification method as in claim 125 further including basing said other plural classification hierarchies at least in part on object metadata.

127. An information classification method as in claim 125 further including specifying said classification object metadata

specified classifications based on at least one of location, name, prices, permissions, ISSN, title, author, publisher and/or date.

128. An information classification method as in claim 125 further including generating said class-based rights management information by classifying classes.

129. An electronic gambling system including a computer that matches gamblers with plural gambling providers based at least in part through classifying the gambling providers using rights management information.

130. An electronic gambling system as in claim 129 wherein the computer includes means for classifying the gamblers based at least in part on rights management information.

131. An electronic gambling system as in claim 129 wherein the computer includes at least one protected processing environment.

132. An electronic gambling system as in claim 129 wherein the computer uses at least one control set to classify, select and/or match at least one of said gambling providers, and/or gamblers.

133. An electronic ticketing system including a computer that matches recipients with tickets to events through classifying said recipients, said system including a computer that matches tickets and/or said events based at least in part on rights management information.

134. An electronic ticketing system as in claim 133 wherein a recipient provides a request containing event and rights management criteria, and the computer matches the recipient with a provider based at least in part on said classifying process.

135. An electronic ticketing system as in claim 133 wherein the rights management information includes method of payment information.

1/96

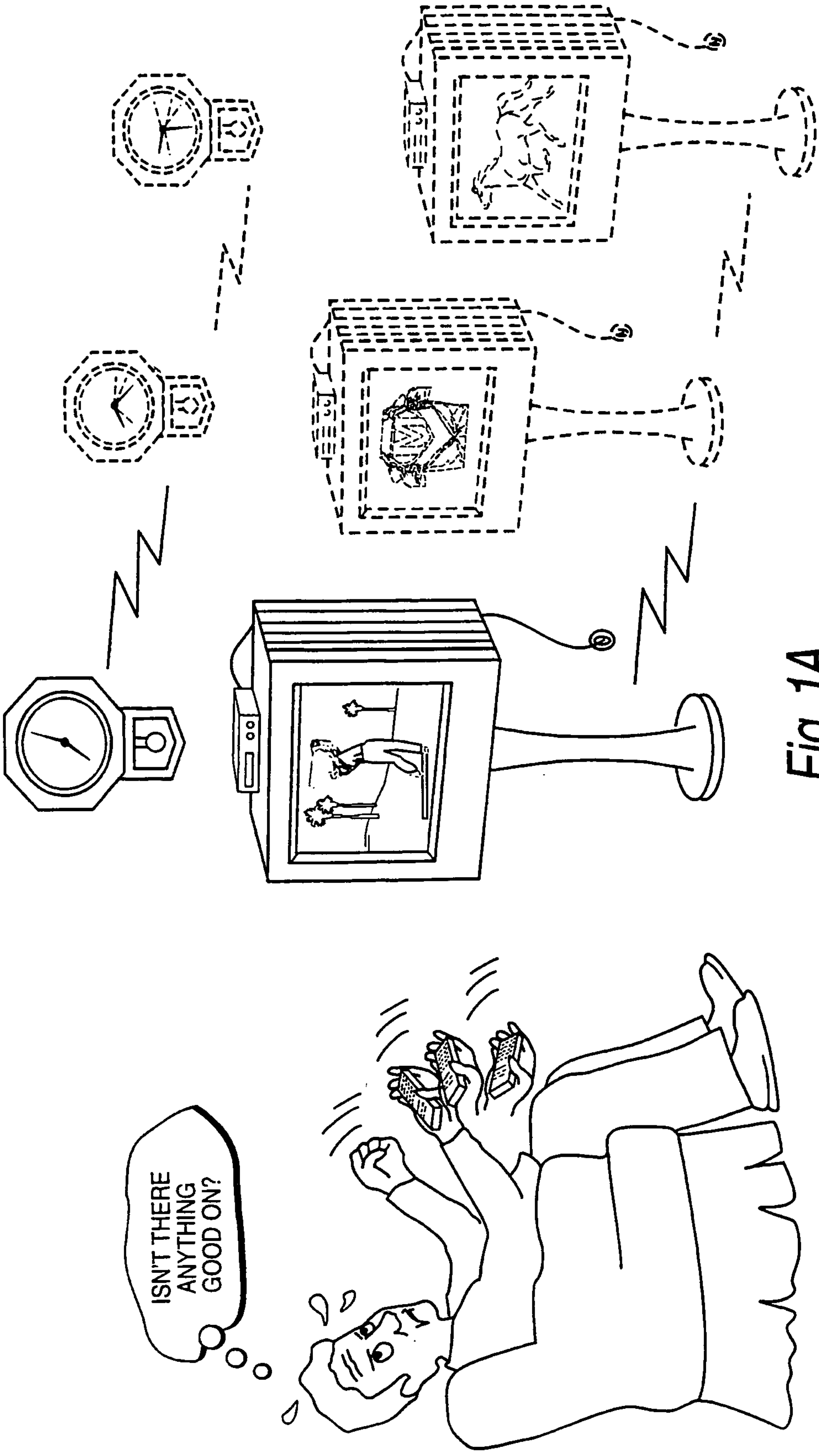

ISN'T THERE ANYTHING GOOD ON?

Fig. 1A
(Prior Art)

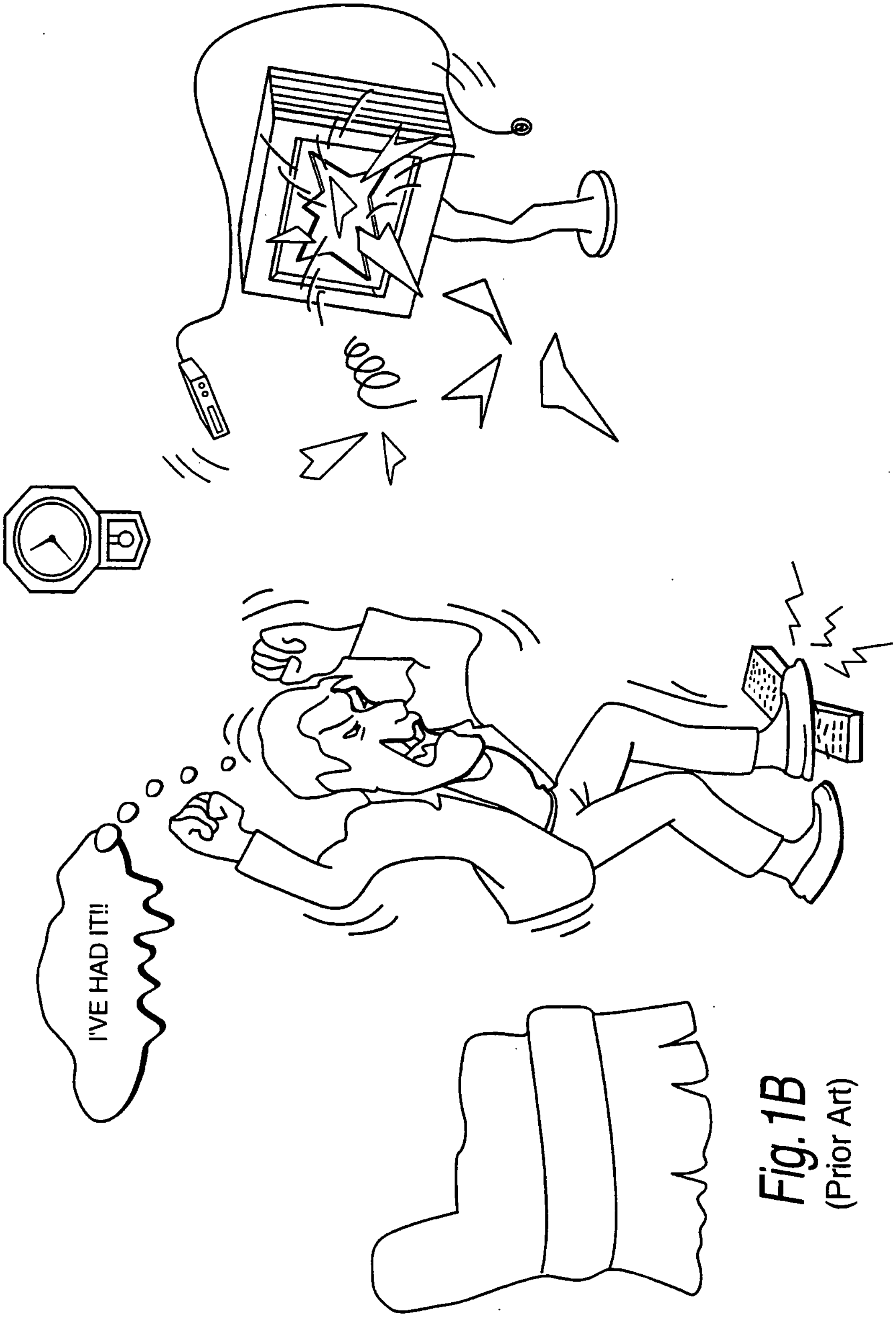


Fig.1B
(Prior Art)

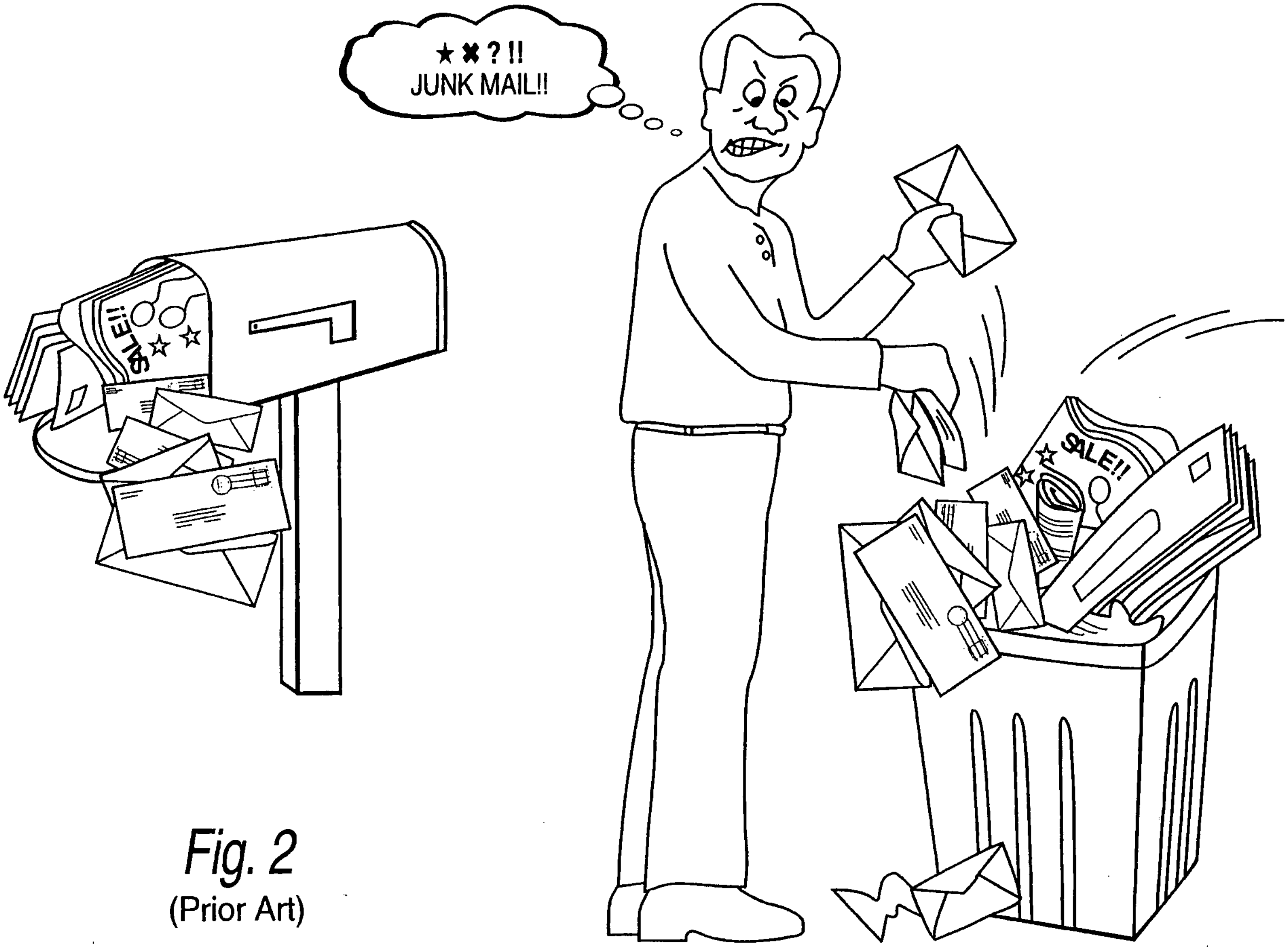

★ ✖ ? !!
JUNK MAIL!!
SALE!!
SALE!!

Fig. 2
(Prior Art)

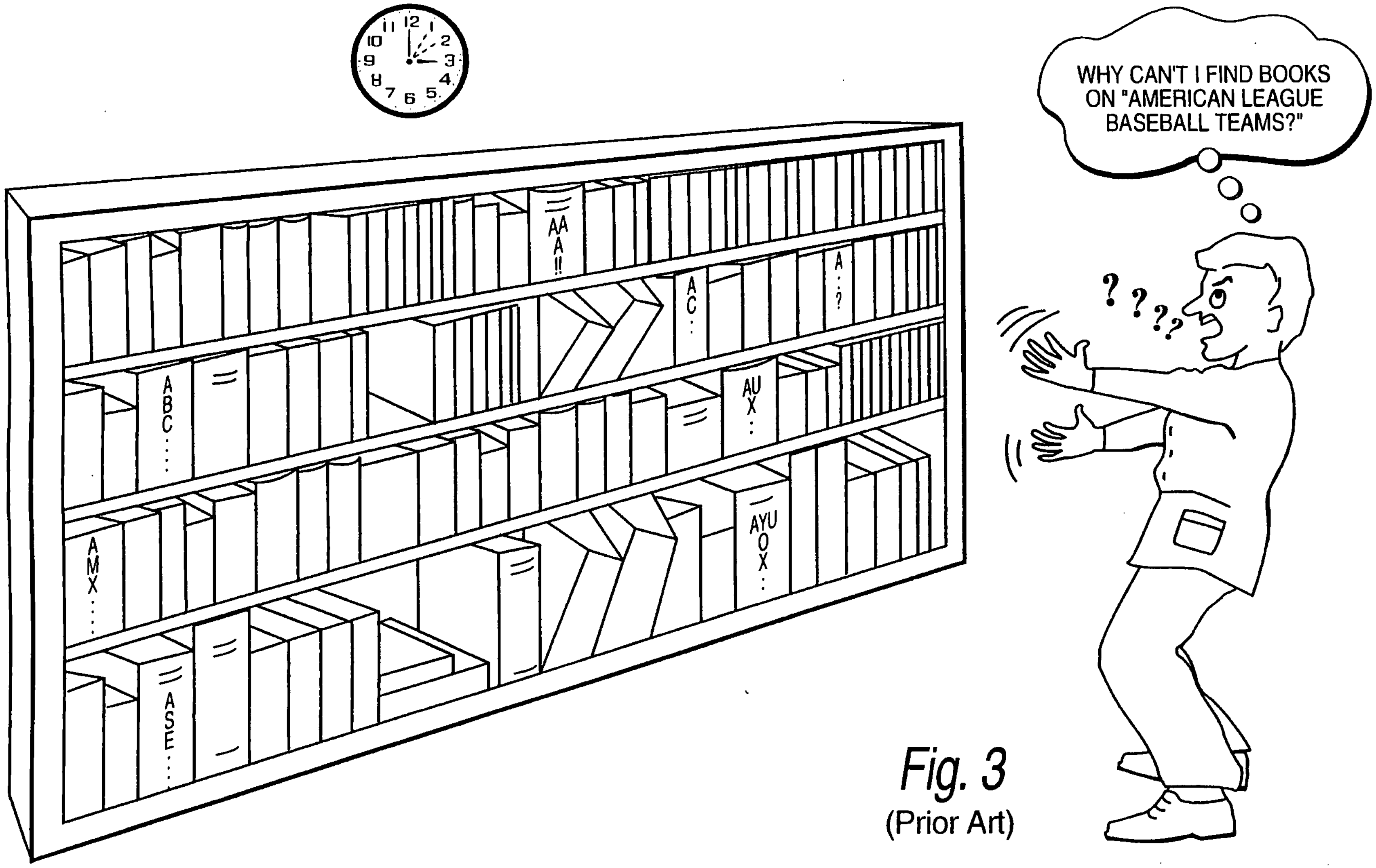

WHY CAN'T I FIND BOOKS ON "AMERICAN LEAGUE BASEBALL TEAMS?"
? ? ? ?
AA A !!
A C
A ?
A B C
AU X
A M X
AYU O X
A S E
12 1 2 3 4 5 6 7 8 9 10 11

Fig. 3
(Prior Art)

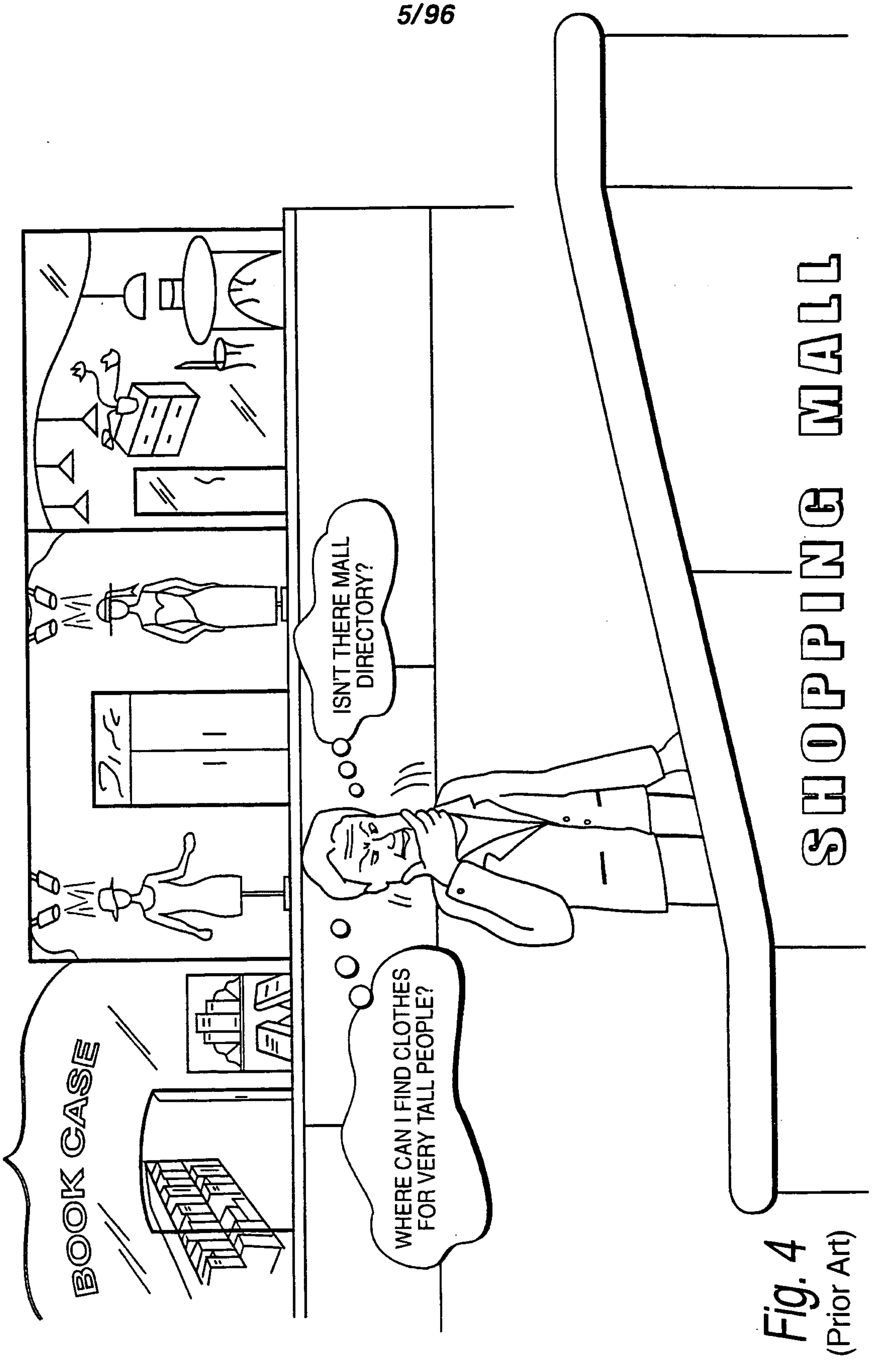

BOOK CASE
WHERE CAN I FIND CLOTHES FOR VERY TALL PEOPLE?
ISN'T THERE MALL DIRECTORY?
SHOPPING MALL

Fig. 4
(Prior Art)

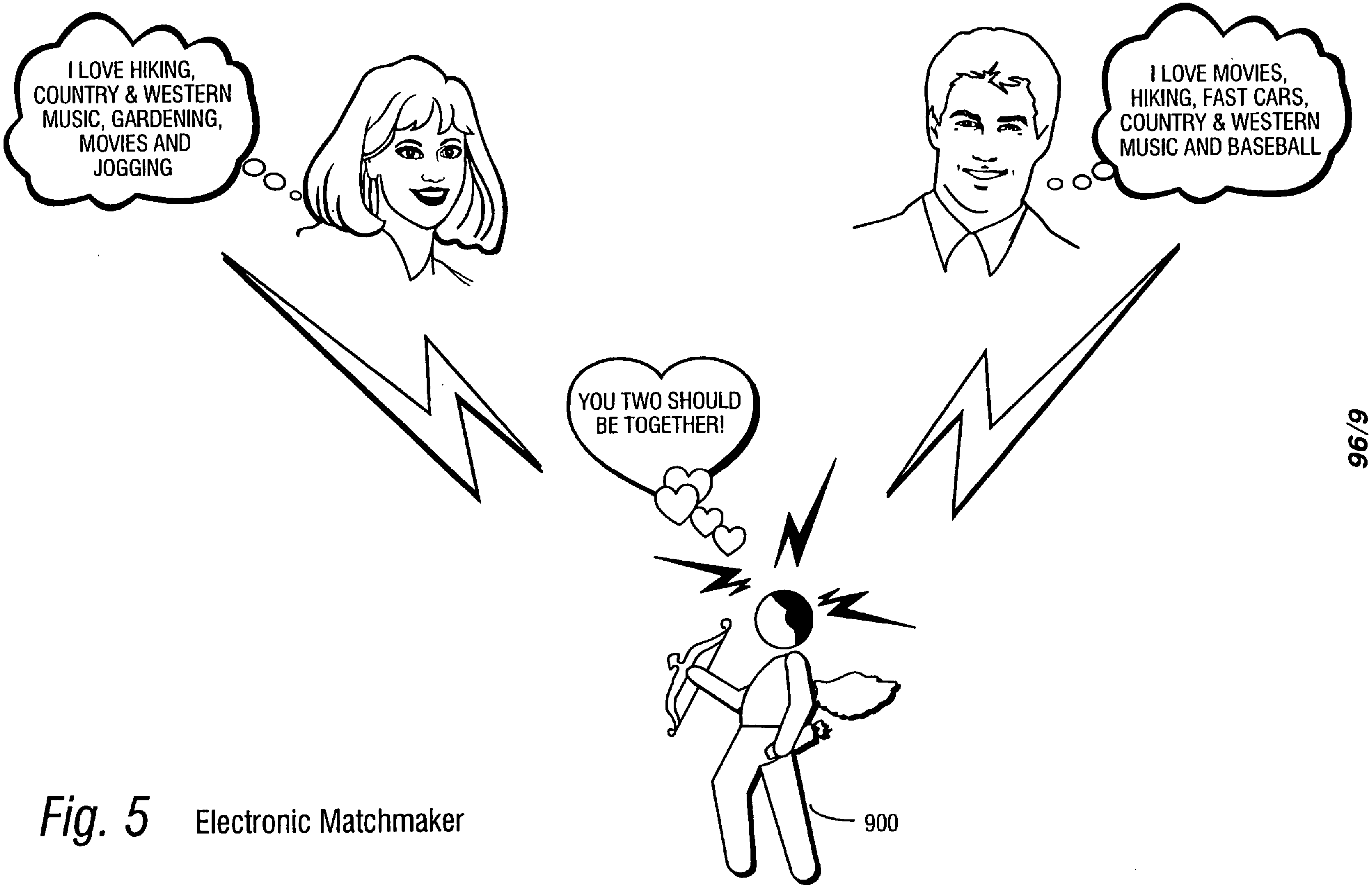

I LOVE HIKING, COUNTRY & WESTERN MUSIC, GARDENING, MOVIES AND JOGGING
I LOVE MOVIES, HIKING, FAST CARS, COUNTRY & WESTERN MUSIC AND BASEBALL
YOU TWO SHOULD BE TOGETHER!
900

Fig. 5 Electronic Matchmaker

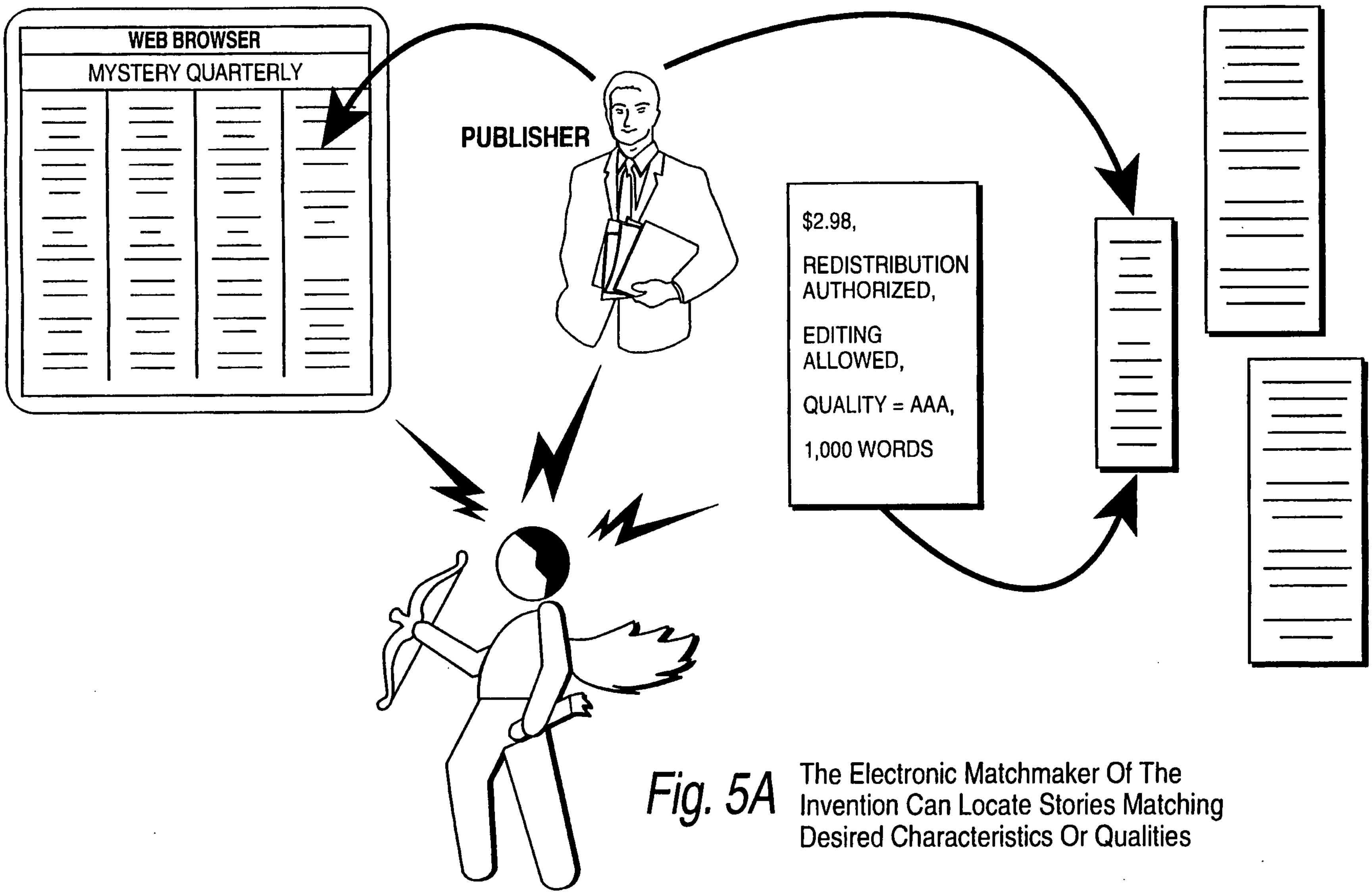

WEB BROWSER
MYSTERY QUARTERLY
PUBLISHER
$2.98,
REDISTRIBUTION AUTHORIZED,
EDITING ALLOWED,
QUALITY = AAA,
1,000 WORDS

Fig. 5A The Electronic Matchmaker Of The Invention Can Locate Stories Matching Desired Characteristics Or Qualities

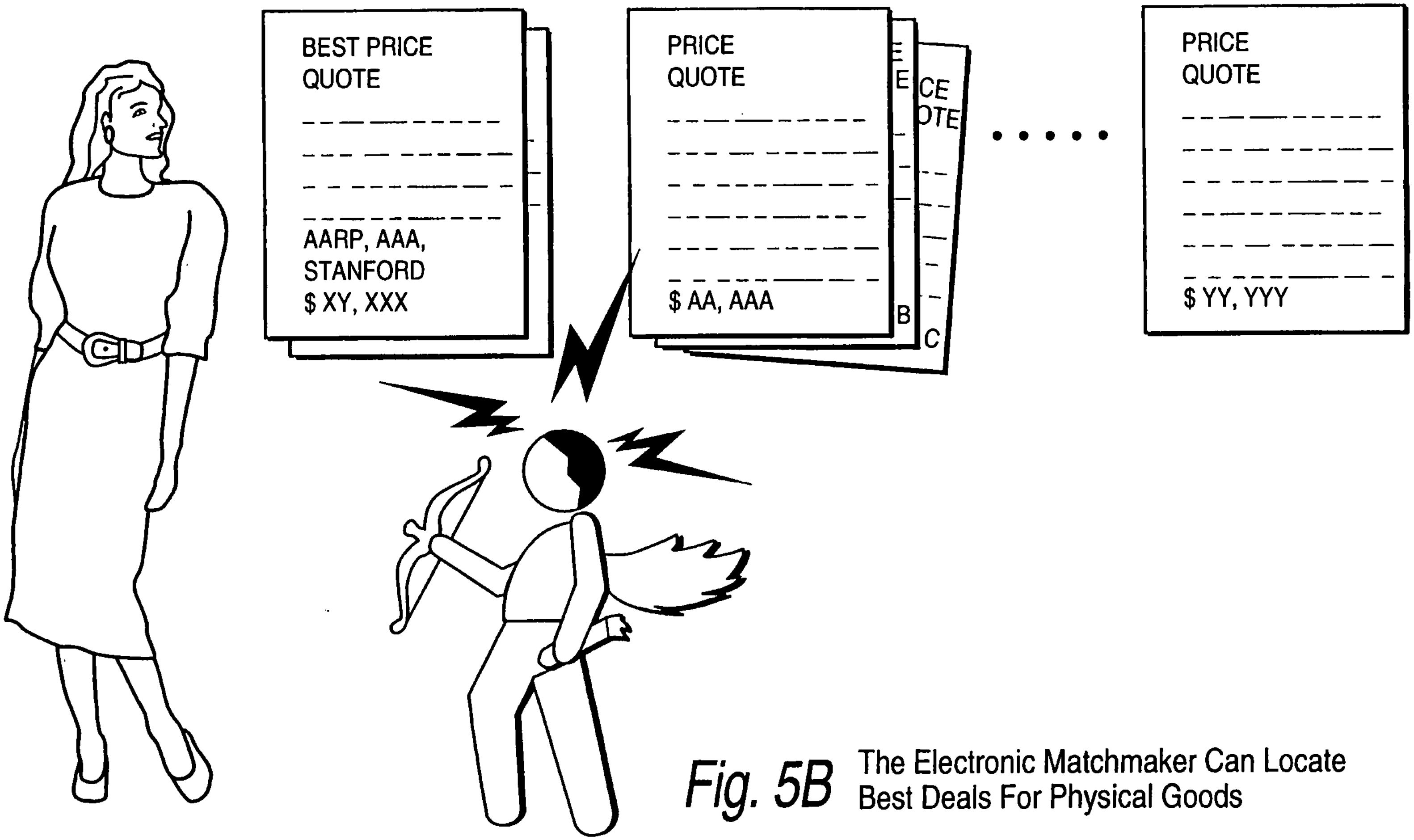

BEST PRICE
QUOTE
AARP, AAA,
STANFORD
$ XY, XXX
PRICE
QUOTE
$ AA, AAA
E
E
CE
OTE
B
C
PRICE
QUOTE
$ YY, YYY

Fig. 5B The Electronic Matchmaker Can Locate Best Deals For Physical Goods

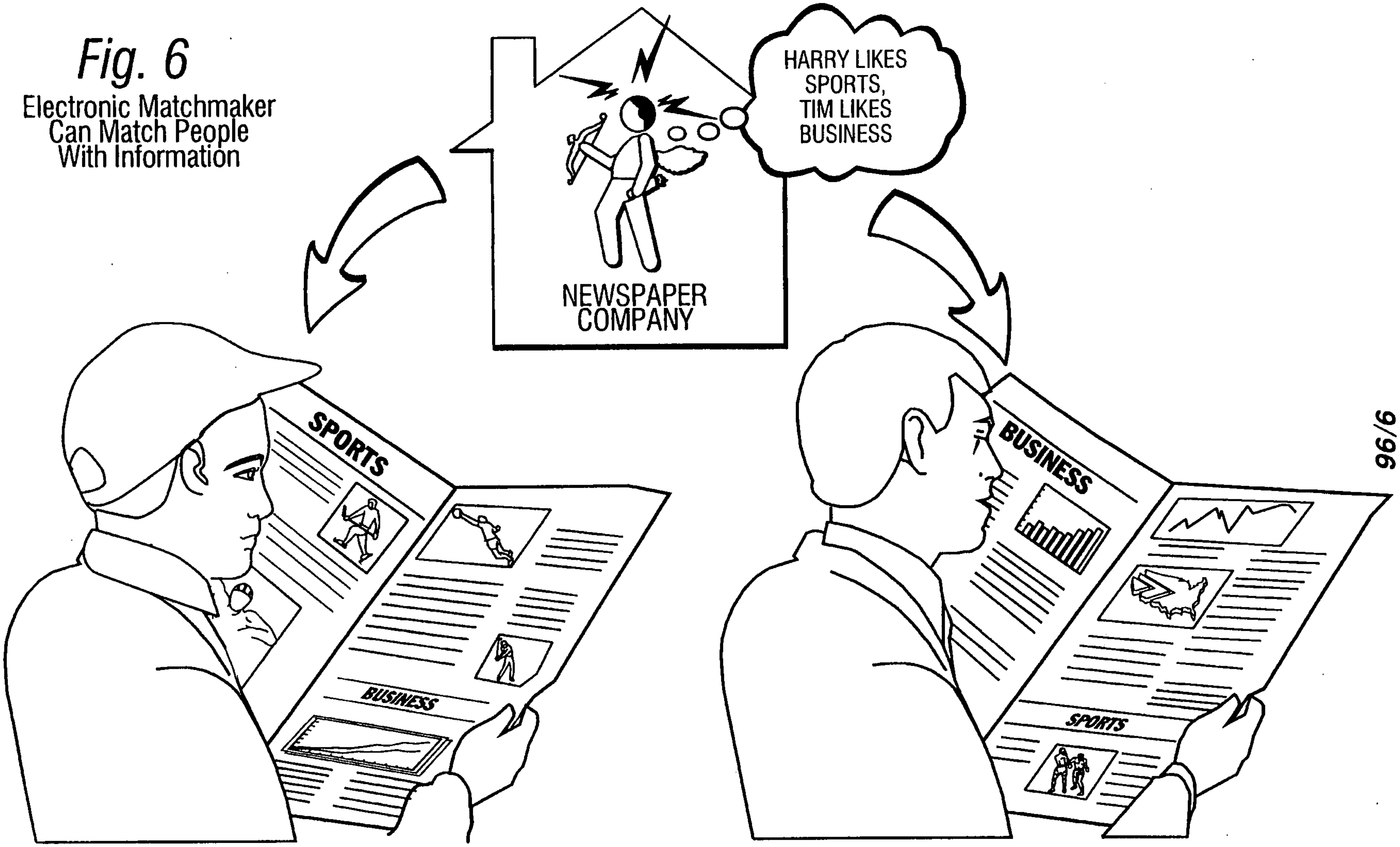

NEWSPAPER COMPANY
HARRY LIKES SPORTS, TIM LIKES BUSINESS
SPORTS
BUSINESS
BUSINESS
SPORTS

Fig. 6

Electronic Matchmaker Can Match People With Information

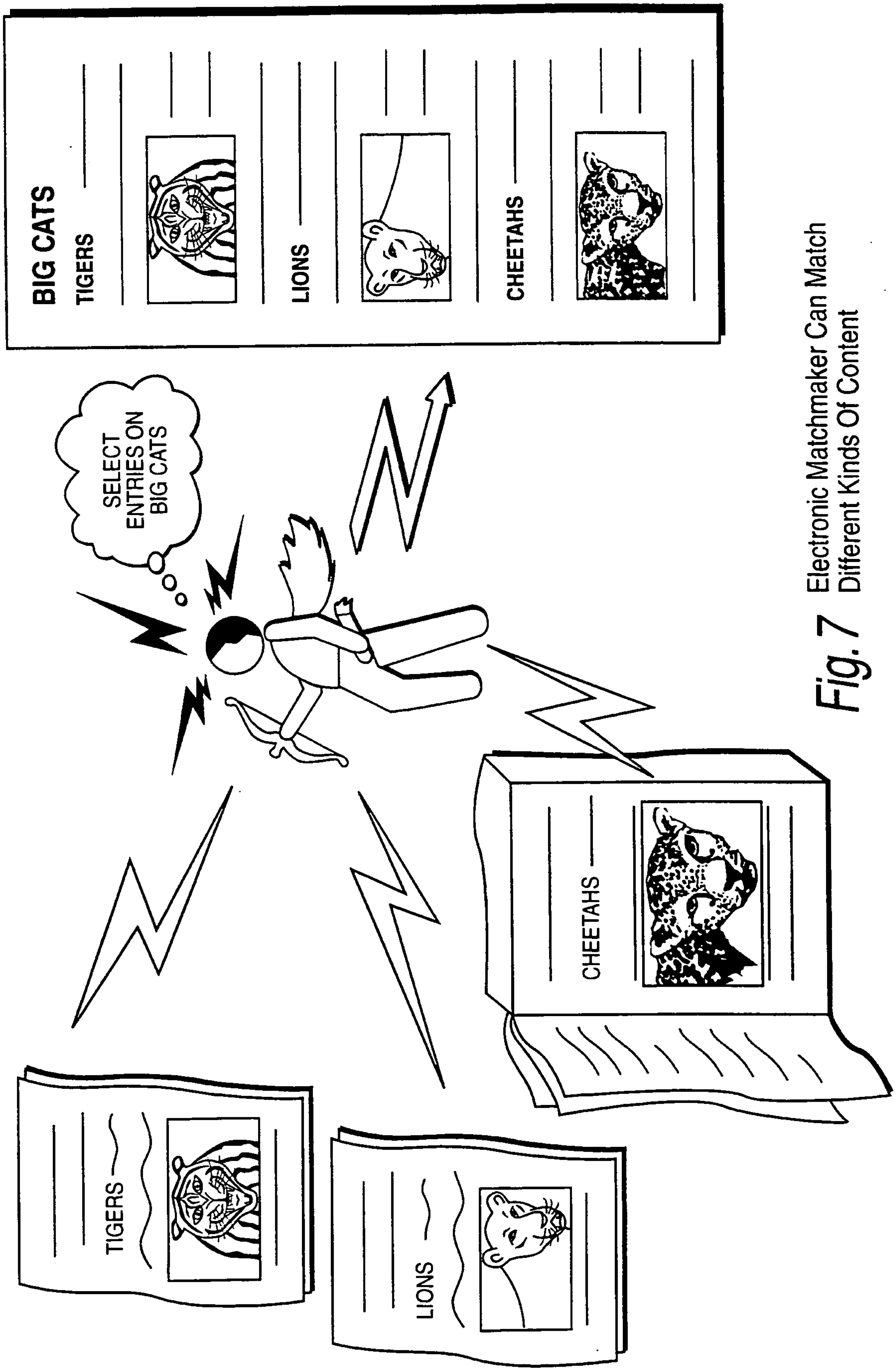

BIG CATS
TIGERS
LIONS
CHEETAHS
SELECT ENTRIES ON BIG CATS
CHEETAHS
TIGERS
LIONS

Fig. 7 Electronic Matchmaker Can Match Different Kinds Of Content

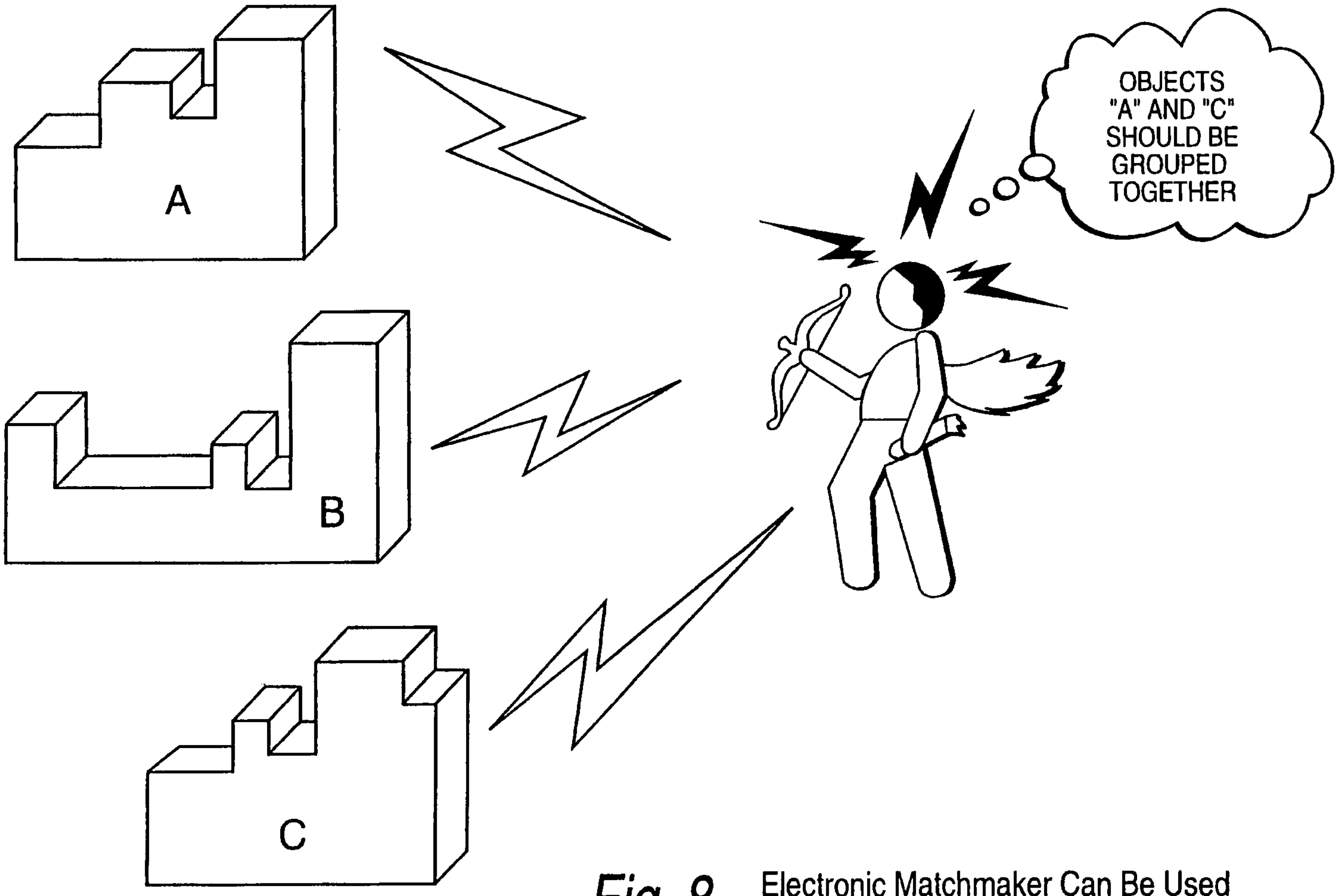

OBJECTS
"A" AND "C"
SHOULD BE
GROUPED
TOGETHER
A
B
C

Fig. 8 Electronic Matchmaker Can Be Used For Matching Any Kinds of Things

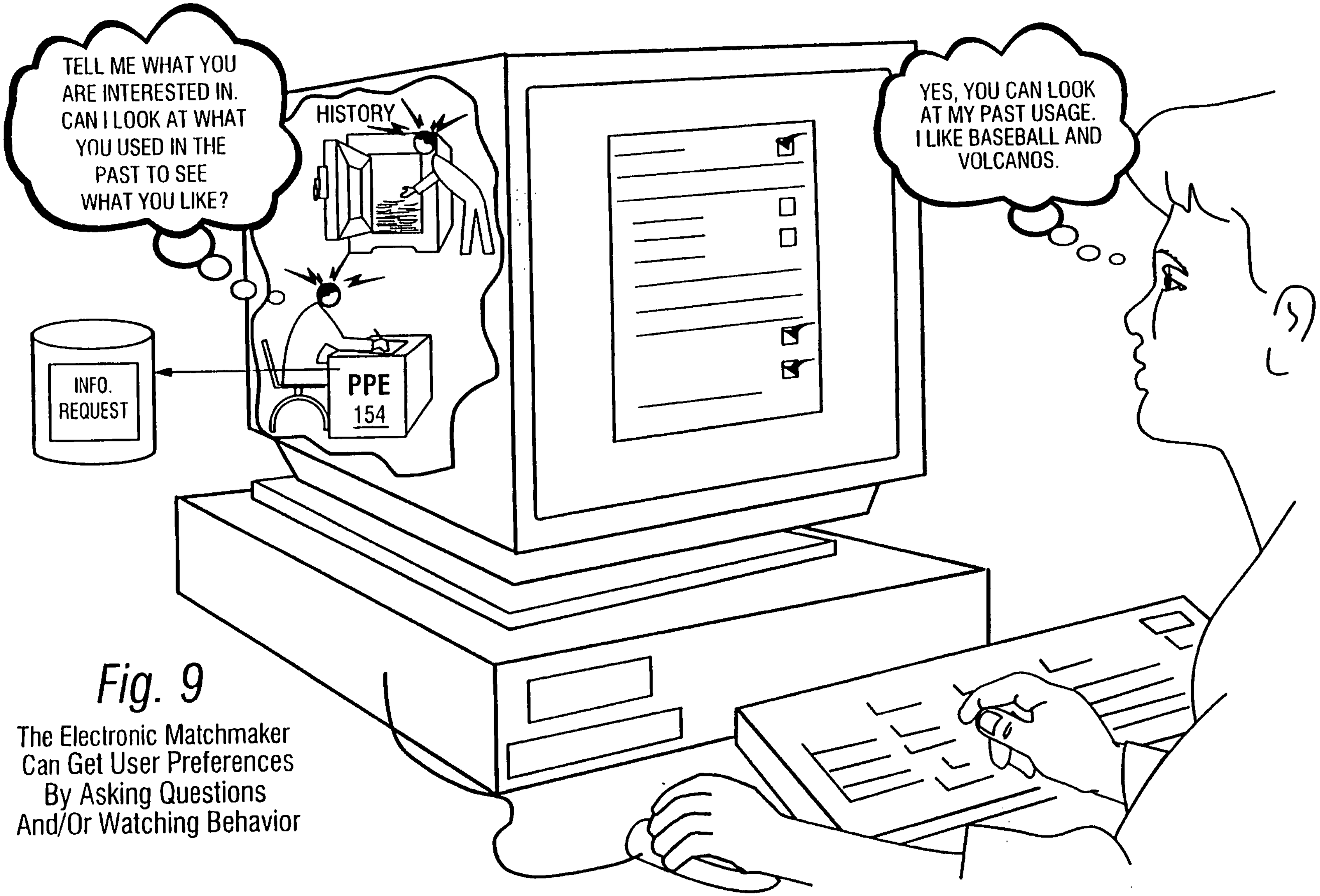

TELL ME WHAT YOU ARE INTERESTED IN. CAN I LOOK AT WHAT YOU USED IN THE PAST TO SEE WHAT YOU LIKE?
HISTORY
PPE
154
INFO. REQUEST
YES, YOU CAN LOOK AT MY PAST USAGE. I LIKE BASEBALL AND VOLCANOS.

Fig. 9

The Electronic Matchmaker Can Get User Preferences By Asking Questions And/Or Watching Behavior

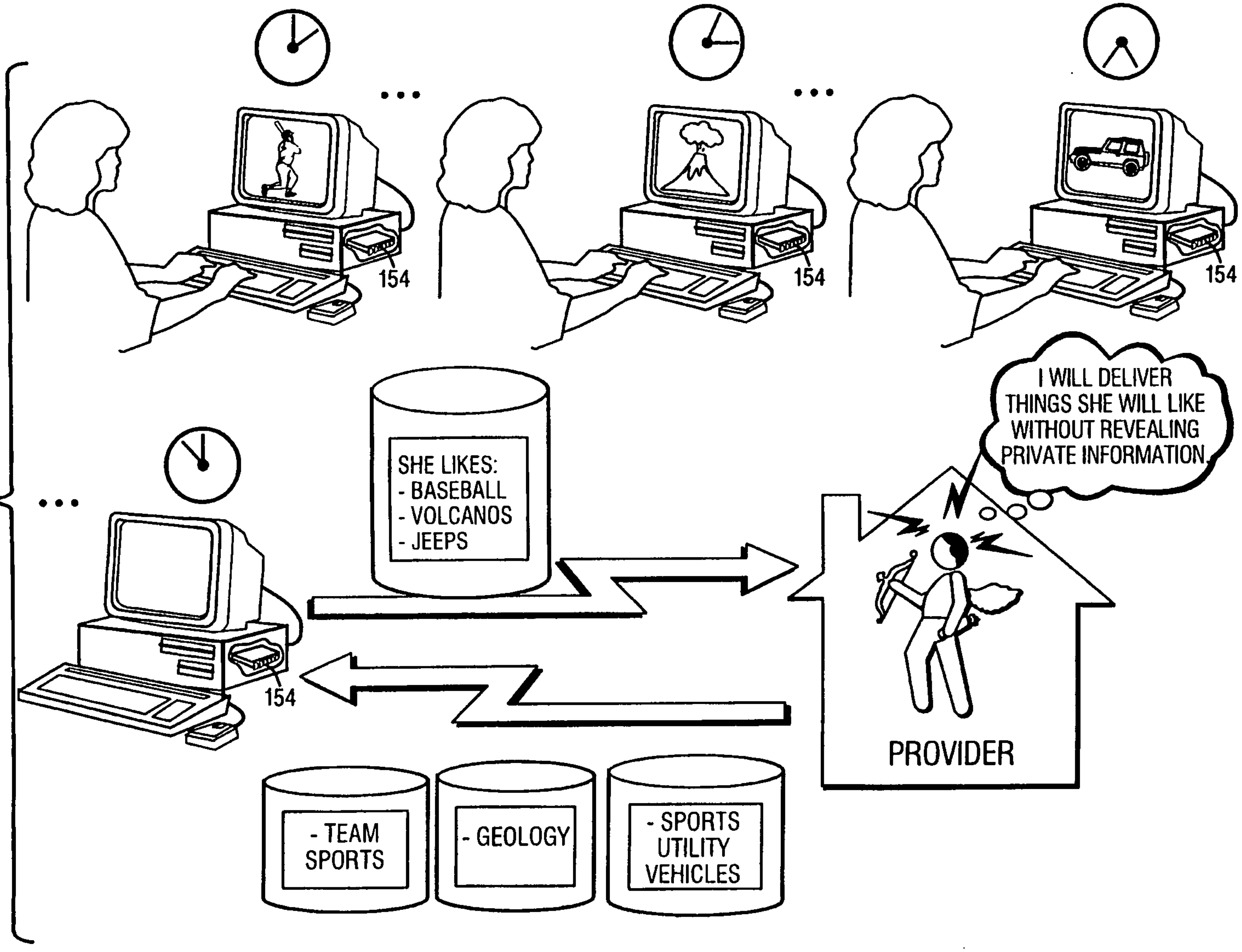
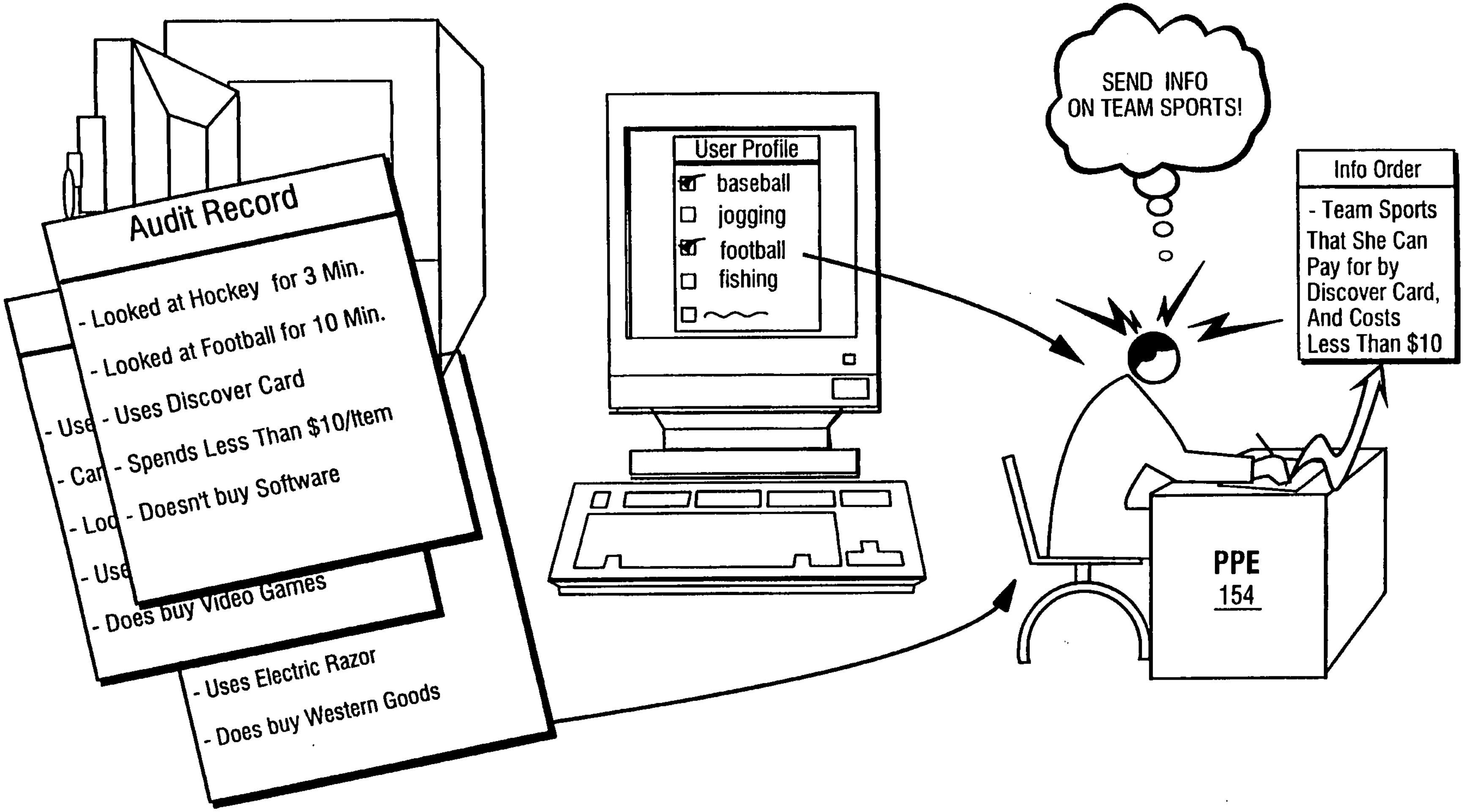

154
154
154
154
SHE LIKES:
- BASEBALL
- VOLCANOS
- JEEPS
I WILL DELIVER THINGS SHE WILL LIKE WITHOUT REVEALING PRIVATE INFORMATION
PROVIDER
- TEAM SPORTS
- GEOLOGY
- SPORTS UTILITY VEHICLES

Fig. 10

Example Electronic Matchmaking Process

Fig. 11 Example User Rights Management Information By Electronic Matchmaker


Audit Record
- Looked at Hockey for 3 Min.
- Looked at Football for 10 Min.
- Uses Discover Card
- Spends Less Than $10/Item
- Doesn't buy Software
- Use
- Car
- Loo
- Use
- Does buy Video Games
- Uses Electric Razor
- Does buy Western Goods
User Profile
☑ baseball
☐ jogging
☑ football
☐ fishing
☐
SEND INFO ON TEAM SPORTS!
Info Order
- Team Sports That She Can Pay for by Discover Card, And Costs Less Than $10
PPE
154

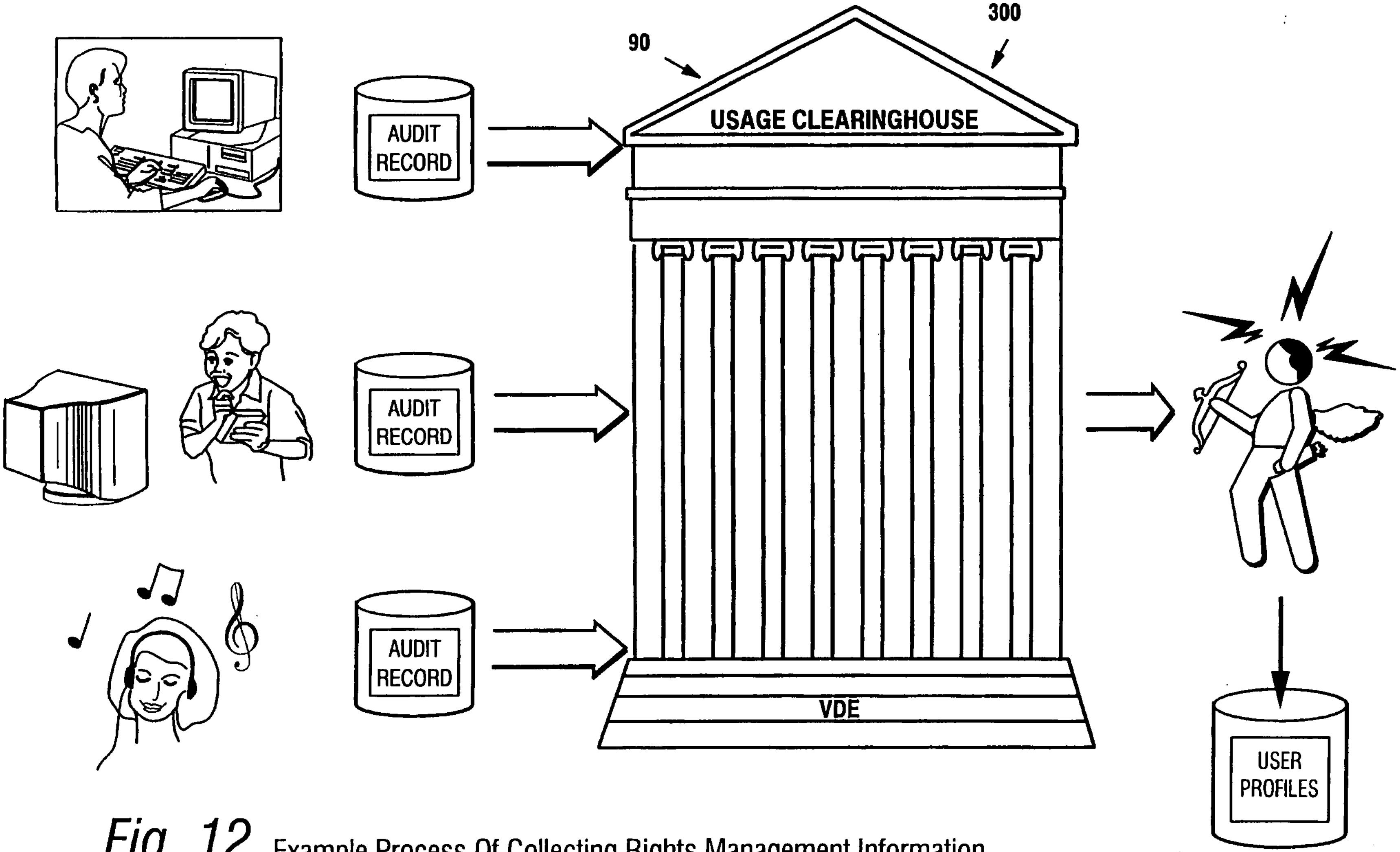
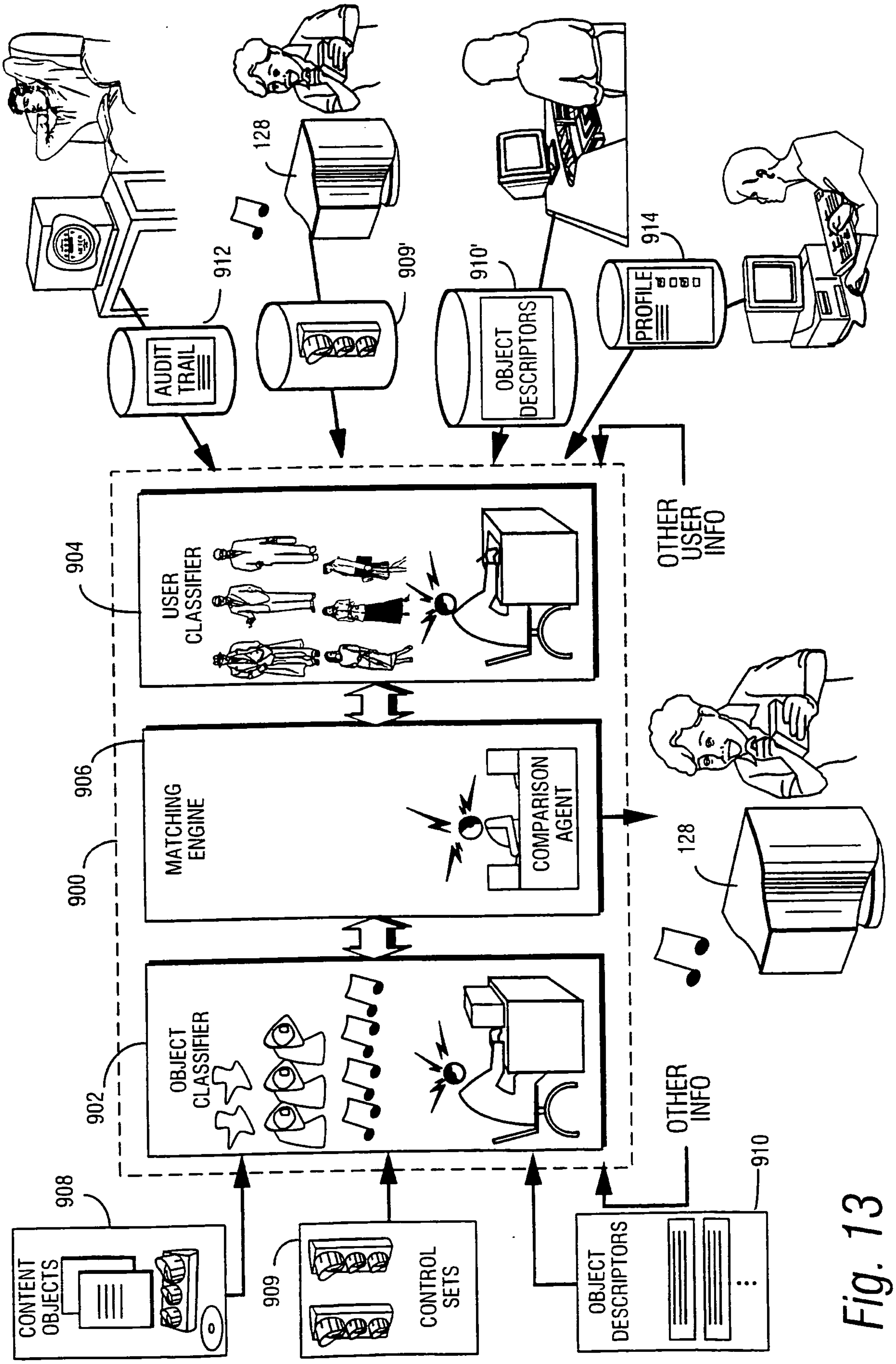

90
300
USAGE CLEARINGHOUSE
AUDIT RECORD
AUDIT RECORD
AUDIT RECORD
VDE
USER PROFILES

Fig. 12 Example Process Of Collecting Rights Management Information



900
902
OBJECT CLASSIFIER
906
MATCHING ENGINE
COMPARISON AGENT
904
USER CLASSIFIER
908
CONTENT OBJECTS
909
CONTROL SETS
910
OBJECT DESCRIPTORS
OTHER INFO
OTHER USER INFO
128
912
AUDIT TRAIL
909'
910'
OBJECT DESCRIPTORS
914
PROFILE
128

Fig. 13

SUBSTITUTE SHEET (RULE 26)

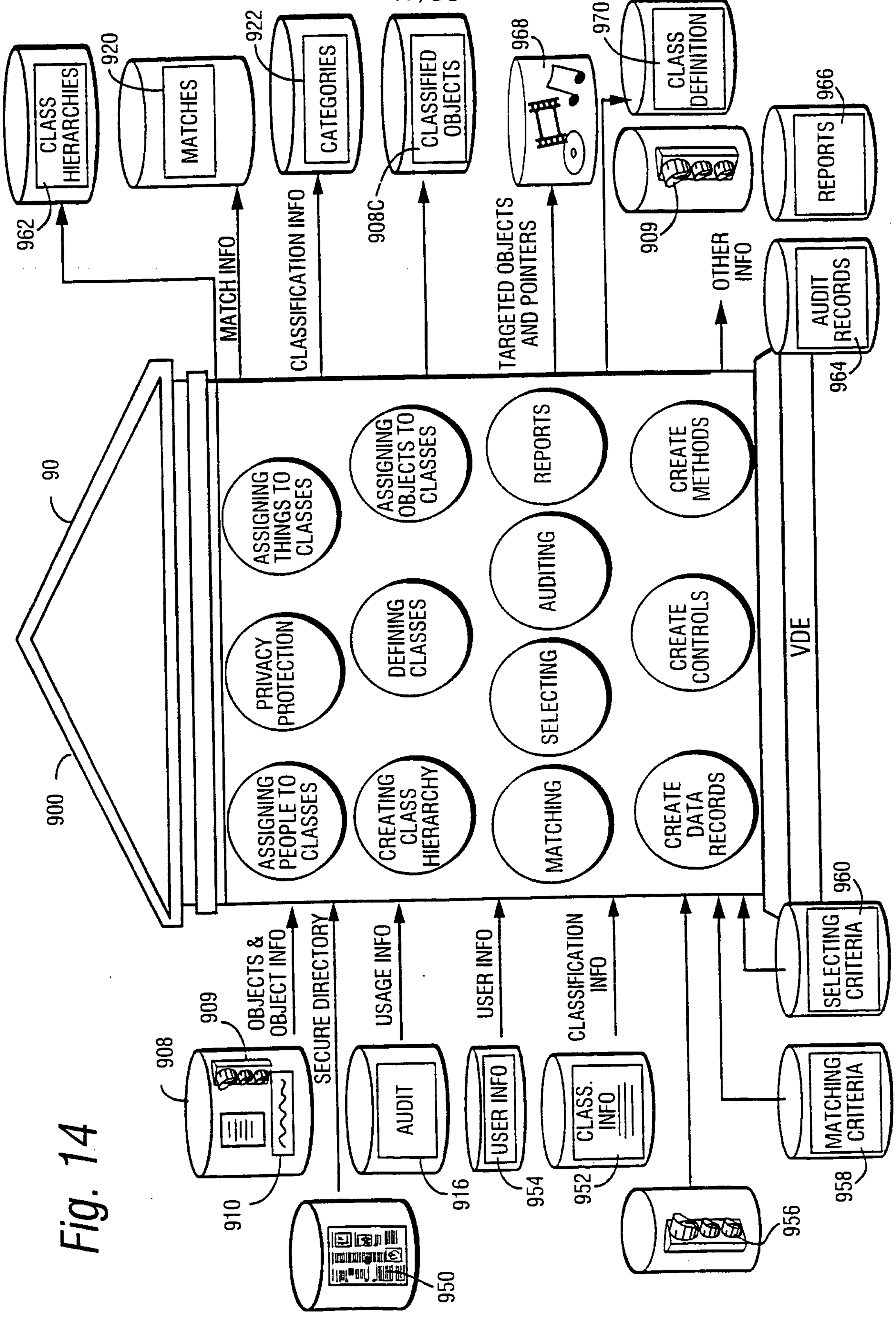
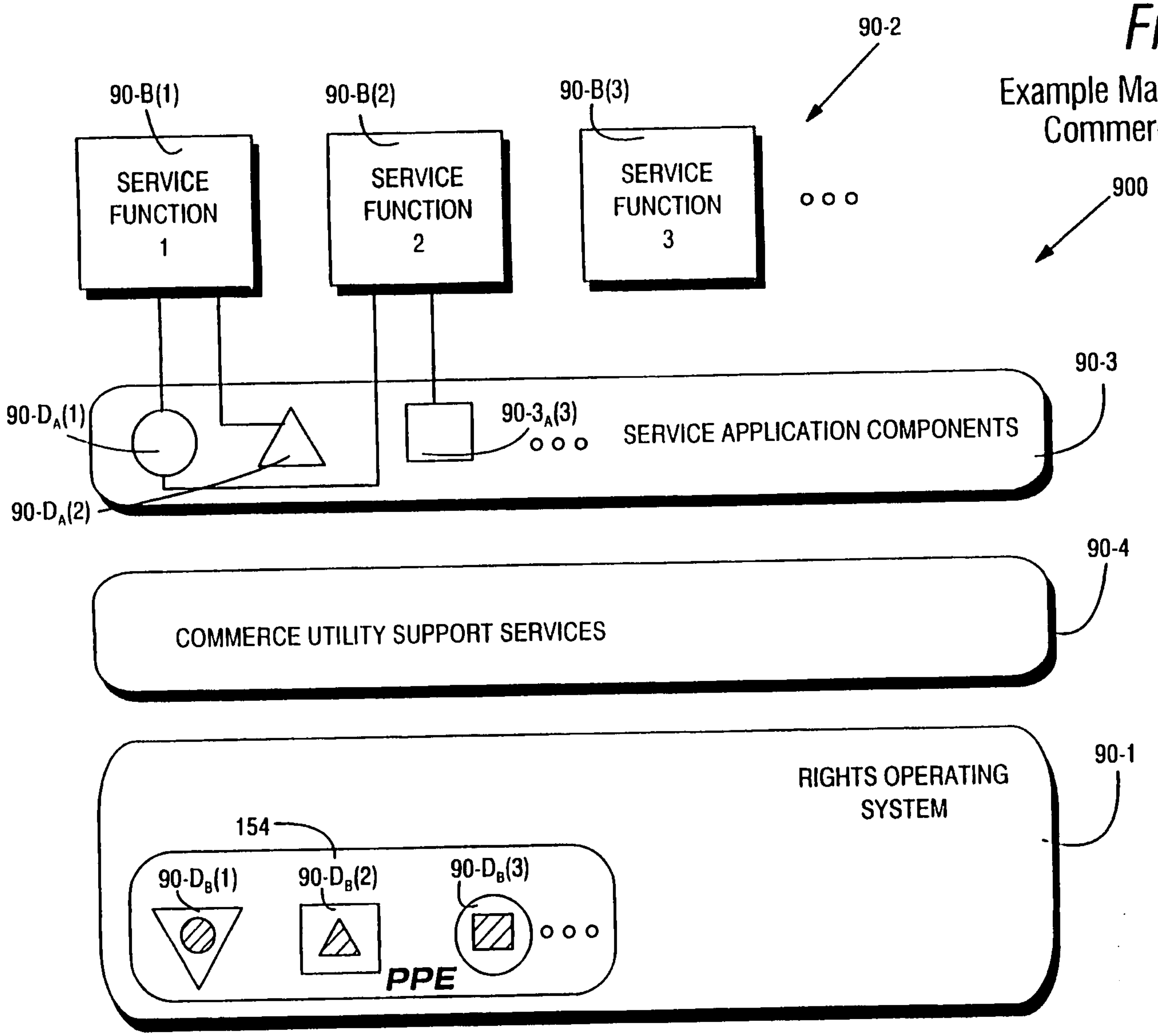

900
90
ASSIGNING PEOPLE TO CLASSES
PRIVACY PROTECTION
ASSIGNING THINGS TO CLASSES
CREATING CLASS HIERARCHY
DEFINING CLASSES
ASSIGNING OBJECTS TO CLASSES
MATCHING
SELECTING
AUDITING
REPORTS
CREATE DATA RECORDS
CREATE CONTROLS
CREATE METHODS
VDE
962
CLASS HIERARCHIES
920
MATCHES
922
CATEGORIES
CLASSIFIED OBJECTS
908C
968
970
CLASS DEFINITION
909
966
REPORTS
AUDIT RECORDS
964
MATCH INFO
CLASSIFICATION INFO
TARGETED OBJECTS AND POINTERS
OTHER INFO
908
909
910
OBJECTS & OBJECT INFO
SECURE DIRECTORY
950
AUDIT
916
USAGE INFO
USER INFO
954
USER INFO
CLASS. INFO
952
CLASSIFICATION INFO
956
MATCHING CRITERIA
958
SELECTING CRITERIA
960

Fig. 14

Fig. 14(A)

Example Matching and Classification Commerce Utility System 900


90-2
90-B(1)
90-B(2)
90-B(3)
SERVICE FUNCTION 1
SERVICE FUNCTION 2
SERVICE FUNCTION 3
900
90-3
$90\text{-}D_A(1)$
$90\text{-}D_A(2)$
$90\text{-}3_A(3)$
SERVICE APPLICATION COMPONENTS
90-4
COMMERCE UTILITY SUPPORT SERVICES
90-1
RIGHTS OPERATING SYSTEM
154
$90\text{-}D_B(1)$
$90\text{-}D_B(2)$
$90\text{-}D_B(3)$
PPE

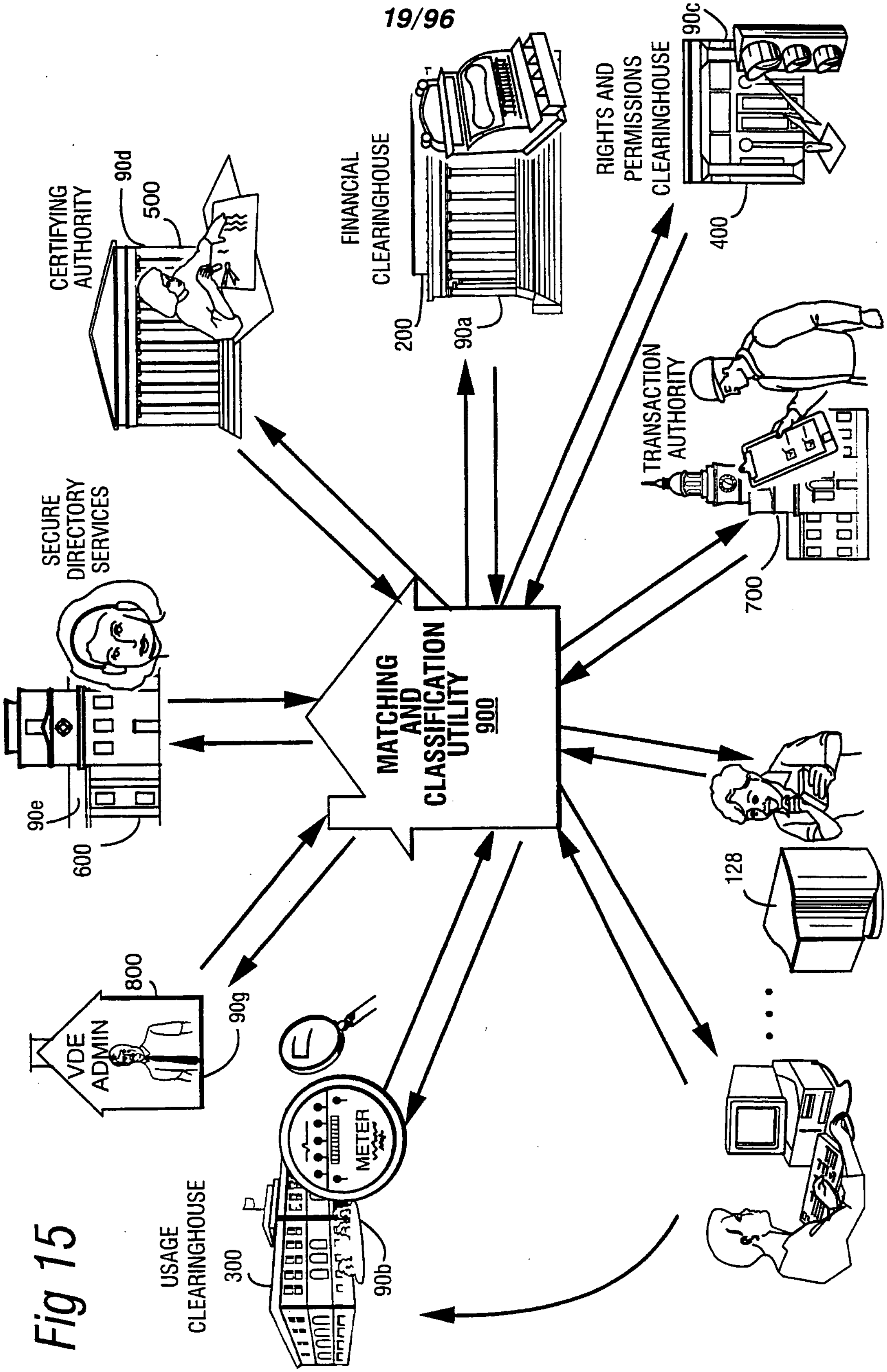

CERTIFYING AUTHORITY
FINANCIAL CLEARINGHOUSE
RIGHTS AND PERMISSIONS CLEARINGHOUSE
SECURE DIRECTORY SERVICES
MATCHING AND CLASSIFICATION UTILITY
TRANSACTION AUTHORITY
VDE ADMIN
METER
USAGE CLEARINGHOUSE

Fig 15

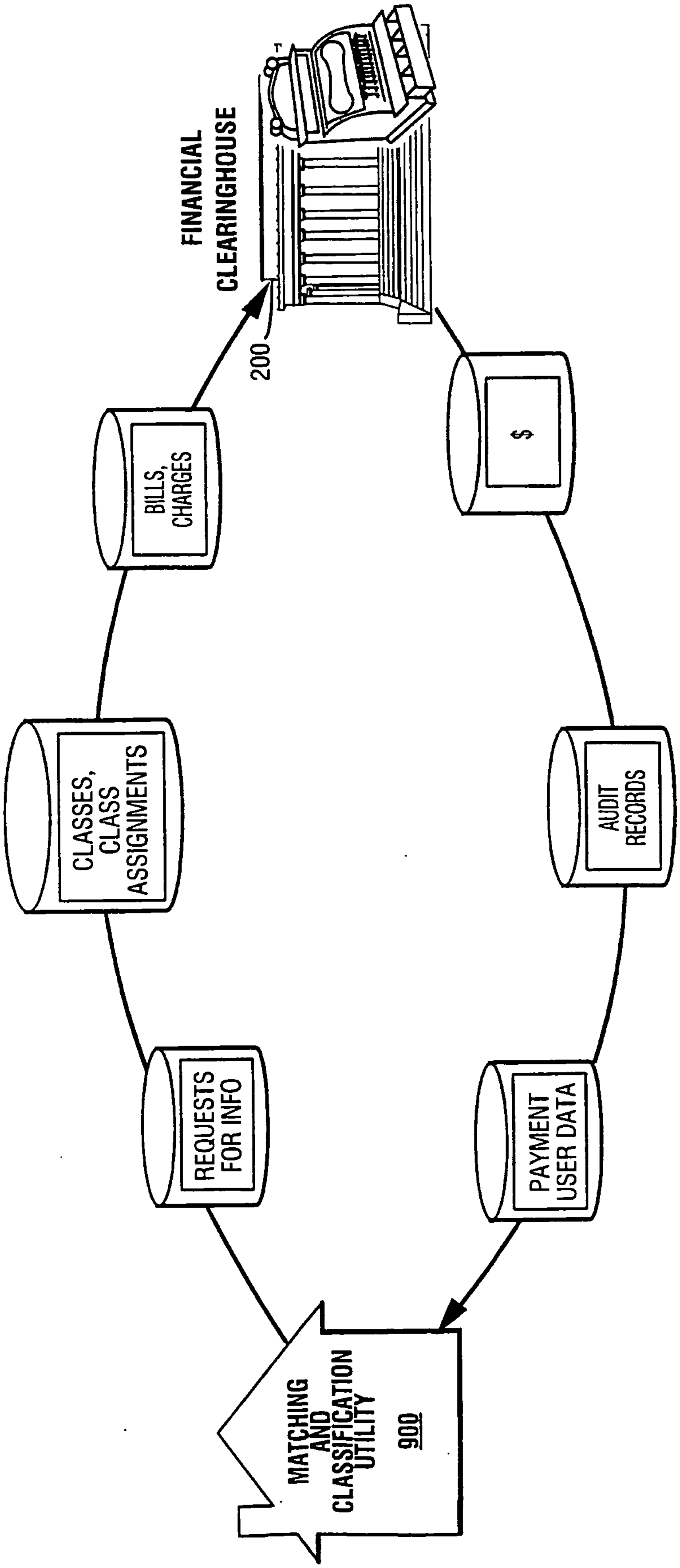

FINANCIAL
CLEARINGHOUSE
200
BILLS,
CHARGES
$
CLASSES,
CLASS
ASSIGNMENTS
AUDIT
RECORDS
REQUESTS
FOR INFO
PAYMENT
USER DATA
MATCHING
AND
CLASSIFICATION
UTILITY
900

Fig. 15A

Fig. 15B

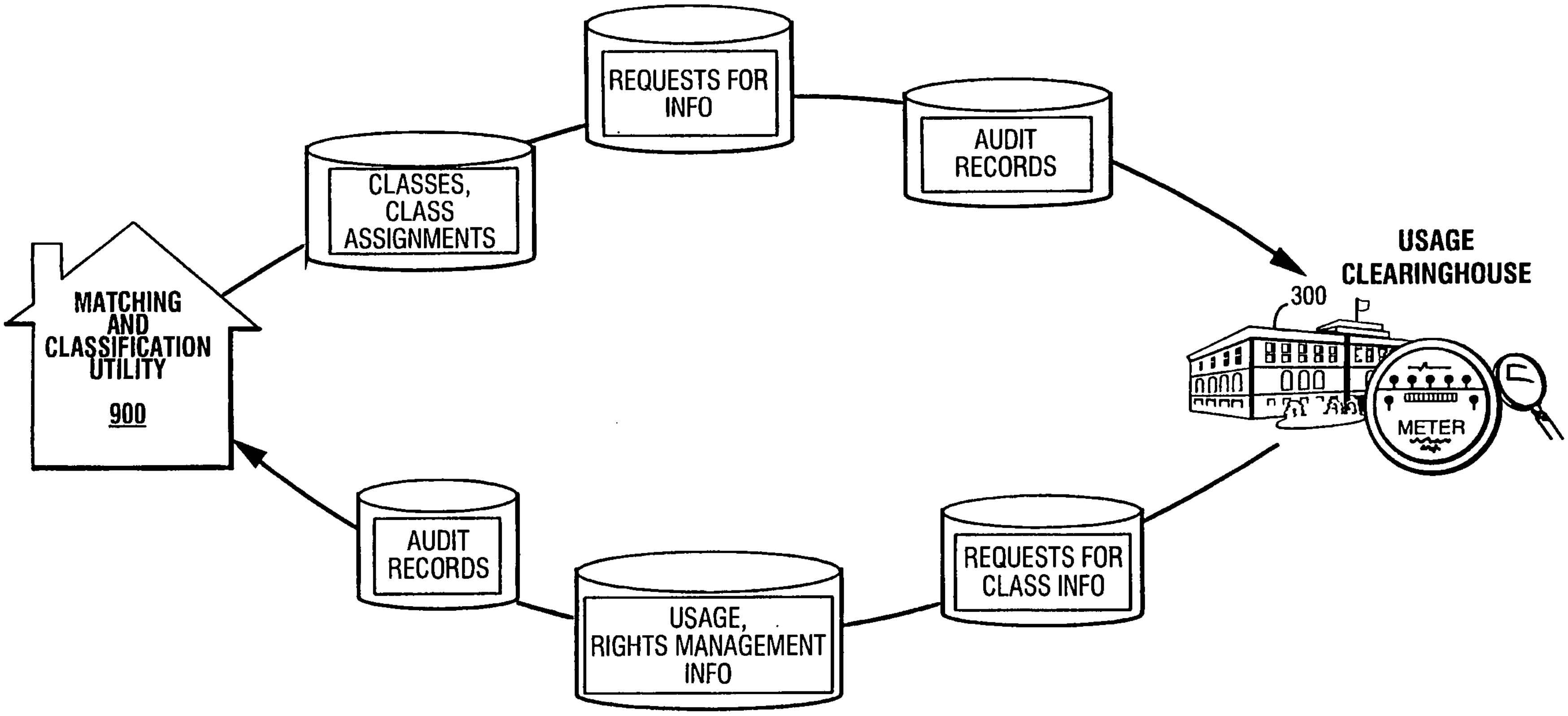
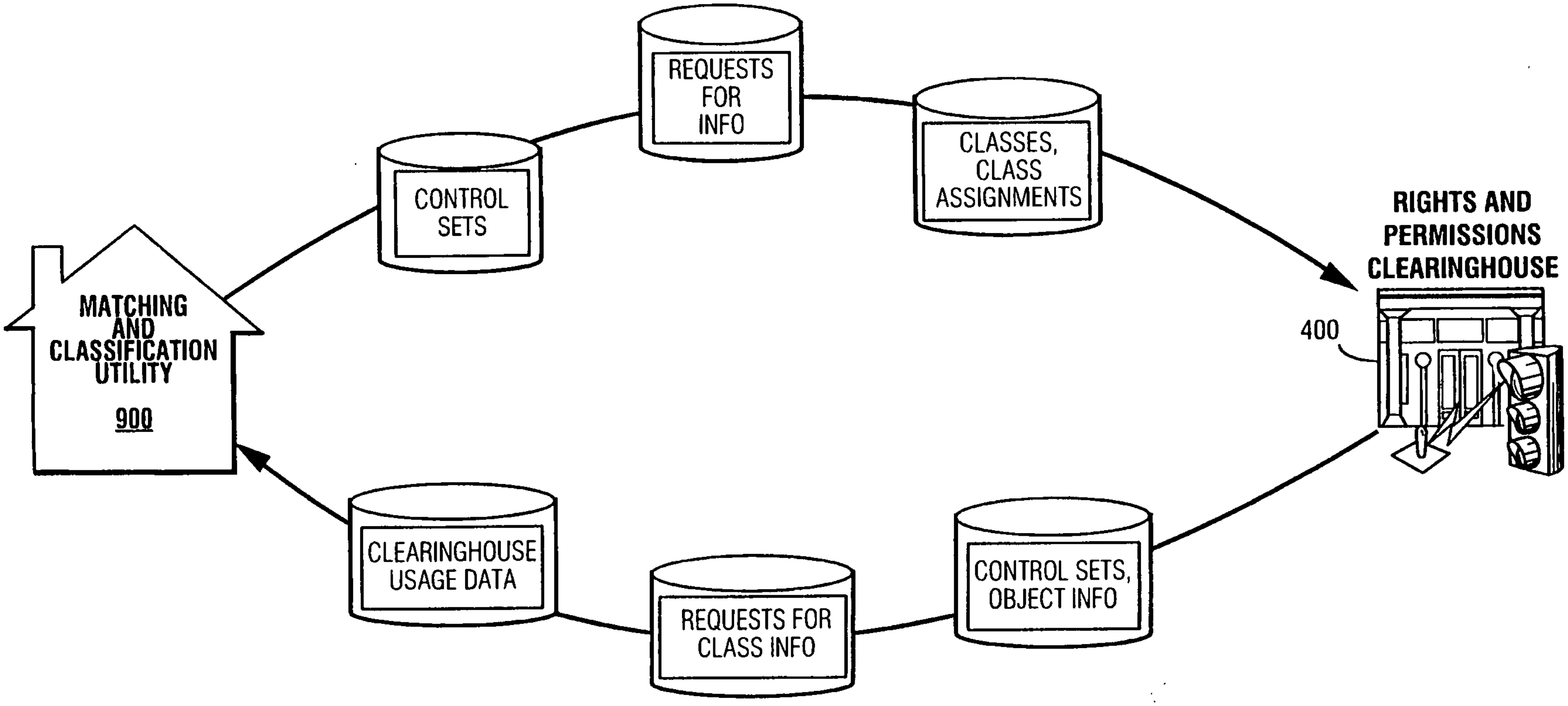

MATCHING AND CLASSIFICATION UTILITY
900
CLASSES, CLASS ASSIGNMENTS
REQUESTS FOR INFO
AUDIT RECORDS
USAGE CLEARINGHOUSE
300
METER
REQUESTS FOR CLASS INFO
USAGE, RIGHTS MANAGEMENT INFO
AUDIT RECORDS

Fig. 15C


RIGHTS AND PERMISSIONS CLEARINGHOUSE
400
CLASSES, CLASS ASSIGNMENTS
REQUESTS FOR INFO
CONTROL SETS
MATCHING AND CLASSIFICATION UTILITY
900
CLEARINGHOUSE USAGE DATA
REQUESTS FOR CLASS INFO
CONTROL SETS, OBJECT INFO

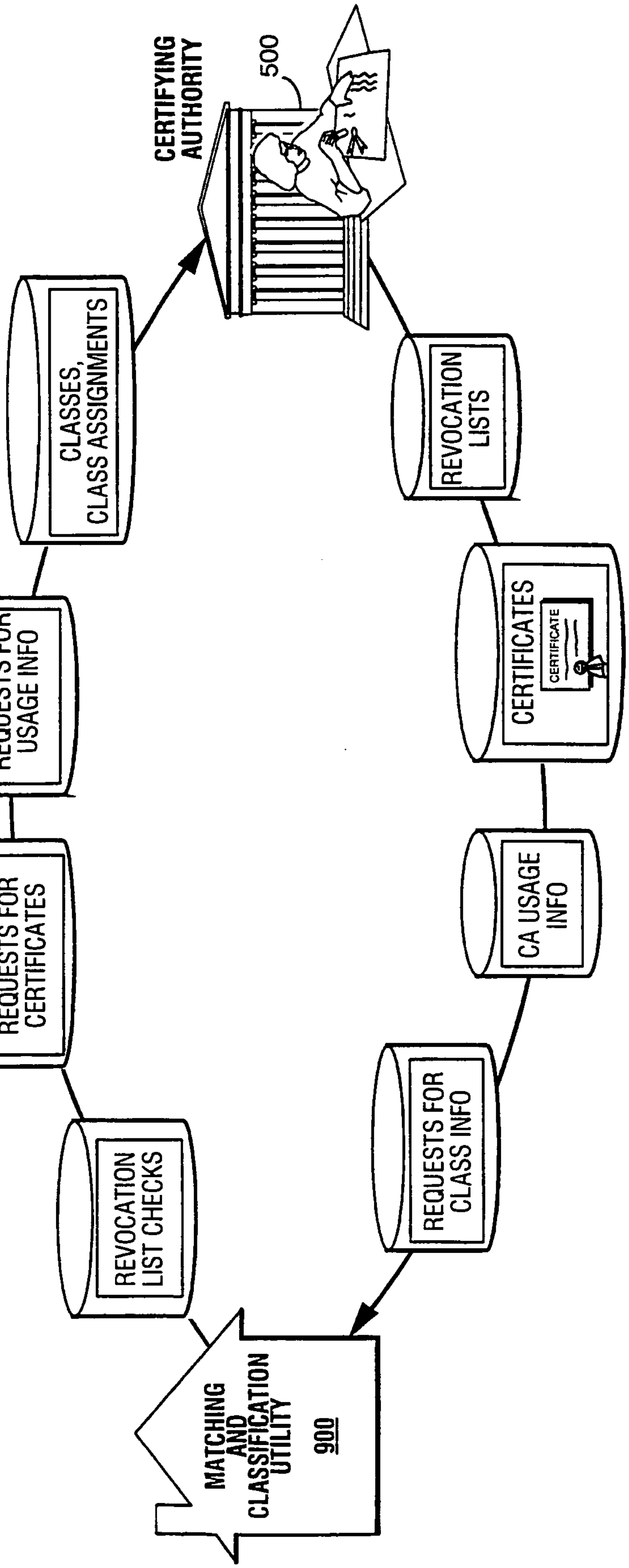

CERTIFYING AUTHORITY
500
CLASSES, CLASS ASSIGNMENTS
REVOCATION LISTS
REQUESTS FOR USAGE INFO
CERTIFICATES
CERTIFICATE
REQUESTS FOR CERTIFICATES
CA USAGE INFO
REVOCATION LIST CHECKS
REQUESTS FOR CLASS INFO
MATCHING AND CLASSIFICATION UTILITY
900

Fig. 15D

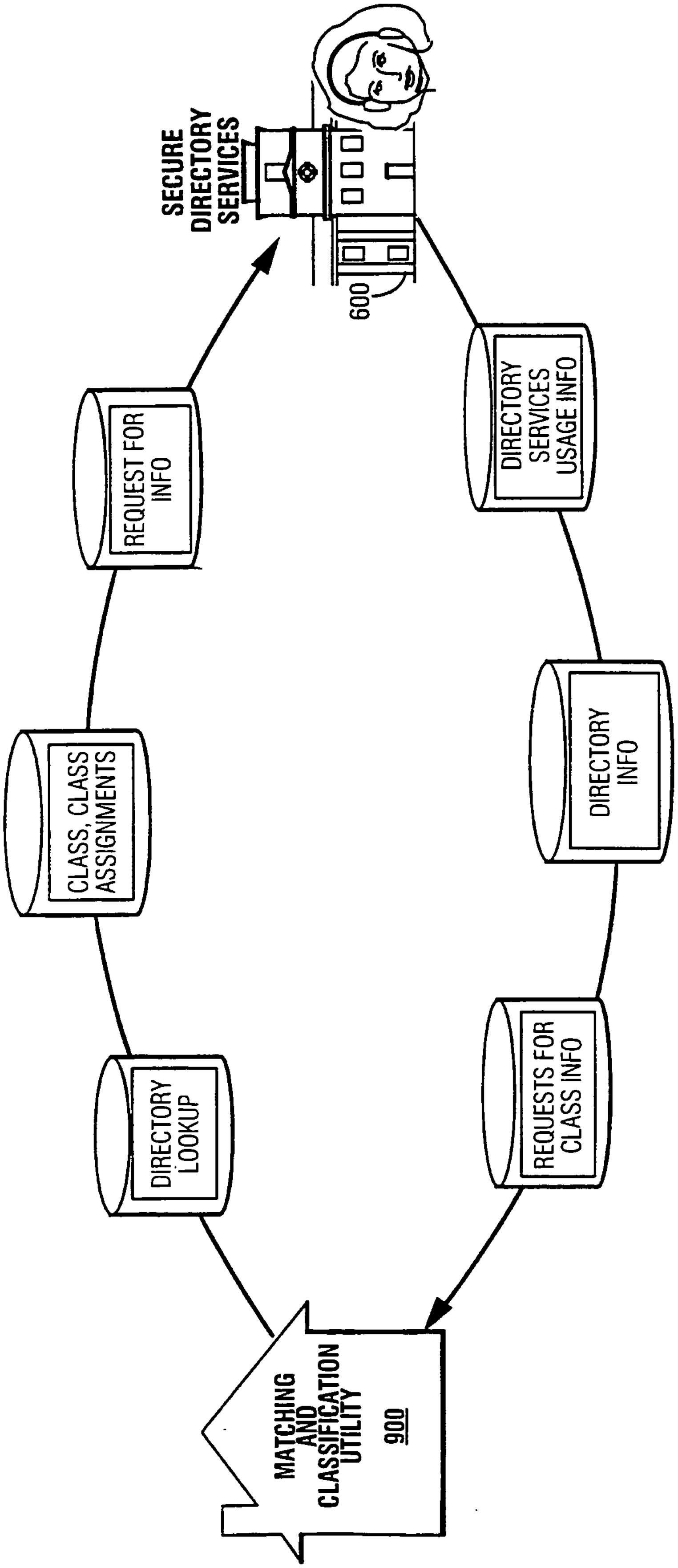

SECURE DIRECTORY SERVICES
600
REQUEST FOR INFO
CLASS, CLASS ASSIGNMENTS
DIRECTORY LOOKUP
MATCHING AND CLASSIFICATION UTILITY
900
REQUESTS FOR CLASS INFO
DIRECTORY INFO
DIRECTORY SERVICES USAGE INFO

Fig. 15E

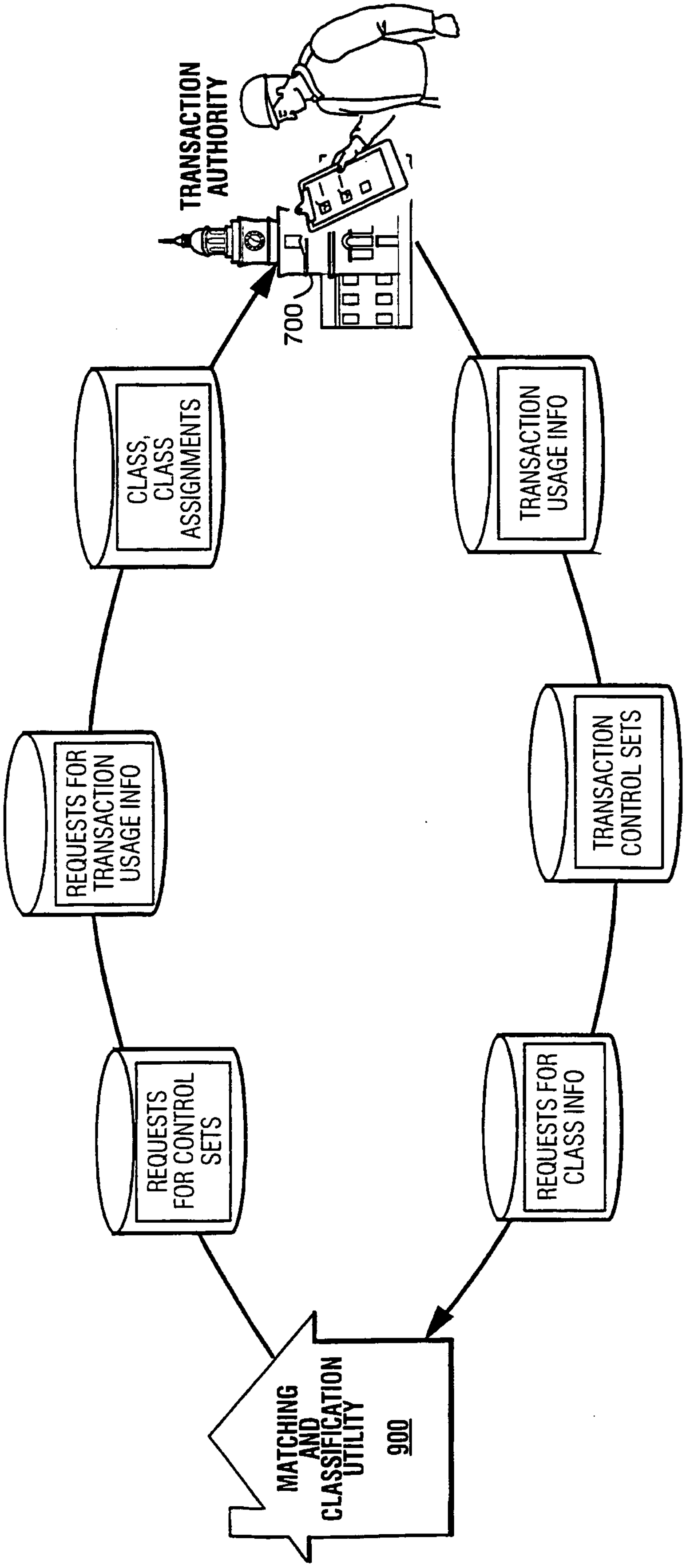

TRANSACTION AUTHORITY
700
CLASS, CLASS ASSIGNMENTS
REQUESTS FOR TRANSACTION USAGE INFO
REQUESTS FOR CONTROL SETS
MATCHING AND CLASSIFICATION UTILITY
900
REQUESTS FOR CLASS INFO
TRANSACTION CONTROL SETS
TRANSACTION USAGE INFO

Fig. 15F

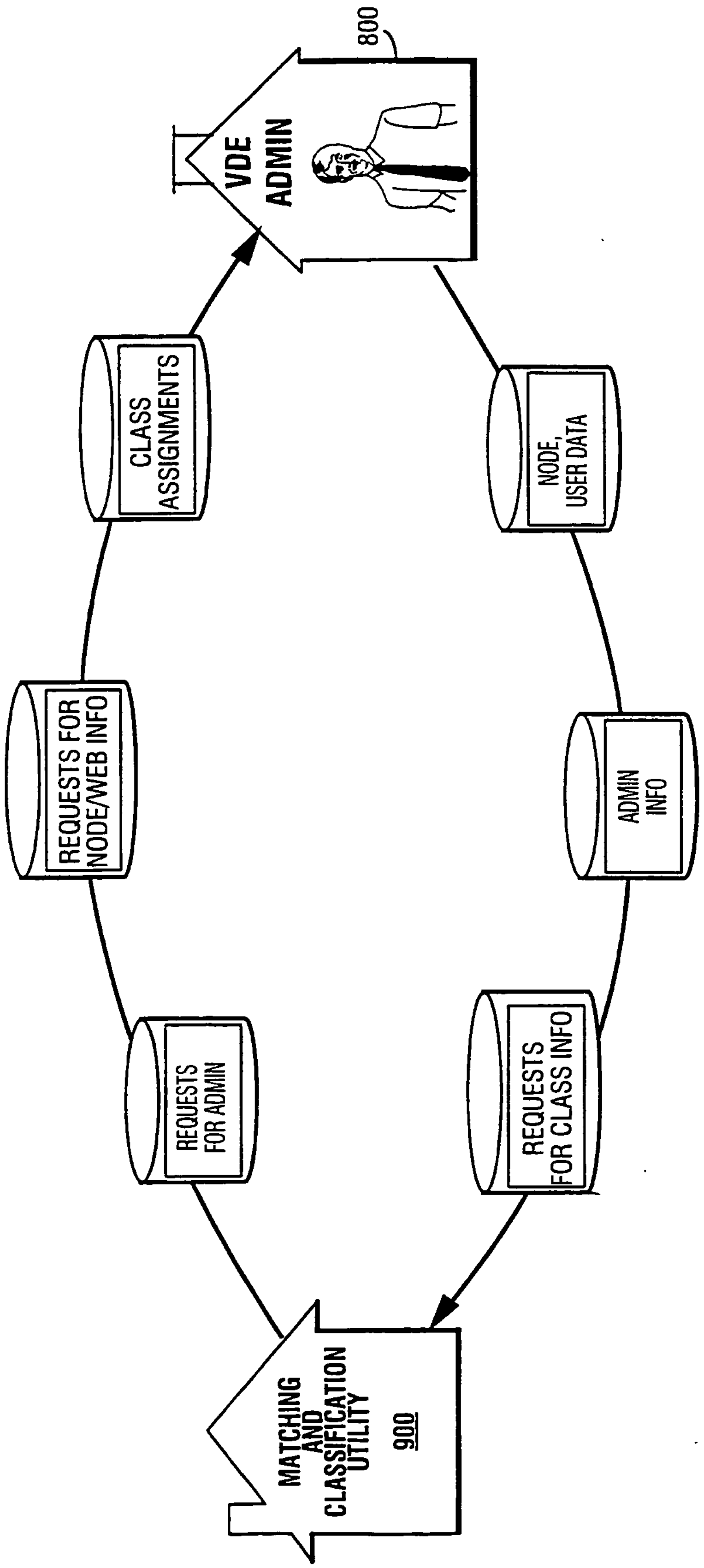

800
VDE
ADMIN
CLASS
ASSIGNMENTS
NODE,
USER DATA
REQUESTS FOR
NODE/WEB INFO
ADMIN
INFO
REQUESTS
FOR ADMIN
REQUESTS
FOR CLASS INFO
MATCHING
AND
CLASSIFICATION
UTILITY
900

Fig. 15G

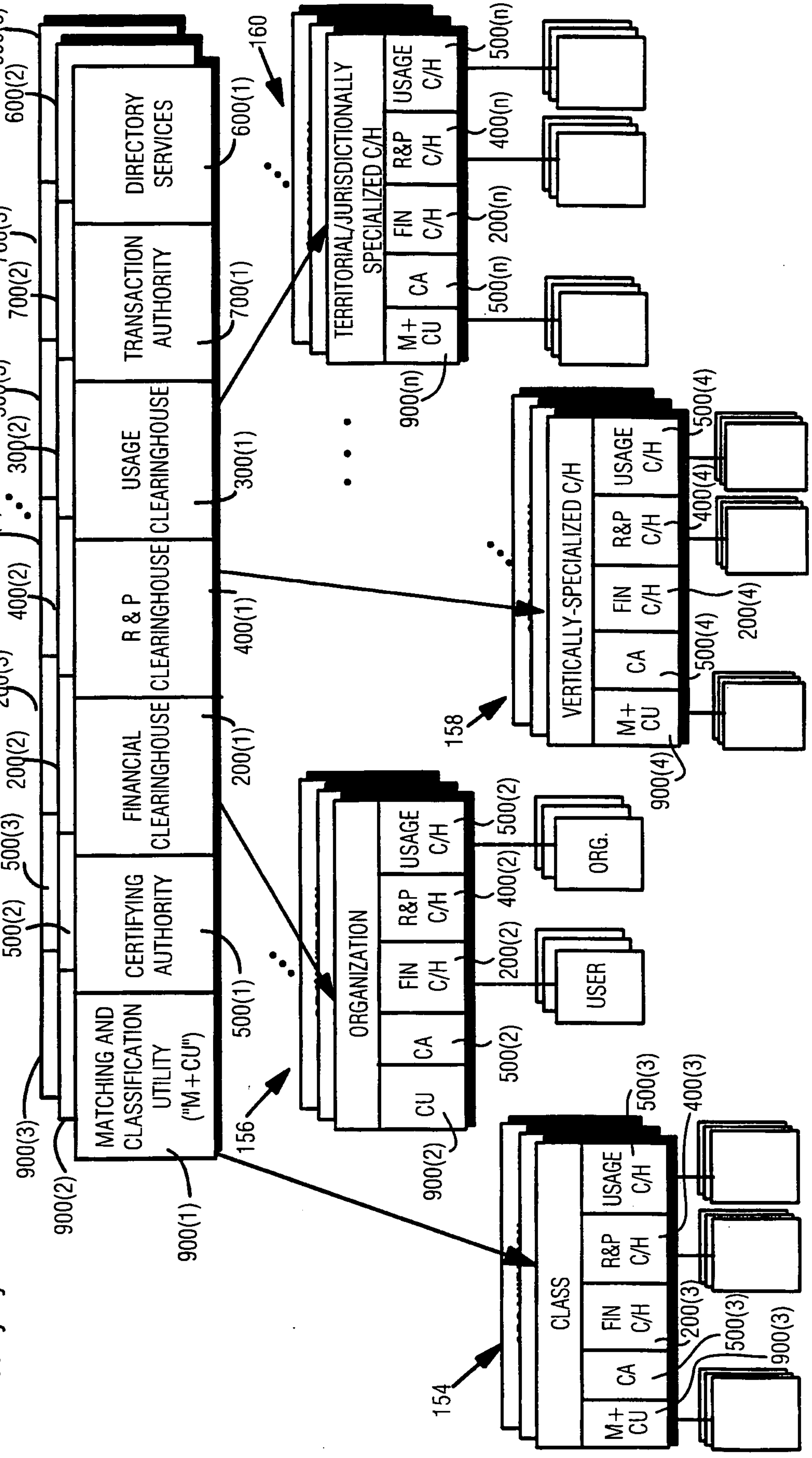

600(3)
600(2)
700(3)
700(2)
300(3)
300(2)
400(3)
400(2)
200(3)
200(2)
500(3)
500(2)
900(3)
900(2)
MATCHING AND CLASSIFICATION UTILITY ("M+CU")
CERTIFYING AUTHORITY
FINANCIAL CLEARINGHOUSE
R & P CLEARINGHOUSE
USAGE CLEARINGHOUSE
TRANSACTION AUTHORITY
DIRECTORY SERVICES
900(1)
500(1)
200(1)
400(1)
300(1)
700(1)
600(1)
160
156
158
154
TERRITORIAL/JURISDICTIONALLY SPECIALIZED C/H
M+ CU
CA
FIN C/H
R&P C/H
USAGE C/H
900(n)
500(n)
200(n)
400(n)
500(n)
ORGANIZATION
CU
CA
FIN C/H
R&P C/H
USAGE C/H
900(2)
500(2)
200(2)
400(2)
500(2)
USER
ORG.
VERTICALLY-SPECIALIZED C/H
M+ CU
CA
FIN C/H
R&P C/H
USAGE C/H
900(4)
500(4)
200(4)
400(4)
500(4)
CLASS
M+ CU
CA
FIN C/H
R&P C/H
USAGE C/H
900(3)
500(3)
200(3)
400(3)
500(3)

Fig. 16A

Hierarchy of Commerce Utility Systems

SUBSTITUTE SHEET (RULE 26)

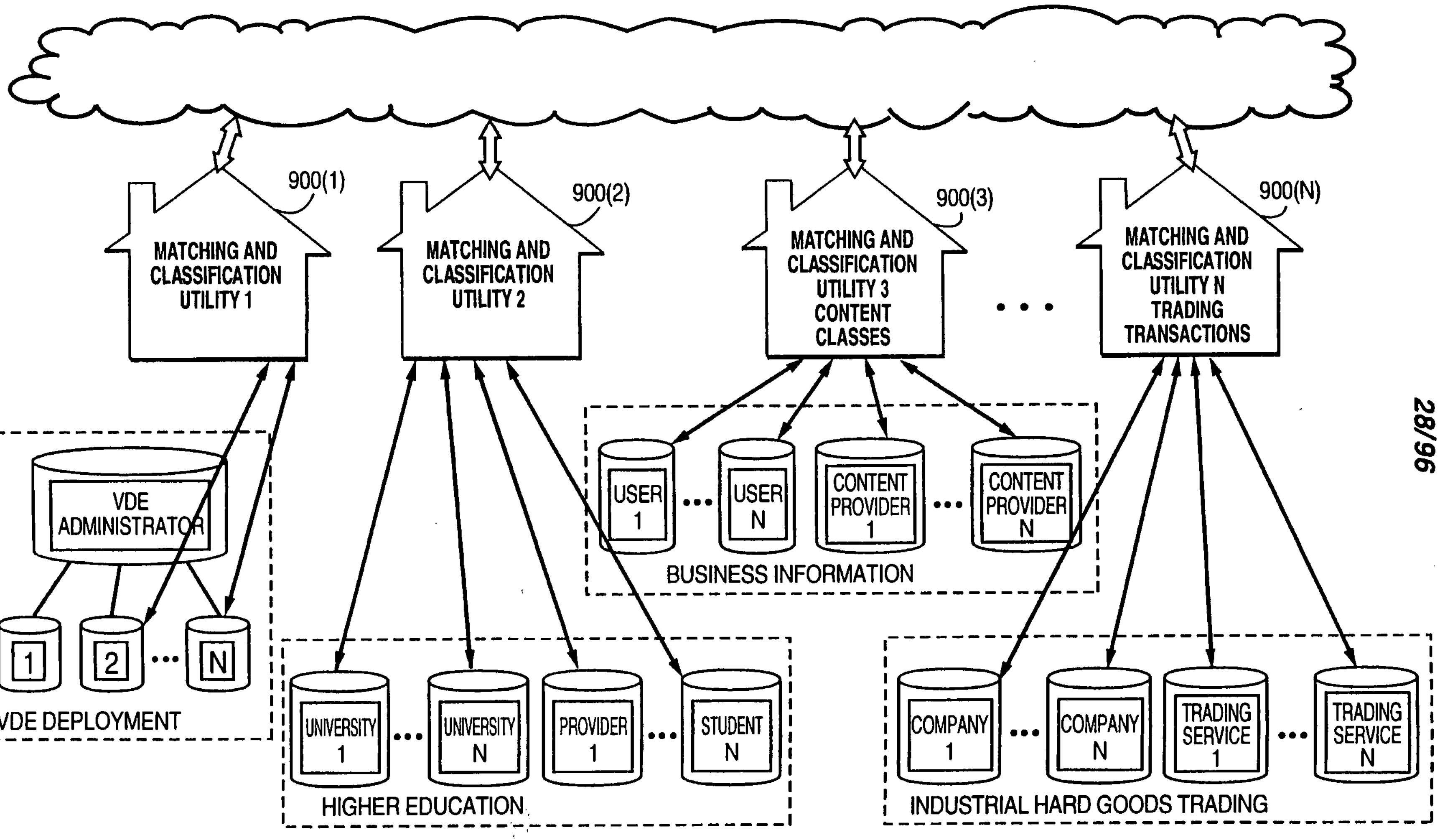

900(1)
MATCHING AND CLASSIFICATION UTILITY 1
900(2)
MATCHING AND CLASSIFICATION UTILITY 2
900(3)
MATCHING AND CLASSIFICATION UTILITY 3 CONTENT CLASSES
900(N)
MATCHING AND CLASSIFICATION UTILITY N TRADING TRANSACTIONS
• • •
VDE ADMINISTRATOR
1
2
N
VDE DEPLOYMENT
UNIVERSITY 1
UNIVERSITY N
PROVIDER 1
STUDENT N
HIGHER EDUCATION
USER 1
USER N
CONTENT PROVIDER 1
CONTENT PROVIDER N
BUSINESS INFORMATION
COMPANY 1
COMPANY N
TRADING SERVICE 1
TRADING SERVICE N
INDUSTRIAL HARD GOODS TRADING

Fig.16B Matching & Classification Utilities Provide Services To Classes Of Nodes, Users, Content Services, Transaction Services.

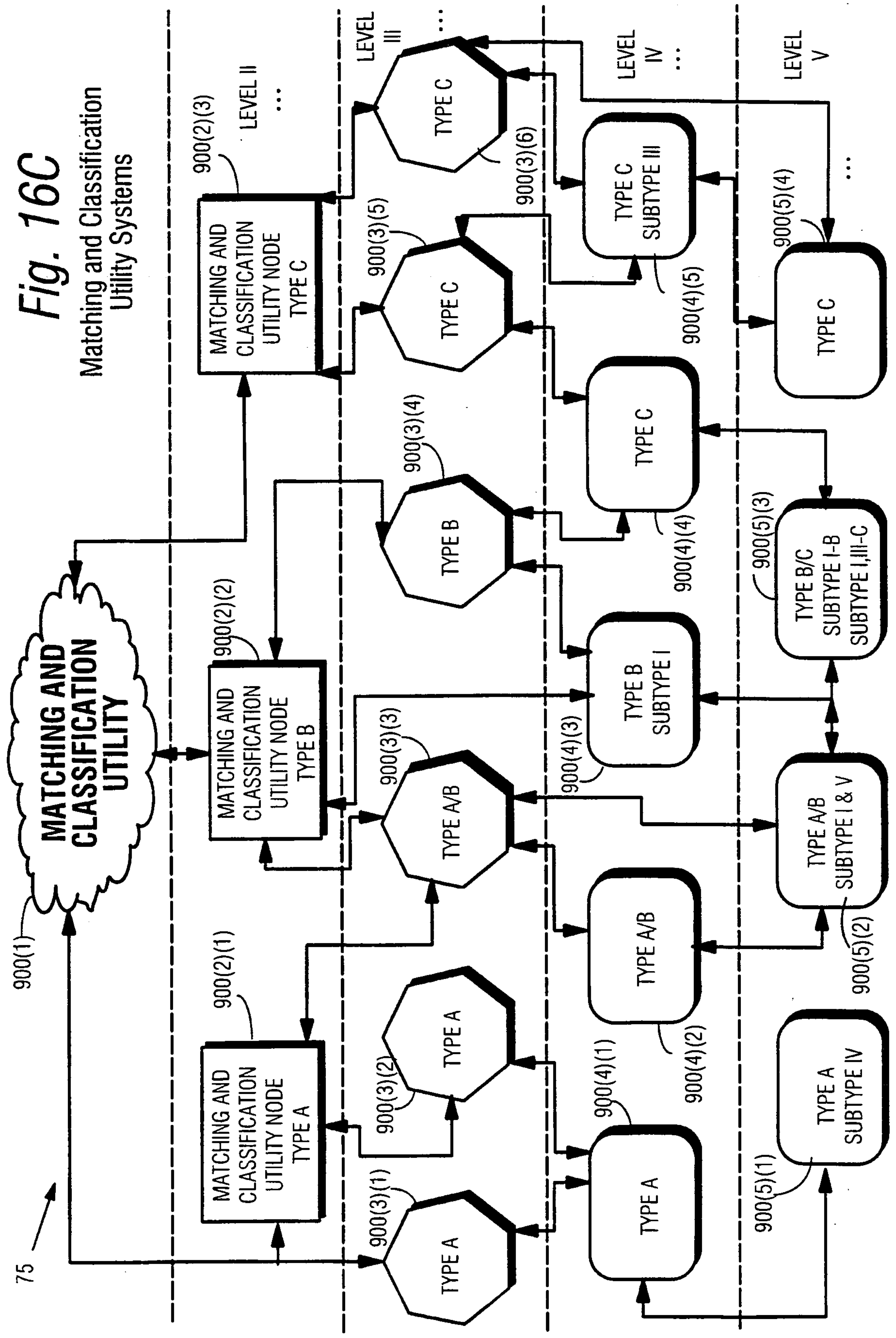
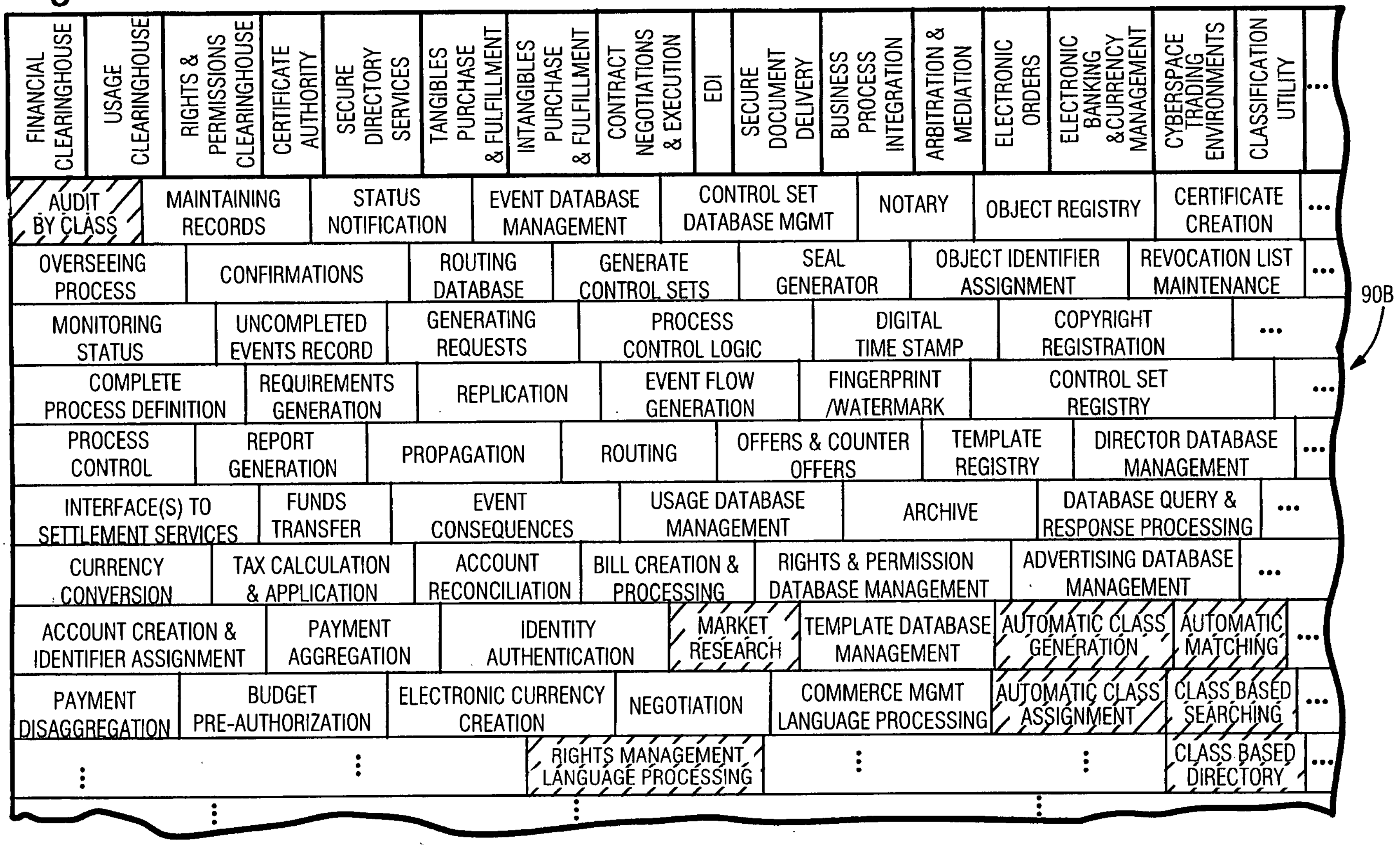


Fig. 16C

Matching and Classification Utility Systems

75
900(1)
MATCHING AND CLASSIFICATION UTILITY
LEVEL II
LEVEL III
LEVEL IV
LEVEL V
900(2)(1)
900(2)(2)
900(2)(3)
MATCHING AND CLASSIFICATION UTILITY NODE TYPE A
MATCHING AND CLASSIFICATION UTILITY NODE TYPE B
MATCHING AND CLASSIFICATION UTILITY NODE TYPE C
900(3)(1)
900(3)(2)
900(3)(3)
900(3)(4)
900(3)(5)
900(3)(6)
TYPE A
TYPE A
TYPE A/B
TYPE B
TYPE C
TYPE C
900(4)(1)
900(4)(2)
900(4)(3)
900(4)(4)
900(4)(5)
TYPE A
TYPE A/B
TYPE B SUBTYPE I
TYPE C
TYPE C SUBTYPE III
900(5)(1)
900(5)(2)
900(5)(3)
900(5)(4)
TYPE A SUBTYPE IV
TYPE A/B SUBTYPE I & V
TYPE B/C SUBTYPE I-B SUBTYPE I,III-C
TYPE C
...

Fig. 17


FINANCIAL CLEARINGHOUSE
USAGE CLEARINGHOUSE
RIGHTS & PERMISSIONS CLEARINGHOUSE
CERTIFICATE AUTHORITY
SECURE DIRECTORY SERVICES
TANGIBLES PURCHASE & FULFILLMENT
INTANGIBLES PURCHASE & FULFILLMENT
CONTRACT NEGOTIATIONS & EXECUTION
EDI
SECURE DOCUMENT DELIVERY
BUSINESS PROCESS INTEGRATION
ARBITRATION & MEDIATION
ELECTRONIC ORDERS
ELECTRONIC BANKING &CURRENCY MANAGEMENT
CYBERSPACE TRADING ENVIRONMENTS
CLASSIFICATION UTILITY
...
AUDIT BY CLASS
MAINTAINING RECORDS
STATUS NOTIFICATION
EVENT DATABASE MANAGEMENT
CONTROL SET DATABASE MGMT
NOTARY
OBJECT REGISTRY
CERTIFICATE CREATION
...
OVERSEEING PROCESS
CONFIRMATIONS
ROUTING DATABASE
GENERATE CONTROL SETS
SEAL GENERATOR
OBJECT IDENTIFIER ASSIGNMENT
REVOCATION LIST MAINTENANCE
...
MONITORING STATUS
UNCOMPLETED EVENTS RECORD
GENERATING REQUESTS
PROCESS CONTROL LOGIC
DIGITAL TIME STAMP
COPYRIGHT REGISTRATION
...
COMPLETE PROCESS DEFINITION
REQUIREMENTS GENERATION
REPLICATION
EVENT FLOW GENERATION
FINGERPRINT /WATERMARK
CONTROL SET REGISTRY
...
PROCESS CONTROL
REPORT GENERATION
PROPAGATION
ROUTING
OFFERS & COUNTER OFFERS
TEMPLATE REGISTRY
DIRECTOR DATABASE MANAGEMENT
...
INTERFACE(S) TO SETTLEMENT SERVICES
FUNDS TRANSFER
EVENT CONSEQUENCES
USAGE DATABASE MANAGEMENT
ARCHIVE
DATABASE QUERY & RESPONSE PROCESSING
...
CURRENCY CONVERSION
TAX CALCULATION & APPLICATION
ACCOUNT RECONCILIATION
BILL CREATION & PROCESSING
RIGHTS & PERMISSION DATABASE MANAGEMENT
ADVERTISING DATABASE MANAGEMENT
...
ACCOUNT CREATION & IDENTIFIER ASSIGNMENT
PAYMENT AGGREGATION
IDENTITY AUTHENTICATION
MARKET RESEARCH
TEMPLATE DATABASE MANAGEMENT
AUTOMATIC CLASS GENERATION
AUTOMATIC MATCHING
...
PAYMENT DISAGGREGATION
BUDGET PRE-AUTHORIZATION
ELECTRONIC CURRENCY CREATION
NEGOTIATION
COMMERCE MGMT LANGUAGE PROCESSING
AUTOMATIC CLASS ASSIGNMENT
CLASS BASED SEARCHING
...
⋮
⋮
RIGHTS MANAGEMENT LANGUAGE PROCESSING
⋮
⋮
CLASS BASED DIRECTORY
...
⋮
⋮
⋮
90B

SUBSTITUTE SHEET (RULE 26)

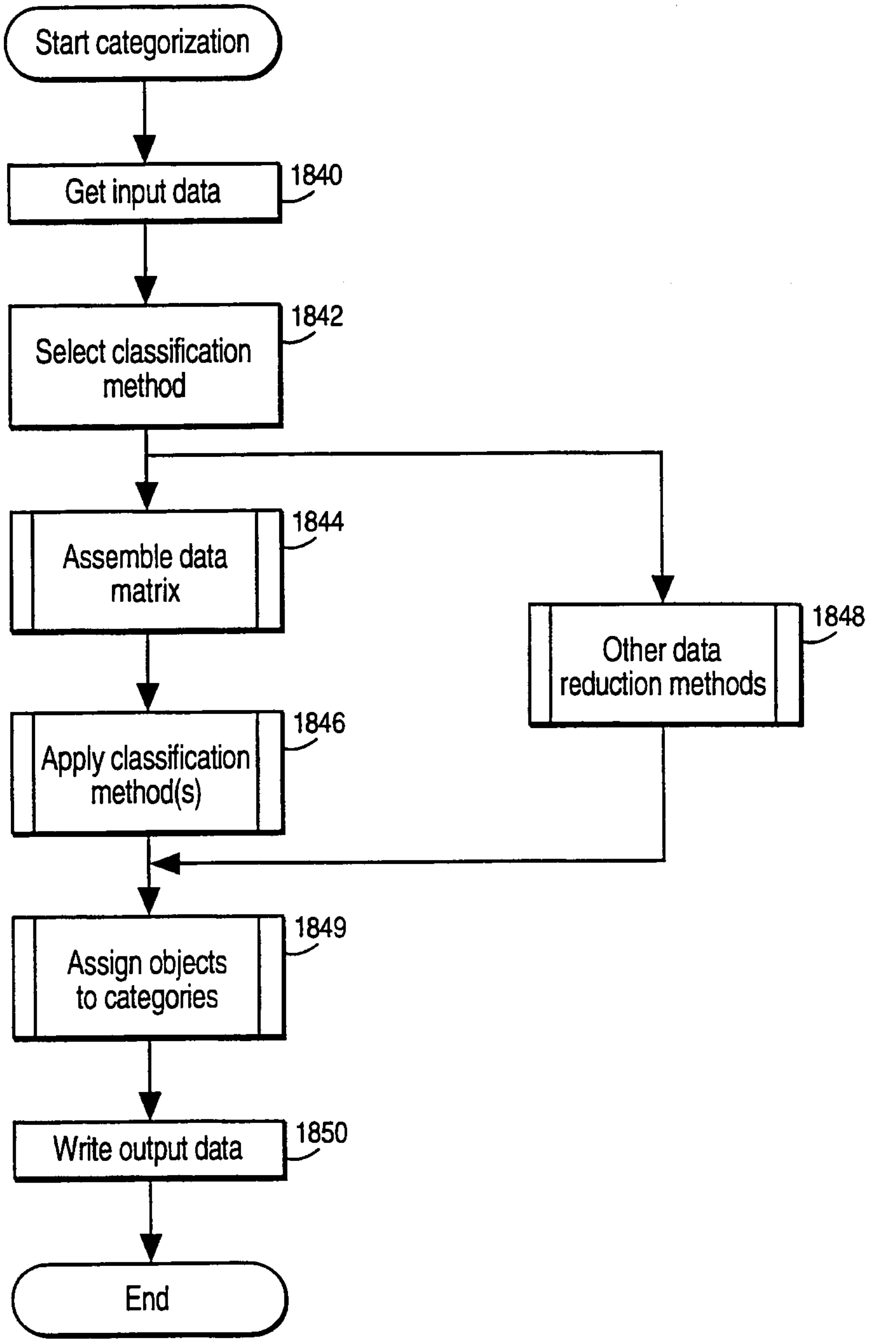

Start categorization
Get input data
1840
Select classification method
1842
Assemble data matrix
1844
Other data reduction methods
1848
Apply classification method(s)
1846
Assign objects to categories
1849
Write output data
1850
End

Fig. 18

Example Steps to Categorize Objects

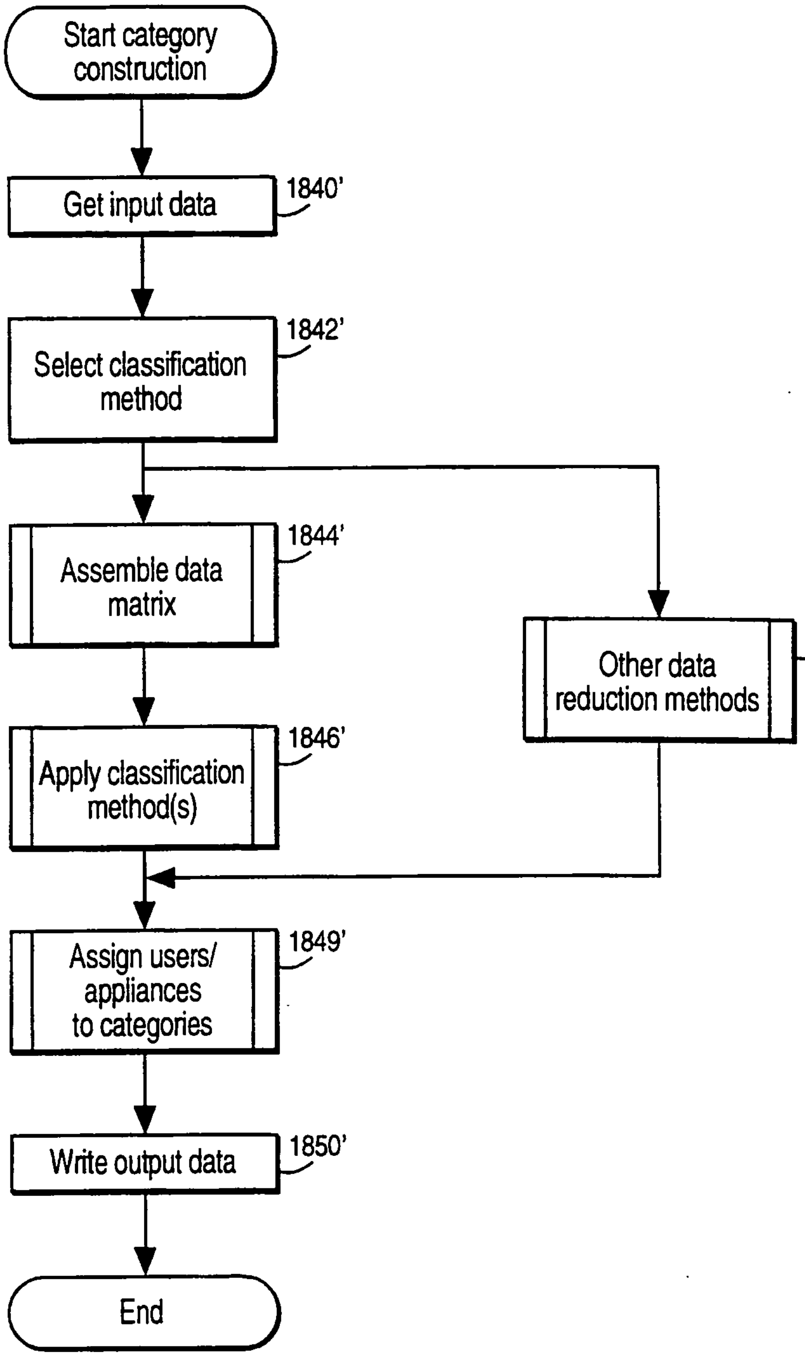
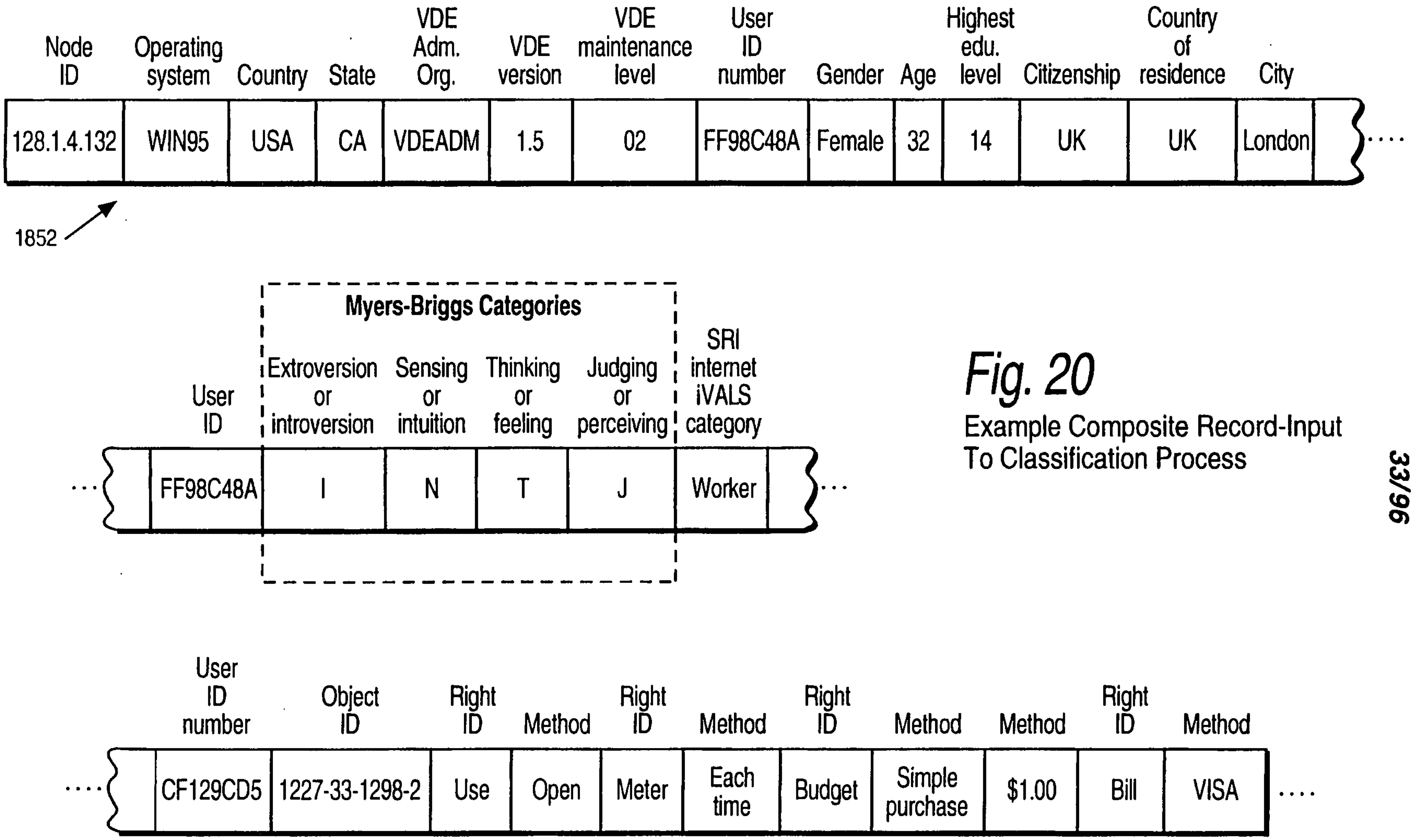

Start category construction
Get input data
1840'
Select classification method
1842'
Assemble data matrix
1844'
Apply classification method(s)
1846'
Other data reduction methods
1848'
Assign users/ appliances to categories
1849'
Write output data
1850'
End

Fig. 19

Example Steps to Categorize Users/Appliances



Node ID	Operating system	Country	State	VDE Adm. Org.	VDE version	VDE maintenance level	User ID number	Gender	Age	Highest edu. level	Citizenship	Country of residence	City
128.1.4.132	WIN95	USA	CA	VDEADM	1.5	02	FF98C48A	Female	32	14	UK	UK	London

User ID	Extroversion or introversion	Sensing or intuition	Thinking or feeling	Judging or perceiving	SRI internet iVALS category
FF98C48A	I	N	T	J	Worker

User ID number	Object ID	Right ID	Method	Right ID	Method	Right ID	Method	Method	Right ID	Method
CF129CD5	1227-33-1298-2	Use	Open	Meter	Each time	Budget	Simple purchase	$1.00	Bill	VISA

Fig. 20

Example Composite Record-Input To Classification Process

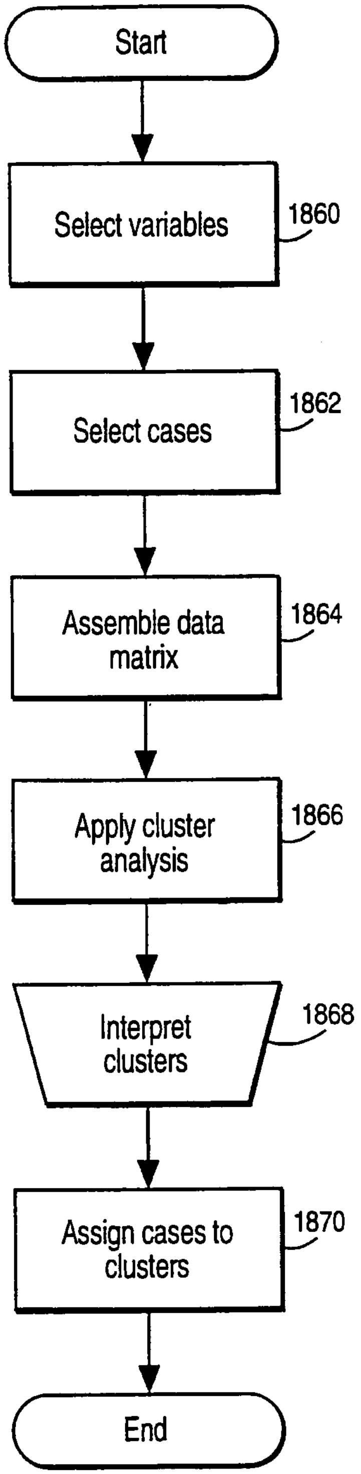

Start
Select variables
1860
Select cases
1862
Assemble data matrix
1864
Apply cluster analysis
1866
Interpret clusters
1868
Assign cases to clusters
1870
End

Fig. 21

Example Cluster Analysis Process

Variables	Typical Class 1-Profile	Typical Class 2-Profile
City	Washington, DC	Knoxville, TN
Av. price of content purchased last 30 days	$8.79	$1.95
Number of trips abroad in last 2 years	3	0
Type of content most frequently purchased	National and international news	Sports
2nd most frequently purchased	Business information	Religious
Third most frequently purchased	Travel information	Movies
Pay per view	No	Yes
Add new controls to content	Yes	No
Stated religious affiliation	None	Methodist
SRI internet lifestyle category	Surfer	Worker
Modification rights purchased	20% of text items	5% of text items

Fig. 22 Example Classification Output Illustrating Different Classes Based Upon Differing Profiles

Variables	Factor 1 Loadings	Factor 2 Loadings
Region of US	.82	.11
Family income	.90	-.09
Av. price of content purchased last 30 days	.72	.15
Number of trips abroad in last 2 years	.91	.09
Percent news, business	.79	-.12
Percent entertainment	-.69	.21
Add new controls to content	.88	.19
Religiosity	-.60	-.22
Participates in sports	-.21	.87
Watches team/individual sports on TV	-.11	.62
Owns a sports utility vehicle	.12	.72
Consumes beer/wine	-.18	.83
Male/female	.21	.92
Education beyond college	.45	-.45
Buys pay per view sports events	-.25	.77
Number of TVs in house	-.11	.66

Fig. 23 Example Classification Output Illustrating Principal Components Analysis On Parameter Data And Categories Data

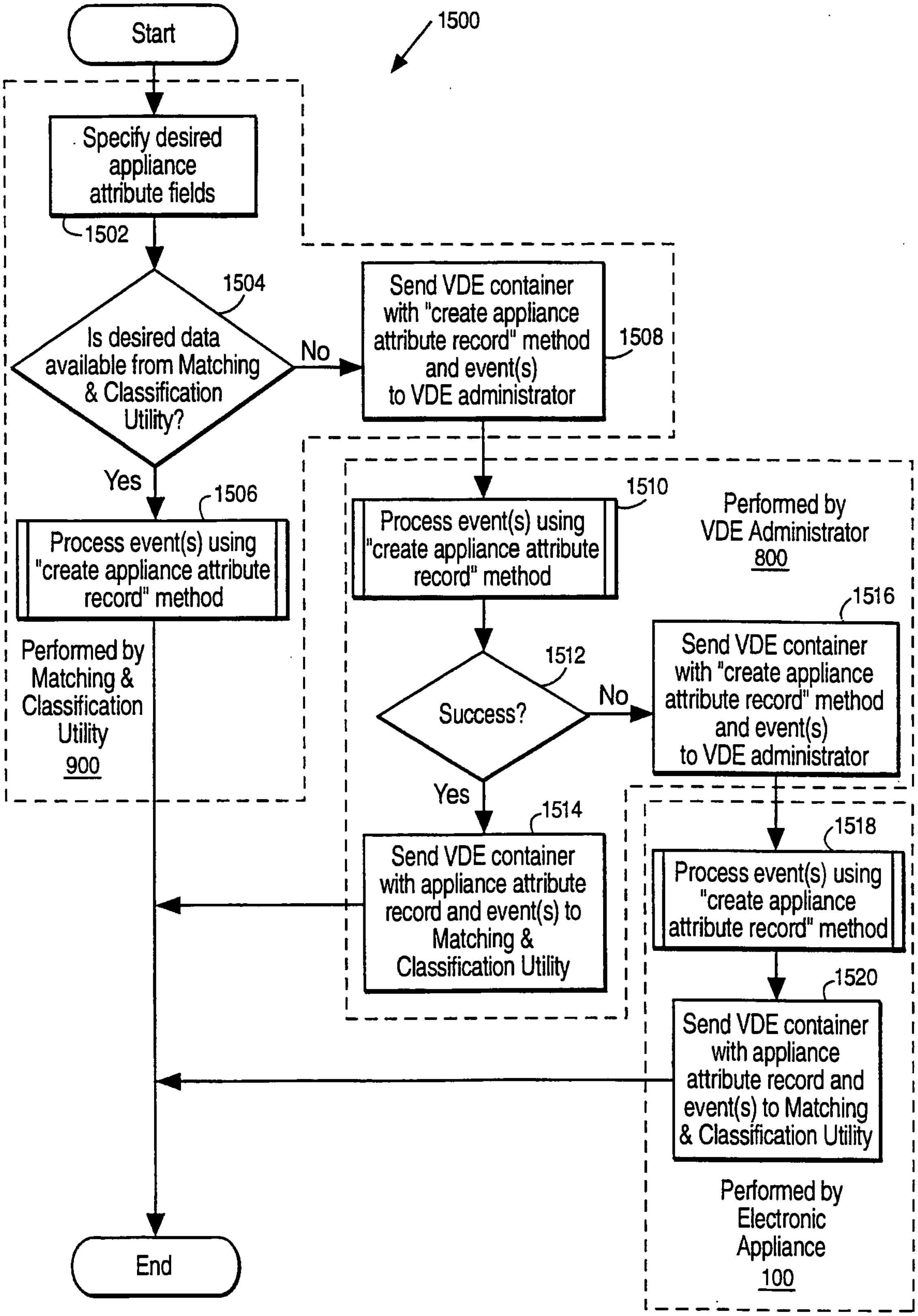

1500
Start
Specify desired appliance attribute fields
1502
1504
Is desired data available from Matching & Classification Utility?
No
Send VDE container with "create appliance attribute record" method and event(s) to VDE administrator
1508
Yes
1506
Process event(s) using "create appliance attribute record" method
Performed by Matching & Classification Utility 900
1510
Process event(s) using "create appliance attribute record" method
Performed by VDE Administrator 800
1516
1512
Success?
No
Send VDE container with "create appliance attribute record" method and event(s) to VDE administrator
Yes
1514
Send VDE container with appliance attribute record and event(s) to Matching & Classification Utility
1518
Process event(s) using "create appliance attribute record" method
1520
Send VDE container with appliance attribute record and event(s) to Matching & Classification Utility
Performed by Electronic Appliance 100
End

Fig. 24

Example Steps for Collecting Appliance Attribute Data

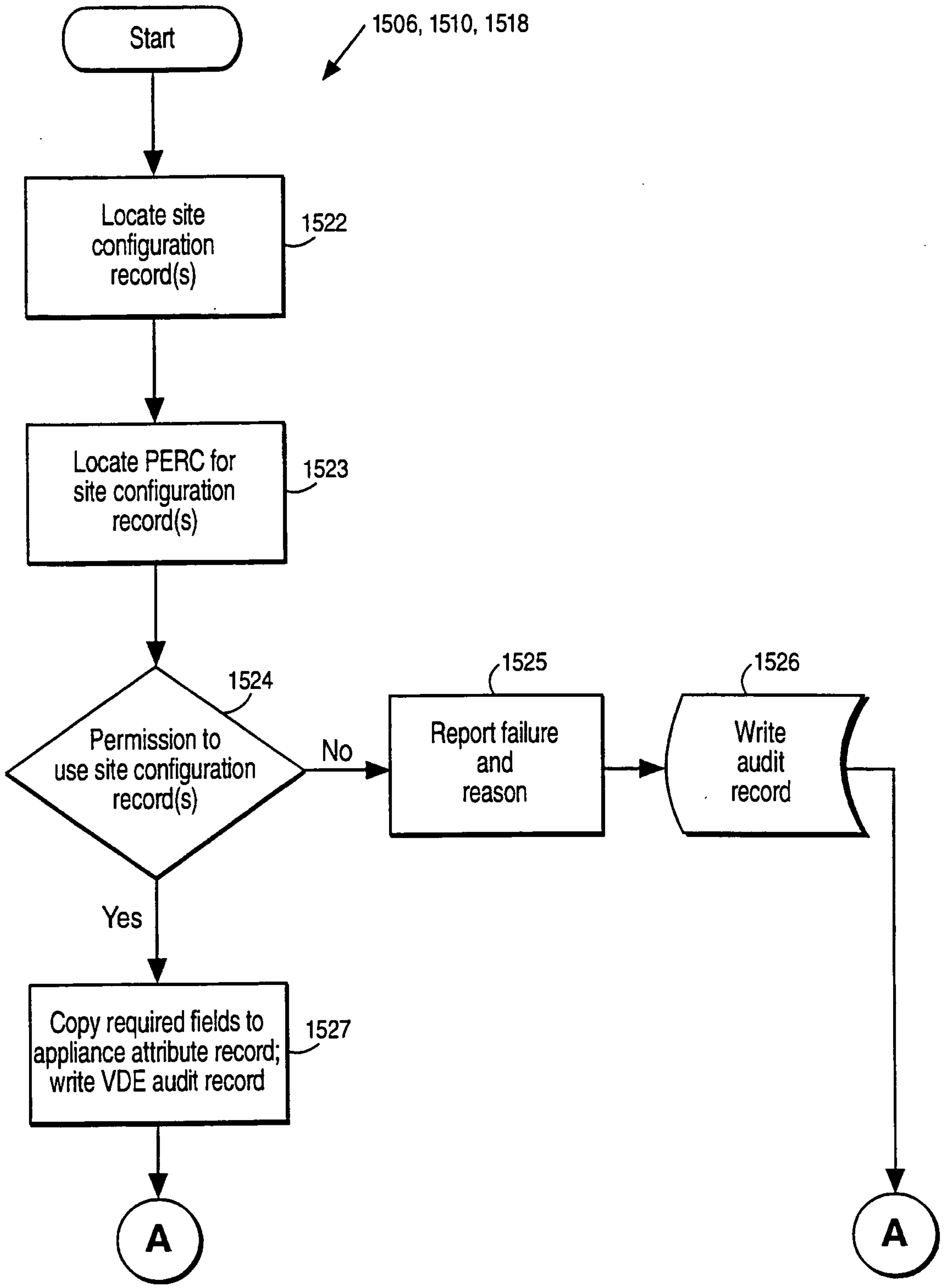

Start
1506, 1510, 1518
Locate site configuration record(s)
1522
Locate PERC for site configuration record(s)
1523
Permission to use site configuration record(s)
1524
No
Report failure and reason
1525
Write audit record
1526
Yes
Copy required fields to appliance attribute record; write VDE audit record
1527
A
A

Fig. 25(A)

Example Create Appliance Attribute Data Method steps

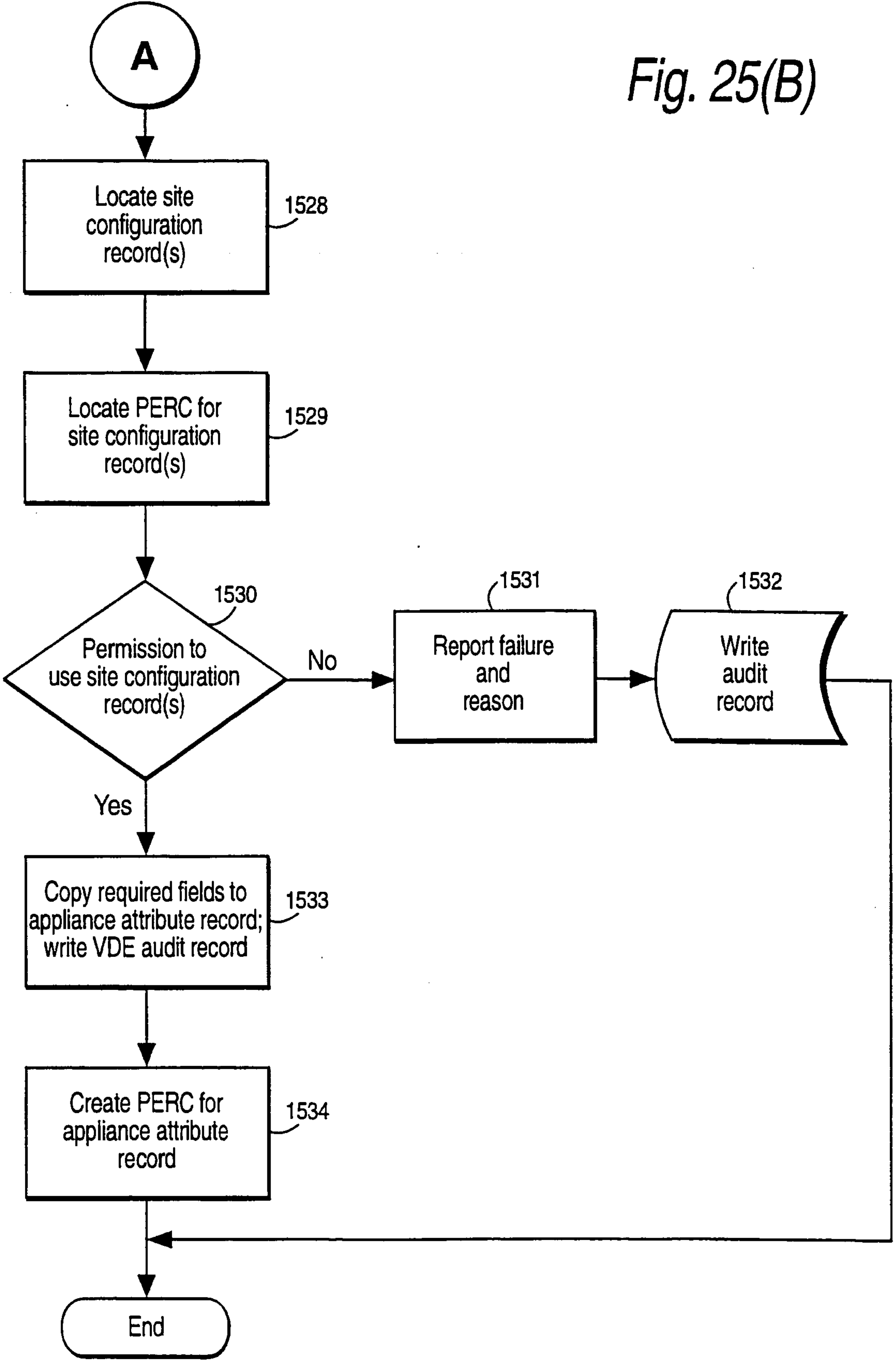

A
Locate site configuration record(s)
1528
Locate PERC for site configuration record(s)
1529
Permission to use site configuration record(s)
1530
No
Report failure and reason
1531
Write audit record
1532
Yes
Copy required fields to appliance attribute record; write VDE audit record
1533
Create PERC for appliance attribute record
1534
End

Fig. 25(B)

Fig. 26(A) Example Appliance Attribute Record

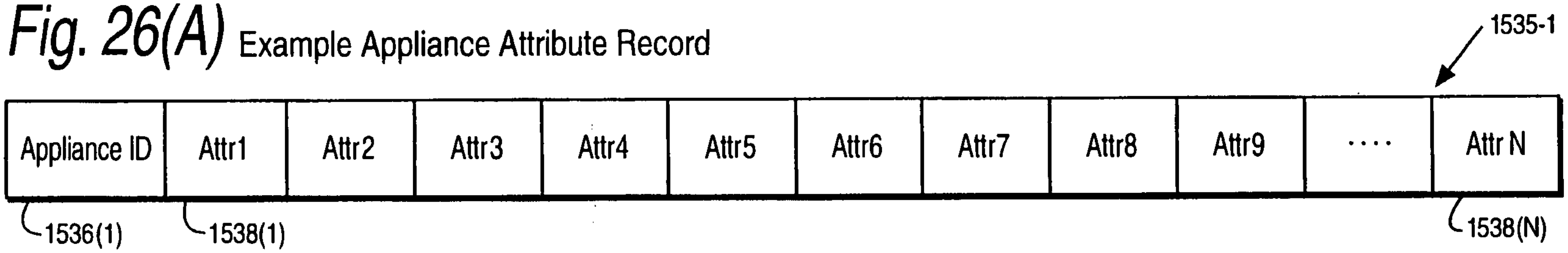
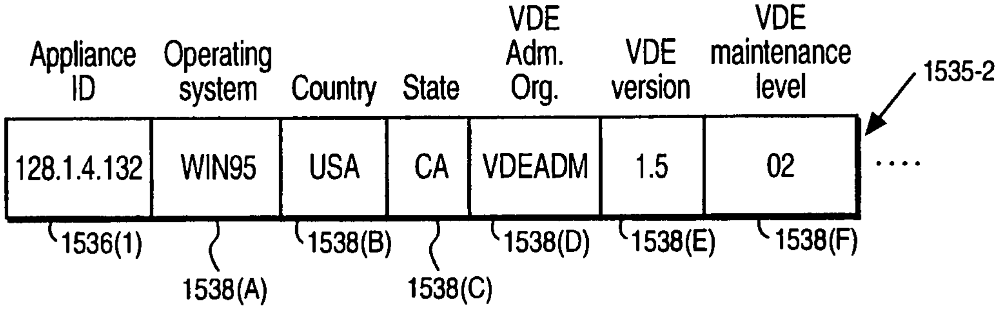

1535-1
Appliance ID
Attr1
Attr2
Attr3
Attr4
Attr5
Attr6
Attr7
Attr8
Attr9
....
Attr N
1536(1)
1538(1)
1538(N)

Fig. 26(B)


Appliance ID
Operating system
Country
State
VDE Adm. Org.
VDE version
VDE maintenance level
1535-2
128.1.4.132
WIN95
USA
CA
VDEADM
1.5
02
....
1536(1)
1538(A)
1538(B)
1538(C)
1538(D)
1538(E)
1538(F)

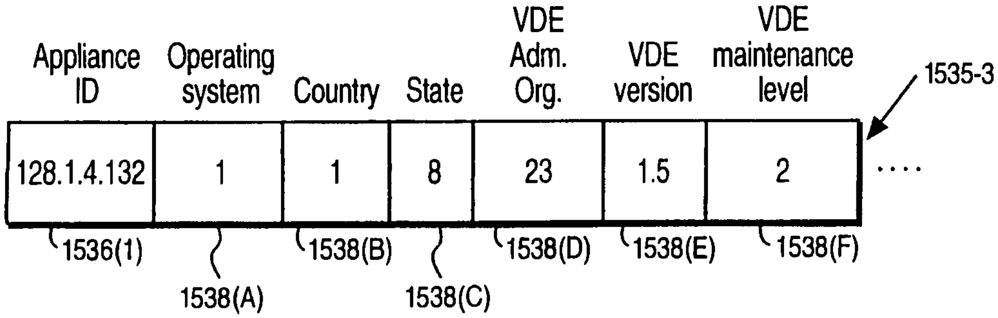

Appliance ID
Operating system
Country
State
VDE Adm. Org.
VDE version
VDE maintenance level
1535-3
128.1.4.132
1
1
8
23
1.5
2
....
1536(1)
1538(A)
1538(B)
1538(C)
1538(D)
1538(E)
1538(F)

Fig. 26(C)
Example Appliance Attribute Record

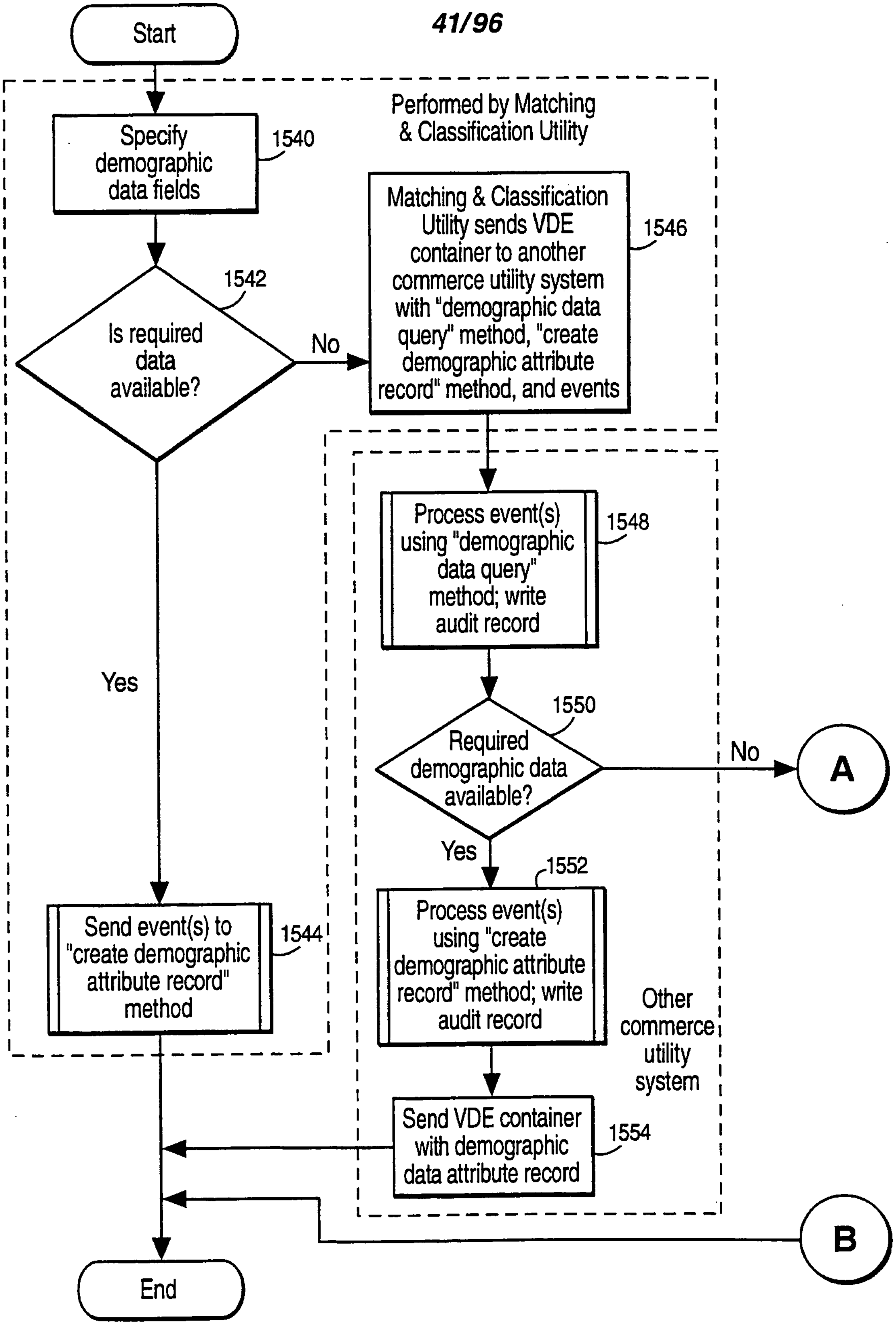
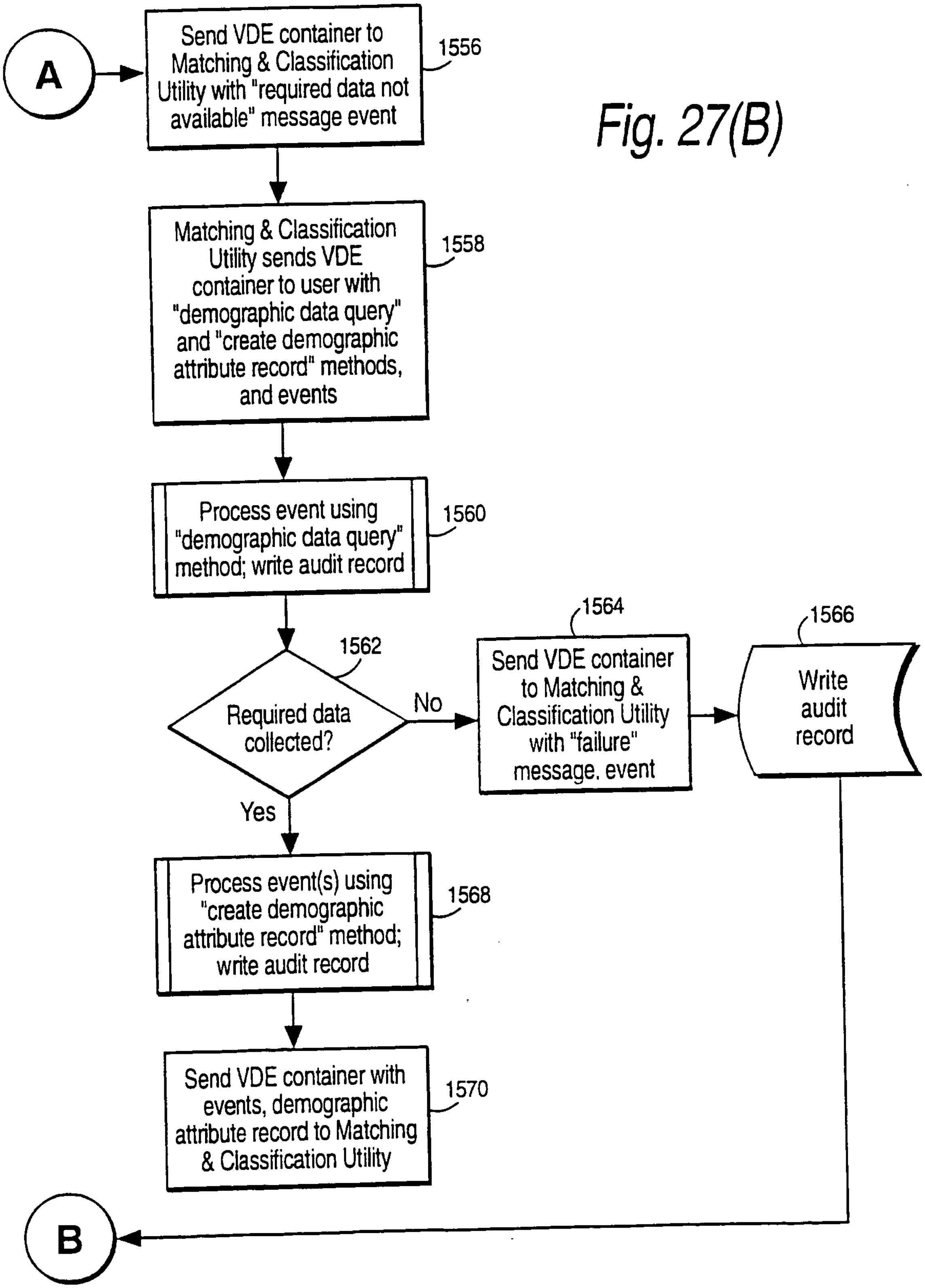
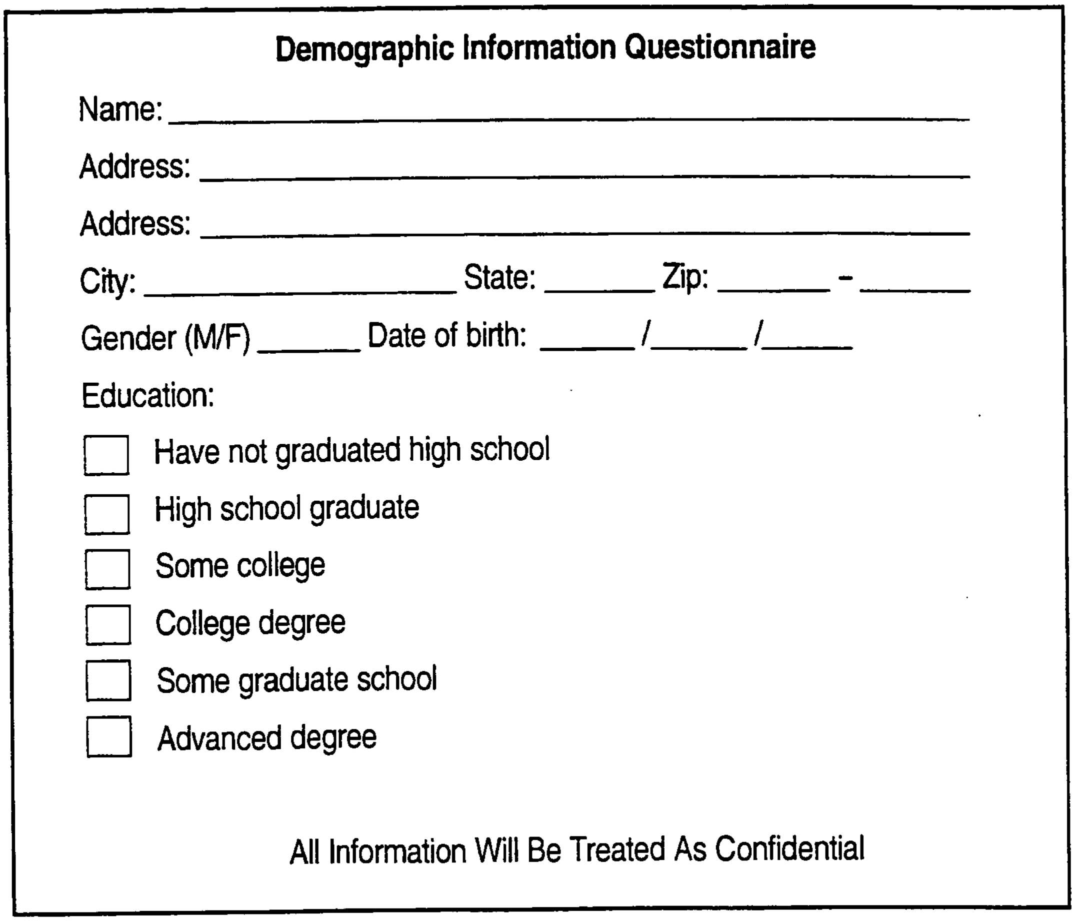

Start
Performed by Matching & Classification Utility
Specify demographic data fields
1540
Is required data available?
1542
No
Matching & Classification Utility sends VDE container to another commerce utility system with "demographic data query" method, "create demographic attribute record" method, and events
1546
Yes
Process event(s) using "demographic data query" method; write audit record
1548
Required demographic data available?
1550
No
A
Yes
Process event(s) using "create demographic attribute record" method; write audit record
1552
Send event(s) to "create demographic attribute record" method
1544
Other commerce utility system
Send VDE container with demographic data attribute record
1554
B
End

Fig. 27(A) Example Steps for Collecting Demographic Data


A
Send VDE container to Matching & Classification Utility with "required data not available" message event
1556
Fig. 27(B)
Matching & Classification Utility sends VDE container to user with "demographic data query" and "create demographic attribute record" methods, and events
1558
Process event using "demographic data query" method; write audit record
1560
1562
Required data collected?
No
1564
Send VDE container to Matching & Classification Utility with "failure" message. event
1566
Write audit record
Yes
Process event(s) using "create demographic attribute record" method; write audit record
1568
Send VDE container with events, demographic attribute record to Matching & Classification Utility
1570
B



Demographic Information Questionnaire

Name: ______________________________

Address: ______________________________

Address: ______________________________

City: ______________ State: ______ Zip: ______ - ______

Gender (M/F) ______ Date of birth: ______/______/______

Education:

☐ Have not graduated high school

☐ High school graduate

☐ Some college

☐ College degree

☐ Some graduate school

☐ Advanced degree

All Information Will Be Treated As Confidential

Fig. 28 Example Demographic Questionnaire “Pop-Up” Screen

Fig. 29(A) Example User Demographic Attribute Information Record

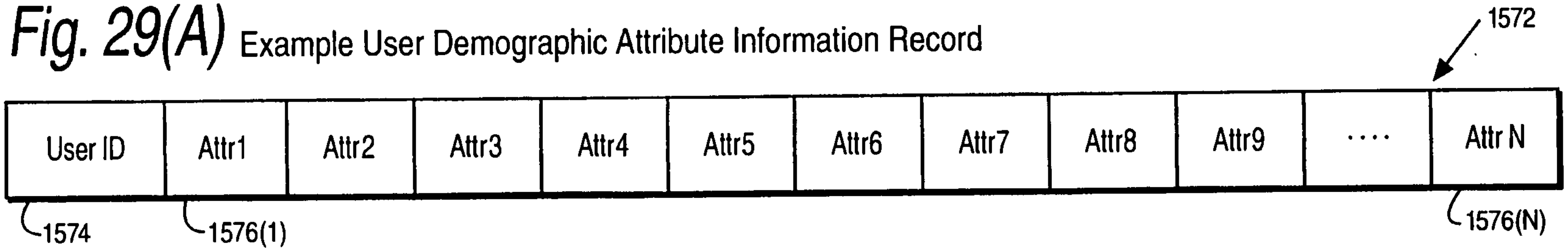

1572
User ID
Attr1
Attr2
Attr3
Attr4
Attr5
Attr6
Attr7
Attr8
Attr9
....
Attr N
1574
1576(1)
1576(N)

Fig. 29(B) Example Demographic Attribute Record

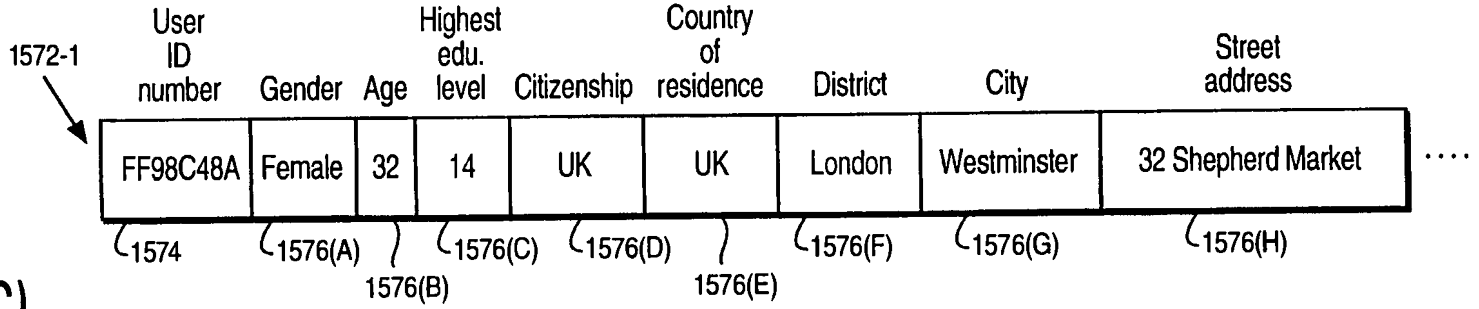
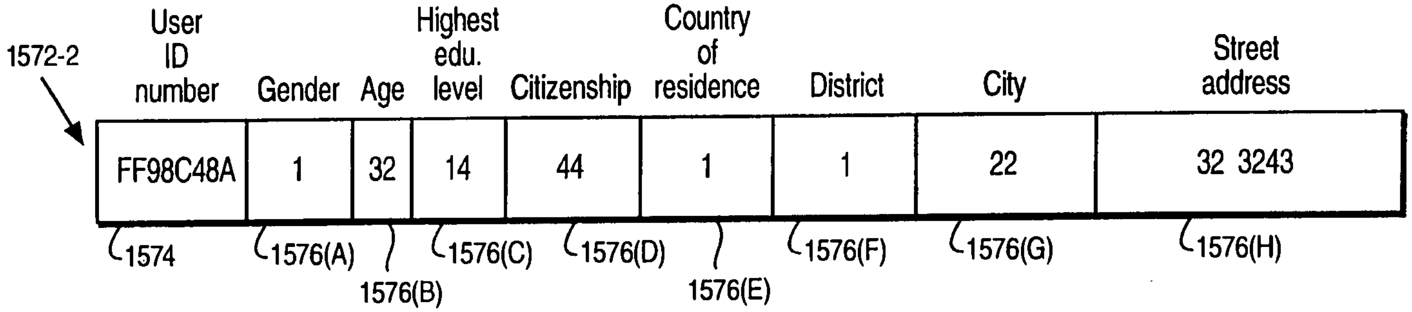

1572-1
User ID number
Gender
Age
Highest edu. level
Citizenship
Country of residence
District
City
Street address
FF98C48A
Female
32
14
UK
UK
London
Westminster
32 Shepherd Market
....
1574
1576(A)
1576(B)
1576(C)
1576(D)
1576(E)
1576(F)
1576(G)
1576(H)

Fig. 29(C)
Example Demographic Attribute Record


1572-2
User ID number
Gender
Age
Highest edu. level
Citizenship
Country of residence
District
City
Street address
FF98C48A
1
32
14
44
1
1
22
32 3243
1574
1576(A)
1576(B)
1576(C)
1576(D)
1576(E)
1576(F)
1576(G)
1576(H)

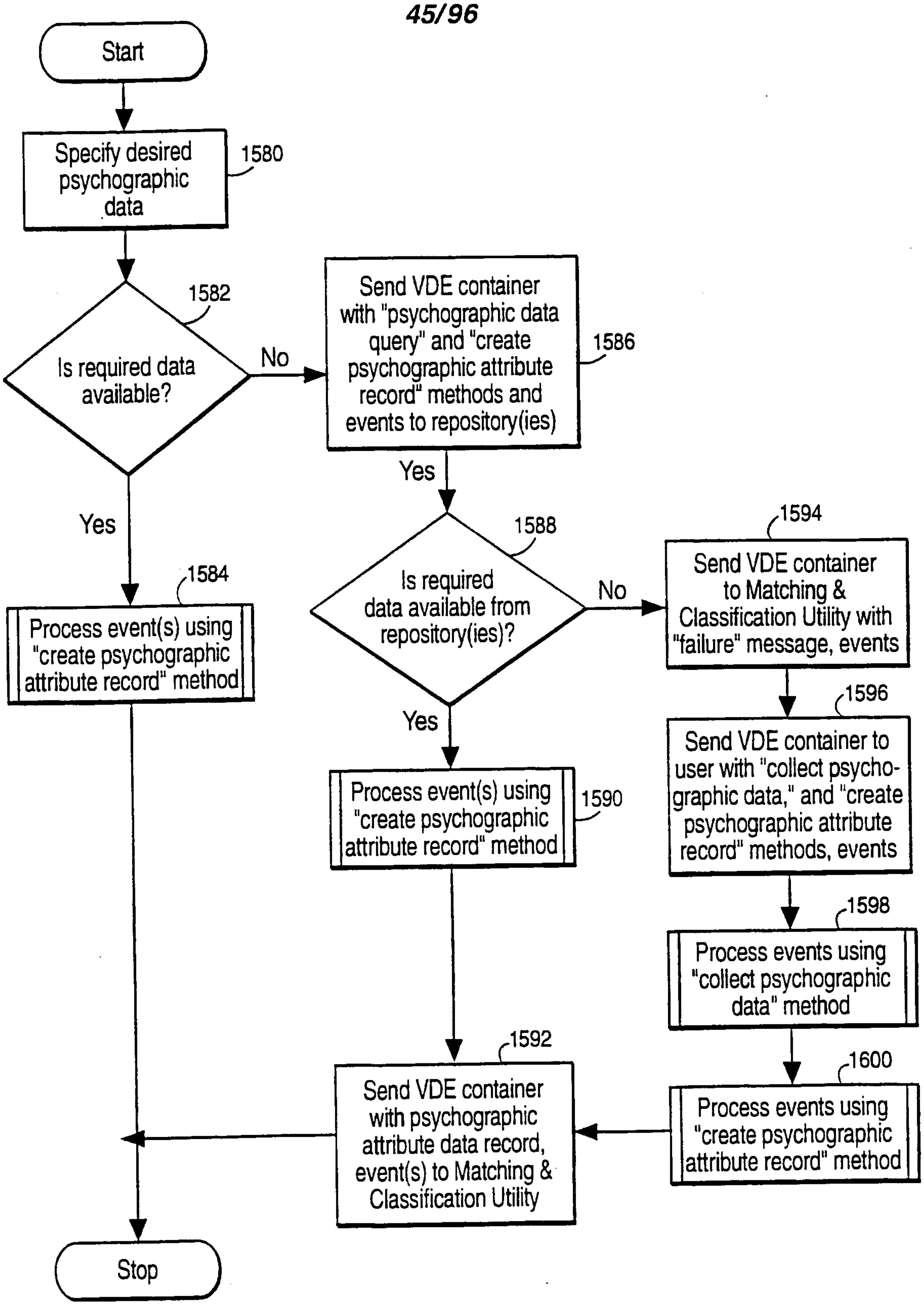
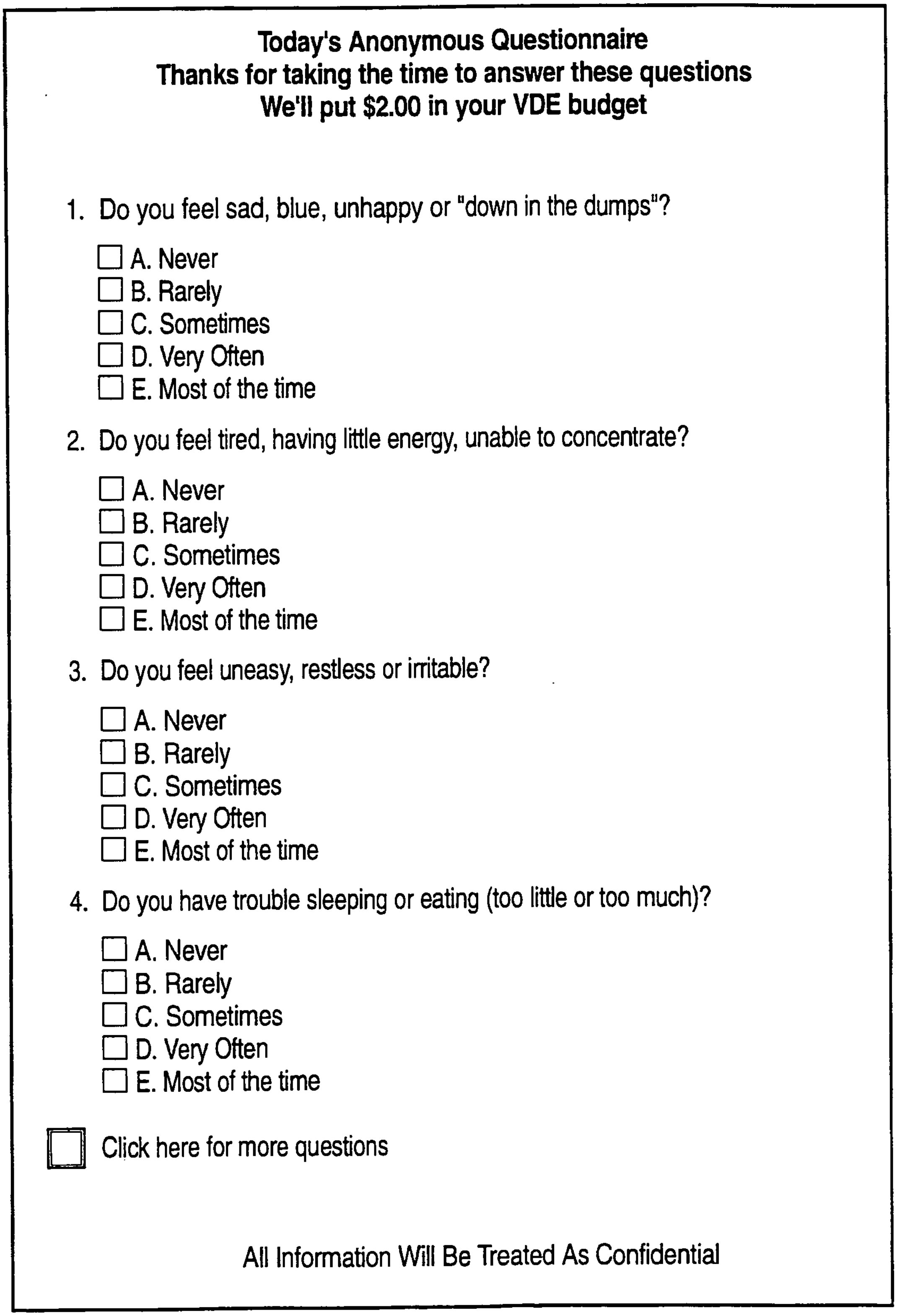

Start
Specify desired psychographic data
1580
Is required data available?
1582
No
Send VDE container with "psychographic data query" and "create psychographic attribute record" methods and events to repository(ies)
1586
Yes
Process event(s) using "create psychographic attribute record" method
1584
Yes
Is required data available from repository(ies)?
1588
No
Send VDE container to Matching & Classification Utility with "failure" message, events
1594
Yes
Process event(s) using "create psychographic attribute record" method
1590
Send VDE container to user with "collect psychographic data," and "create psychographic attribute record" methods, events
1596
Process events using "collect psychographic data" method
1598
Send VDE container with psychographic attribute data record, event(s) to Matching & Classification Utility
1592
Process events using "create psychographic attribute record" method
1600
Stop

Fig. 30 Example Steps for Collecting Psychographic Data



Today's Anonymous Questionnaire
Thanks for taking the time to answer these questions
We'll put $2.00 in your VDE budget

1. Do you feel sad, blue, unhappy or "down in the dumps"?

   ☐ A. Never
   ☐ B. Rarely
   ☐ C. Sometimes
   ☐ D. Very Often
   ☐ E. Most of the time

2. Do you feel tired, having little energy, unable to concentrate?

   ☐ A. Never
   ☐ B. Rarely
   ☐ C. Sometimes
   ☐ D. Very Often
   ☐ E. Most of the time

3. Do you feel uneasy, restless or irritable?

   ☐ A. Never
   ☐ B. Rarely
   ☐ C. Sometimes
   ☐ D. Very Often
   ☐ E. Most of the time

4. Do you have trouble sleeping or eating (too little or too much)?

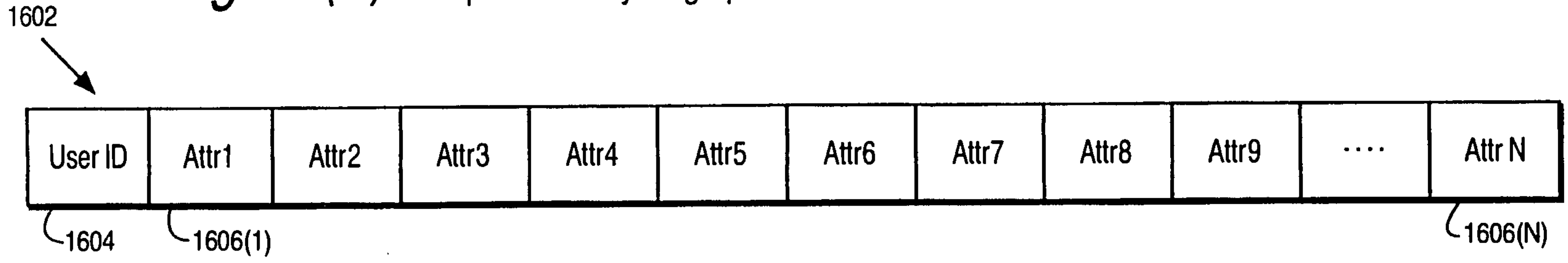
   ☐ A. Never
   ☐ B. Rarely
   ☐ C. Sometimes
   ☐ D. Very Often
   ☐ E. Most of the time

☐ Click here for more questions

All Information Will Be Treated As Confidential

Fig. 31 Example Psychographic Questionnaire “Pop-Up” Screen

Fig. 32(A) Example User Psychographic Attribute Information Record


1602
User ID
Attr1
Attr2
Attr3
Attr4
Attr5
Attr6
Attr7
Attr8
Attr9
....
Attr N
1604
1606(1)
1606(N)

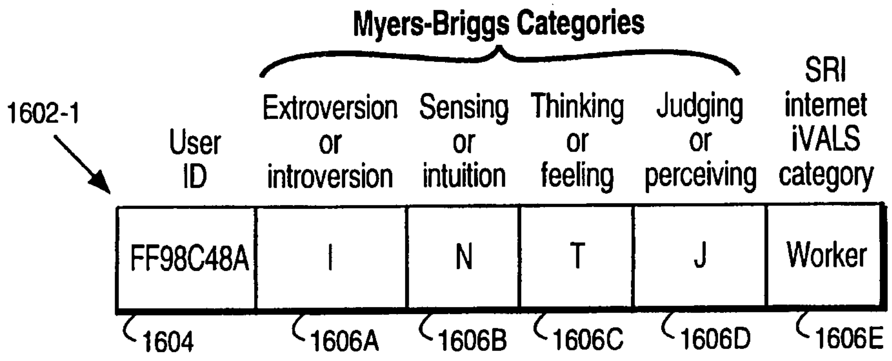

Myers-Briggs Categories
1602-1
User ID
Extroversion or introversion
Sensing or intuition
Thinking or feeling
Judging or perceiving
SRI internet iVALS category
FF98C48A
I
N
T
J
Worker
1604
1606A
1606B
1606C
1606D
1606E

Fig. 32(B)
Example User Psychographic Attribute Record

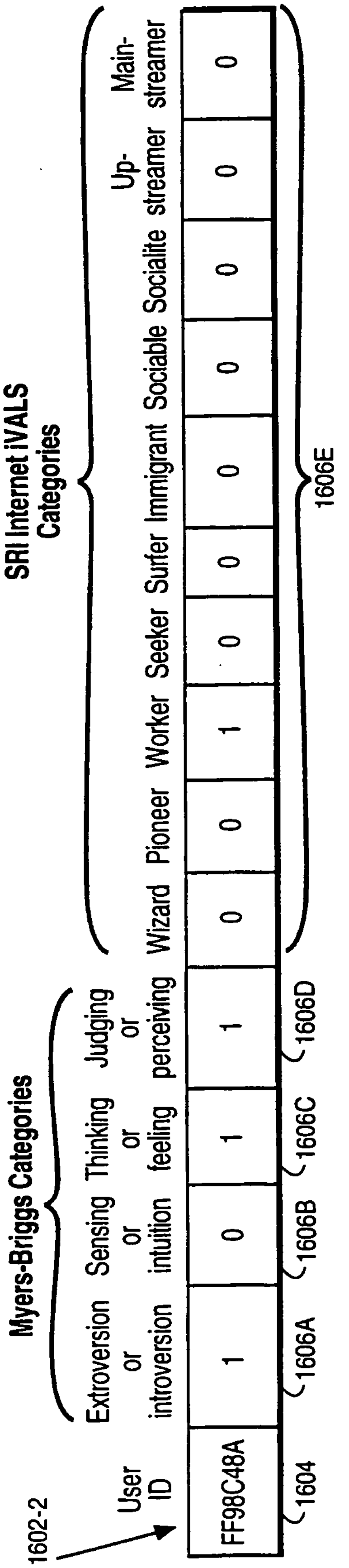

SRI Internet iVALS Categories
Myers-Briggs Categories
1602-2
User ID
FF98C48A
1604
Extroversion or introversion
1
1606A
Sensing or intuition
0
1606B
Thinking or feeling
1
1606C
Judging or perceiving
1
1606D
Wizard
0
Pioneer
0
Worker
1
Seeker
0
Surfer
0
Immigrant
0
Sociable
0
Socialite
0
Up-streamer
0
Main-streamer
0
1606E

Fig. 32(C) Example Psychographic Attribute Record

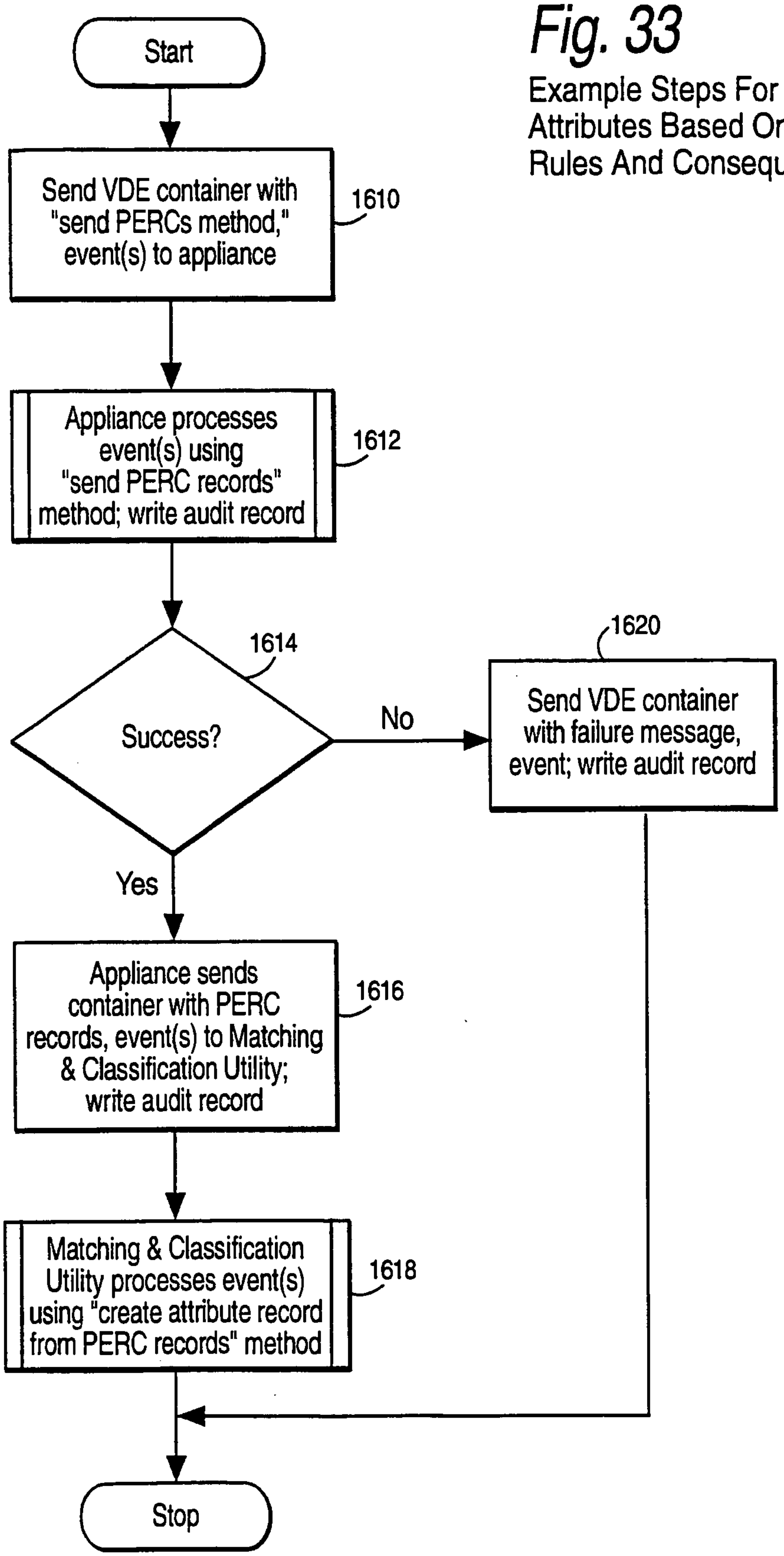

Start
Send VDE container with "send PERCs method," event(s) to appliance
1610
Appliance processes event(s) using "send PERC records" method; write audit record
1612
Success?
1614
No
Yes
Send VDE container with failure message, event; write audit record
1620
Appliance sends container with PERC records, event(s) to Matching & Classification Utility; write audit record
1616
Matching & Classification Utility processes event(s) using "create attribute record from PERC records" method
1618
Stop

Fig. 33

Example Steps For Determining Attributes Based On Available Rules And Consequences

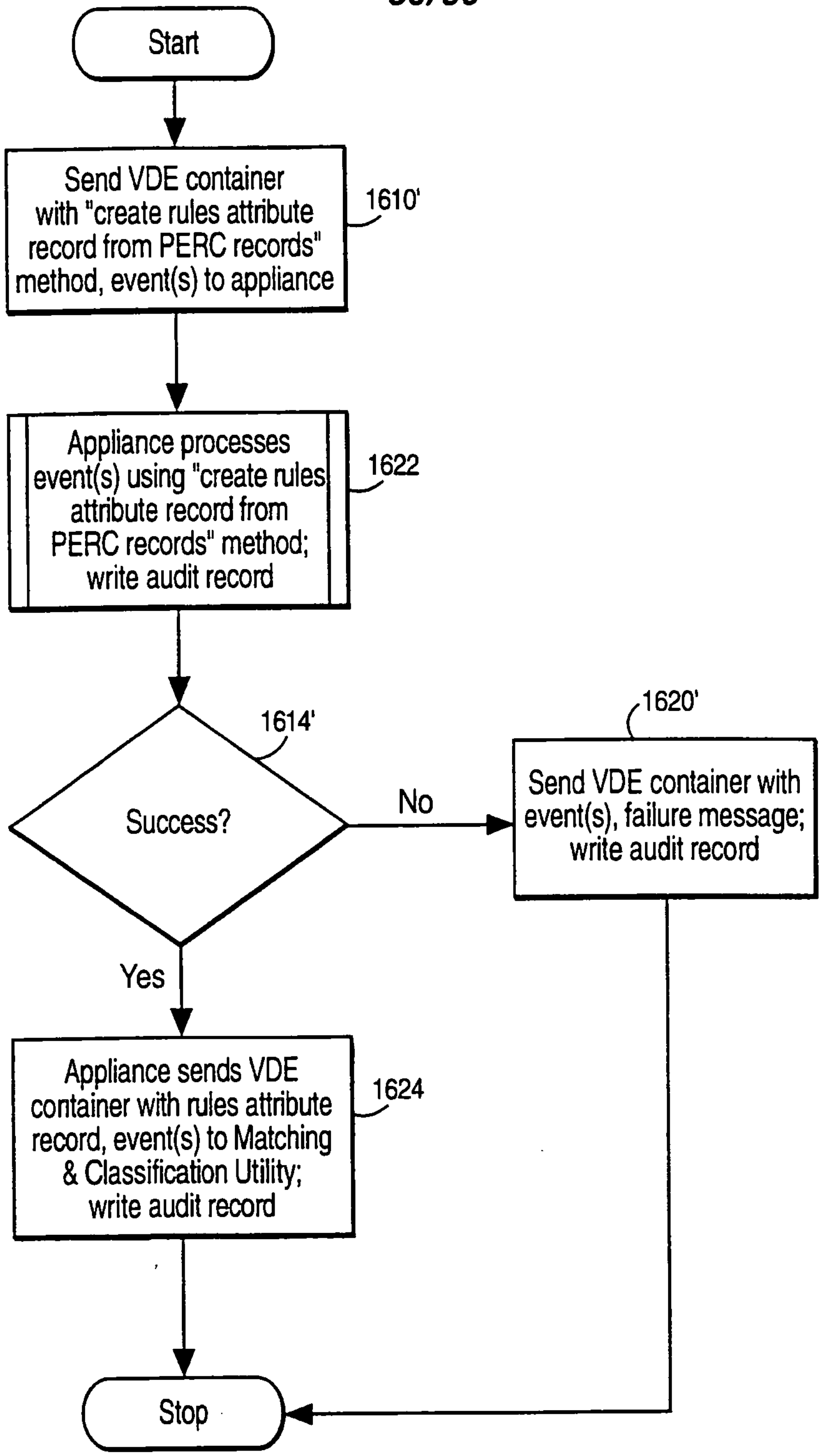

Start
Send VDE container with "create rules attribute record from PERC records" method, event(s) to appliance
1610'
Appliance processes event(s) using "create rules attribute record from PERC records" method; write audit record
1622
Success?
1614'
No
Send VDE container with event(s), failure message; write audit record
1620'
Yes
Appliance sends VDE container with rules attribute record, event(s) to Matching & Classification Utility; write audit record
1624
Stop

Fig. 34

Example Steps For Determining Attributes Based On Available Rules And Consequences

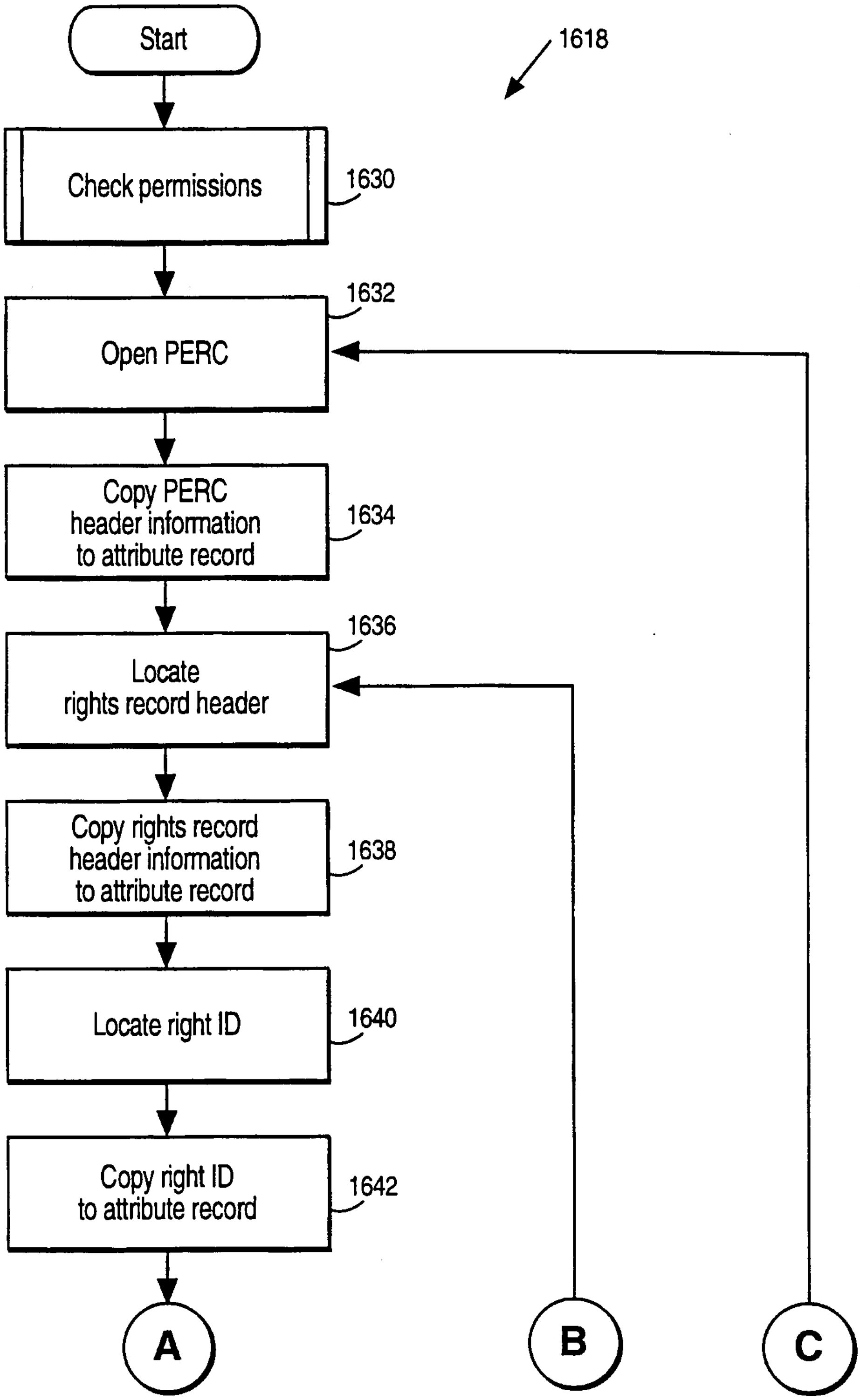

Start
1618
Check permissions
1630
Open PERC
1632
Copy PERC header information to attribute record
1634
Locate rights record header
1636
Copy rights record header information to attribute record
1638
Locate right ID
1640
Copy right ID to attribute record
1642
A
B
C

Fig. 35(A)

Construct Attribute Records From PERC Records Example Method

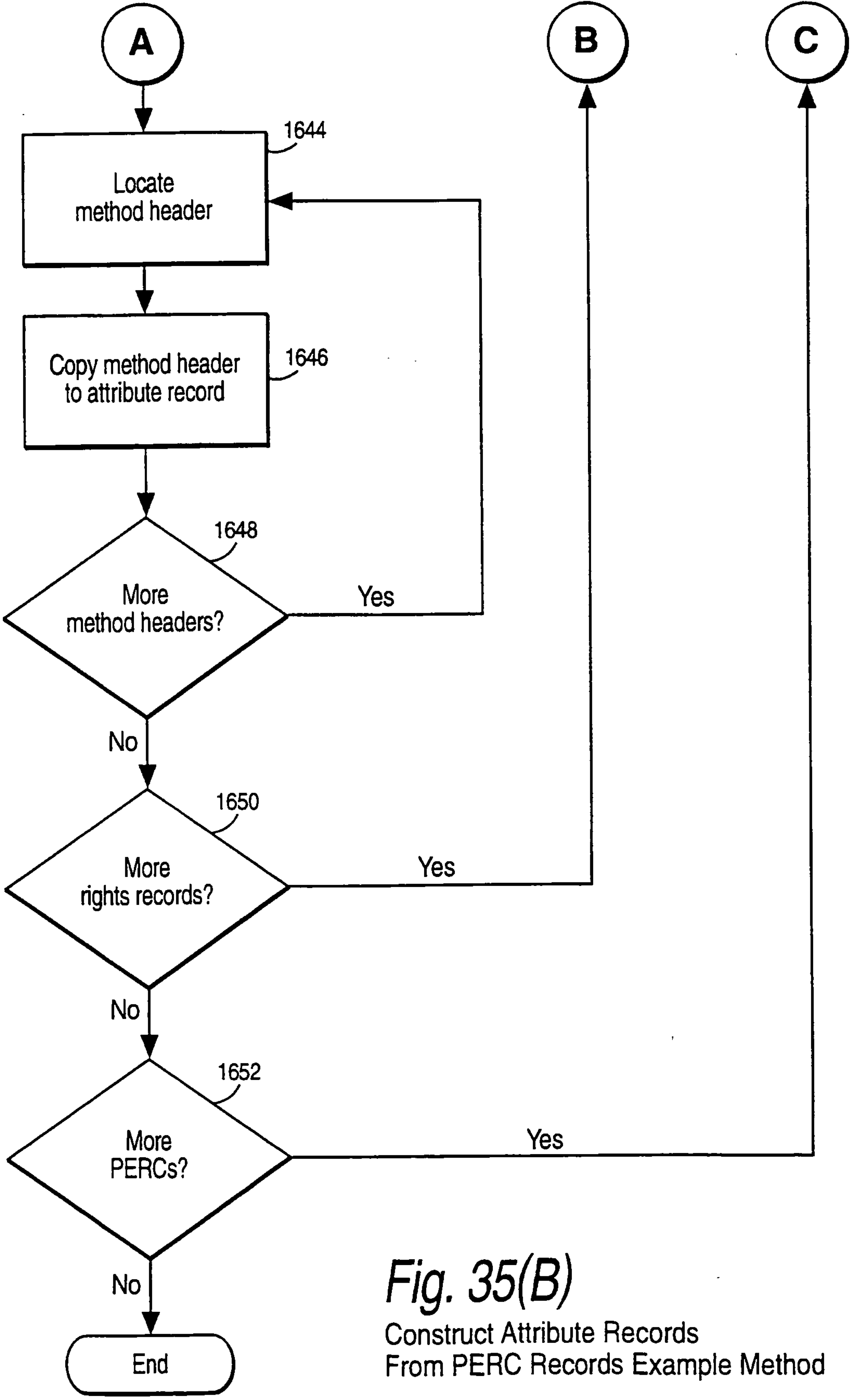

A
B
C
Locate method header
1644
Copy method header to attribute record
1646
More method headers?
1648
Yes
No
More rights records?
1650
Yes
No
More PERCs?
1652
Yes
No
End

Fig. 35(B)

Construct Attribute Records
From PERC Records Example Method

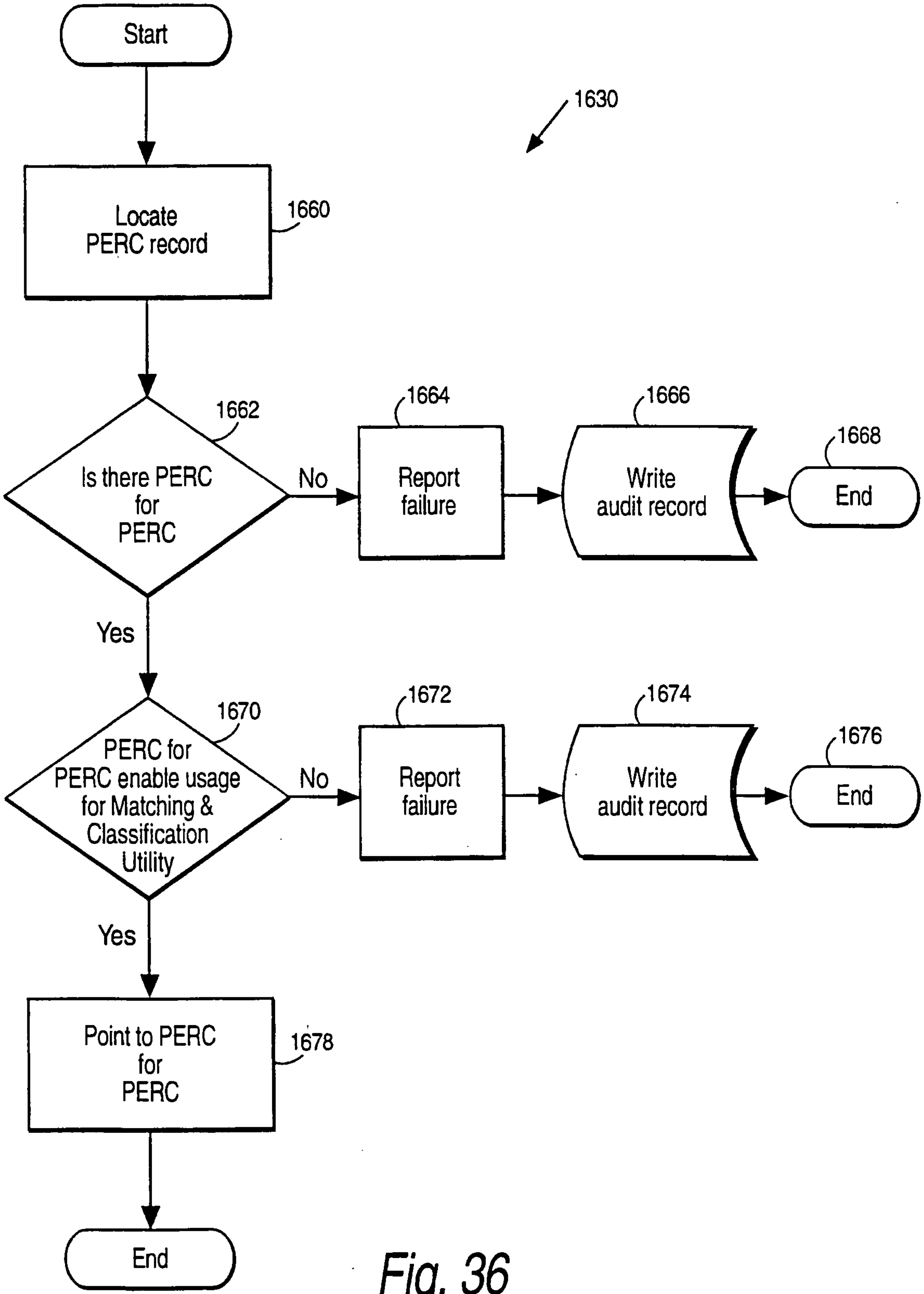
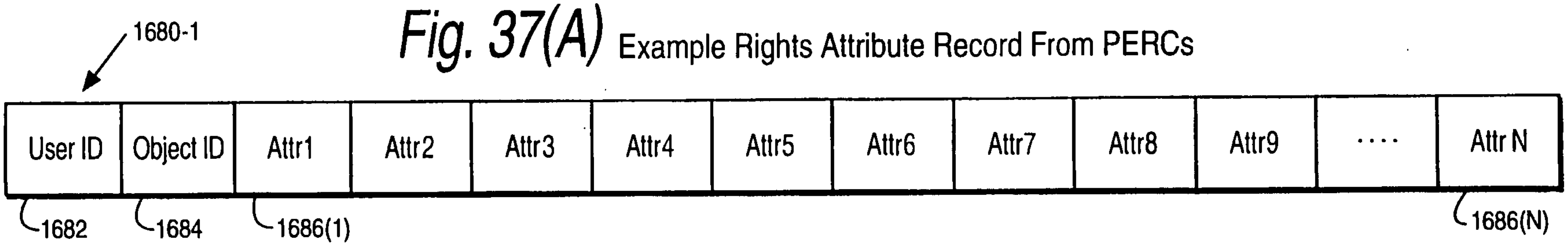

Start
1630
Locate PERC record
1660
Is there PERC for PERC
1662
No
Report failure
1664
Write audit record
1666
End
1668
Yes
PERC for PERC enable usage for Matching & Classification Utility
1670
No
Report failure
1672
Write audit record
1674
End
1676
Yes
Point to PERC for PERC
1678
End

Fig. 36

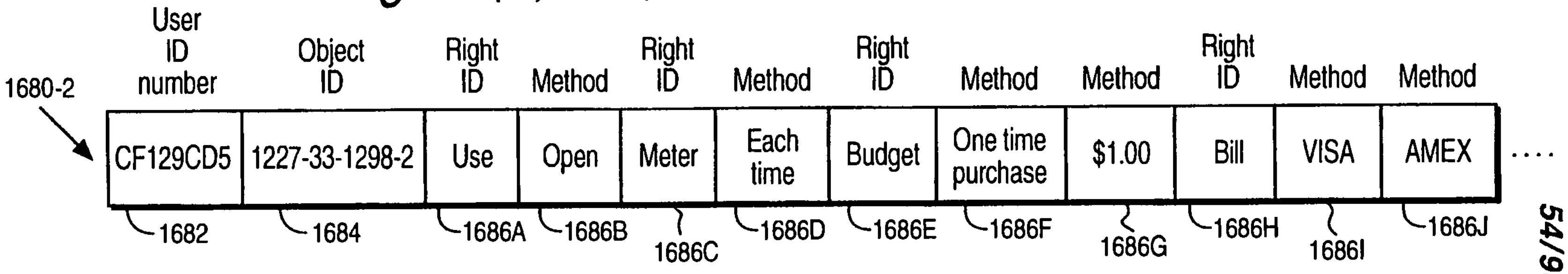
Check Permissions Record Example Steps

Fig. 37(A) Example Rights Attribute Record From PERCs


1680-1
1682
1684
1686(1)
1686(N)

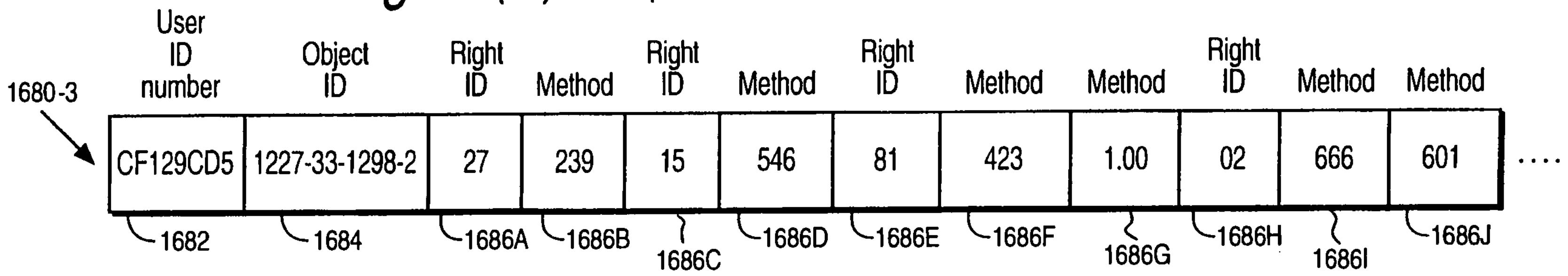
User ID	Object ID	Attr1	Attr2	Attr3	Attr4	Attr5	Attr6	Attr7	Attr8	Attr9	....	Attr N

Fig. 37(B) Example Attribute Record


1680-2
1682
1684
1686A
1686B
1686C
1686D
1686E
1686F
1686G
1686H
1686I
1686J

User ID number	Object ID	Right ID	Method	Right ID	Method	Right ID	Method	Method	Right ID	Method	Method	
CF129CD5	1227-33-1298-2	Use	Open	Meter	Each time	Budget	One time purchase	$1.00	Bill	VISA	AMEX	....

Fig. 37(C) Example Attribute Records From PERC Record


1680-3
1682
1684
1686A
1686B
1686C
1686D
1686E
1686F
1686G
1686H
1686I
1686J

User ID number	Object ID	Right ID	Method	Right ID	Method	Right ID	Method	Method	Right ID	Method	Method	
CF129CD5	1227-33-1298-2	27	239	15	546	81	423	1.00	02	666	601	....

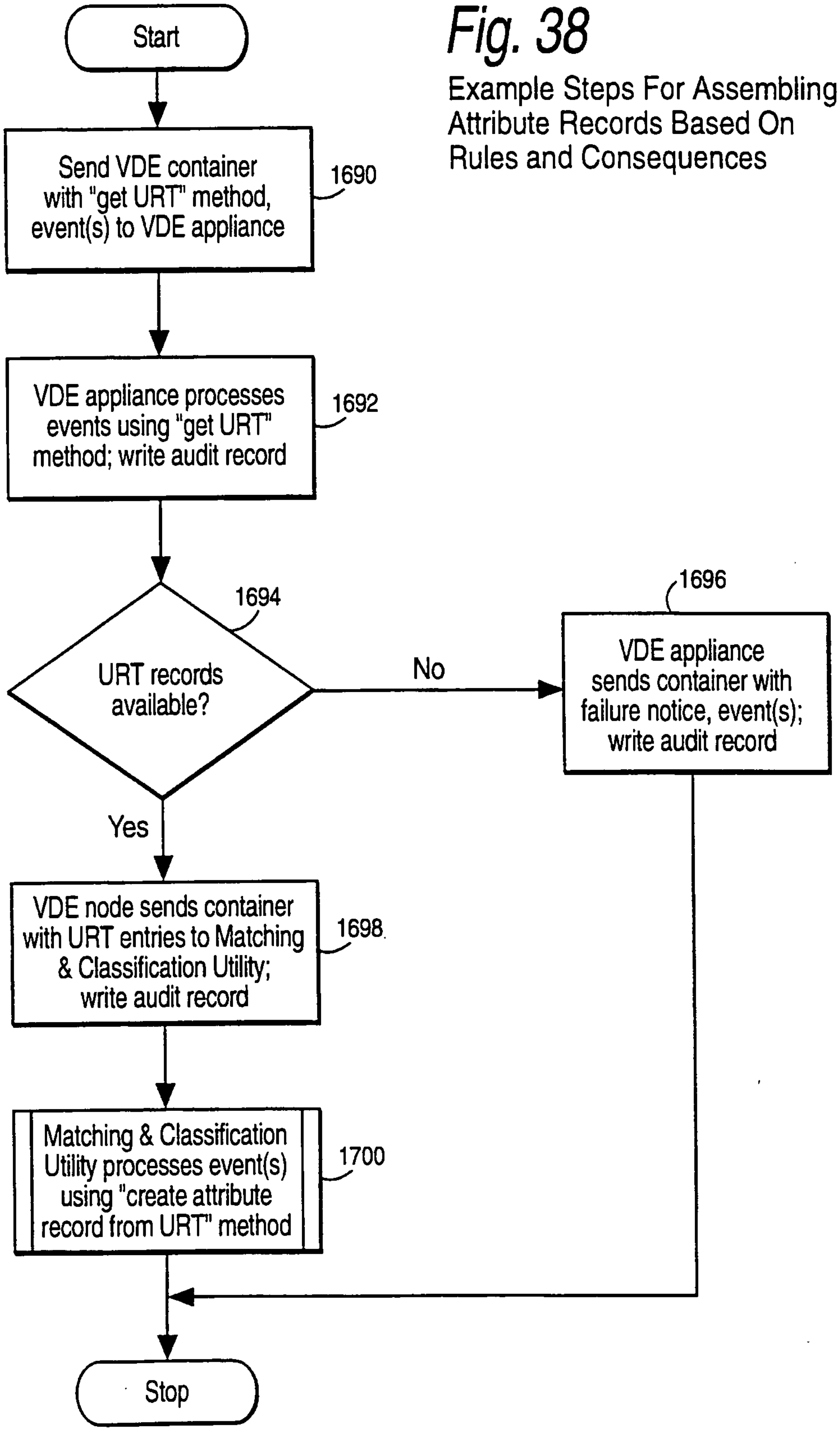

Start
Send VDE container with "get URT" method, event(s) to VDE appliance
1690
VDE appliance processes events using "get URT" method; write audit record
1692
URT records available?
1694
No
1696
VDE appliance sends container with failure notice, event(s); write audit record
Yes
VDE node sends container with URT entries to Matching & Classification Utility; write audit record
1698
Matching & Classification Utility processes event(s) using "create attribute record from URT" method
1700
Stop

Fig. 38

Example Steps For Assembling Attribute Records Based On Rules and Consequences

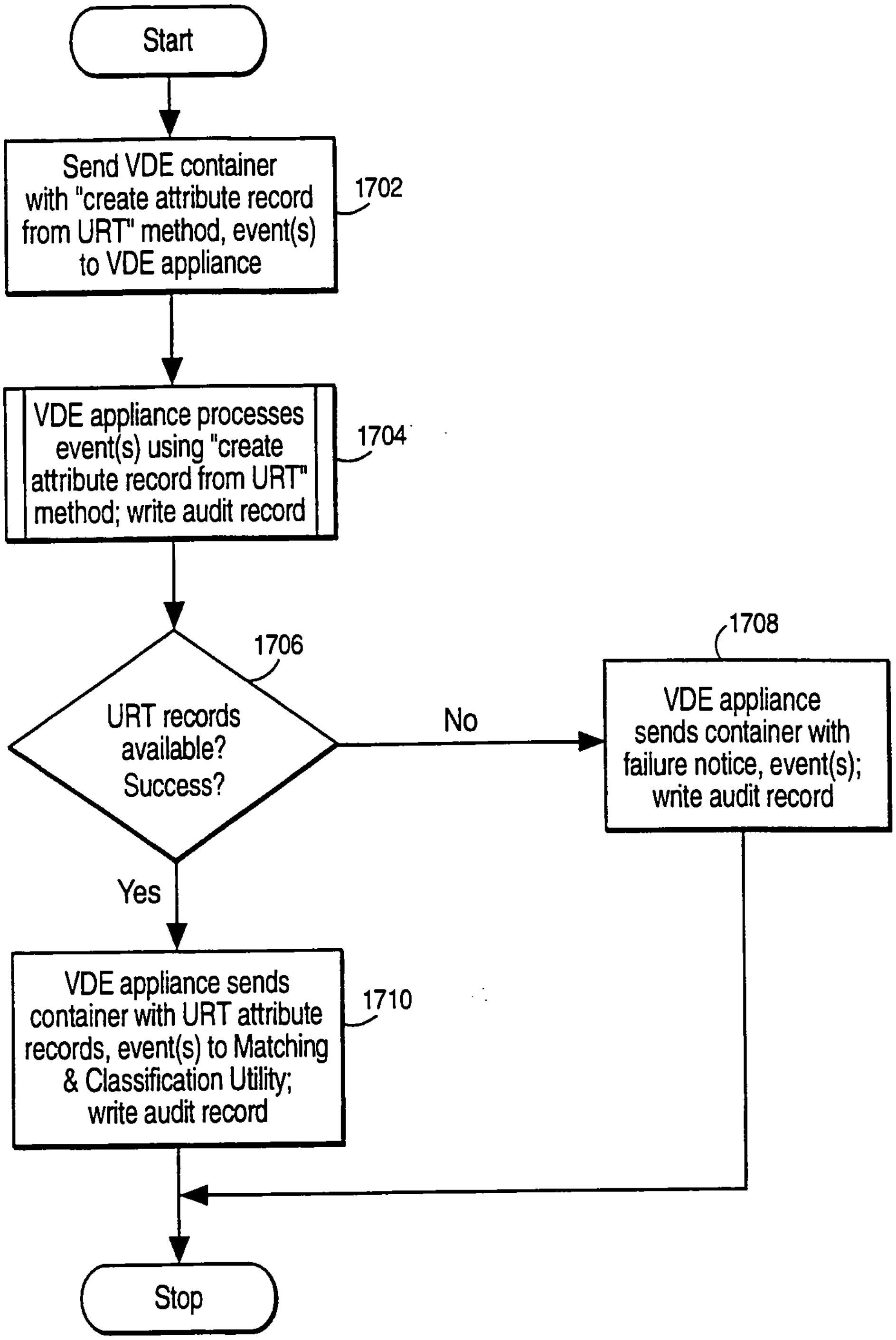

Start
Send VDE container with "create attribute record from URT" method, event(s) to VDE appliance
1702
VDE appliance processes event(s) using "create attribute record from URT" method; write audit record
1704
URT records available? Success?
1706
No
VDE appliance sends container with failure notice, event(s); write audit record
1708
Yes
VDE appliance sends container with URT attribute records, event(s) to Matching & Classification Utility; write audit record
1710
Stop

Fig. 39

Example Steps For Assembling Attribute Records Based On Rules and Consequences

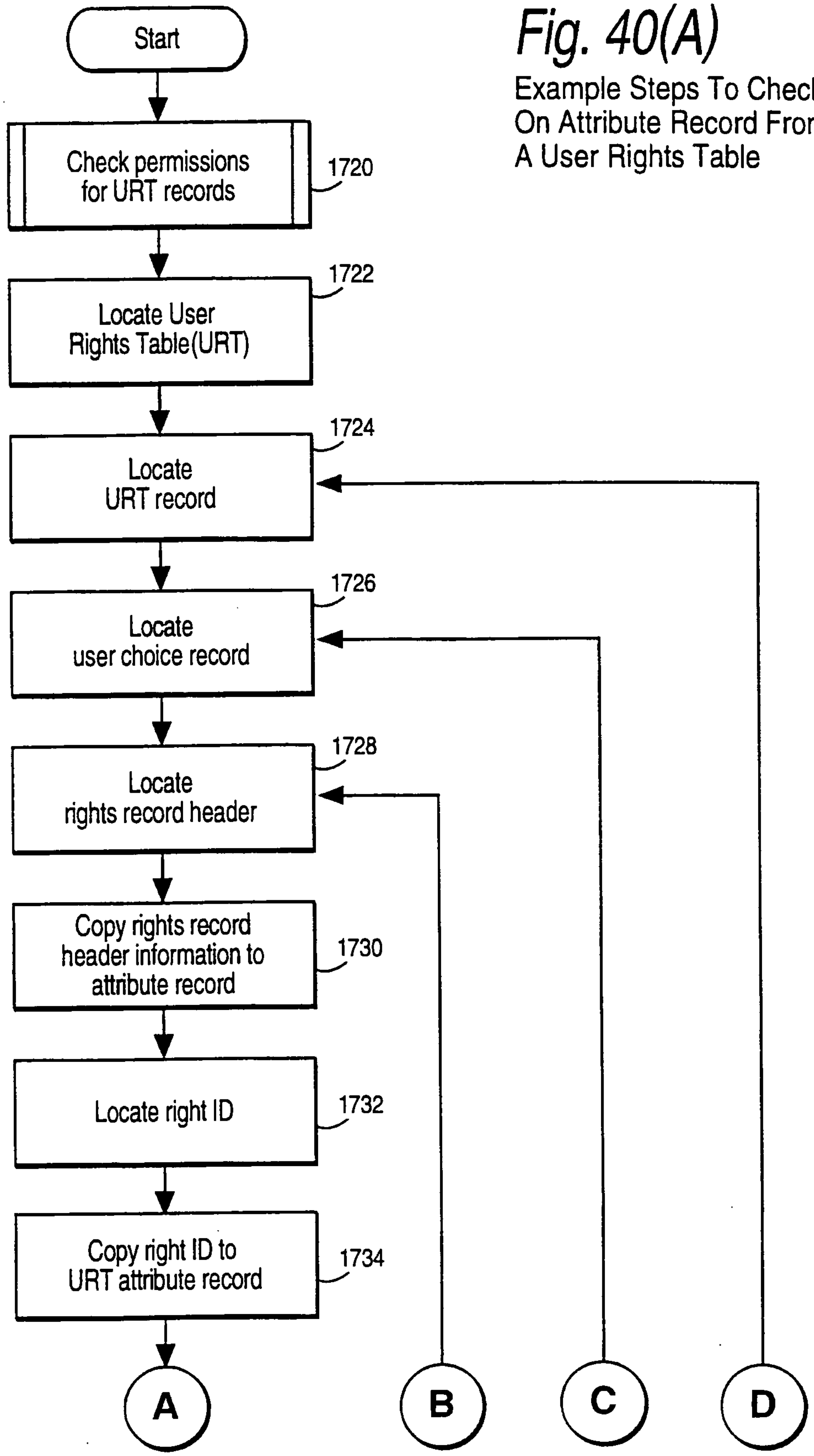

Start
Check permissions for URT records
1720
Locate User Rights Table(URT)
1722
Locate URT record
1724
Locate user choice record
1726
Locate rights record header
1728
Copy rights record header information to attribute record
1730
Locate right ID
1732
Copy right ID to URT attribute record
1734
A
B
C
D

Fig. 40(A)

Example Steps To Check On Attribute Record From A User Rights Table

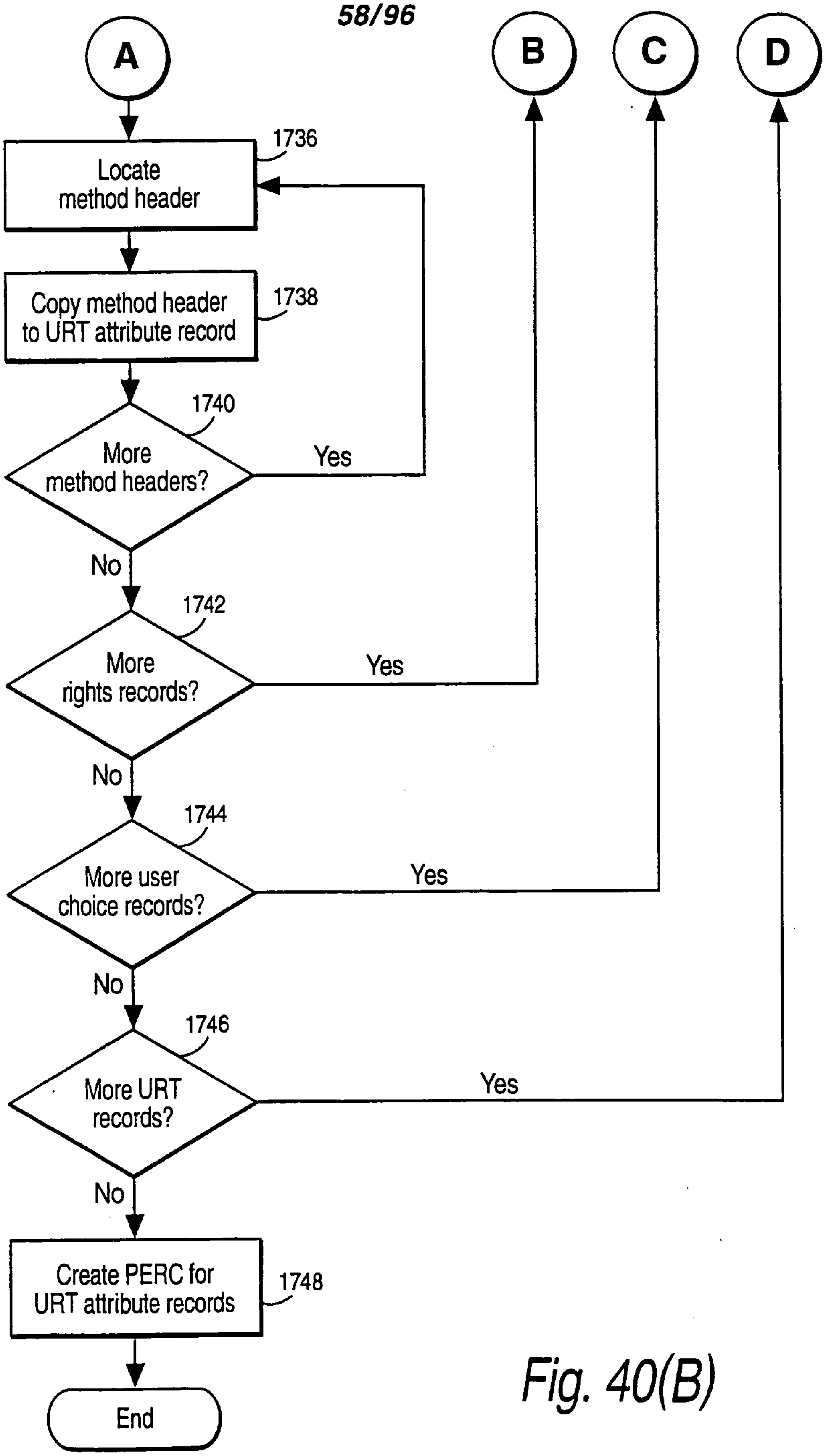

A
B
C
D
Locate method header
1736
Copy method header to URT attribute record
1738
More method headers?
1740
Yes
No
More rights records?
1742
Yes
No
More user choice records?
1744
Yes
No
More URT records?
1746
Yes
No
Create PERC for URT attribute records
1748
End

Fig. 40(B)

Fig. 41

Contruct attribute records from PERC records method example

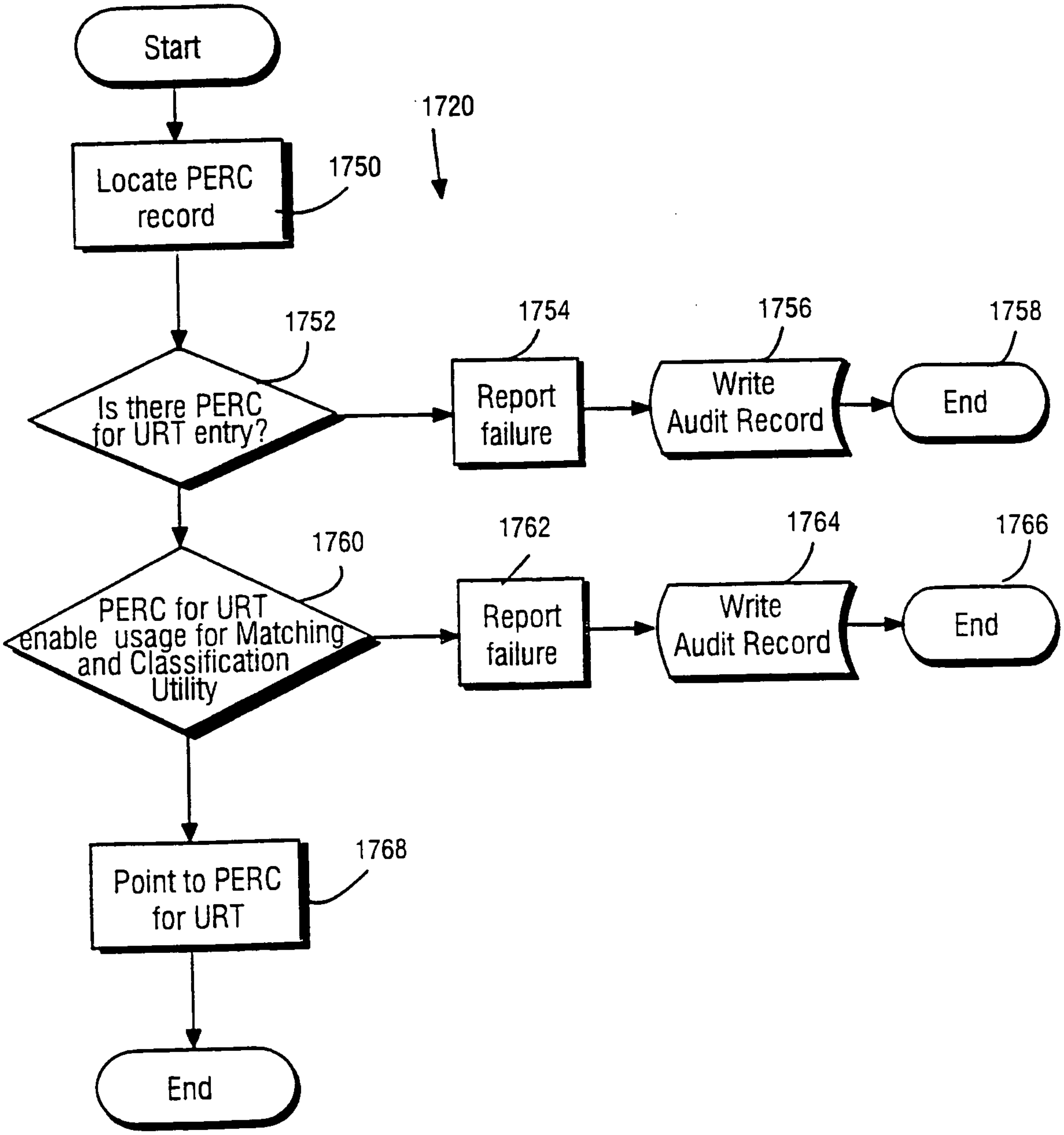

Start
1720
Locate PERC record
1750
1752
Is there PERC for URT entry?
1754
Report failure
1756
Write Audit Record
1758
End
1760
PERC for URT enable usage for Matching and Classification Utility
1762
Report failure
1764
Write Audit Record
1766
End
Point to PERC for URT
1768
End

Fig. 42(A) Example Rights Attribute Record From URT

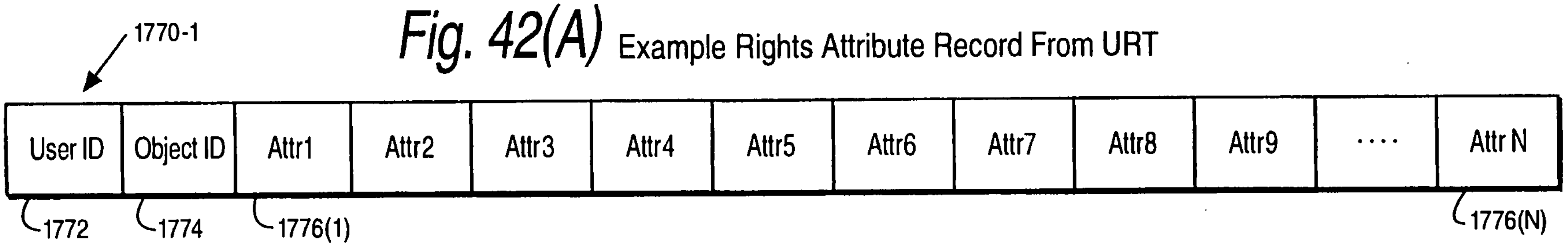

1770-1
User ID
Object ID
Attr1
Attr2
Attr3
Attr4
Attr5
Attr6
Attr7
Attr8
Attr9
....
Attr N
1772
1774
1776(1)
1776(N)

Fig. 42(B) Example Rights Attribute Record From URT

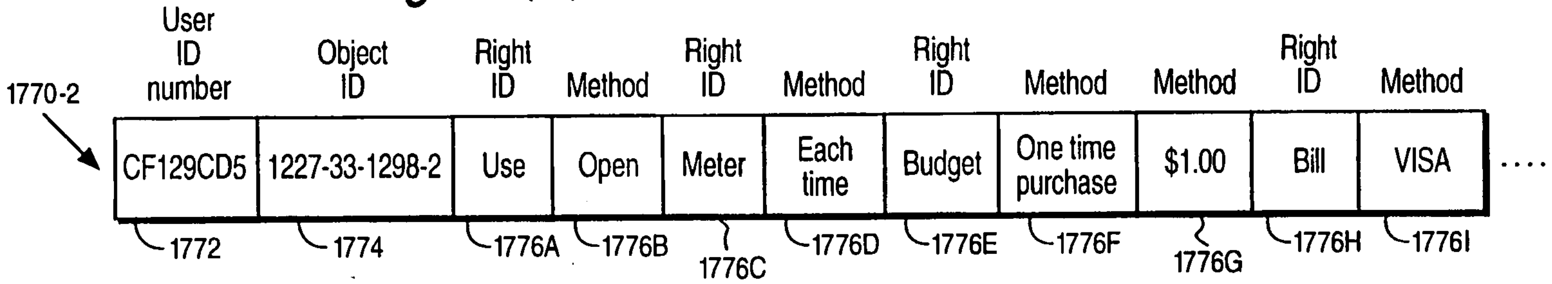

User ID number
Object ID
Right ID
Method
Right ID
Method
Right ID
Method
Method
Right ID
Method
1770-2
CF129CD5
1227-33-1298-2
Use
Open
Meter
Each time
Budget
One time purchase
$1.00
Bill
VISA
....
1772
1774
1776A
1776B
1776C
1776D
1776E
1776F
1776G
1776H
1776I

Fig. 42(C) Example User Choice Attribute Records (From URT)

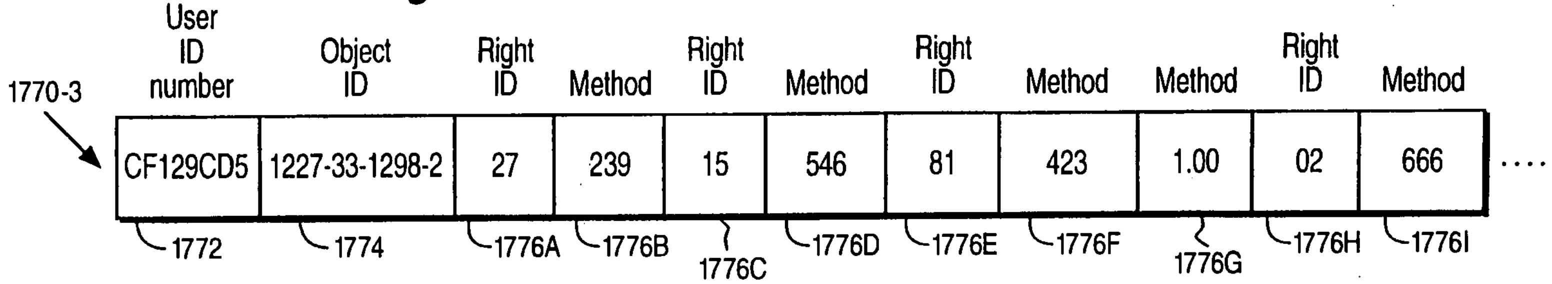

User ID number
Object ID
Right ID
Method
Right ID
Method
Right ID
Method
Method
Right ID
Method
1770-3
CF129CD5
1227-33-1298-2
27
239
15
546
81
423
1.00
02
666
....
1772
1774
1776A
1776B
1776C
1776D
1776E
1776F
1776G
1776H
1776I

Fig. 43

Example steps for assembling usage audit records

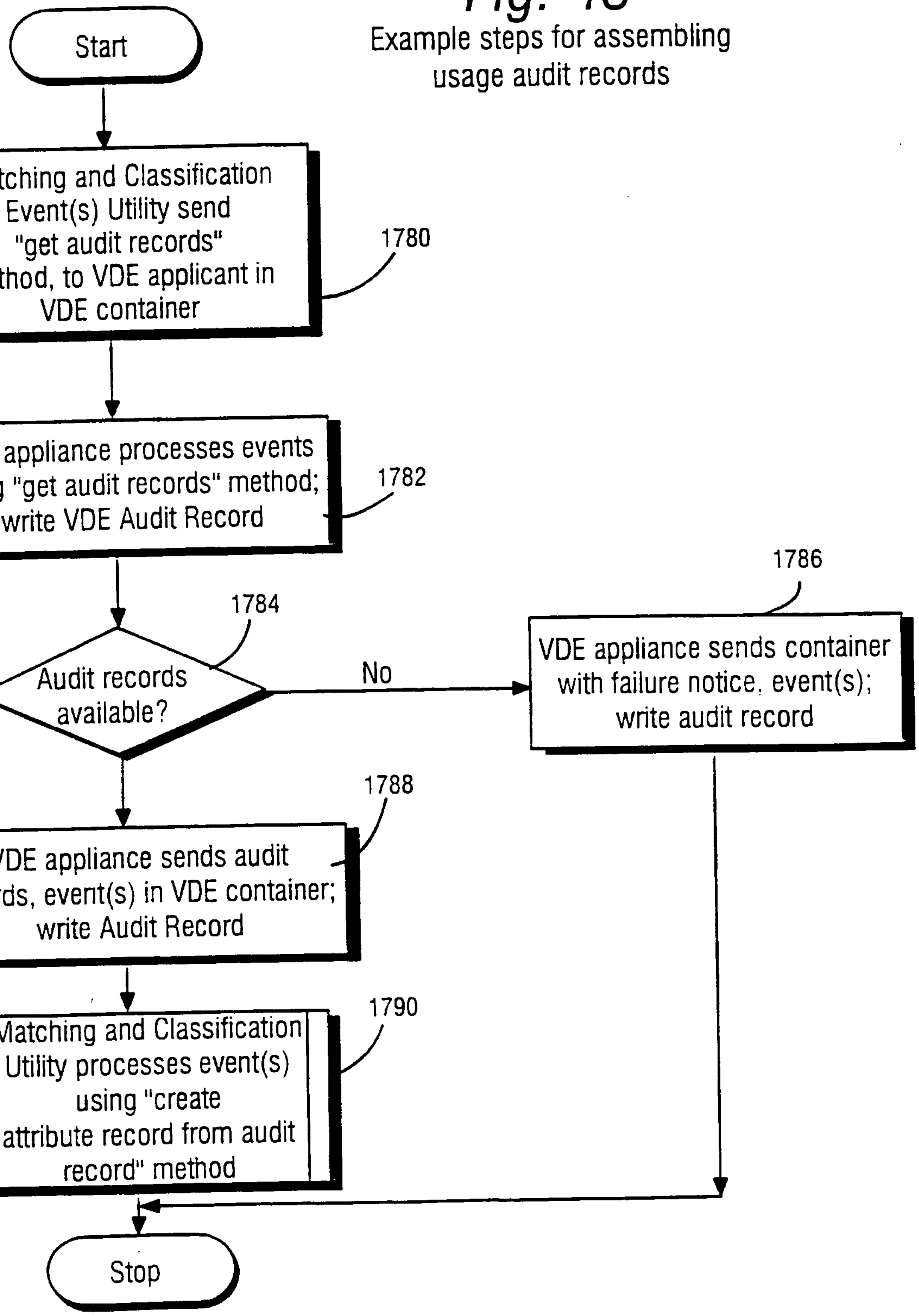

Start
Matching and Classification Event(s) Utility send "get audit records" method, to VDE applicant in VDE container
1780
VDE appliance processes events using "get audit records" method; write VDE Audit Record
1782
1784
Audit records available?
No
1786
VDE appliance sends container with failure notice, event(s); write audit record
1788
VDE appliance sends audit records, event(s) in VDE container; write Audit Record
1790
Matching and Classification Utility processes event(s) using "create attribute record from audit record" method
Stop

Fig. 44
Example steps for assembling usage audit records

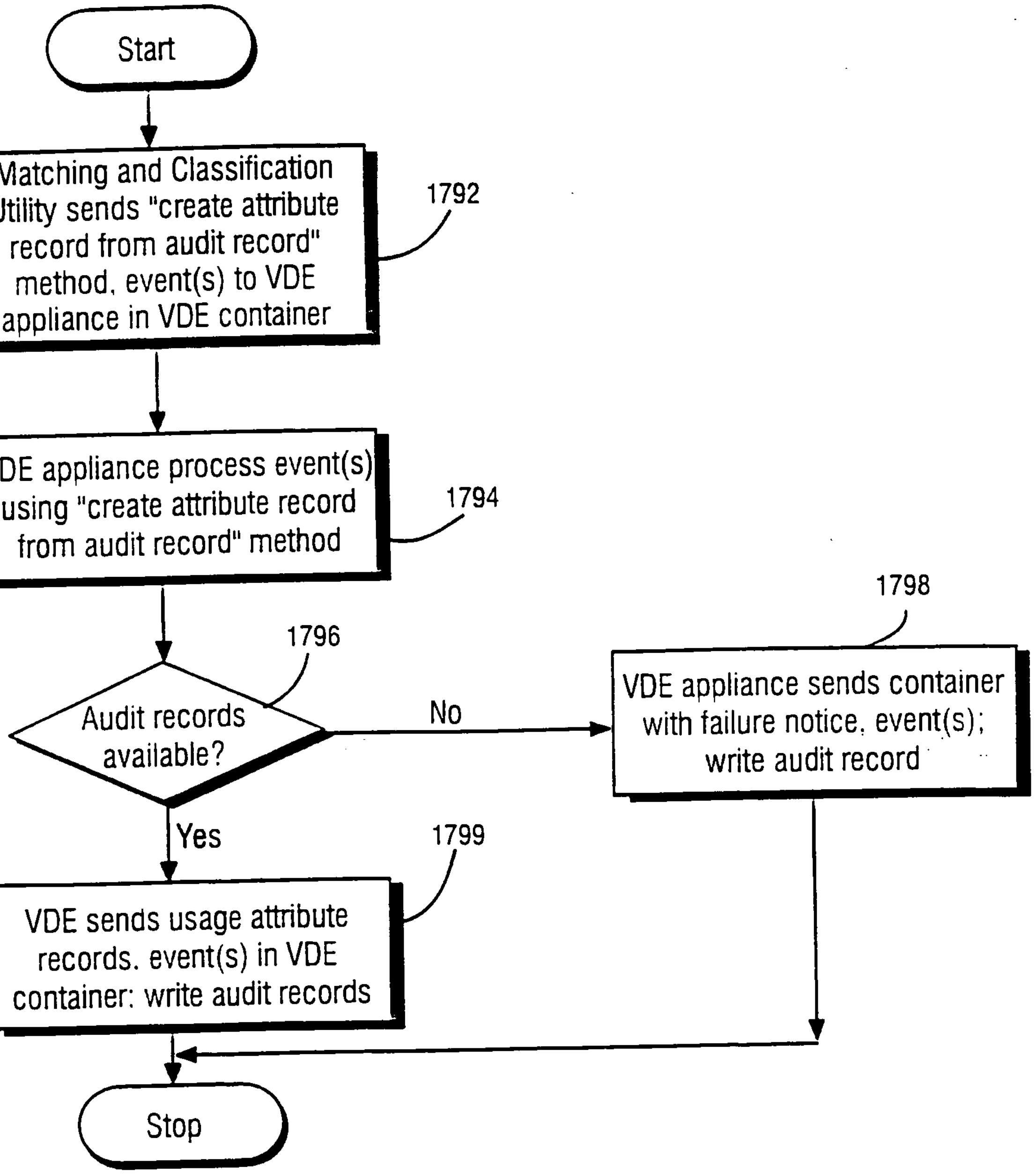

Start
Matching and Classification Utility sends "create attribute record from audit record" method, event(s) to VDE appliance in VDE container
1792
VDE appliance process event(s) using "create attribute record from audit record" method
1794
1796
Audit records available?
No
1798
VDE appliance sends container with failure notice, event(s); write audit record
Yes
1799
VDE sends usage attribute records, event(s) in VDE container; write audit records
Stop

Fig. 45(A)

Example steps to create audit attribute records

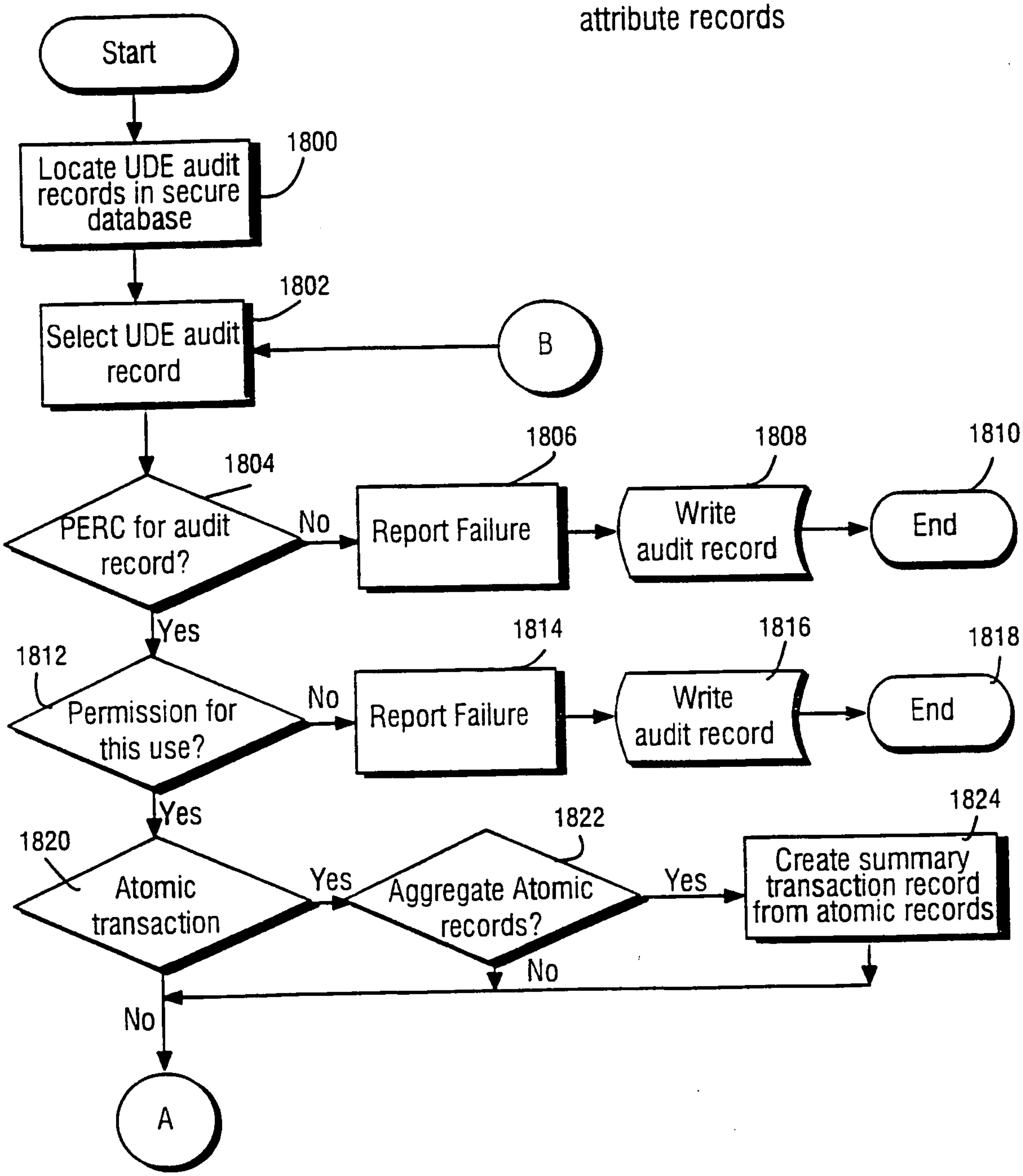

Start
Locate UDE audit records in secure database
1800
1802
Select UDE audit record
B
1804
PERC for audit record?
No
1806
Report Failure
1808
Write audit record
1810
End
Yes
1812
Permission for this use?
No
1814
Report Failure
1816
Write audit record
1818
End
Yes
1820
Atomic transaction
Yes
1822
Aggregate Atomic records?
Yes
1824
Create summary transaction record from atomic records
No
No
A

Fig. 45(B)

Example steps to create audit attribute records

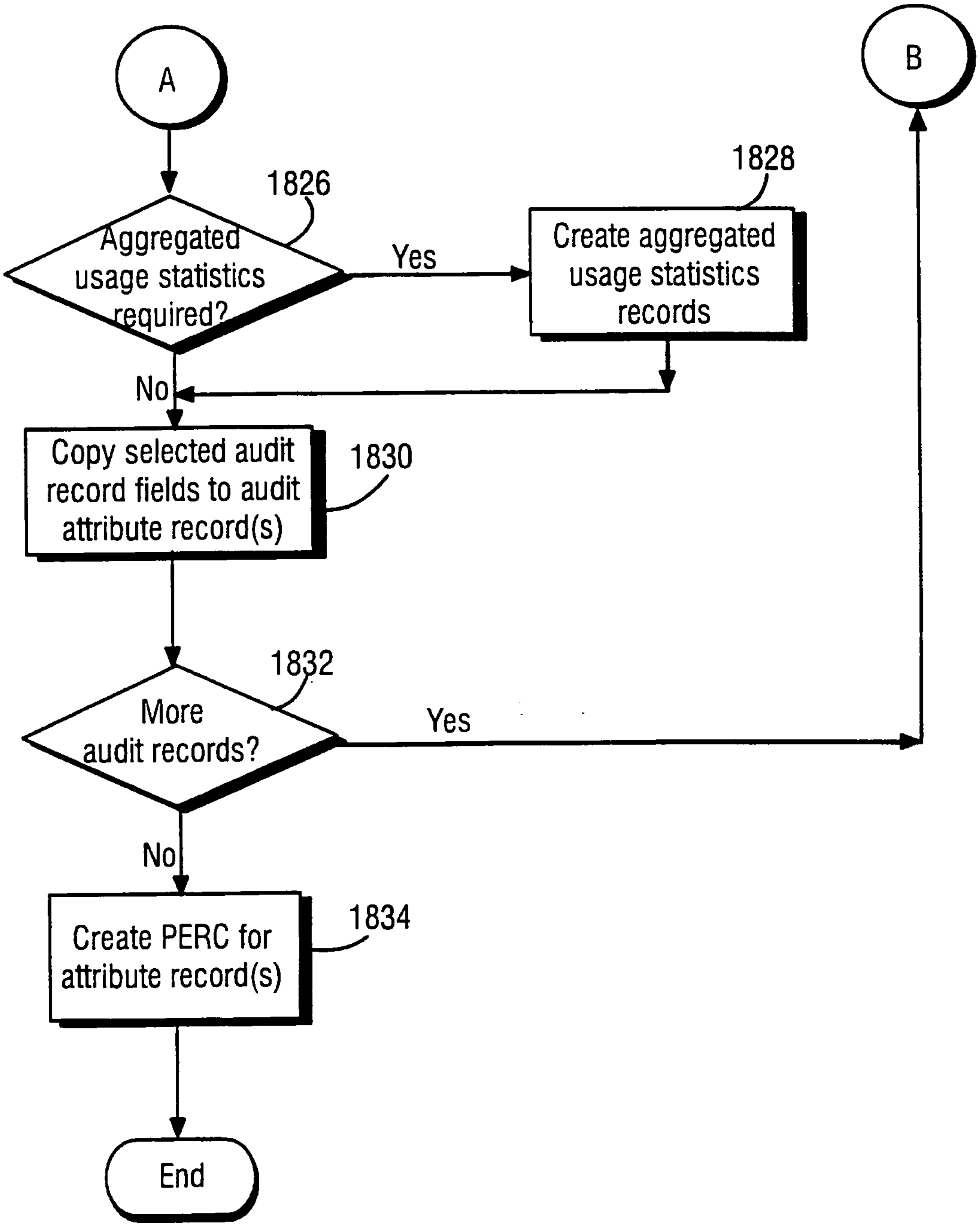

A
1826
Aggregated usage statistics required?
Yes
1828
Create aggregated usage statistics records
B
No
Copy selected audit record fields to audit attribute record(s)
1830
1832
More audit records?
Yes
No
Create PERC for attribute record(s)
1834
End

Fig. 46(A) Example Usage Attribute Record From UDE Audit Record

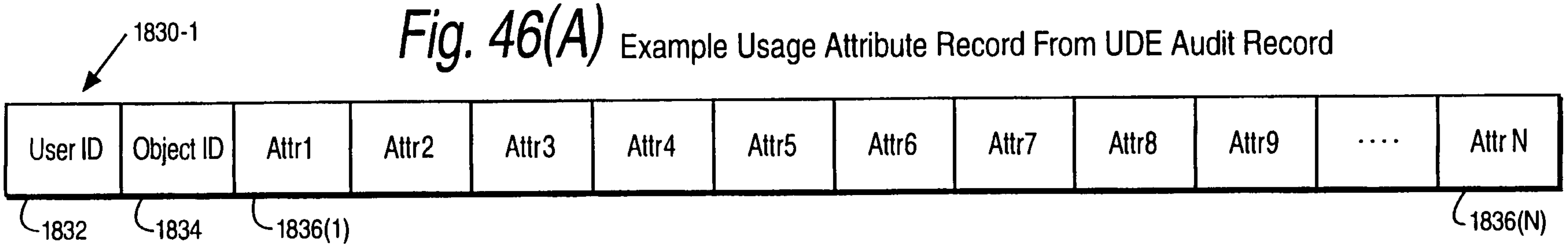
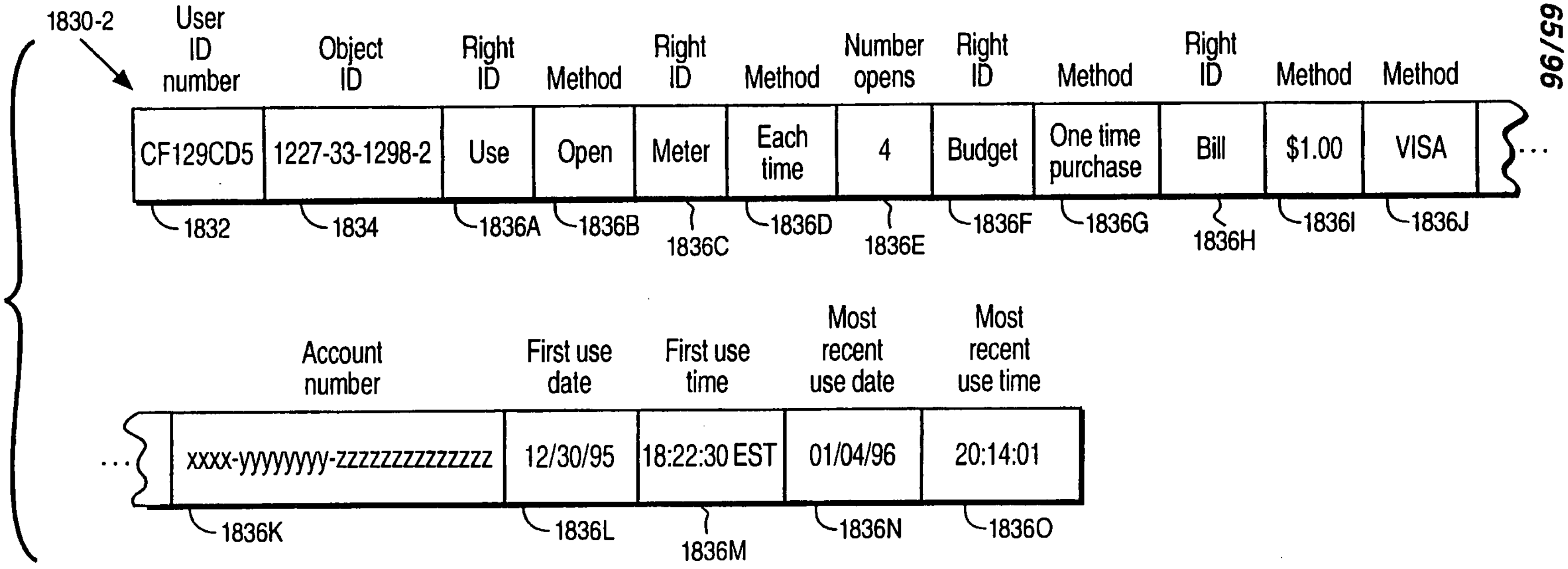
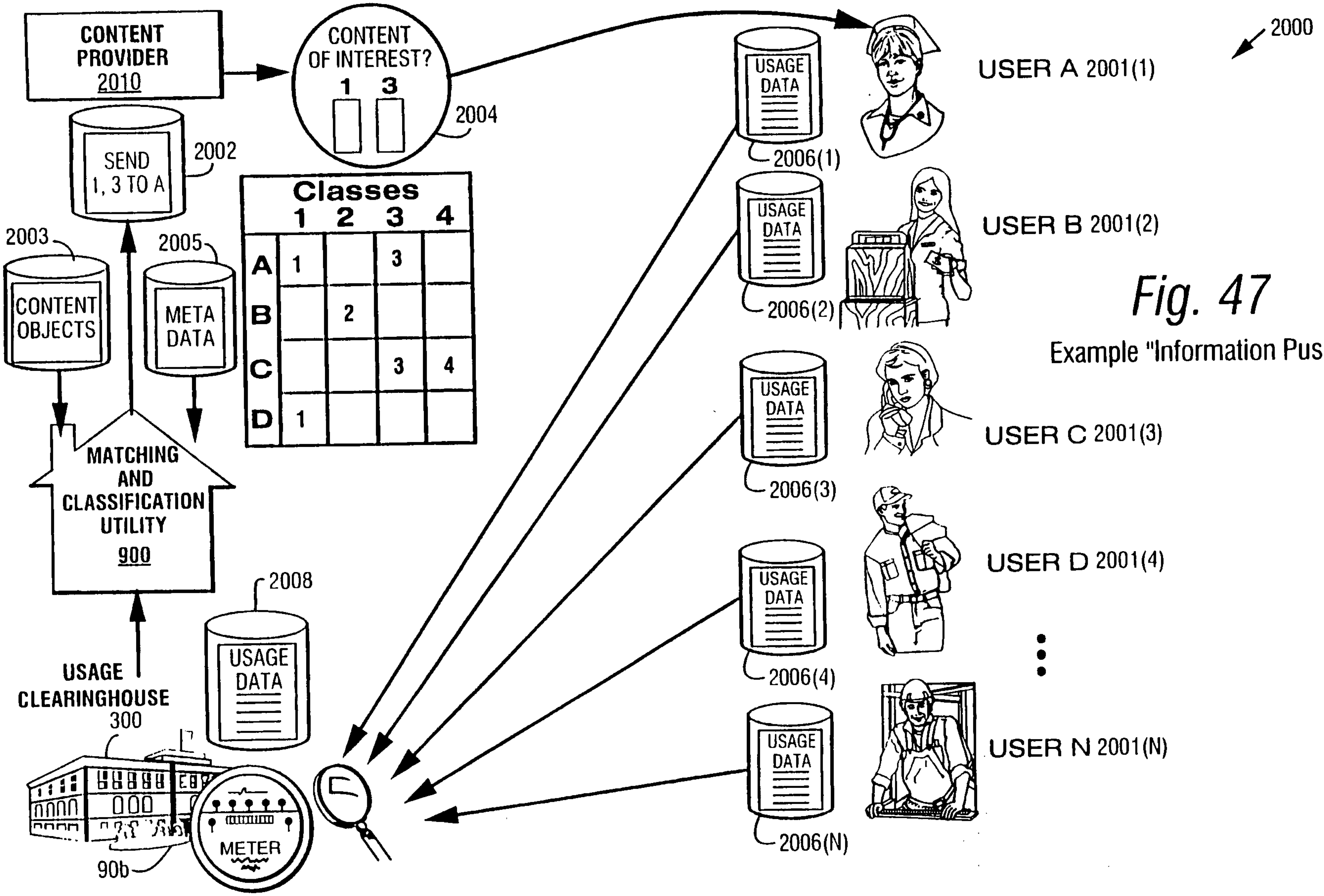

1830-1
User ID
Object ID
Attr1
Attr2
Attr3
Attr4
Attr5
Attr6
Attr7
Attr8
Attr9
....
Attr N
1832
1834
1836(1)
1836(N)

Fig. 46(B) Example Usage Attribute/Statistics Records From UDE Audit Records


1830-2
User ID number
Object ID
Right ID
Method
Right ID
Method
Number opens
Right ID
Method
Right ID
Method
Method
CF129CD5
1227-33-1298-2
Use
Open
Meter
Each time
4
Budget
One time purchase
Bill
$1.00
VISA
...
1832
1834
1836A
1836B
1836C
1836D
1836E
1836F
1836G
1836H
1836I
1836J
Account number
First use date
First use time
Most recent use date
Most recent use time
...
xxxx-yyyyyyy-zzzzzzzzzzzzzz
12/30/95
18:22:30 EST
01/04/96
20:14:01
1836K
1836L
1836M
1836N
1836O


2000
CONTENT PROVIDER 2010
CONTENT OF INTEREST? 1 3
2004
SEND 1, 3 TO A
2002
2003
CONTENT OBJECTS
2005
META DATA
MATCHING AND CLASSIFICATION UTILITY 900
USAGE CLEARINGHOUSE 300
90b
METER
USAGE DATA
2008
USAGE DATA
2006(1)
USER A 2001(1)
USAGE DATA
2006(2)
USER B 2001(2)
USAGE DATA
2006(3)
USER C 2001(3)
USAGE DATA
2006(4)
USER D 2001(4)
USAGE DATA
2006(N)
USER N 2001(N)

Classes	1	2	3	4
A	1		3	
B		2		
C			3	4
D	1			

Fig. 47

Example "Information Push"

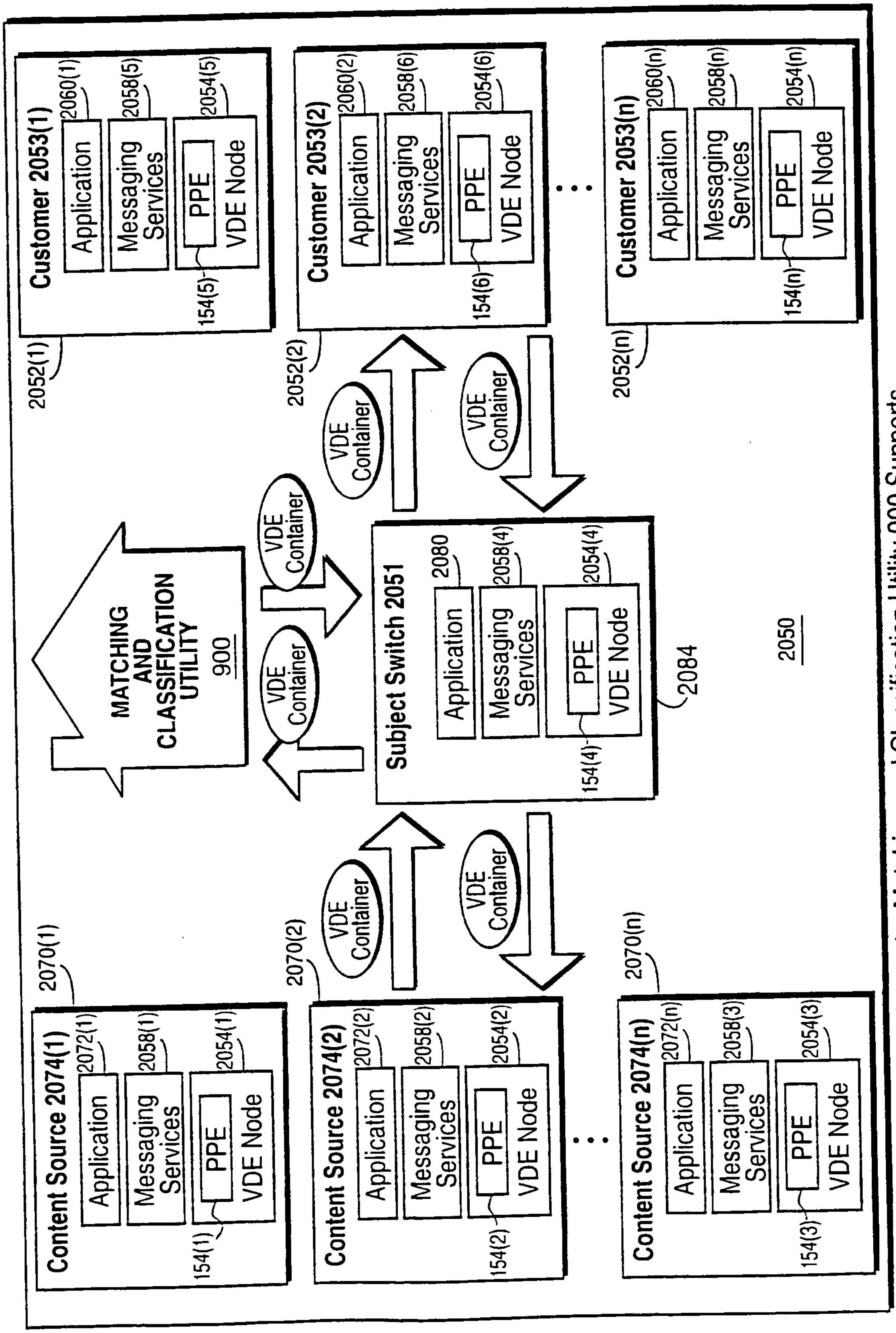

Content Source 2074(1)
Application
2072(1)
Messaging Services
2058(1)
PPE
VDE Node
2054(1)
154(1)
2070(1)
Content Source 2074(2)
Application
2072(2)
Messaging Services
2058(2)
PPE
VDE Node
2054(2)
154(2)
2070(2)
Content Source 2074(n)
Application
2072(n)
Messaging Services
2058(3)
PPE
VDE Node
2054(3)
154(3)
2070(n)
MATCHING AND CLASSIFICATION UTILITY
900
VDE Container
VDE Container
VDE Container
VDE Container
VDE Container
VDE Container
Subject Switch 2051
Application
2080
Messaging Services
2058(4)
PPE
VDE Node
2054(4)
154(4)
2084
2050
Customer 2053(1)
Application
2060(1)
Messaging Services
2058(5)
PPE
VDE Node
2054(5)
154(5)
2052(1)
Customer 2053(2)
Application
2060(2)
Messaging Services
2058(6)
PPE
VDE Node
2054(6)
154(6)
2052(2)
Customer 2053(n)
Application
2060(n)
Messaging Services
2058(n)
PPE
VDE Node
2054(n)
154(n)
2052(n)

Fig. 47(A) Matching and Classification Utility 900 Supports "Push" models using Subject Switching and Messaging Services

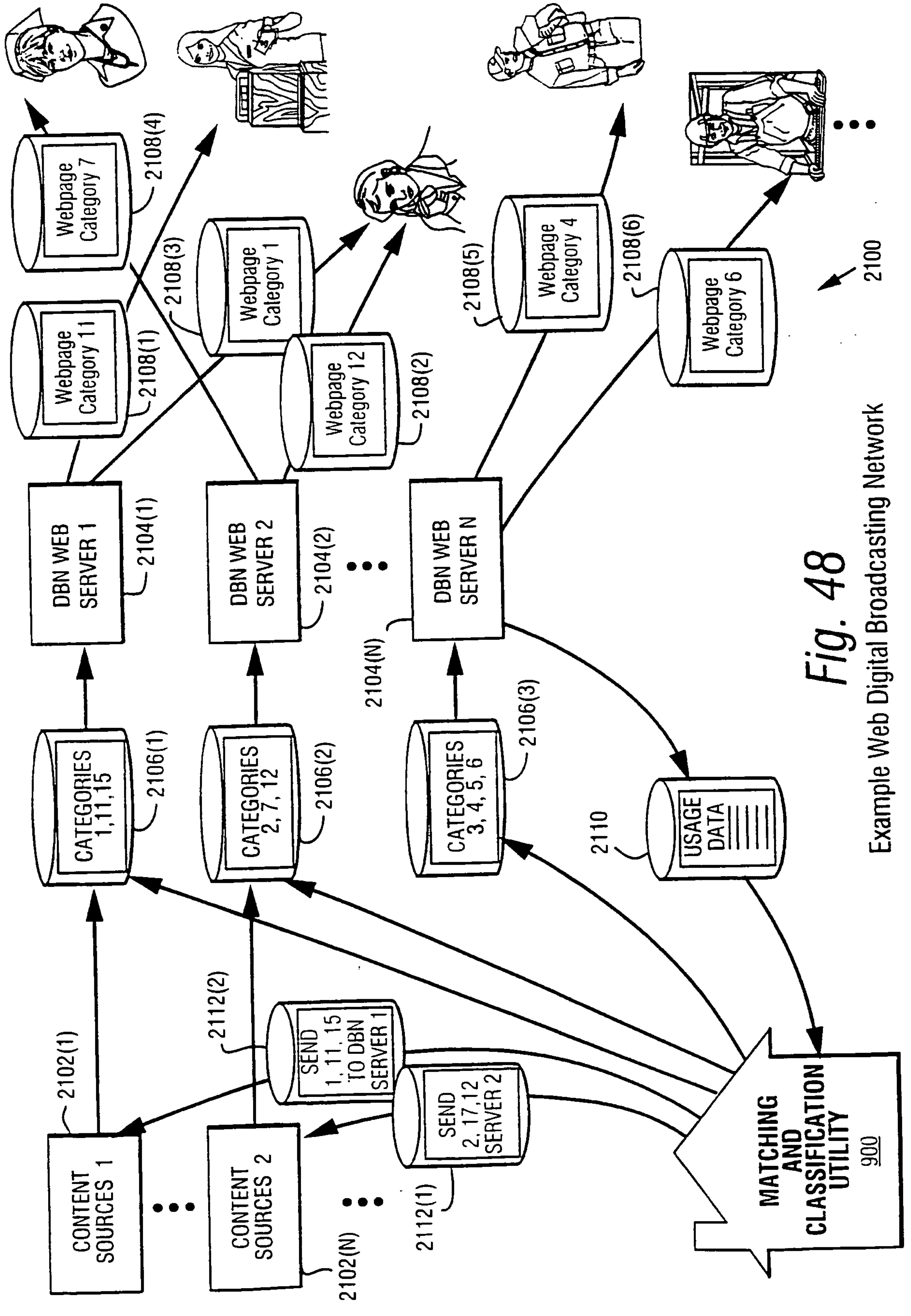

CONTENT SOURCES 1
2102(1)
CONTENT SOURCES 2
2102(N)
SEND 1, 11, 15 TO DBN SERVER 1
2112(2)
SEND 2, 17, 12 SERVER 2
2112(1)
CATEGORIES 1, 11, 15
2106(1)
CATEGORIES 2, 7, 12
2106(2)
CATEGORIES 3, 4, 5, 6
2106(3)
DBN WEB SERVER 1
2104(1)
DBN WEB SERVER 2
2104(2)
DBN WEB SERVER N
2104(N)
Webpage Category 11
2108(1)
Webpage Category 7
2108(4)
Webpage Category 1
2108(3)
Webpage Category 12
2108(2)
Webpage Category 4
2108(5)
Webpage Category 6
2108(6)
USAGE DATA
2110
MATCHING AND CLASSIFICATION UTILITY
900
2100

Fig. 48

Example Web Digital Broadcasting Network

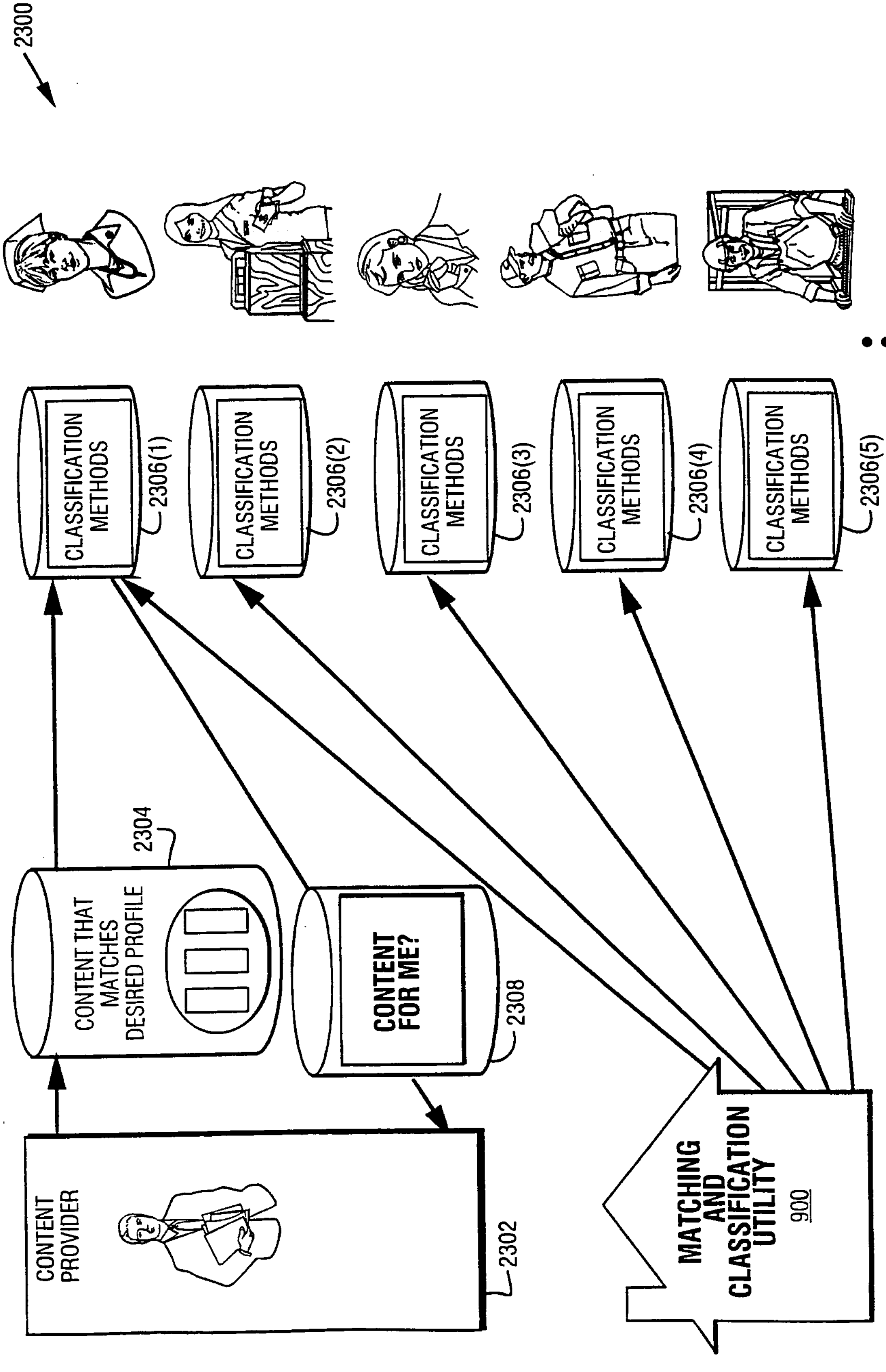

2300
CLASSIFICATION METHODS
2306(1)
CLASSIFICATION METHODS
2306(2)
CLASSIFICATION METHODS
2306(3)
CLASSIFICATION METHODS
2306(4)
CLASSIFICATION METHODS
2306(5)
CONTENT THAT MATCHES DESIRED PROFILE
2304
CONTENT FOR ME?
2308
CONTENT PROVIDER
2302
MATCHING AND CLASSIFICATION UTILITY
900

Fig. 49 Example "Consumer Pull"

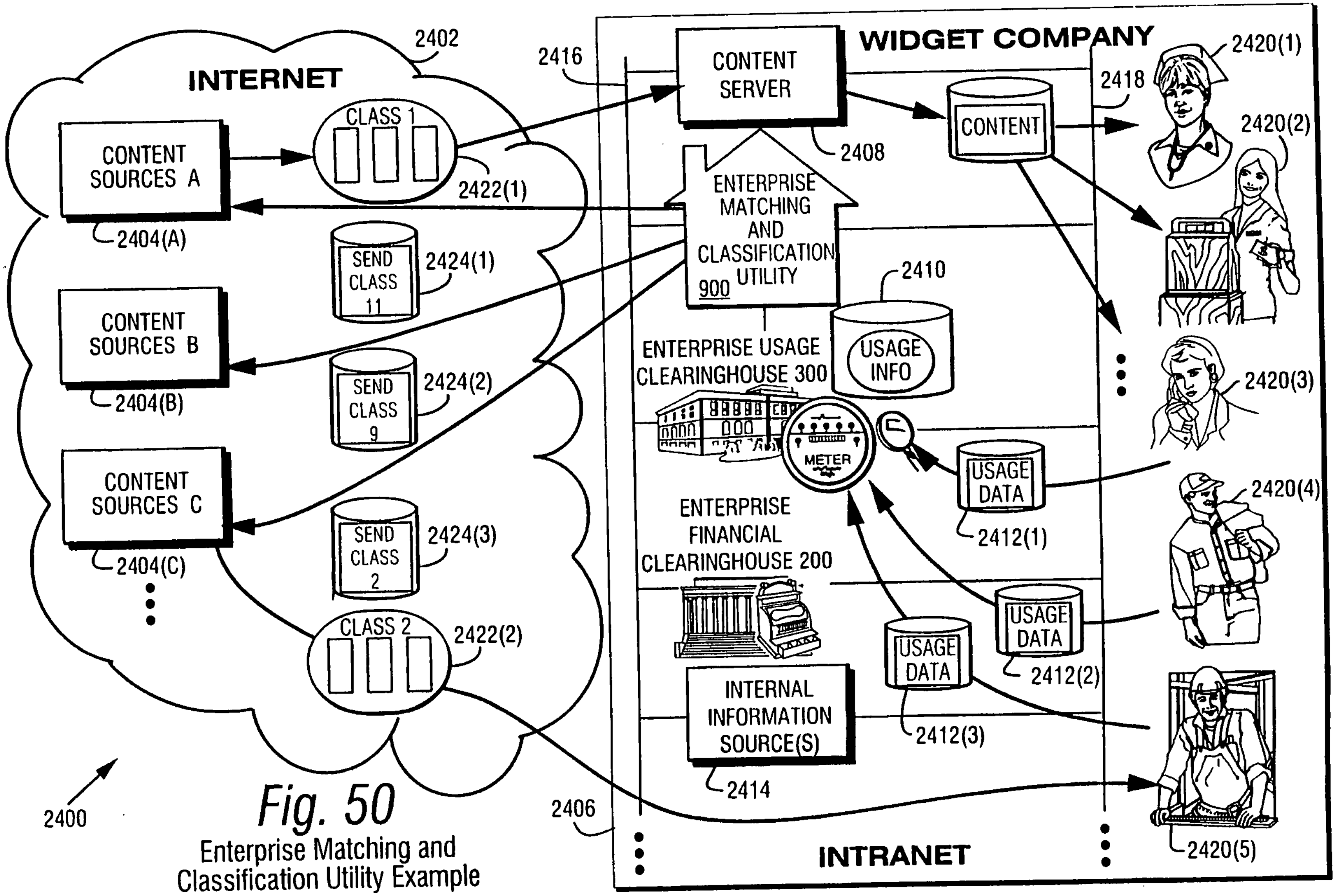

INTERNET
2402
CONTENT SOURCES A
2404(A)
CLASS 1
2422(1)
SEND CLASS 11
2424(1)
CONTENT SOURCES B
2404(B)
SEND CLASS 9
2424(2)
CONTENT SOURCES C
2404(C)
SEND CLASS 2
2424(3)
CLASS 2
2422(2)
2400
2416
2406
WIDGET COMPANY
CONTENT SERVER
2408
CONTENT
2418
ENTERPRISE MATCHING AND CLASSIFICATION UTILITY 900
2410
USAGE INFO
ENTERPRISE USAGE CLEARINGHOUSE 300
METER
USAGE DATA
2412(1)
ENTERPRISE FINANCIAL CLEARINGHOUSE 200
USAGE DATA
2412(2)
USAGE DATA
2412(3)
INTERNAL INFORMATION SOURCE(S)
2414
INTRANET
2420(1)
2420(2)
2420(3)
2420(4)
2420(5)

Fig. 50

Enterprise Matching and Classification Utility Example

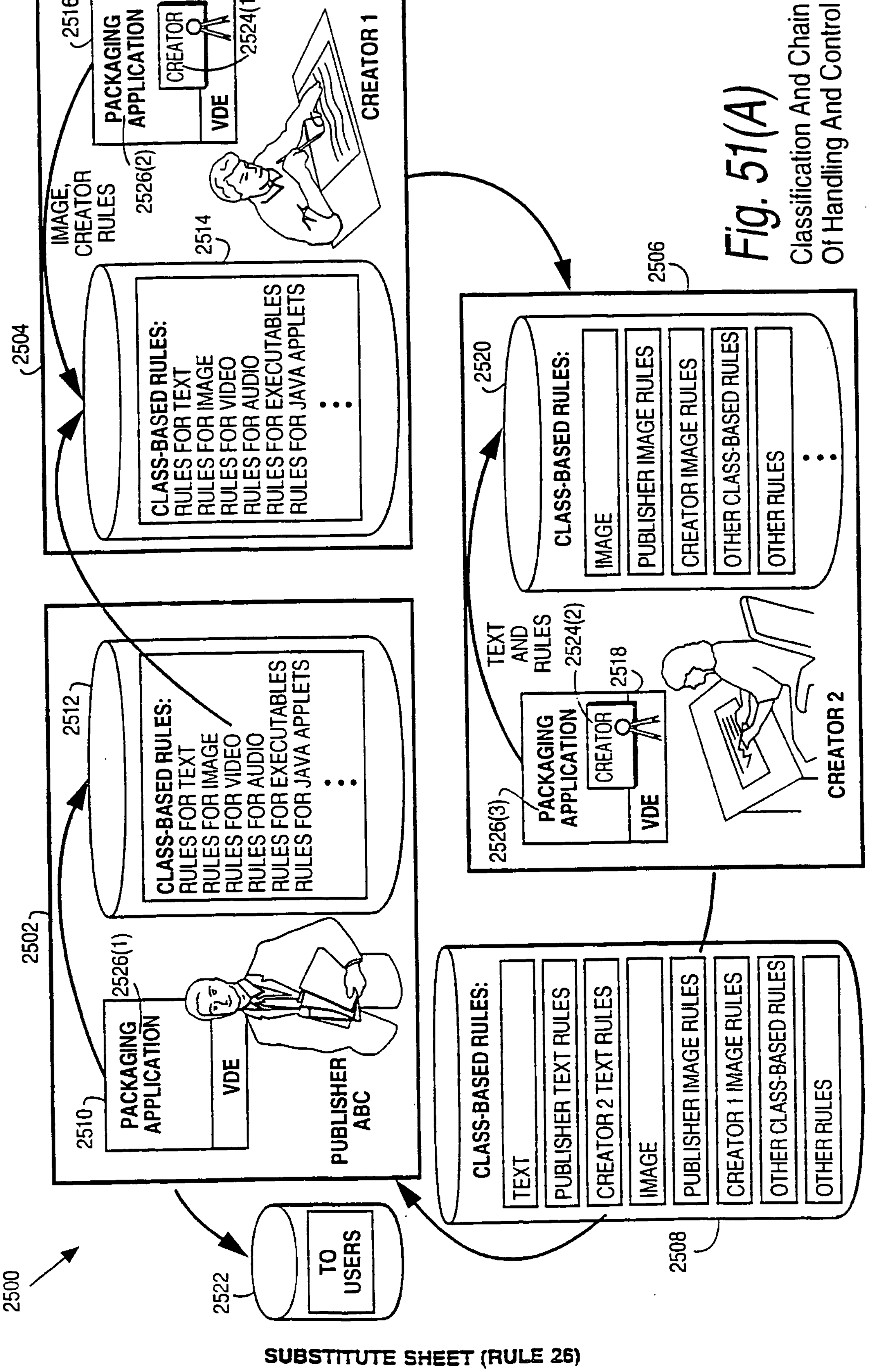

2500
2502
2504
2506
2508
2510
PACKAGING APPLICATION
VDE
2526(1)
PUBLISHER ABC
2512
CLASS-BASED RULES:
RULES FOR TEXT
RULES FOR IMAGE
RULES FOR VIDEO
RULES FOR AUDIO
RULES FOR EXECUTABLES
RULES FOR JAVA APPLETS
2514
IMAGE, CREATOR RULES
2526(2)
2516
PACKAGING APPLICATION
CREATOR
VDE
2524(1)
CREATOR 1
2520
CLASS-BASED RULES:
IMAGE
PUBLISHER IMAGE RULES
CREATOR IMAGE RULES
OTHER CLASS-BASED RULES
OTHER RULES
TEXT AND RULES
2524(2)
2518
2526(3)
PACKAGING APPLICATION
CREATOR
VDE
CREATOR 2
CLASS-BASED RULES:
TEXT
PUBLISHER TEXT RULES
CREATOR 2 TEXT RULES
IMAGE
PUBLISHER IMAGE RULES
CREATOR 1 IMAGE RULES
OTHER CLASS-BASED RULES
OTHER RULES
2522
TO USERS

Fig. 51(A)
Classification And Chain Of Handling And Control

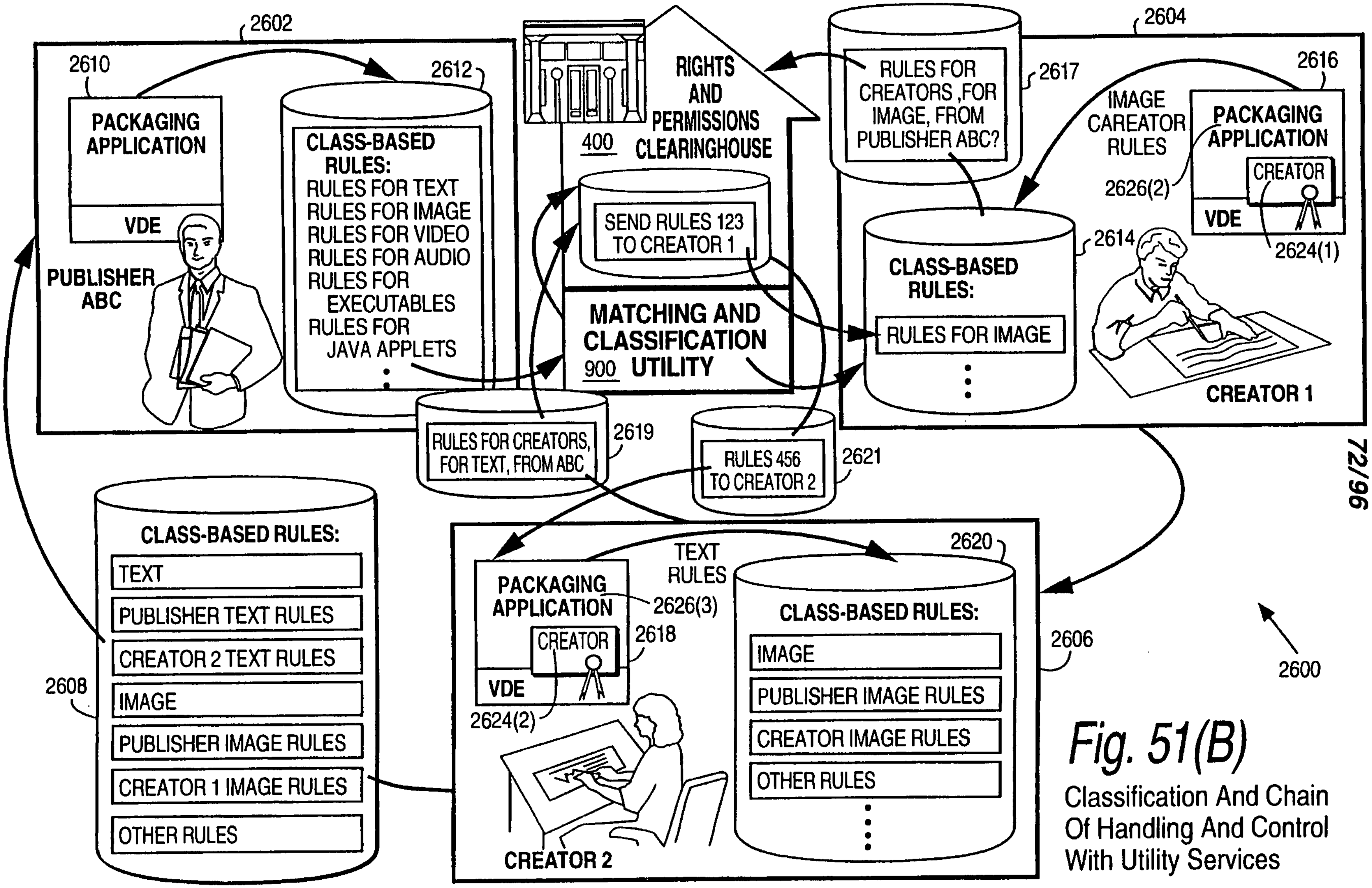

2602
2610
PACKAGING APPLICATION
VDE
PUBLISHER ABC
2612
CLASS-BASED RULES:
RULES FOR TEXT
RULES FOR IMAGE
RULES FOR VIDEO
RULES FOR AUDIO
RULES FOR EXECUTABLES
RULES FOR JAVA APPLETS
400
RIGHTS AND PERMISSIONS CLEARINGHOUSE
SEND RULES 123 TO CREATOR 1
MATCHING AND CLASSIFICATION
900 UTILITY
RULES FOR CREATORS, FOR TEXT, FROM ABC
2619
RULES 456 TO CREATOR 2
2621
2604
RULES FOR CREATORS ,FOR IMAGE, FROM PUBLISHER ABC?
2617
IMAGE CAREATOR RULES
2626(2)
2616
PACKAGING APPLICATION
CREATOR
VDE
2624(1)
2614
CLASS-BASED RULES:
RULES FOR IMAGE
CREATOR 1
2608
CLASS-BASED RULES:
TEXT
PUBLISHER TEXT RULES
CREATOR 2 TEXT RULES
IMAGE
PUBLISHER IMAGE RULES
CREATOR 1 IMAGE RULES
OTHER RULES
PACKAGING APPLICATION
TEXT RULES
2626(3)
CREATOR
2618
VDE
2624(2)
CREATOR 2
2620
CLASS-BASED RULES:
IMAGE
PUBLISHER IMAGE RULES
CREATOR IMAGE RULES
OTHER RULES
2606
2600

Fig. 51(B)

Classification And Chain Of Handling And Control With Utility Services

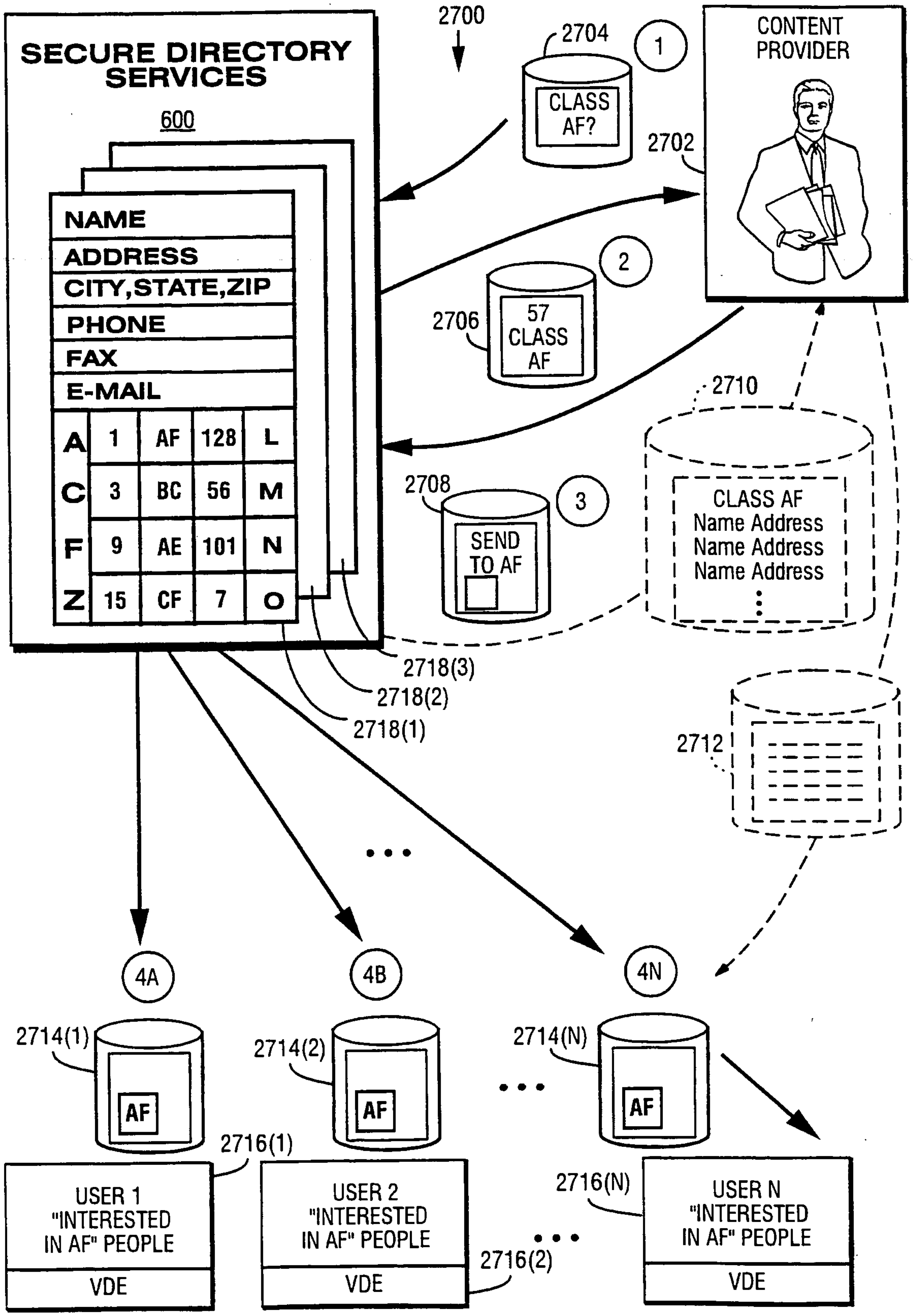
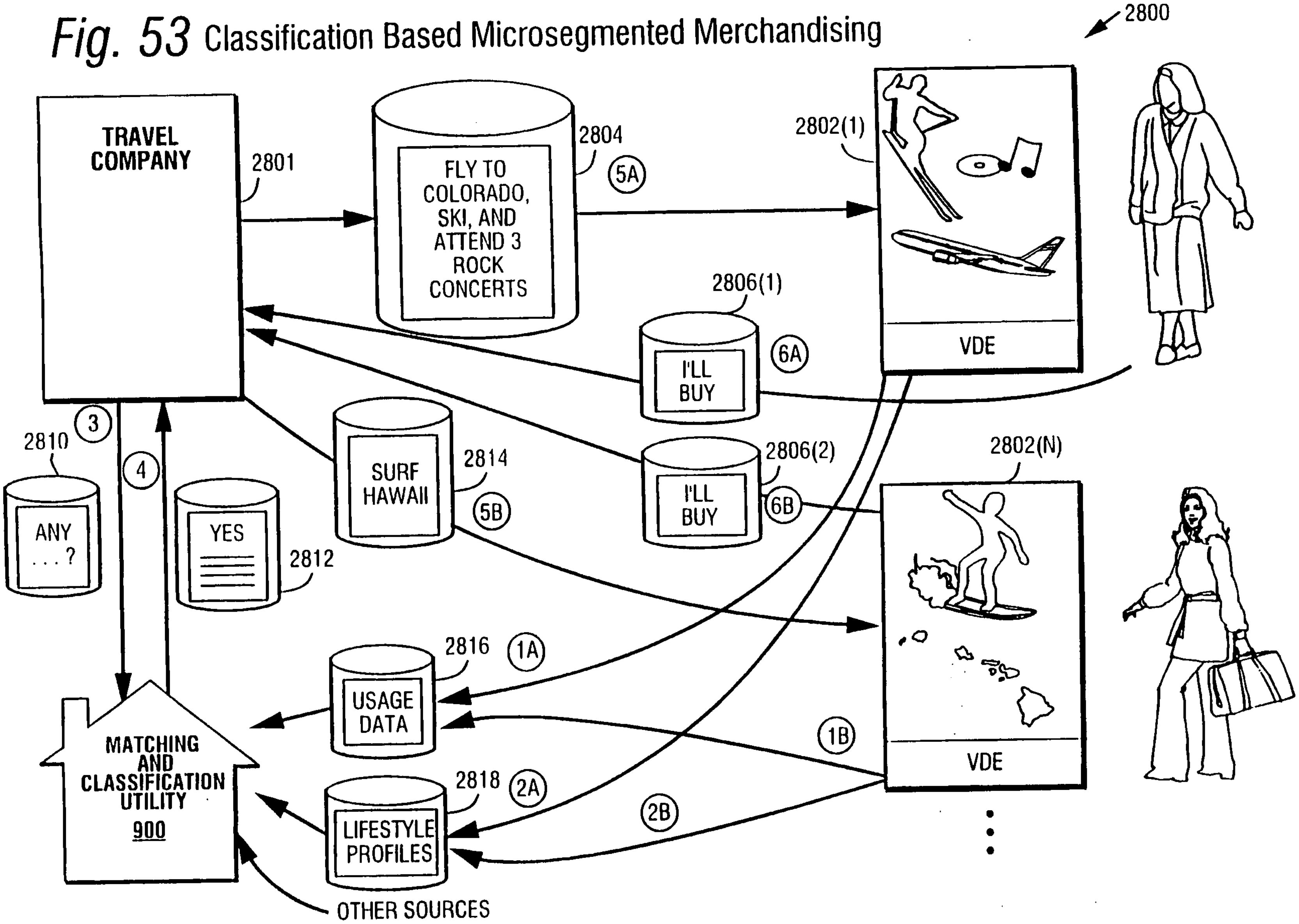

2700
SECURE DIRECTORY SERVICES
600
NAME
ADDRESS
CITY,STATE,ZIP
PHONE
FAX
E-MAIL
A 1 AF 128 L
C 3 BC 56 M
F 9 AE 101 N
Z 15 CF 7 O
2704
1
CLASS AF?
CONTENT PROVIDER
2702
2
2706
57 CLASS AF
2710
2708
3
SEND TO AF
CLASS AF
Name Address
Name Address
Name Address
2718(3)
2718(2)
2718(1)
2712
4A
4B
4N
2714(1)
2714(2)
2714(N)
AF
AF
AF
2716(1)
2716(2)
2716(N)
USER 1 "INTERESTED IN AF" PEOPLE
VDE
USER 2 "INTERESTED IN AF" PEOPLE
VDE
USER N "INTERESTED IN AF" PEOPLE
VDE

Fig. 52 Secure Directory Services

Fig. 53 Classification Based Microsegmented Merchandising


2800
TRAVEL COMPANY
2801
FLY TO COLORADO, SKI, AND ATTEND 3 ROCK CONCERTS
2804
5A
2802(1)
VDE
2806(1)
I'LL BUY
6A
2806(2)
I'LL BUY
6B
2802(N)
VDE
SURF HAWAII
2814
5B
3
2810
ANY ...?
4
YES
2812
USAGE DATA
2816
1A
1B
LIFESTYLE PROFILES
2818
2A
2B
MATCHING AND CLASSIFICATION UTILITY
900
OTHER SOURCES

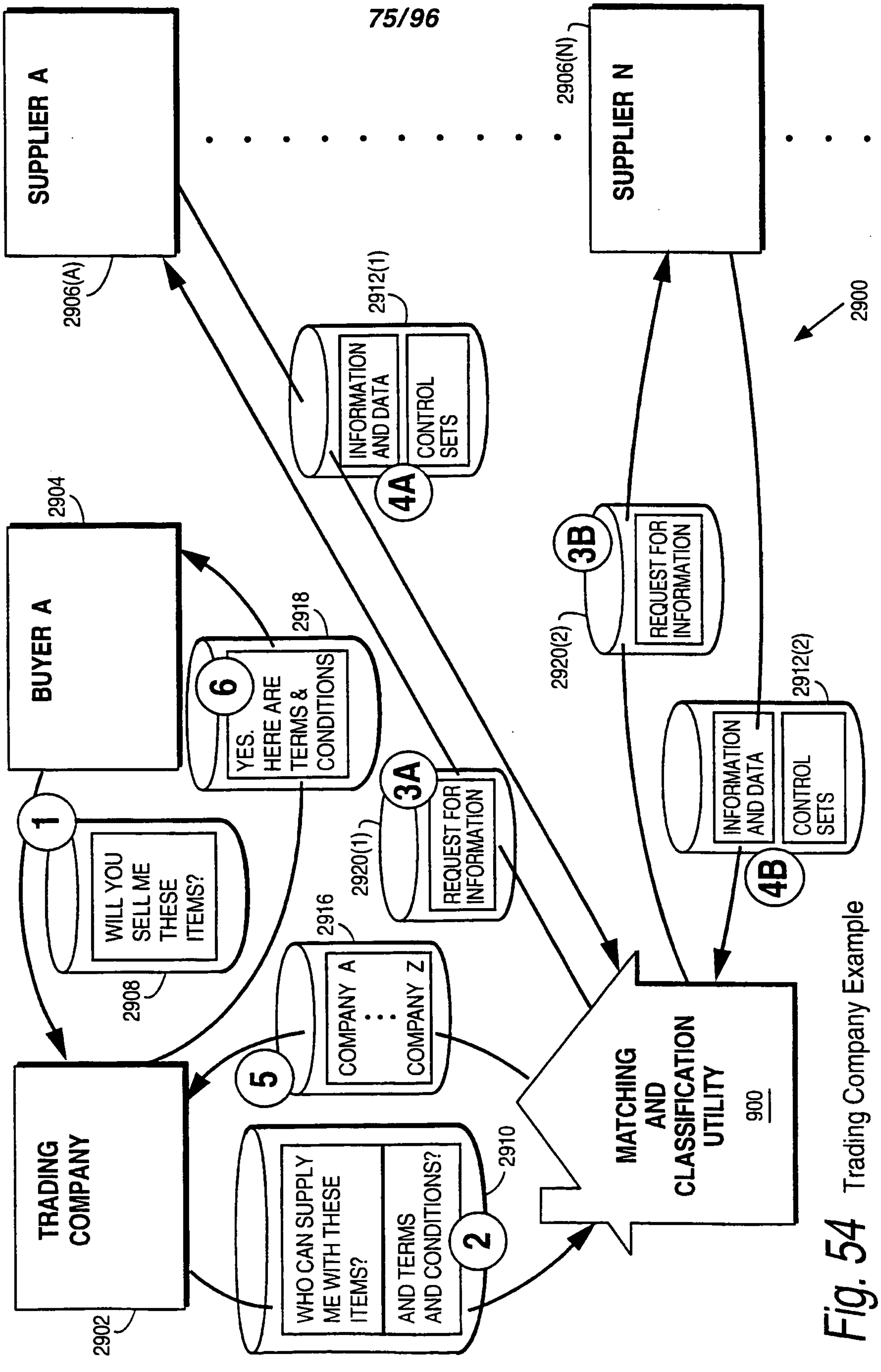

SUPPLIER A
2906(A)
SUPPLIER N
2906(N)
BUYER A
2904
TRADING COMPANY
2902
MATCHING AND CLASSIFICATION UTILITY
900
2900
1
WILL YOU SELL ME THESE ITEMS?
2908
2
WHO CAN SUPPLY ME WITH THESE ITEMS?
AND TERMS AND CONDITIONS?
2910
3A
REQUEST FOR INFORMATION
2920(1)
3B
REQUEST FOR INFORMATION
2920(2)
4A
INFORMATION AND DATA
CONTROL SETS
2912(1)
4B
INFORMATION AND DATA
CONTROL SETS
2912(2)
5
COMPANY A
COMPANY Z
2916
6
YES. HERE ARE TERMS & CONDITIONS
2918

Fig. 54 Trading Company Example

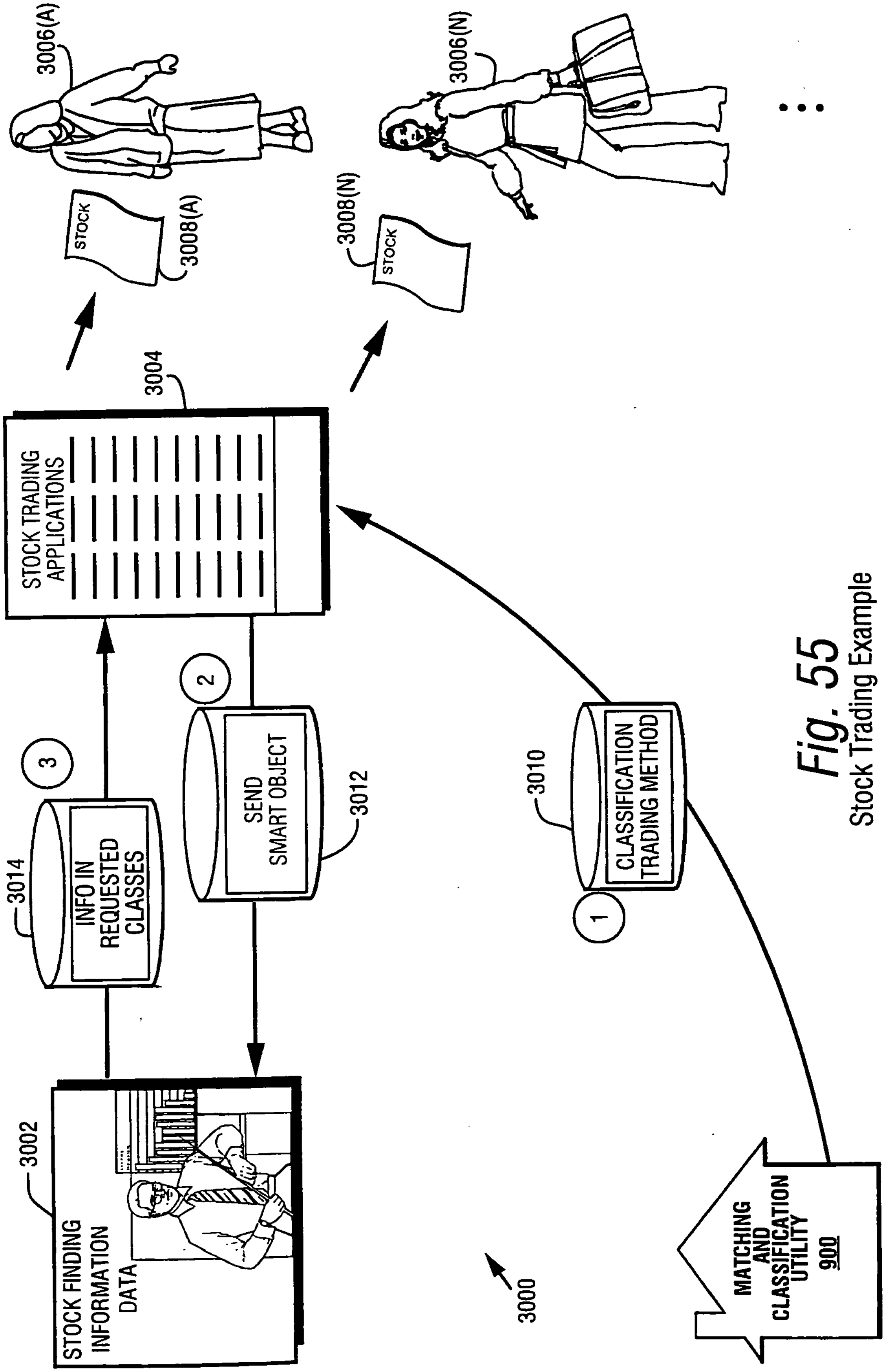

STOCK FINDING INFORMATION DATA
3002
3014
INFO IN REQUESTED CLASSES
3
SEND SMART OBJECT
2
3012
STOCK TRADING APPLICATIONS
3004
3008(A)
STOCK
3006(A)
3008(N)
STOCK
3006(N)
3010
CLASSIFICATION TRADING METHOD
1
MATCHING AND CLASSIFICATION UTILITY
900
3000

Fig. 55
Stock Trading Example

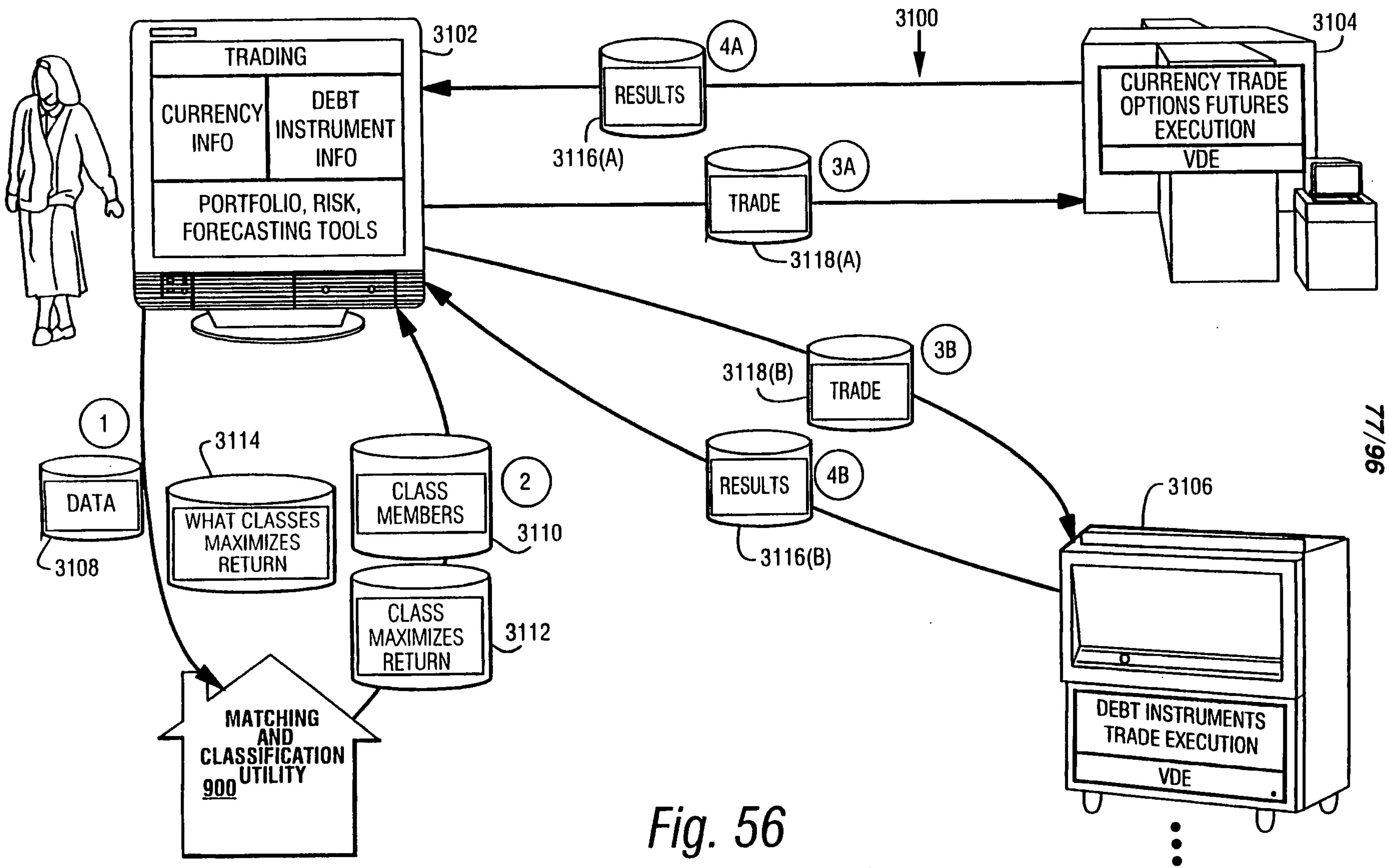

3100
3102
TRADING
CURRENCY INFO
DEBT INSTRUMENT INFO
PORTFOLIO, RISK, FORECASTING TOOLS
4A
RESULTS
3116(A)
3A
TRADE
3118(A)
3104
CURRENCY TRADE OPTIONS FUTURES EXECUTION
VDE
3B
TRADE
3118(B)
4B
RESULTS
3116(B)
3106
DEBT INSTRUMENTS TRADE EXECUTION
VDE
1
DATA
3108
3114
WHAT CLASSES MAXIMIZES RETURN
2
CLASS MEMBERS
3110
CLASS MAXIMIZES RETURN
3112
MATCHING AND CLASSIFICATION UTILITY
900

Fig. 56

Currency Trading Example

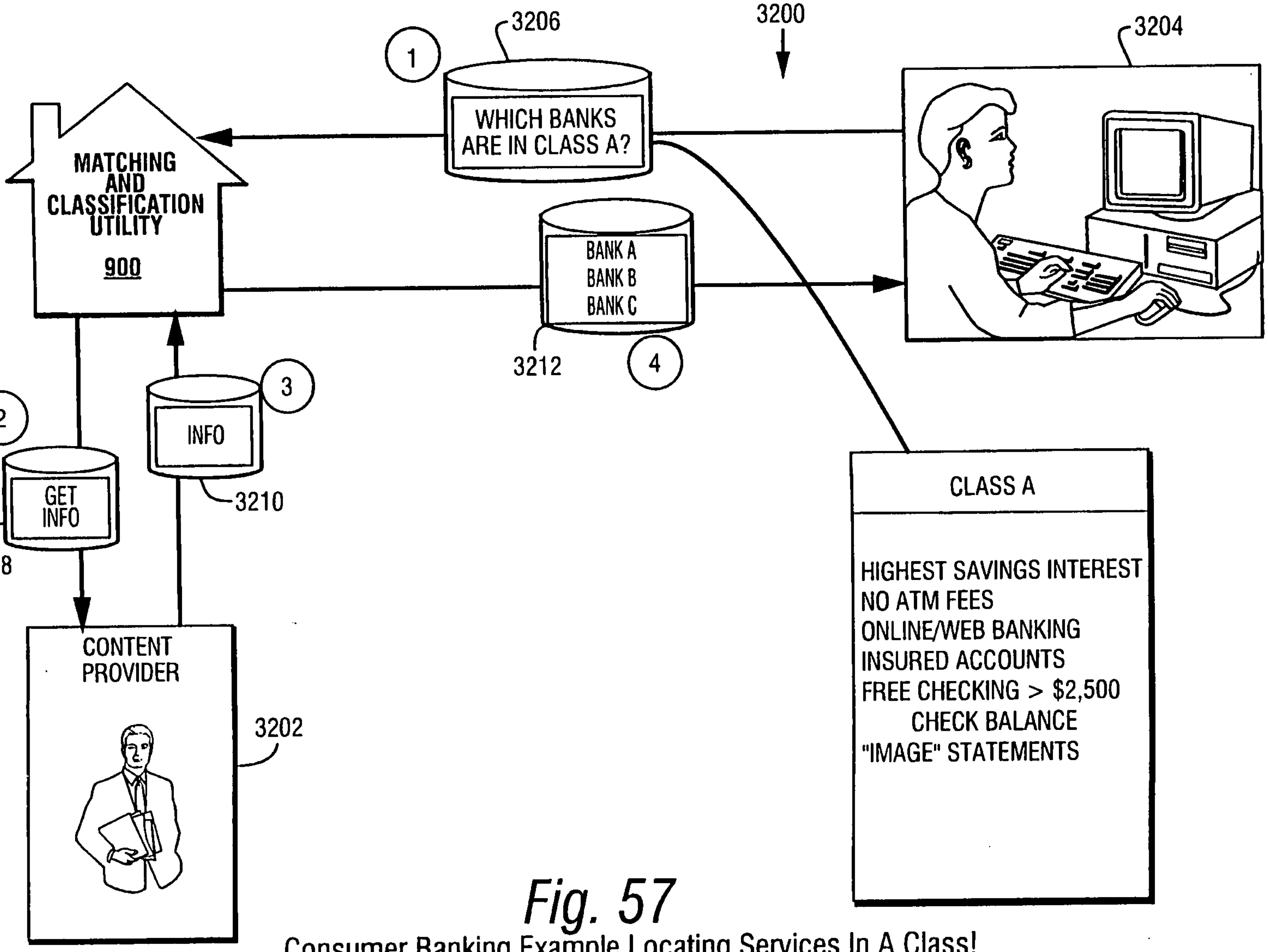

3206
3200
3204
1
WHICH BANKS
ARE IN CLASS A?
MATCHING
AND
CLASSIFICATION
UTILITY
900
BANK A
BANK B
BANK C
3212
4
2
3
INFO
GET
INFO
3210
3208
CONTENT
PROVIDER
3202
CLASS A
HIGHEST SAVINGS INTEREST
NO ATM FEES
ONLINE/WEB BANKING
INSURED ACCOUNTS
FREE CHECKING > $2,500
CHECK BALANCE
"IMAGE" STATEMENTS

Fig. 57
Consumer Banking Example Locating Services In A Class!

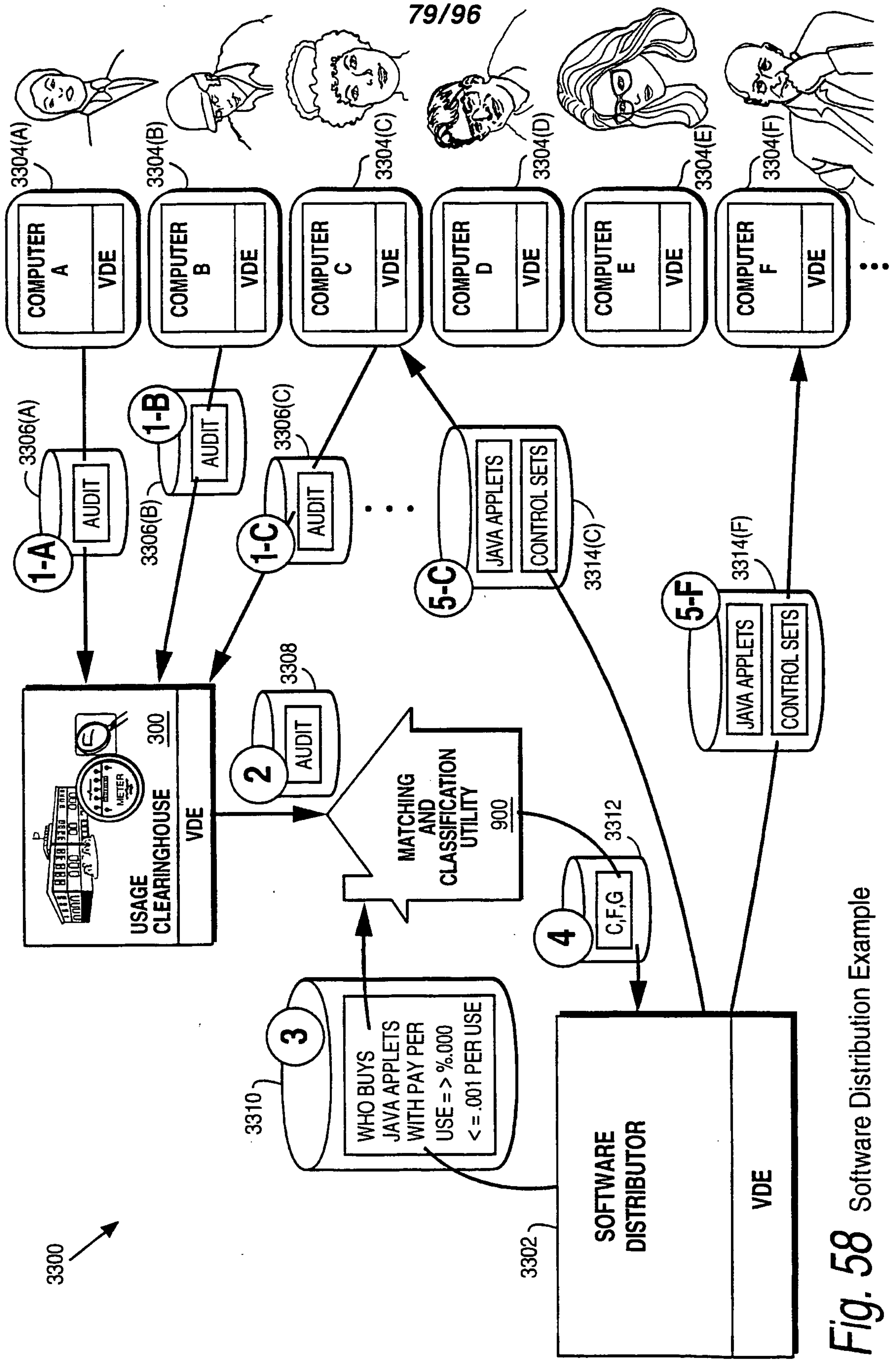

3300
USAGE CLEARINGHOUSE
300
VDE
METER
1-A
AUDIT
3306(A)
1-B
AUDIT
3306(B)
1-C
AUDIT
3306(C)
2
AUDIT
3308
MATCHING AND CLASSIFICATION UTILITY
900
3
WHO BUYS JAVA APPLETS WITH PAY PER USE => %.000 <= .001 PER USE
3310
4
C,F,G
3312
SOFTWARE DISTRIBUTOR
VDE
3302
5-C
JAVA APPLETS
CONTROL SETS
3314(C)
5-F
JAVA APPLETS
CONTROL SETS
3314(F)
COMPUTER A
VDE
3304(A)
COMPUTER B
VDE
3304(B)
COMPUTER C
VDE
3304(C)
COMPUTER D
VDE
3304(D)
COMPUTER E
VDE
3304(E)
COMPUTER F
VDE
3304(F)

Fig. 58 Software Distribution Example

SUBSTITUTE SHEET (RULE 26)

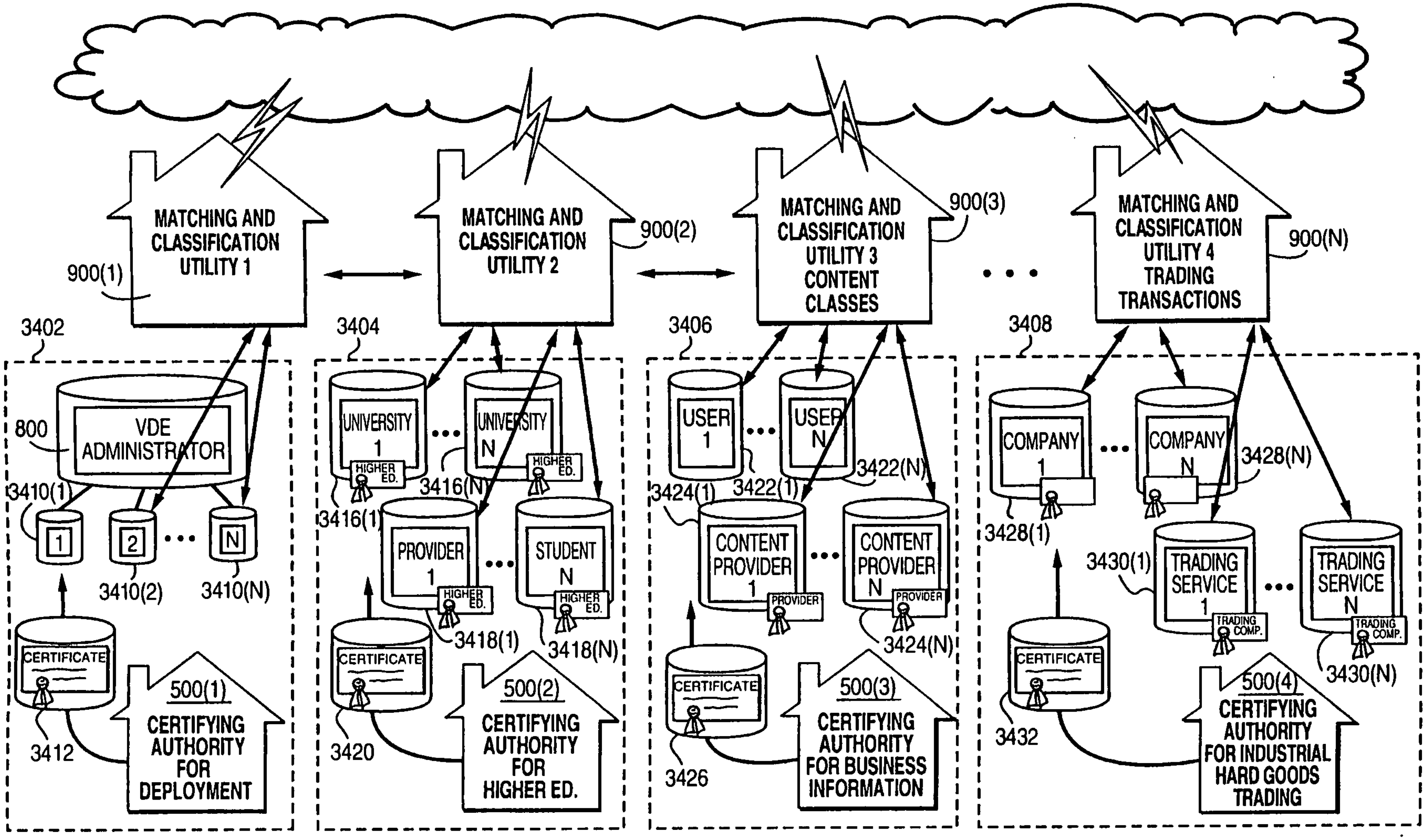

MATCHING AND CLASSIFICATION UTILITY 1
900(1)
MATCHING AND CLASSIFICATION UTILITY 2
900(2)
MATCHING AND CLASSIFICATION UTILITY 3 CONTENT CLASSES
900(3)
MATCHING AND CLASSIFICATION UTILITY 4 TRADING TRANSACTIONS
900(N)
3402
800
VDE ADMINISTRATOR
3410(1)
1
2
N
3410(2)
3410(N)
CERTIFICATE
3412
500(1)
CERTIFYING AUTHORITY FOR DEPLOYMENT
3404
UNIVERSITY 1
HIGHER ED.
3416(1)
UNIVERSITY N
3416(N)
PROVIDER 1
3418(1)
STUDENT N
3418(N)
3420
500(2)
CERTIFYING AUTHORITY FOR HIGHER ED.
3406
USER 1
USER N
3422(1)
3422(N)
CONTENT PROVIDER 1
CONTENT PROVIDER N
PROVIDER
3424(1)
3424(N)
3426
500(3)
CERTIFYING AUTHORITY FOR BUSINESS INFORMATION
3408
COMPANY 1
COMPANY N
3428(1)
3428(N)
TRADING SERVICE 1
TRADING SERVICE N
TRADING COMP.
3430(1)
3430(N)
3432
500(4)
CERTIFYING AUTHORITY FOR INDUSTRIAL HARD GOODS TRADING
3400

Fig. 59 Matching & Classification Utilities Provide Services To Authenticated Classes Of Nodes, Users, Content Services, Transaction Services.

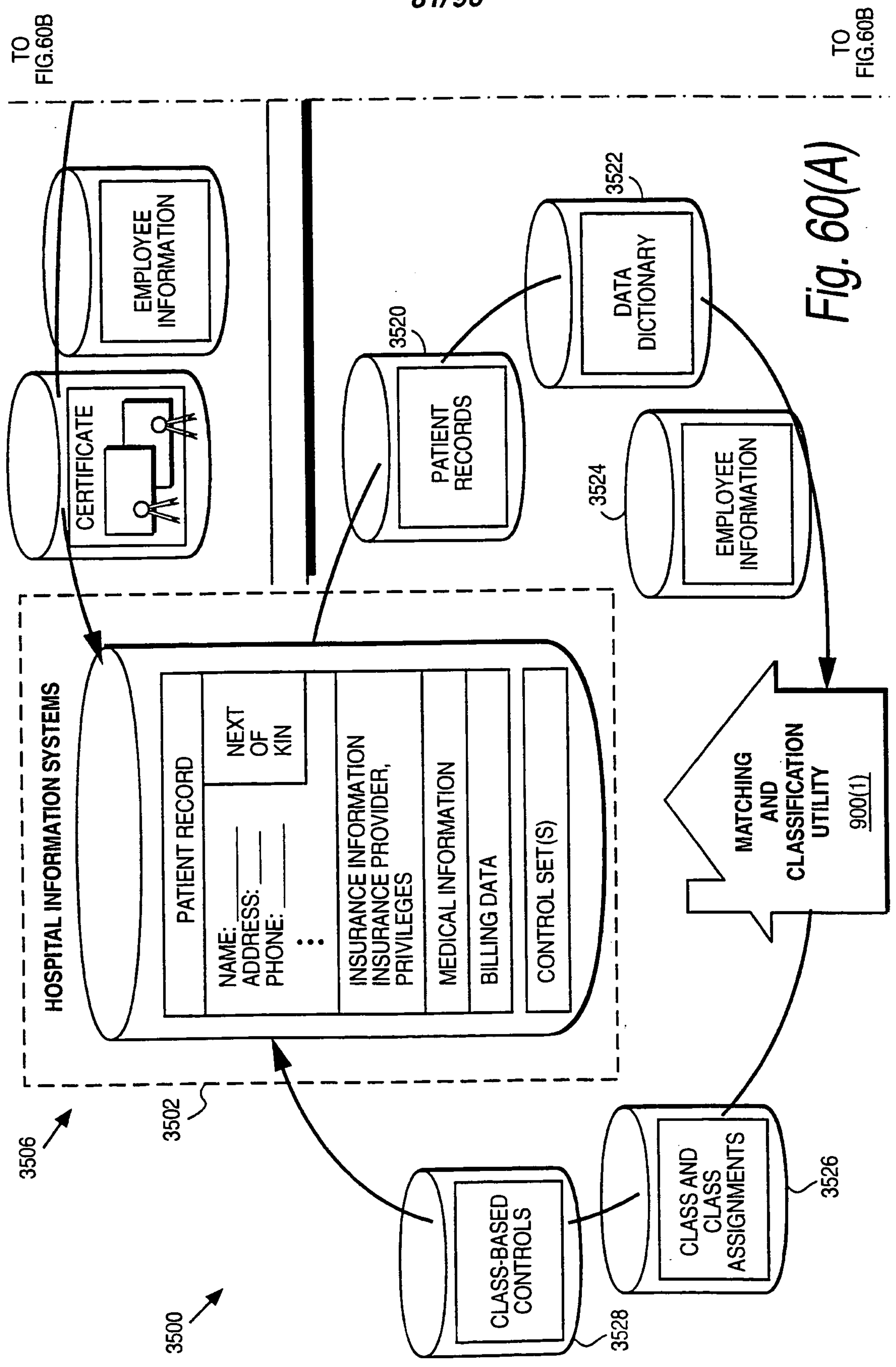

TO FIG.60B
TO FIG.60B
EMPLOYEE INFORMATION
CERTIFICATE
3520
PATIENT RECORDS
3522
DATA DICTIONARY
3524
EMPLOYEE INFORMATION
HOSPITAL INFORMATION SYSTEMS
PATIENT RECORD
NAME:
ADDRESS:
PHONE:
NEXT OF KIN
INSURANCE INFORMATION INSURANCE PROVIDER, PRIVILEGES
MEDICAL INFORMATION
BILLING DATA
CONTROL SET(S)
MATCHING AND CLASSIFICATION UTILITY
900(1)
3506
3502
3500
CLASS-BASED CONTROLS
3528
CLASS AND CLASS ASSIGNMENTS
3526

Fig. 60(A)

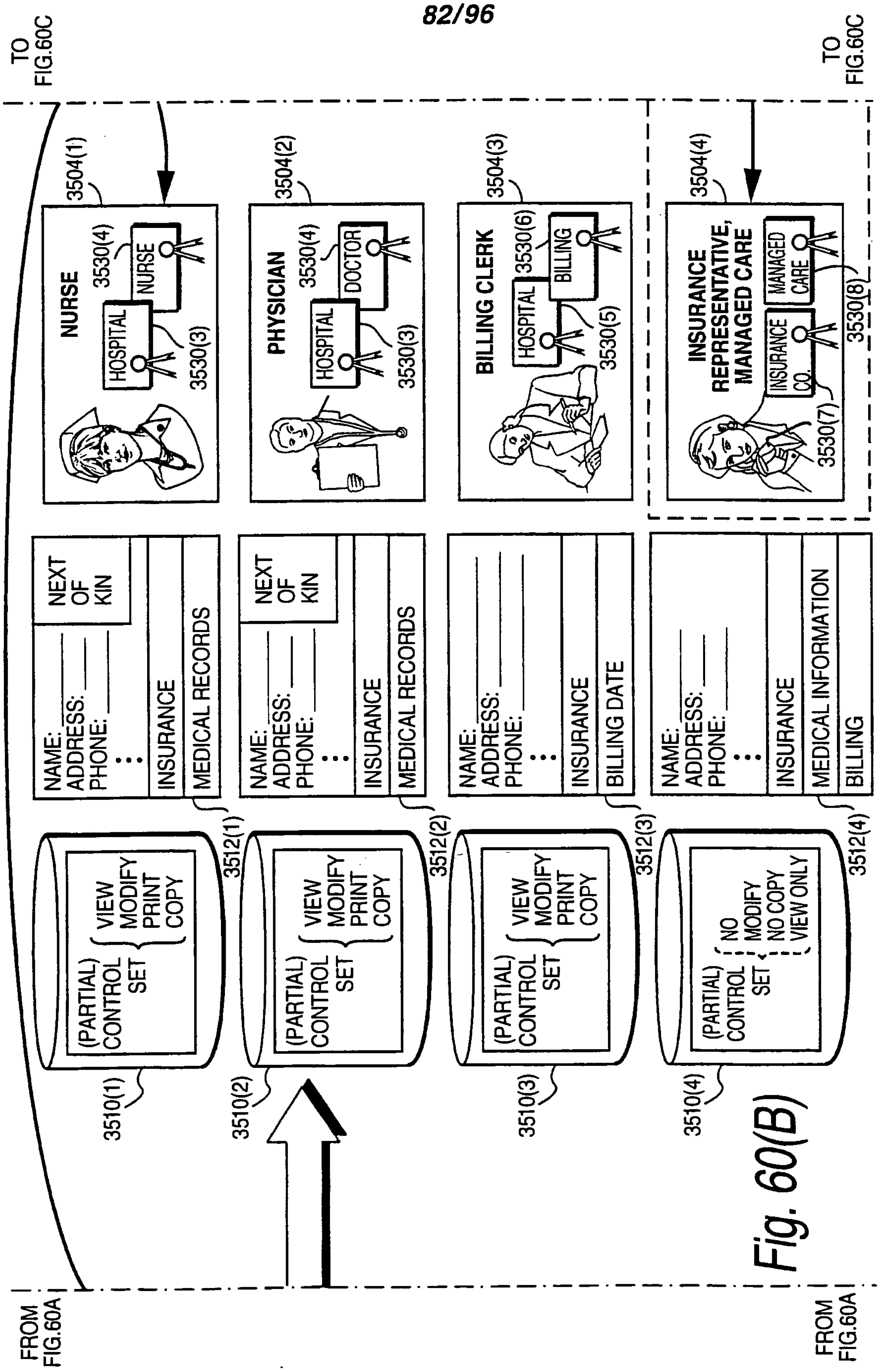

TO FIG.60C
TO FIG.60C
FROM FIG.60A
FROM FIG.60A
3504(1)
NURSE
HOSPITAL
NURSE
3530(4)
3530(3)
3504(2)
PHYSICIAN
HOSPITAL
DOCTOR
3530(4)
3530(3)
3504(3)
BILLING CLERK
HOSPITAL
BILLING
3530(6)
3530(5)
3504(4)
INSURANCE REPRESENTATIVE, MANAGED CARE
INSURANCE CO.
MANAGED CARE
3530(7)
3530(8)
NAME:
ADDRESS:
PHONE:
NEXT OF KIN
INSURANCE
MEDICAL RECORDS
NAME:
ADDRESS:
PHONE:
NEXT OF KIN
INSURANCE
MEDICAL RECORDS
NAME:
ADDRESS:
PHONE:
INSURANCE
BILLING DATE
NAME:
ADDRESS:
PHONE:
INSURANCE
MEDICAL INFORMATION
BILLING
3512(1)
3512(2)
3512(3)
3512(4)
(PARTIAL) CONTROL SET
VIEW
MODIFY
PRINT
COPY
(PARTIAL) CONTROL SET
VIEW
MODIFY
PRINT
COPY
(PARTIAL) CONTROL SET
VIEW
MODIFY
PRINT
COPY
(PARTIAL) CONTROL SET
NO MODIFY
NO COPY
VIEW ONLY
3510(1)
3510(2)
3510(3)
3510(4)

Fig. 60(B)

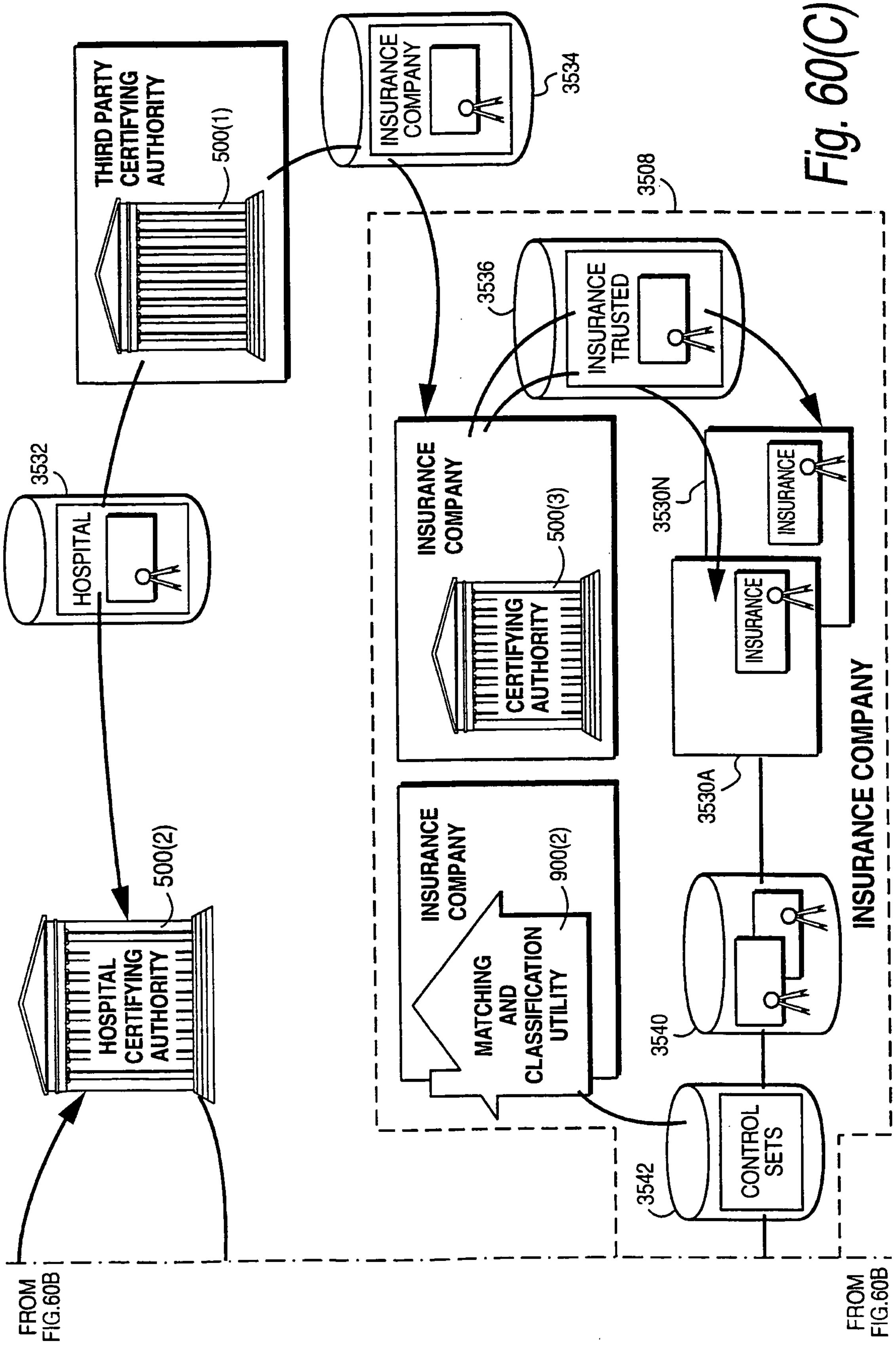


Fig. 60(C)
FROM FIG.60B
HOSPITAL CERTIFYING AUTHORITY
500(2)
HOSPITAL
3532
THIRD PARTY CERTIFYING AUTHORITY
500(1)
INSURANCE COMPANY
3534
3508
INSURANCE COMPANY
MATCHING AND CLASSIFICATION UTILITY
900(2)
INSURANCE COMPANY
CERTIFYING AUTHORITY
500(3)
3536
INSURANCE TRUSTED
3542
CONTROL SETS
3540
3530A
INSURANCE
3530N
INSURANCE
INSURANCE COMPANY
FROM FIG.60B

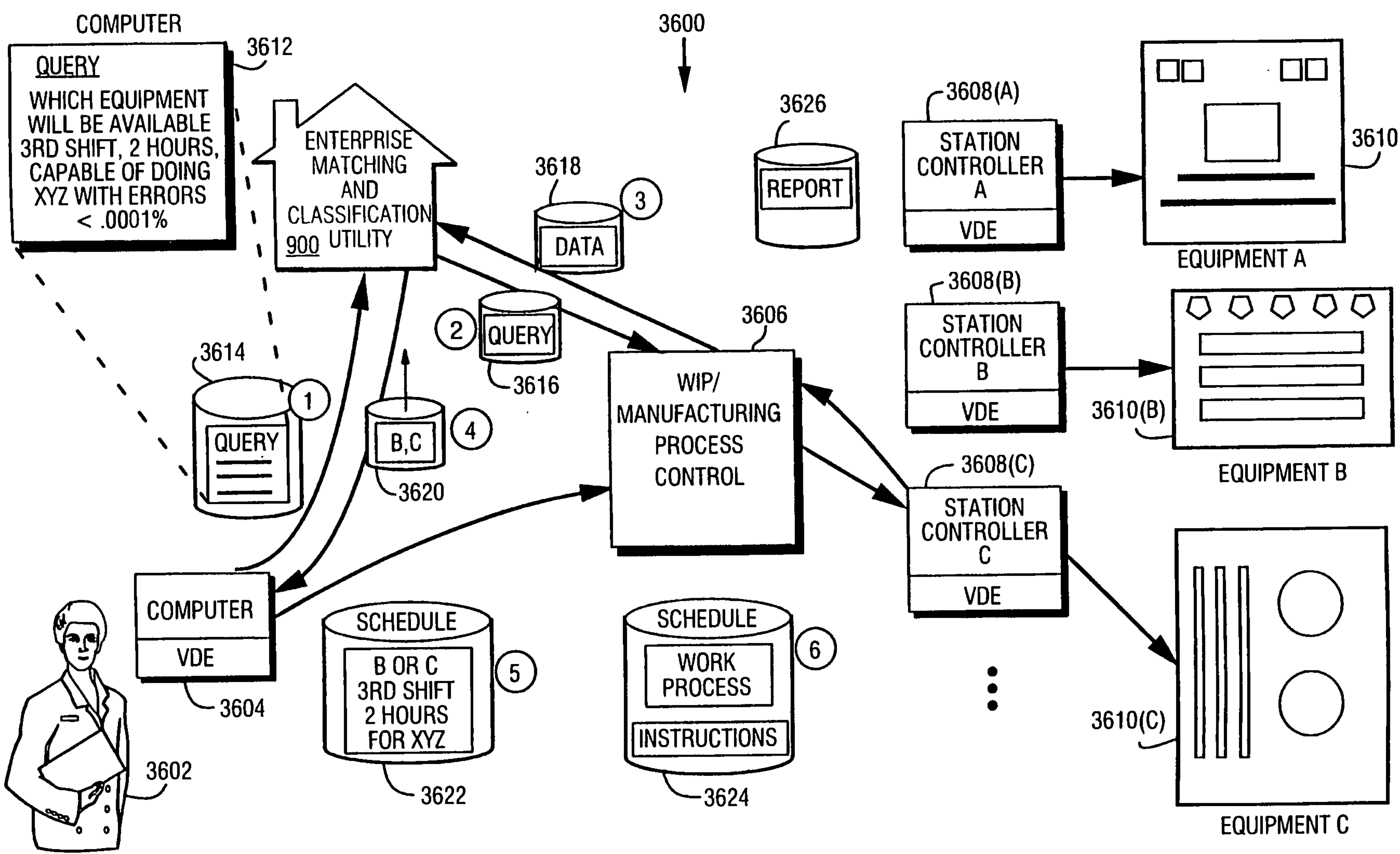

COMPUTER
QUERY
WHICH EQUIPMENT WILL BE AVAILABLE 3RD SHIFT, 2 HOURS, CAPABLE OF DOING XYZ WITH ERRORS < .0001%
3612
ENTERPRISE MATCHING AND CLASSIFICATION 900 UTILITY
3600
3626
REPORT
3608(A)
STATION CONTROLLER A
VDE
3610(A)
EQUIPMENT A
3618
DATA
3
3614
QUERY
1
2
QUERY
3616
3606
WIP/ MANUFACTURING PROCESS CONTROL
3608(B)
STATION CONTROLLER B
VDE
3610(B)
EQUIPMENT B
B,C
4
3620
3608(C)
STATION CONTROLLER C
VDE
3610(C)
EQUIPMENT C
COMPUTER
VDE
3604
3602
SCHEDULE
B OR C 3RD SHIFT 2 HOURS FOR XYZ
5
3622
SCHEDULE
WORK PROCESS
INSTRUCTIONS
6
3624

Fig. 61 Workflow Example

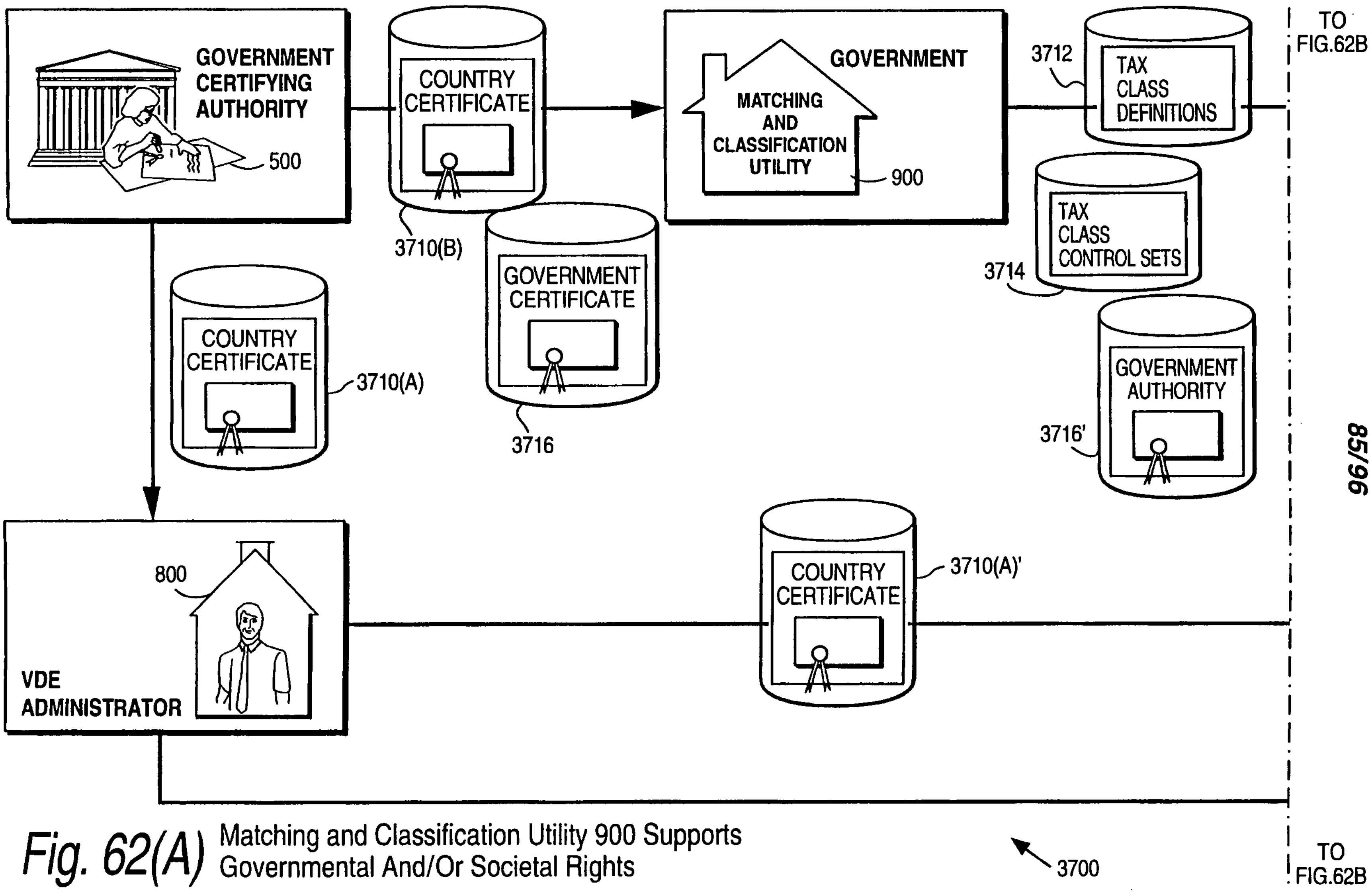

GOVERNMENT CERTIFYING AUTHORITY
500
COUNTRY CERTIFICATE
3710(B)
GOVERNMENT CERTIFICATE
3716
GOVERNMENT
MATCHING AND CLASSIFICATION UTILITY
900
3712
TAX CLASS DEFINITIONS
TAX CLASS CONTROL SETS
3714
GOVERNMENT AUTHORITY
3716'
TO FIG.62B
COUNTRY CERTIFICATE
3710(A)
800
VDE ADMINISTRATOR
COUNTRY CERTIFICATE
3710(A)'
3700
TO FIG.62B

Fig. 62(A) Matching and Classification Utility 900 Supports Governmental And/Or Societal Rights

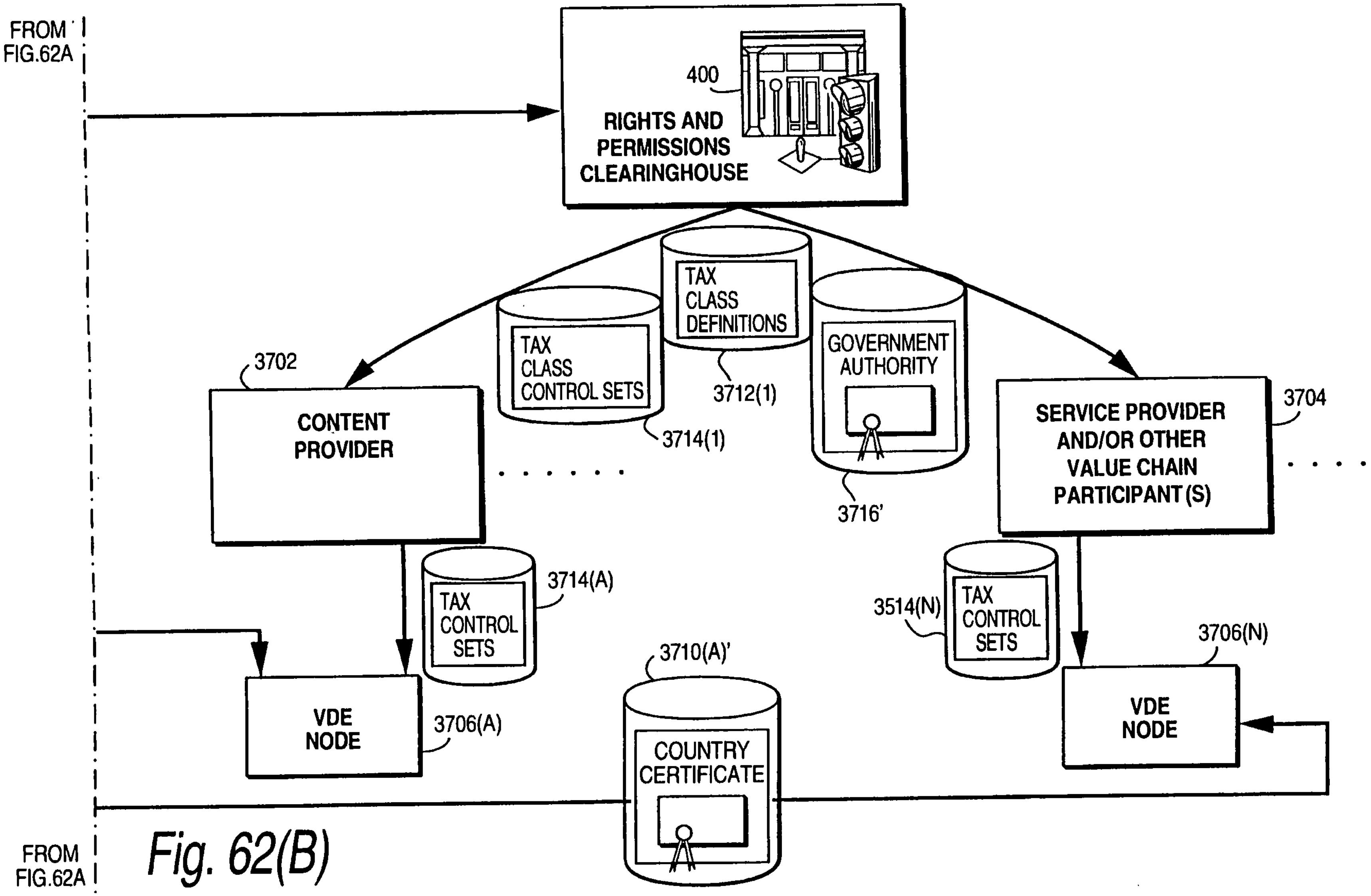

FROM FIG.62A
400
RIGHTS AND PERMISSIONS CLEARINGHOUSE
TAX CLASS DEFINITIONS
3712(1)
TAX CLASS CONTROL SETS
3714(1)
GOVERNMENT AUTHORITY
3716'
3702
CONTENT PROVIDER
3704
SERVICE PROVIDER AND/OR OTHER VALUE CHAIN PARTICIPANT(S)
TAX CONTROL SETS
3714(A)
TAX CONTROL SETS
3514(N)
VDE NODE
3706(A)
VDE NODE
3706(N)
COUNTRY CERTIFICATE
3710(A)'
FROM FIG.62A

Fig. 62(B)

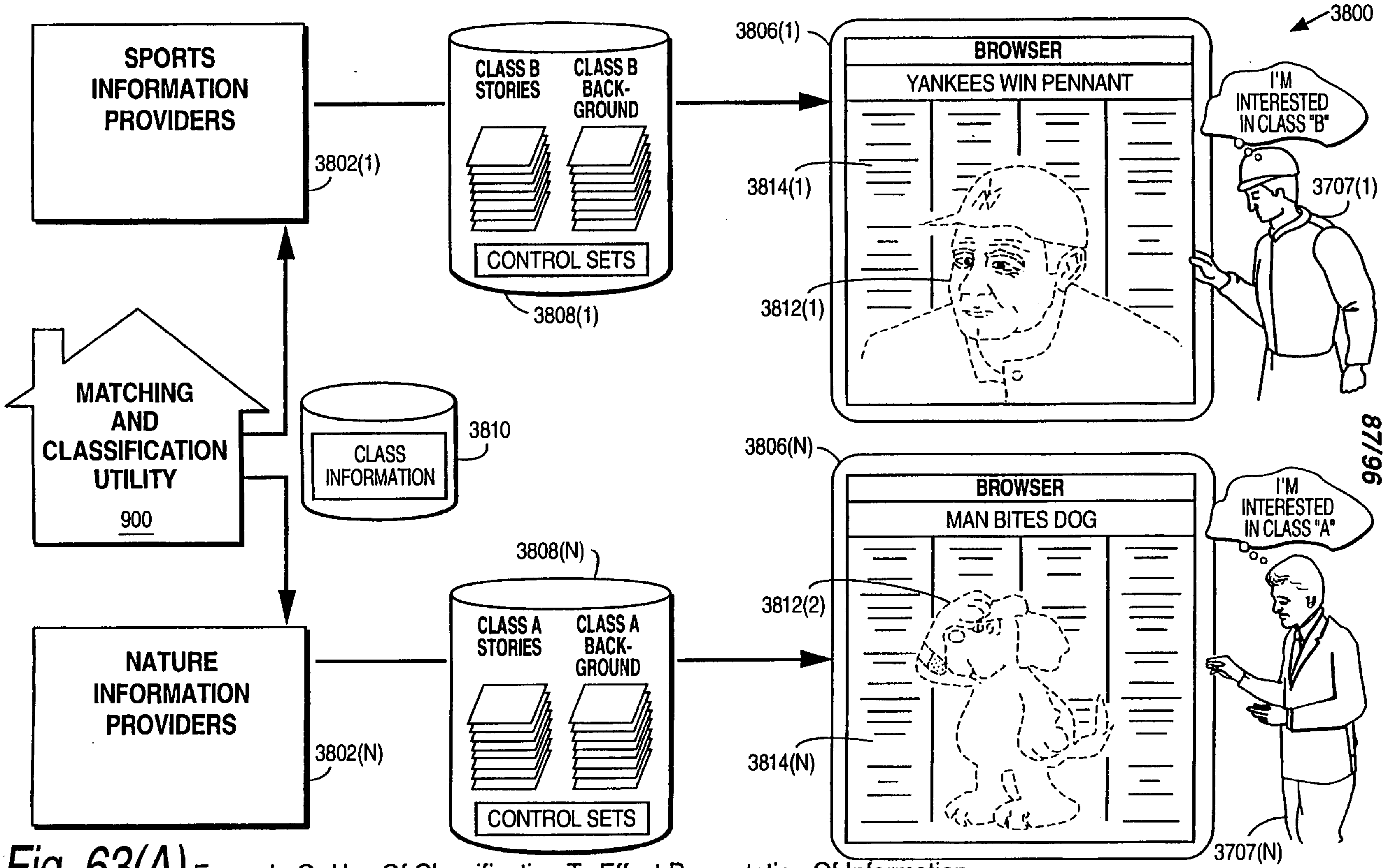

3800
SPORTS INFORMATION PROVIDERS
3802(1)
CLASS B STORIES
CLASS B BACK-GROUND
CONTROL SETS
3808(1)
3806(1)
BROWSER
YANKEES WIN PENNANT
3814(1)
3812(1)
I'M INTERESTED IN CLASS "B"
3707(1)
MATCHING AND CLASSIFICATION UTILITY
900
CLASS INFORMATION
3810
3808(N)
NATURE INFORMATION PROVIDERS
3802(N)
CLASS A STORIES
CLASS A BACK-GROUND
CONTROL SETS
3806(N)
BROWSER
MAN BITES DOG
3812(2)
3814(N)
I'M INTERESTED IN CLASS "A"
3707(N)

Fig. 63(A) Example Or Use Of Classification To Effect Presentation Of Information.

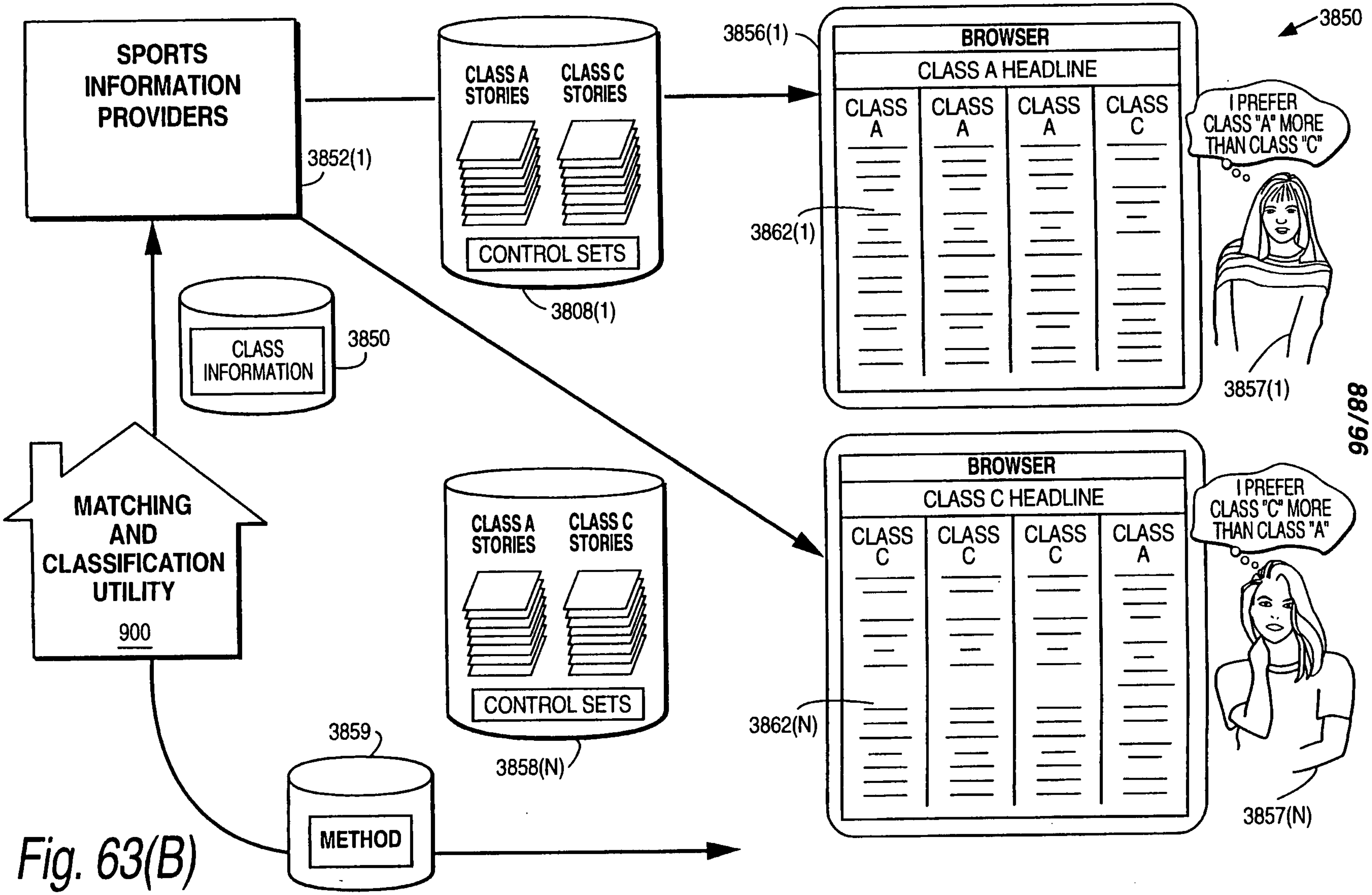

3850
SPORTS INFORMATION PROVIDERS
3852(1)
CLASS A STORIES
CLASS C STORIES
CONTROL SETS
3808(1)
3856(1)
BROWSER
CLASS A HEADLINE
CLASS A
CLASS A
CLASS A
CLASS C
3862(1)
I PREFER CLASS "A" MORE THAN CLASS "C"
3857(1)
CLASS INFORMATION
3850
MATCHING AND CLASSIFICATION UTILITY
900
CLASS A STORIES
CLASS C STORIES
CONTROL SETS
3858(N)
3859
METHOD
BROWSER
CLASS C HEADLINE
CLASS C
CLASS C
CLASS C
CLASS A
3862(N)
I PREFER CLASS "C" MORE THAN CLASS "A"
3857(N)

Fig. 63(B)

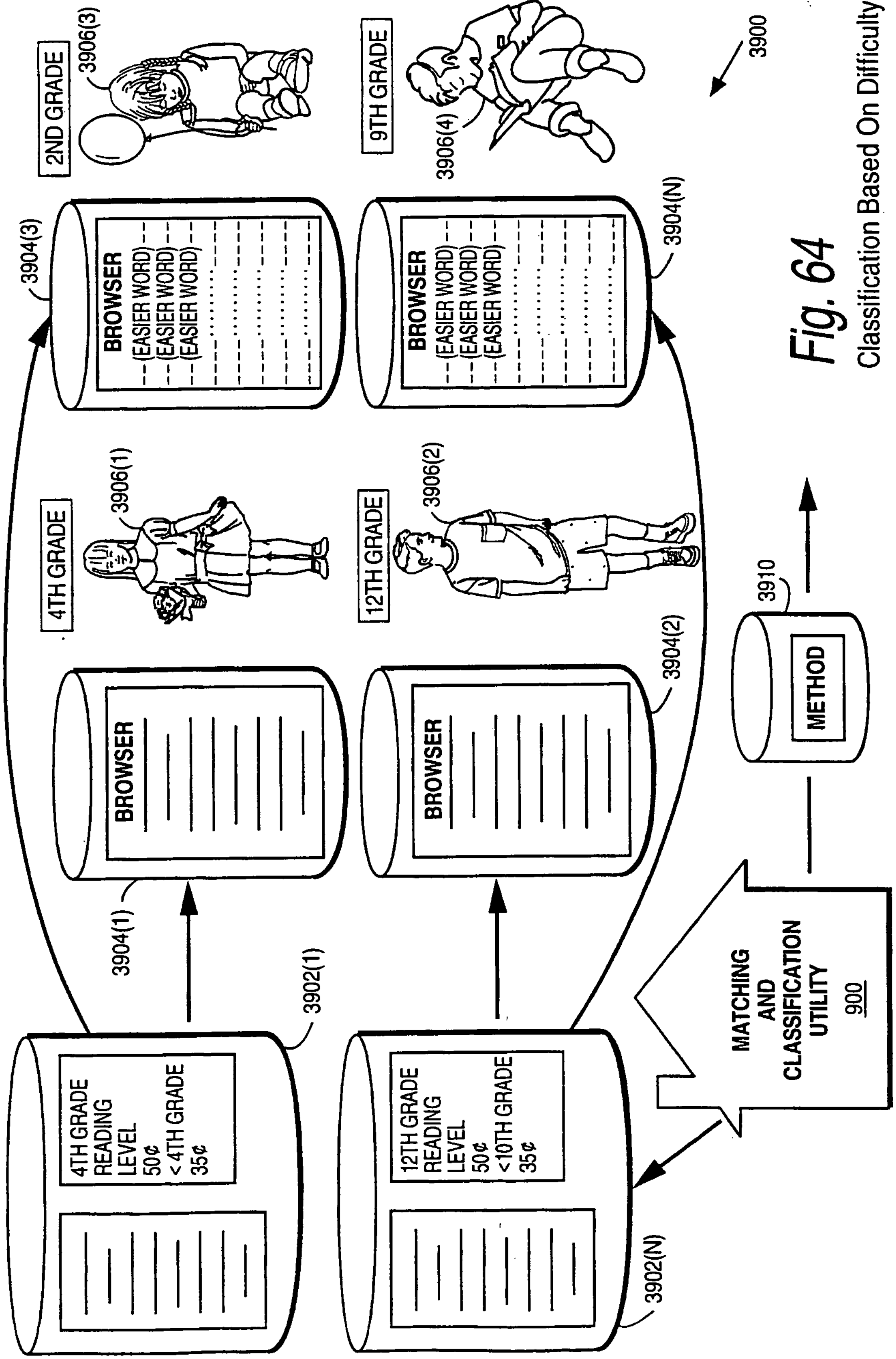

2ND GRADE
3906(3)
9TH GRADE
3906(4)
3900
3904(3)
BROWSER
(EASIER WORD)
(EASIER WORD)
(EASIER WORD)
3904(N)
4TH GRADE
3906(1)
12TH GRADE
3906(2)
BROWSER
3904(1)
3904(2)
3910
METHOD
MATCHING AND CLASSIFICATION UTILITY
900
3902(1)
4TH GRADE READING LEVEL 50¢ <4TH GRADE 35¢
12TH GRADE READING LEVEL 50¢ <10TH GRADE 35¢
3902(N)

Fig. 64
Classification Based On Difficulty

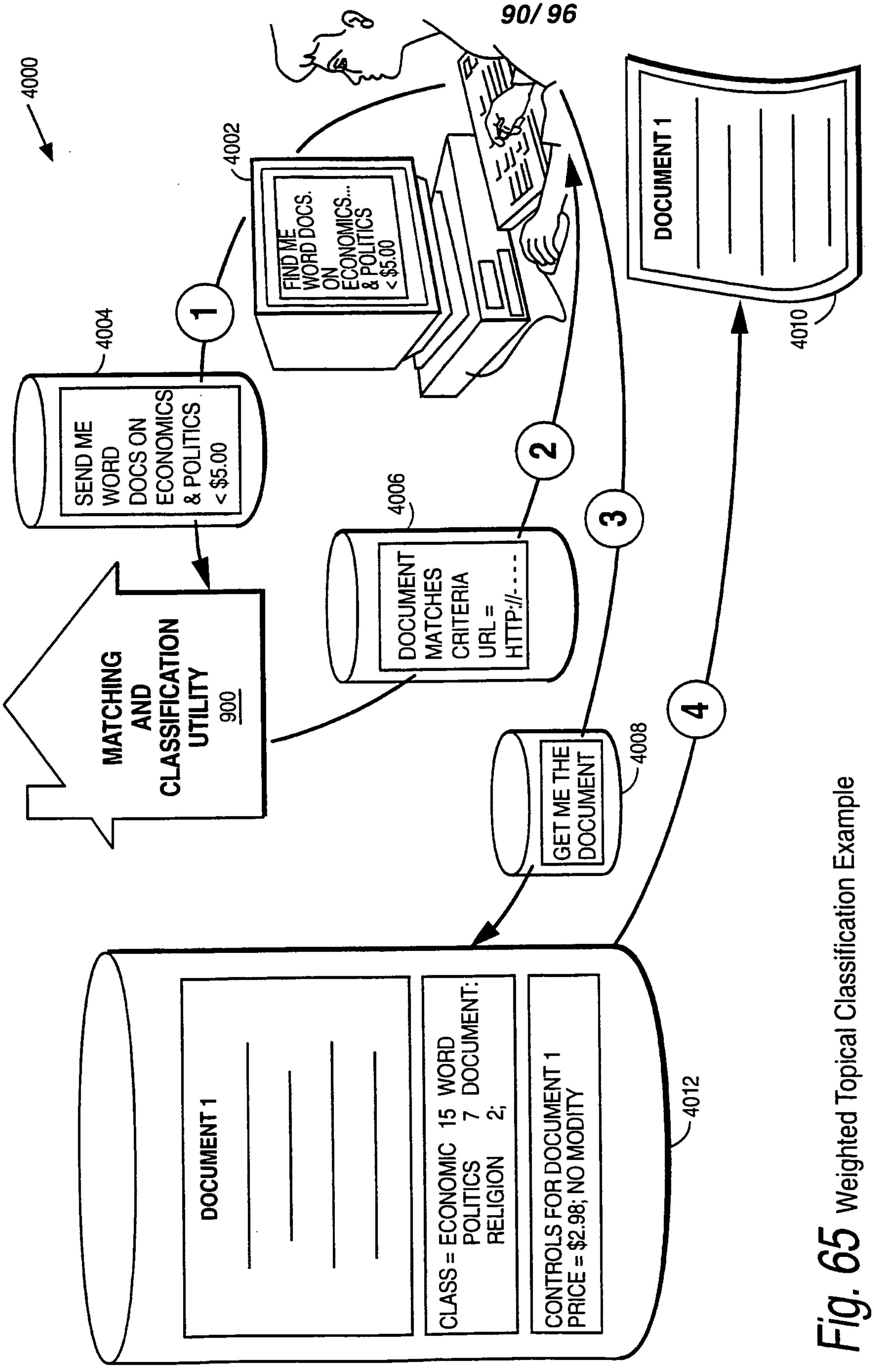

4000
4002
FIND ME WORD DOCS. ON ECONOMICS... & POLITICS < $5.00
1
4004
SEND ME WORD DOCS ON ECONOMICS & POLITICS < $5.00
MATCHING AND CLASSIFICATION UTILITY 900
2
4006
DOCUMENT MATCHES CRITERIA URL = HTTP://- - - -
3
4008
GET ME THE DOCUMENT
4
DOCUMENT 1
4010
DOCUMENT 1
CLASS = ECONOMIC 15 WORD POLITICS 7 DOCUMENT: RELIGION 2;
CONTROLS FOR DOCUMENT 1 PRICE = $2.98; NO MODITY
4012

Fig. 65 Weighted Topical Classification Example

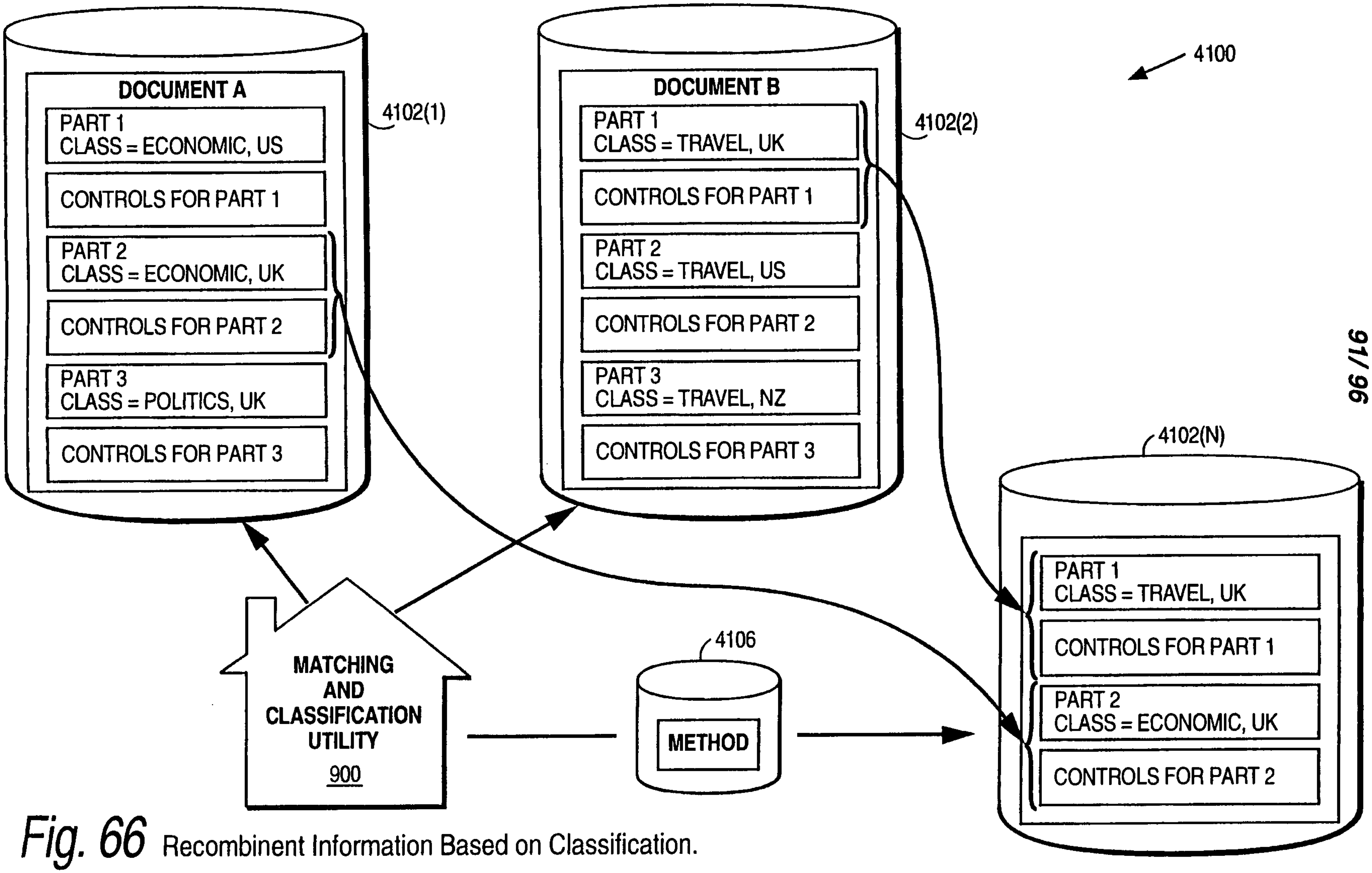

4100
DOCUMENT A
PART 1
CLASS = ECONOMIC, US
CONTROLS FOR PART 1
PART 2
CLASS = ECONOMIC, UK
CONTROLS FOR PART 2
PART 3
CLASS = POLITICS, UK
CONTROLS FOR PART 3
4102(1)
DOCUMENT B
PART 1
CLASS = TRAVEL, UK
CONTROLS FOR PART 1
PART 2
CLASS = TRAVEL, US
CONTROLS FOR PART 2
PART 3
CLASS = TRAVEL, NZ
CONTROLS FOR PART 3
4102(2)
4102(N)
PART 1
CLASS = TRAVEL, UK
CONTROLS FOR PART 1
PART 2
CLASS = ECONOMIC, UK
CONTROLS FOR PART 2
MATCHING
AND
CLASSIFICATION
UTILITY
900
4106
METHOD

Fig. 66 Recombinent Information Based on Classification.

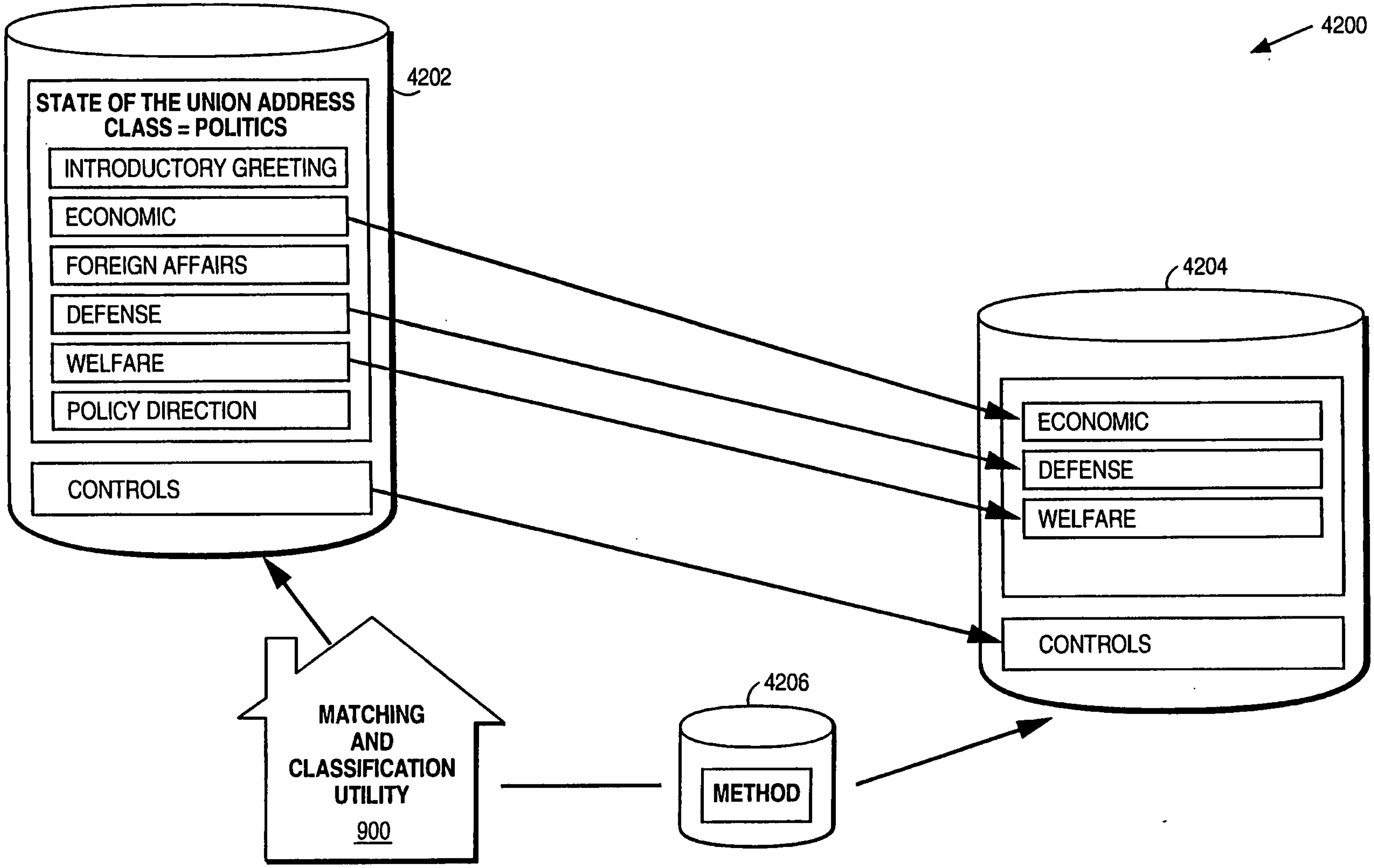

4200
4202
STATE OF THE UNION ADDRESS
CLASS = POLITICS
INTRODUCTORY GREETING
ECONOMIC
FOREIGN AFFAIRS
DEFENSE
WELFARE
POLICY DIRECTION
CONTROLS
4204
ECONOMIC
DEFENSE
WELFARE
CONTROLS
4206
METHOD
MATCHING
AND
CLASSIFICATION
UTILITY
900

Fig. 67 Nested Classification.

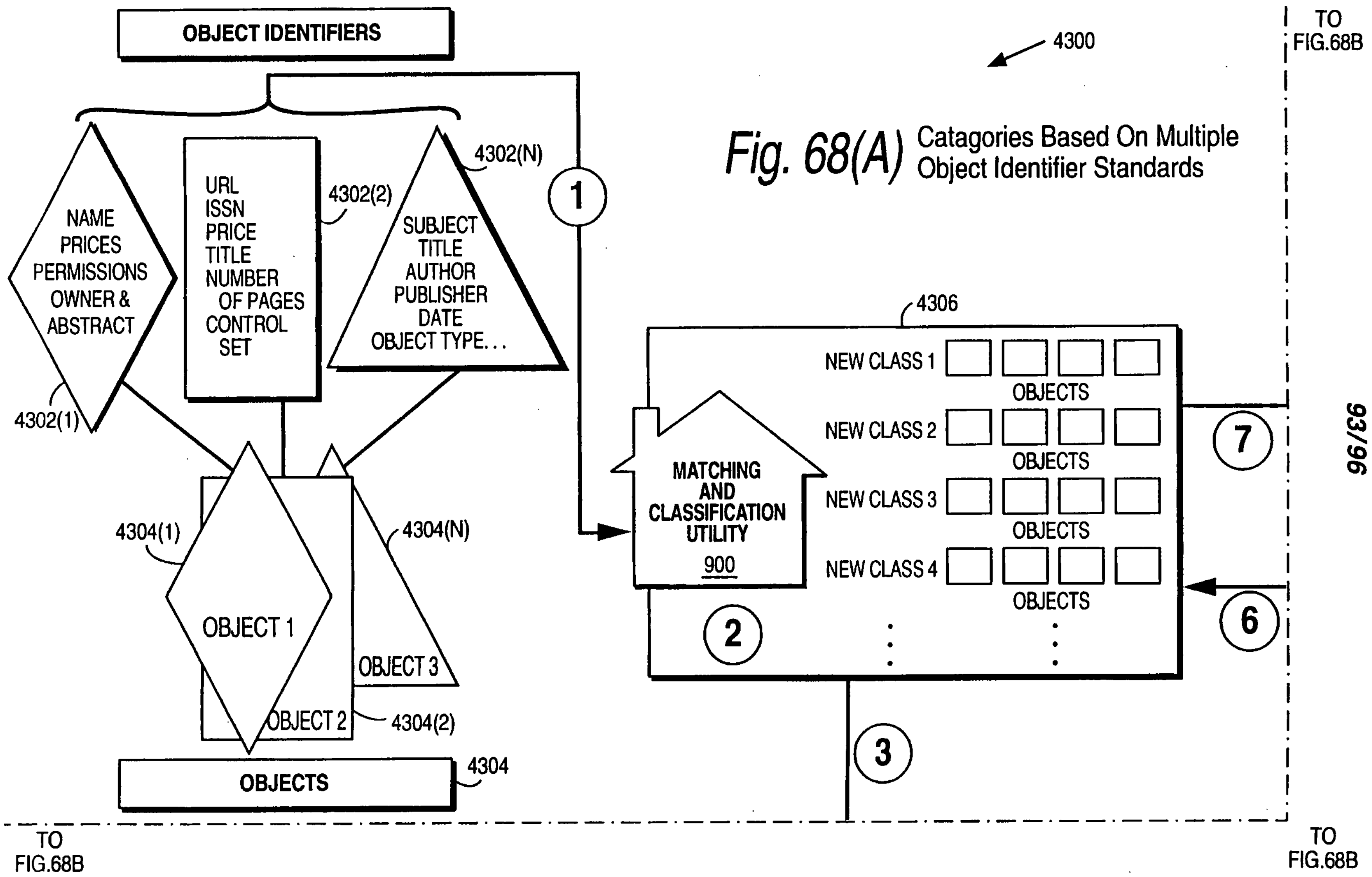

OBJECT IDENTIFIERS
NAME
PRICES
PERMISSIONS
OWNER &
ABSTRACT
4302(1)
URL
ISSN
PRICE
TITLE
NUMBER
OF PAGES
CONTROL
SET
4302(2)
SUBJECT
TITLE
AUTHOR
PUBLISHER
DATE
OBJECT TYPE...
4302(N)
1
OBJECT 1
OBJECT 2
OBJECT 3
4304(1)
4304(2)
4304(N)
OBJECTS
4304
4300
MATCHING
AND
CLASSIFICATION
UTILITY
900
2
4306
NEW CLASS 1
NEW CLASS 2
NEW CLASS 3
NEW CLASS 4
OBJECTS
OBJECTS
OBJECTS
OBJECTS
3
6
7
TO
FIG.68B
TO
FIG.68B
TO
FIG.68B

Fig. 68(A) Catagories Based On Multiple Object Identifier Standards

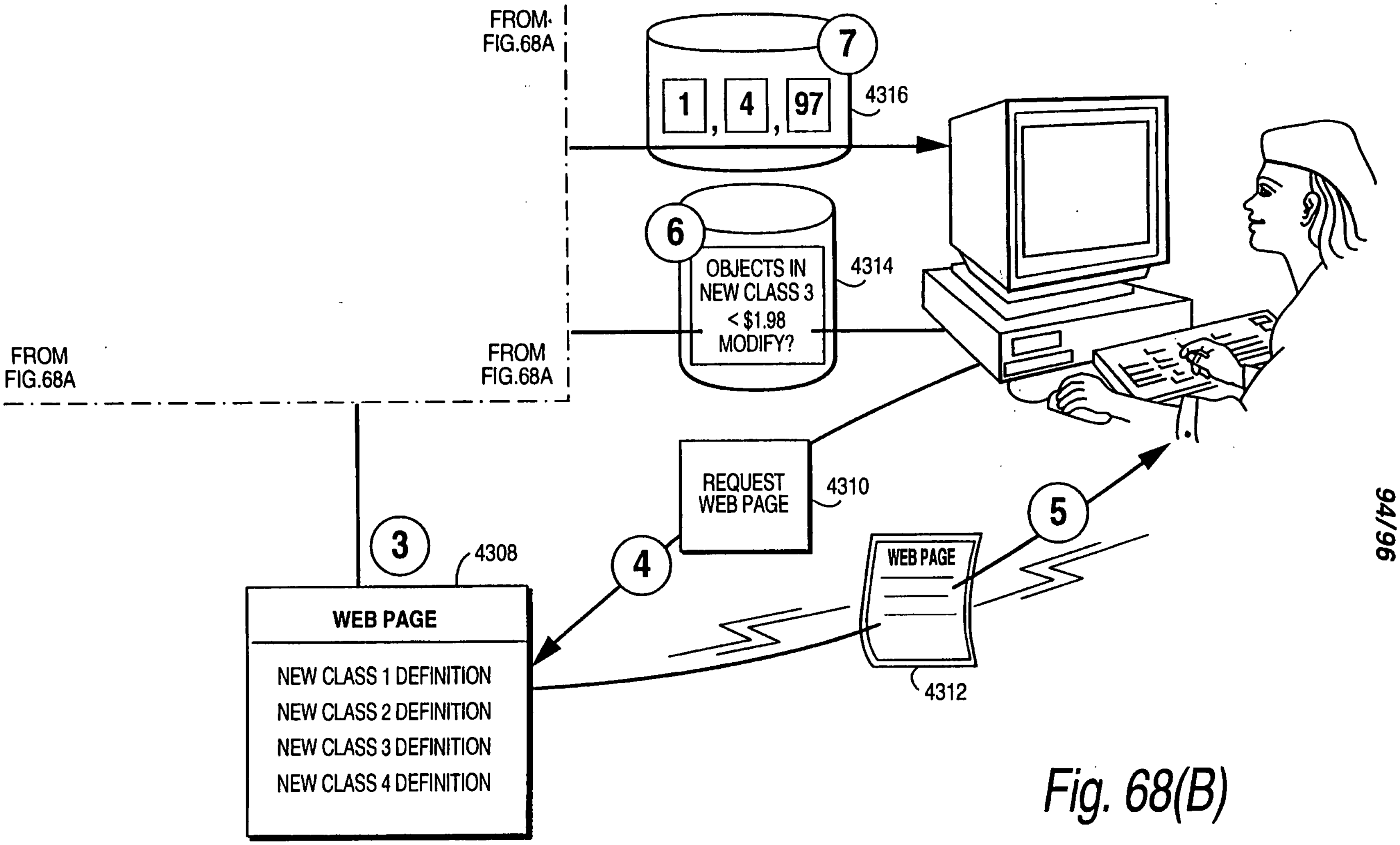
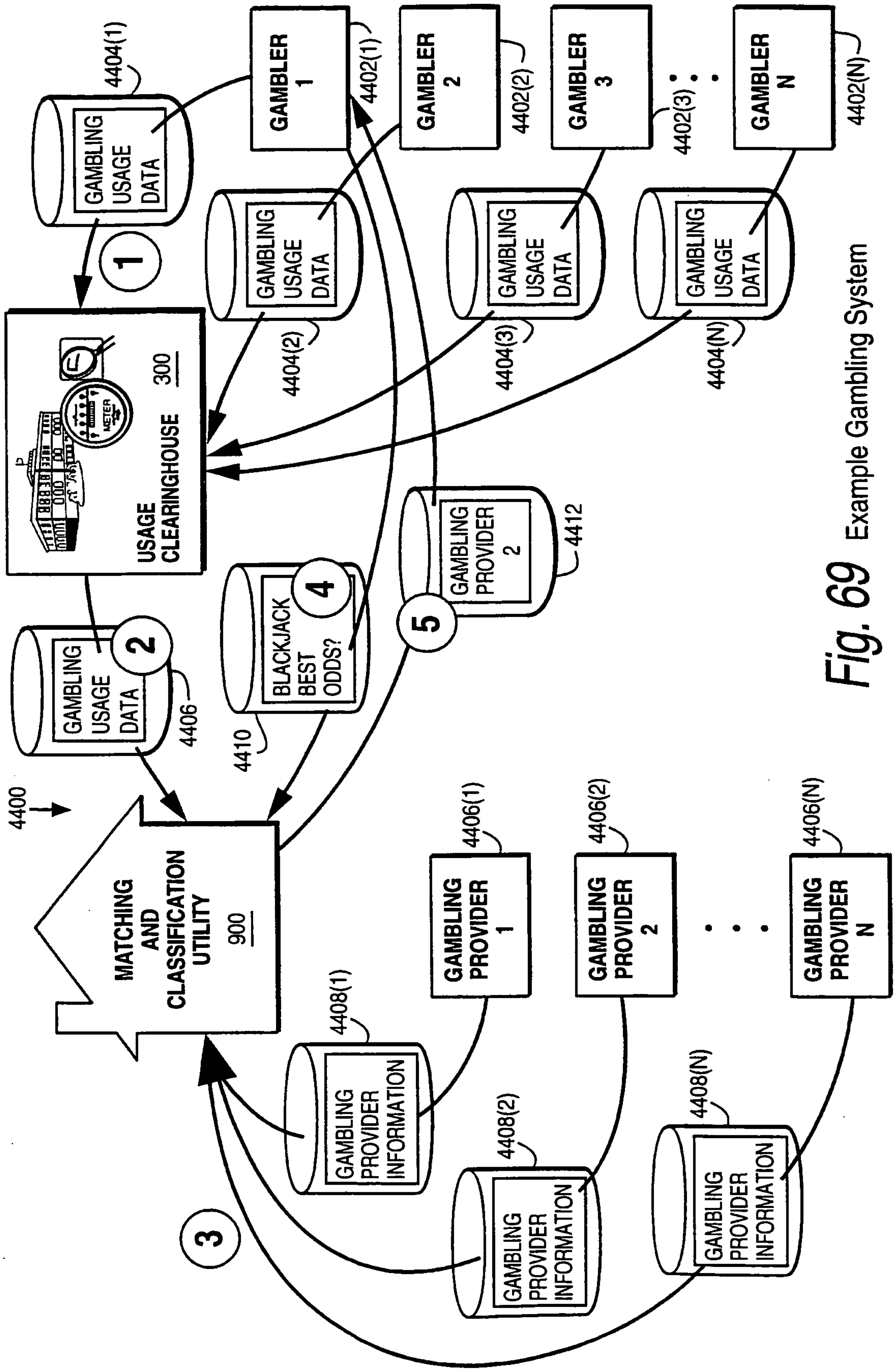

FROM
FIG.68A
7
1 , 4 , 97
4316
6
OBJECTS IN
NEW CLASS 3
< $1.98
MODIFY?
4314
FROM
FIG.68A
FROM
FIG.68A
REQUEST
WEB PAGE
4310
3
4308
4
5
WEB PAGE
4312
WEB PAGE
NEW CLASS 1 DEFINITION
NEW CLASS 2 DEFINITION
NEW CLASS 3 DEFINITION
NEW CLASS 4 DEFINITION

Fig. 68(B)



4400
GAMBLING USAGE DATA
4404(1)
1
USAGE CLEARINGHOUSE
300
METER
GAMBLER 1
4402(1)
GAMBLER 2
4402(2)
GAMBLER 3
4402(3)
GAMBLER N
4402(N)
GAMBLING USAGE DATA
4404(2)
GAMBLING USAGE DATA
4404(3)
GAMBLING USAGE DATA
4404(N)
GAMBLING USAGE DATA
2
4406
BLACKJACK BEST ODDS?
4
4410
GAMBLING PROVIDER 2
4412
5
MATCHING AND CLASSIFICATION UTILITY
900
GAMBLING PROVIDER 1
4406(1)
GAMBLING PROVIDER 2
4406(2)
GAMBLING PROVIDER N
4406(N)
GAMBLING PROVIDER INFORMATION
4408(1)
GAMBLING PROVIDER INFORMATION
4408(2)
GAMBLING PROVIDER INFORMATION
4408(N)
3

Fig. 69 Example Gambling System

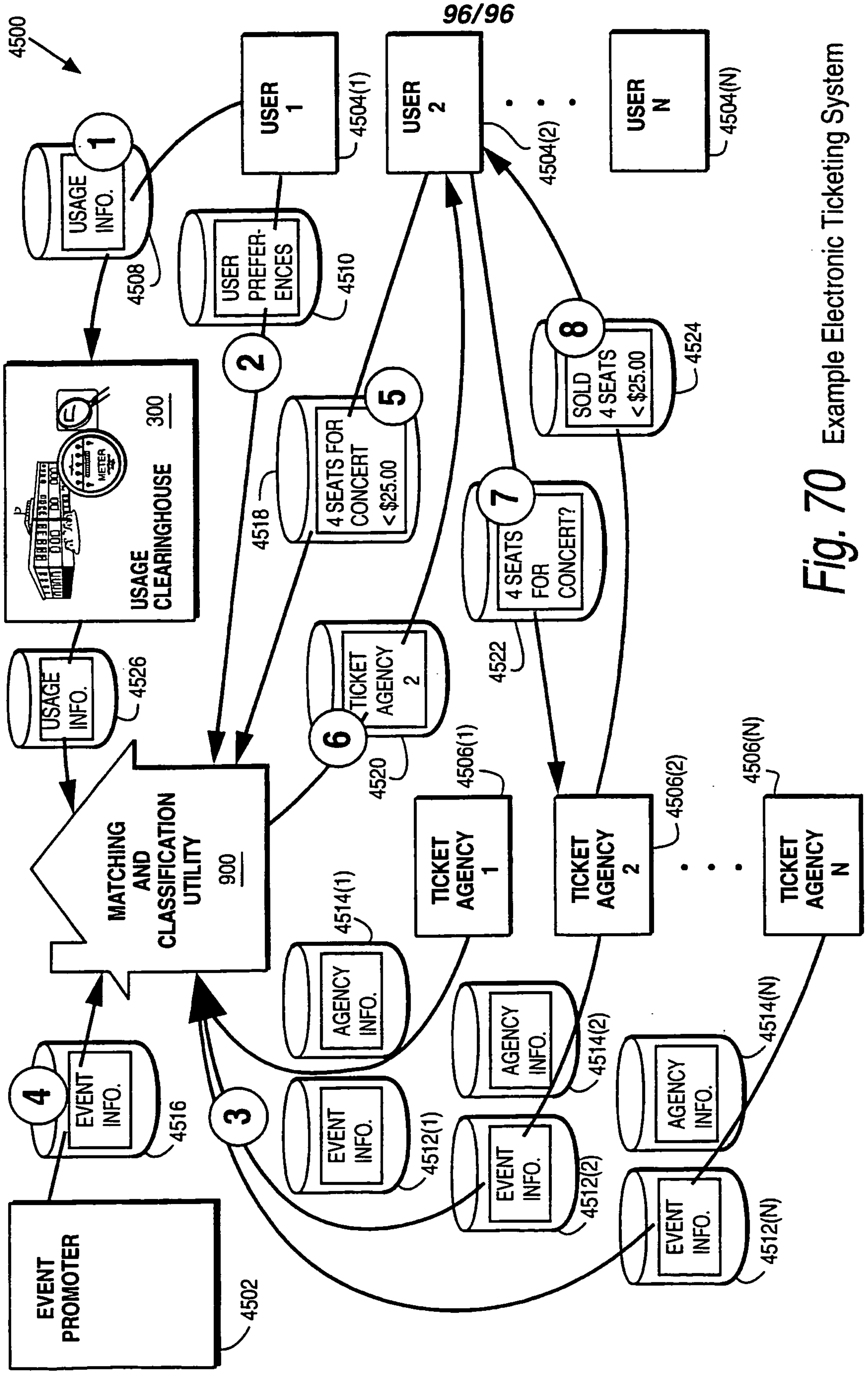

4500
USER 1
4504(1)
USER 2
4504(2)
USER N
4504(N)
USAGE INFO.
4508
USER PREFERENCES
4510
1
2
USAGE CLEARINGHOUSE
300
METER
4 SEATS FOR CONCERT < $25.00
4518
5
SOLD 4 SEATS < $25.00
4524
8
4 SEATS FOR CONCERT?
4522
7
TICKET AGENCY 2
4520
6
USAGE INFO.
4526
MATCHING AND CLASSIFICATION UTILITY
900
TICKET AGENCY 1
4506(1)
TICKET AGENCY 2
4506(2)
TICKET AGENCY N
4506(N)
AGENCY INFO.
4514(1)
AGENCY INFO.
4514(2)
AGENCY INFO.
4514(N)
EVENT INFO.
4512(1)
EVENT INFO.
4512(2)
EVENT INFO.
4512(N)
EVENT INFO.
4516
4
3
EVENT PROMOTER
4502

Fig. 70 Example Electronic Ticketing System

⑲ **Europäisches Patentamt**
European Patent Office
Office européen des brevets

⑪ Publication number: **0 529 261 A2**

⑫ **EUROPEAN PATENT APPLICATION**

㉑ Application number: **92111758.6**

(51) Int. Cl.^{5}: **H04L 9/08**

㉒ Date of filing: **10.07.92**

㉚ Priority: **22.08.91 US 748407**

㊸ Date of publication of application:
03.03.93 Bulletin 93/09

(84) Designated Contracting States:
CH DE FR GB IT LI NL SE

(71) Applicant: **International Business Machines Corporation**
Old Orchard Road
Armonk, N.Y. 10504(US)

(72) Inventor: **Matyas, Stephen M.**
10298 Cedar Ridge Drive
Manassas, VA 22110(US)
Inventor: **Johnson, Donald B.**
11635 Crystal Creek Lane
Manassa, VA 22111(US)
Inventor: **Le, An V.**
10227 Battlefield Drive
Manassas, Va 22110(US)
Inventor: **Martin, William C.**
1835 Hilliard Lane
Concord, NC 28025(US)
Inventor: **Prymak, Rostislaw**
15900 Fairway Drive
Dumfries, VA 22026(US)
Inventor: **Rohland, William S.**
4234 Rotunda Road
Charlotte, NC 28226(US)
Inventor: **Wilkins, John D.**
P.O. Box 8
Somerville, VA 22739(US)

(74) Representative: **Herzog, Friedrich Joachim, Dipl.-Ing.**
IBM Deutschland GmbH, Patentwesen und Urheberrecht, Schönaicher Strasse 220
W-7030 Böblingen (DE)

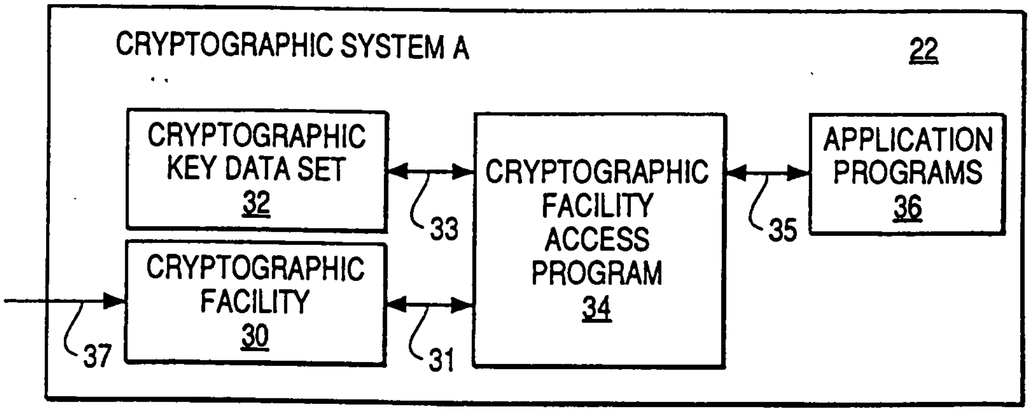
(54) A hybrid public key algorithm/data encryption algorithm key distribution method based on control vectors.

(57) The patent describes a method and apparatus for securely distributing an initial Data Encryption Algorithm (DEA) key-encrypting key by encrypting a key record (consisting of the key-encrypting key and control information associated with that key-encrypting key) using a public key algorithm and a public key belonging to the intended recipient of the key record. The patent further describes a method and apparatus for securely recovering the distributed key-encrypting key by the recipient by decrypting the received key record using the same public key algorithm and private key associated with the public key and re-encrypting the key-encrypting key under a key formed by arithmetically combining the recipient's master key with a control vector contained in the control information of the received key record. Thus the type and usage attributes assigned by the originator of the key-encrypting key in the form of a control vector are cryptographically coupled to the key-encrypting key such that the recipient may only use the received key-encrypting key in a manner defined by the key originator.

The patent further describes a method and apparatus to improve the integrity of the key distribution process by applying a digital signature to the key record and by including identifying information (i.e., an originator identifier) in the control information of the key record. The integrity of the distribution process is enhanced by verifying the digital signature and originator identifier at the recipient node.

Rank Xerox (UK) Business Services
(3.10/3.5x/3.0.1)

FIG. 2


CRYPTOGRAPHIC SYSTEM A
CRYPTOGRAPHIC KEY DATA SET
CRYPTOGRAPHIC FACILITY ACCESS PROGRAM
APPLICATION PROGRAMS
CRYPTOGRAPHIC FACILITY

The invention disclosed broadly relates to data processing systems and methods and more particularly relates to cryptographic systems and methods for use in data processing systems to enhance security.

The following patents are related to this invention and are incorporated herein by reference:

B. Brachtl, et al., "Controlled Use of Cryptographic Keys Via Generating Stations Established Control Values," USP 4,850,017, issued July 18, 1989, assigned to IBM Corporation, and incorporated herein by reference.

S. M. Matyas, et al., "Secure Management of Keys Using Control Vectors," USP 4,941,176, issued July 10, 1990, assigned to IBM Corporation and incorporated herein by reference.

S. M. Matyas, et al., "Data Cryptography Operaions Using Control Vectors," USP 4,918,728, issued April 17, 1990, assigned to IBM Corporation, and incorporated herein by reference.

S. M. Matyas, et al., "Personal Identification Number Processing Using Control Vectors," USP 4,924,514, issued May 8, 1990, assigned to IBM Corporation and incorporated herein by reference.

S. M. Matyas, et al., "Secure Management of Keys Using Extended Control Vectors," USP 4,924,515, issued May 8, 1990, assigned to IBM Corporation and incorporated herein by reference.

S. M. Matyas, et al., "Secure Key Management Using Programmable Control Vector Checking," USP 5,007,089, issued April 9, 1991, assigned to IBM Corporation and incorporated herein by reference.

B. Brachtl, et al., "Data Authentication Using Modification Detection Codes Based on a Public One Way Encryption Function," USP 4,908,861, issued March 13, 1990, assigned to IBM Corporation and incorporated herein by reference.

D. Abraham, et al., "Smart Card Having External Programming Capability and Method of Making Same," serial number 004,501, filed January 19, 1987, assigned to IBM Corporation, and incorporated herein by reference.

S. M. Matyas, et al., "Method and Apparatus for Controlling the Use of a Public Key, Based on the Level of Import Integrity for the Key," serial number 07/602,989, filed October 24, 1990, assigned to the IBM Corporation.

S. M. Matyas, et al., "Secure Key Management Using Programmable Control Vector Checking," USP 5,007,089, issued April 9, 1991, assigned to IBM Corporation and incorporated herein by reference.

The cryptographic architecture described in the cited patents by S. M. Matyas, et al. is based on associating with a cryptographic key, a control vector which provides the authorization for the uses of the key intended by the originator of the key. The cryptographic architecture described in the cited patents by S. M. Matyas, et al. is based on the Data Encryption Algorithm (DEA), whereas the present invention is based on both a secret key algorithm, such as the DEA, and a public key algorithm. Various key management functions, data cryptography functions, and other data processing functions are possible using control vectors, in accordance with the invention. A system administrator can exercise flexibility in the implementation of his security policy by selecting appropriate control vectors in accordance with the invention. A cryptographic facility (CF) in the cryptographic architecture is described in the above cited patents by S. M. Matyas, et al. The CF is an instruction processor for a set of cryptographic instructions, implementing encryption methods and key generation methods. A memory in the crypto facility stores a set of internal cryptographic variables. Each cryptographic instruction is described in terms of a sequence of processing steps required to transform a set of input-parameters to a set of output parameters. A cryptographic facility application program is also described in the referenced patents and patent applications, which defines an invocation method, as a calling sequence, for each cryptographic instruction consisting of an instruction mnemonic and an address with corresponding input and output parameters.

Public key encryption algorithms are described in a paper by W. Diffie and M. E. Hellman entitled "Privacy and Authentication: An Introduction to Cryptography," Proceedings of the IEEE, Vol. 67, No. 3, March 1979, pp. 397-427. Public key systems are based on dispensing with the secret key distribution channel, as long as the channel has a sufficient level of integrity. In a public key crypto system, two keys are used, one for enciphering and one for deciphering. Public key algorithm systems are designed so that it is easy to generate a random pair of inverse keys PU for enciphering and PR for deciphering and it is easy to operate with PU and PR, but is computationally infeasible to compute PR from PU. Each user generates a pair of inverse transforms, PU and PR. He keeps the deciphering transformation PR secret, and makes the enciphering transformation PU public by placing it in a public directory. Anyone can now encrypt messages and send them to the user, but no one else can decipher messages intended for him. It is possible, and often desirable, to encipher with PU and decipher with PR. For this reason, PU is usually referred to as a public key and PR is usually referred to as a private key. A corollary feature of public key crypto systems is the provision of a digital signature which uniquely identifies the sender of a message. If user A wishes to send a signed message M to user B, he operates on it with his private key PR to produce the signed message S. PR was used as A's deciphering key when privacy was desired, but it is now used as his "enciphering" key. When user B receives the message S, he can recover the message M by operating on the ciphertext S with A's public PU. By successfully decrypting A's message, the receiver B has conclusive proof it came from the sender A. Examples of public key cryptography are provided in the following U. S. patents:

USP 4,218,582 to Hellman, et al., "Public Key Cryptographic Apparatus and Method;" USP 4,200,770 to Hellman, et al., "Cryptographic Apparatus and Method;" and USP 4,405,829 to Rivest, et al., "Cryptographic Communications System and Method," which discloses the RSA public-key algorithm.

In general, it is preferable for performance reasons to use symmetric algorithms such as the Data Encryption Algorithm (DEA) bulk data encryption rather to use a public key algorithm for such purposes. However to use DEA both the data originator and intended recipient must first share a common, secret key. This requires the secure distribution of at least one DEA key for each secure "channel" between originator and recipient. The problem can be reduced to distributing one secret DEA key-encrypting key (KEK) between the originating node and receiving node, and thereafter transmitting all other DEA keys encrypted under this common KEK. The usual method of distributing the initial KEK is via trusted couriers.

It is well-known that a hybrid system employing a public key algorithm and the DEA may be effective in solving the initial KEK distribution problem, while still retaining the faster bulk data encryption capabilities of the DEA. In such a hybrid cryptographic system A, a public key PU is transmitted with integrity (see S. M. Matyas, et al., "Method and Apparatus for Controlling the Use of a Public Key, Based on the Level of Import Integrity for the Key", serial number 07/602,989, filed October 24, 1990) to a second hybrid cryptographic system B. A secret DEA KEK, say KK, is generated and encrypted under PU at system B and transmitted to system A. System A uses the corresponding private key PR to decrypt KK. KK may then be used with the DEA algorithm to distribute additional DEA keys for use by systems A and B.

Prior art, however, has not provided a cryptographically secure means to define the type and to control the usage of the generated KEK to insure that the type and uses defined by the originator of the key (system B) are enforced at both the originating node and the recipient node (system A). Without such controls (as described in S. M. Matyas, et al., "Secure Management of Keys Using Control Vectors", USP 4,941,176, issued July 10, 1990), the distributed KEK may be subject to misuse by either party to weaken the security of the system (e.g., by allowing the KEK to be used in a data decrypt operation and thus allowing DEA keys encrypted under the KEK to be decrypted and exposed in the clear).

While the prior art addresses the concept of unidirectional key-encrypting keys, i.e., key-encrypting keys that establish a key distribution channel in one direction only, the method for establishing, with integrity, such a unidirectional channel using a public key algorithm has not been addressed. To accomplish this, a unique Environment Identifier (EID) is stored at each cryptographic device such that a distributed key-encrypting key can be imported only at the designated receiving device, but it does not allow the key-encrypting key to be imported or re-imported at the sending device, as described below.

The originating node B generates the KEK in two forms: one form to be exported to the recipient node A (encrypted under the PU received from A) and a second form to be used at B ultimately to either export or import additional DEA keys (encrypted under some form of the local master key). As was described in the above reference U. S. patent 4,941,176, "Secure Management of Keys Using Control Vectors," it is critical to the security of each cryptographic system that the type and usage attributes of a given KEK on one system be limited to either EXPORTER usage or IMPORTER usage, but never both. Correspondingly, it must not be possible to generate or introduce two copies of the same KEK into the system, one with EXPORTER usage and one with IMPORTER usage. Such a pair of key forms is known as a bi-functional key pair.

Prior art has provided no cryptographically secure means to insure that the generated KEK cannot be re-imported into the originating node to form a bi-functional key pair. Since key PU is public, system A cannot be certain that system B is the originator of the generated KEK.

It is therefore a main object of the invention to provide an improved method for distributing DEA keys using a public key crypto system.

It is another object of the invention to provide an improved method of distributing a DEA key-encrypting key using a public key crypto system.

It is another object of the invention to provide an improved method of distributing a DEA key-encrypting key that does not require the use of couriers.

It is another object of the invention to provide control information associated with a distributed key, which defines the type and usage of the distributed key.

It is another object of the invention to provide a means to cryptographically couple the control information and key using a public-key algorithm.

It is another object of the invention to provide control information that prevents a distributed key from being imported at the originating device.

It is another object of the invention to provide a method of key distribution which is compatible with a key management based on control vectors (in the above referenced patents).

It is another object of the invention to provide a method of key distribution that does not also provide a covert privacy channel.

It is another object of the invention to provide a means for a receiving device to validate that a received distributed key has originated with an expected originating device.

It is another object of the invention to provide a means for a distributed key to be authenticated on the basis of a signature generated on the distributed key by the cryptographic system software.

It is still a further object of the invention to provide a higher integrity means for a distributed key to be authenticated on the basis of a signature generated on the distributed key as an integral part of the cryptographic system hardware export function.

These and other objects, features, and advantages are accomplished by the invention disclosed herein. A method and apparatus are disclosed for generating and distributing a DEA key-encrypting key from a sending device implementing a public-key cryptographic system to a receiving device implementing a public-key cryptographic system. The method and apparatus find application in a cryptographic system implementing both a symmetric encryption algorithm, such as the Data Encryption Standard, and an asymmetric encryption algorithm, such as the RSA public-key algorithm. The method begins by generating a key-encrypting key at a sending device and producing two encrypted copies of the generated key. The generated key is encrypted first under the public key of a designated receiving device and the encrypted key is then electronically transmitted to the receiving device. The generated key is also encrypted under the master key of the sending device and stored in a key storage for later use in a DEA key management scheme for distributing further DEA keys to the designated receiving device. At the receiving device, the encrypted key is decrypted using the private key of the receiving device and the clear key is then re-encrypted under the master key of the receiving device and the encrypted key is stored in a key storage for later use in a DEA key management scheme for receiving further DEA keys from the same sending device. In accordance with the invention, the method of key distribution makes use of a key block containing the distributed key-encrypting key and control information associated with the distributed key, which includes a control vector to limit uses of the key and an environment ID to identify the sender of the key. The method of key distribution also makes use of an optional digital signature generated on the encrypted key block at the originating device and validated at the receiving device.

These and other objects, features, and advantages of the invention will be more fully appreciated with reference to the accompanying figures.

Fig. 1 illustrates a communications network 10 including a plurality of data processors, each of which includes a cryptographic system;

Fig. 2 is a block diagram of a cryptographic system 22;

Fig. 3 is a block diagram of a cryptographic facility 30;

Fig. 4 is a block diagram showing the public and private keys that must first be initialized at two cryptographic systems A and B in order that they may electronically distribute DEA keys using a public key algorithm;

Fig. 5 is a block diagram illustrating DEA key distribution using the GKSP and IDK instructions without digital signatures;

Fig. 6 is a block diagram of a key block;

Fig. 7 is a block diagram of an external key token;

Fig. 8 is a block diagram illustrating DEA key distribution using the GKSP and IDK instructions with digital signatures;

Fig. 9 is a block diagram of the Generate Key Set PKA (GKSP) instruction;

Fig. 10 is a block diagram of the Import DEA Key (IDK) instruction;

Fig. 11 is a block diagram of control vectors for public and private keys used for key distribution (i.e., key management purposes);

Fig. 12 is a block diagram depicting an encrypted channel and a clear channel between two cryptographic systems A and B;

Fig. 13 is a block diagram illustrating the processing of control information at a receiving cryptographic device;

Fig. 14 is a block diagram of a cryptographic facility at a sending location, in accordance with the invention;

Fig. 15 is a block diagram of a cryptographic facility at a receiving location, in accordance with the invention;

Fig. 16 is a block diagram of the crypto-variable retrieval means 40 which is a component of the cryptographic facility shown in Fig. 14.

Environment Description: Fig. 1 illustrates a network block diagram showing a communications network 10 to which is connected a plurality of data processors including data processor 20, data processor 20', and data processor 20". Also included in each data processor is a cryptographic system, as shown in Fig. 1. Data processor 20 includes cryptographic system 22, data processor 20' includes cryptographic system 22' and data processor 20" includes cryptographic system 22". Each data processor supports the processing of one or more applications which require access to cryptographic services such as for the encryption, decryption and authenticating of application data and the generation and installation of cryptographic keys. The cryptographic services are provided by a secure cryptographic facility in each cryptographic system. The network provides the means for the data processors to send and receive encrypted data and keys. Various protocols, that is, formats and procedural rules, govern the exchange of cryptographic quantities between communicating data processors in order to ensure the interoperability between them.

Fig. 2 illustrates the cryptographic system 22. In the cryptographic system 22, the cryptographic facility (CF) 30 has an input 37 from a physical interface. The cryptographic facility access program (CFAP) 34 is coupled to the cryptographic facility 30 by means of the interface 31. The cryptographic key data set (CKDS) 32 is connected to the cryptographic facility access program 34 by means of the interface 33. The application programs (APPL) 36 are connected to the cryptographic facility access program 34 by means of the interface 35.

A typical request for cryptographic service is initiated by APPL 36 via a function call to the CFAP 34 at the interface 35. The service request includes key and data parameters, as well as key identifiers which the CFAP 34 uses to access encrypted keys from the CKDS 32 at the interface 33. The CFAP 34 processes the service request by issuing one or more cryptographic access instructions to the CF 30 at the interface 31. The CF 30 may also have an optional physical interface 37 for direct entry of cryptographic variables into the CF 30. Each cryptographic access instruction invoked at the interface 31 has a set of input parameters processed by the CF 30 to produce a set of output parameters returned by the CF 30 to the CFAP 34. In turn, the CFAP 34 may return output parameters to the APPL 36. The CFAP 34 may also use the output parameters and input parameters to subsequently invoke instructions. If the output parameters contain encrypted keys, then the CFAP 34, in many cases, may store these encrypted keys in the CKDS 32.

Fig. 3 illustrates the cryptographic facility 30. The cryptographic facility 30 is maintained within a secure boundary 140. The cryptographic facility 30 includes the instruction processor 142 which is coupled to the cryptographic algorithms 144 which are embodied as executable code. The cryptographic facility environment memory 146 is coupled to the instruction processor 142. The physical interface can be coupled over line 37 to the CF environment memory 146, as shown in the figure. The instruction processor 142 is coupled to the cryptographic facility access program (CFAP) 34 by means of the interface at 31.

The instruction processor 142 is a functional element which executes cryptographic microinstructions invoked by the CFAP access instruction at the interface 31. For each access instruction, the interface 31 first defines an instruction mnemonic or operation code used to select particular microinstructions for execution. Secondly a set of input parameters is passed from the CFAP 34 to the CF 30. Thirdly, a set of output parameters is returned by the CF 30 to the CFAP 34. The instruction processor 142 executes the selected instruction by performing an instruction specific sequence of cryptographic processing steps embodied as microinstructions stored in cryptographic microinstruction memory 144. The control flow and subsequent output of the cryptographic processing steps depend on the values of the input parameters and the contents of the CF environment memory 146. The CF environment memory 146 consists of a set of cryptographic variables, for example keys, flags, counters, CF configuration data, etc., which are collectively stored within the CF 30. The CF environment variables in memory 146 are initialized via the interface 31, that is by execution of certain CF microinstructions which read input parameters and load them into the CF environment memory 146. Alternately, initialization can be done via an optional physical interface which permits cryptographic variables to be loaded directly into the CF environment memory 146, for example via an attached key entry device.

The physical embodiment of the cryptographic facility secure boundary 140, incorporates the following physical security features. The physical embodiment resists probing by an insider adversary who has limited access to the cryptographic facility 30. The term "limited" is measured in minutes or hours as opposed to days or weeks. The adversary is constrained to a probing attack at the customer's site using limited electronic devices as opposed to a laboratory attack launched at a site under the control of the adversary using sophisticated electronic and mechanical equipment. The physical embodiment also detects attempts at physical probing or intruding, through the use of a variety of electro-mechanical sensing devices. Also, the physical embodiment of the cryptographic facility 30 provides for the zeroization of all internally stored secret cryptographic variables. Such zeroization is done automatically whenever an attempted probing or intrusion has been detected. The physical embodiment also provides a manual facility for a zeroization of internally stored secret cryptographic variables. Reference to the Abraham, et al. patent application cited above, will give an example of how such physical security features can be implemented.

Initialization of Public-Key Cryptographic System: Fig. 4 illustrates two cryptographic systems, A and B, that wish to communicate cryptographically using public key cryptography. Cryptographic system A generates a public and private key pair (PUa, PRa), where PUa is the public key of A and PRa is the private key of A. In like manner, cryptographic system B generates a public and private key pair (PUb, PRb), where PUb is the public key of B and PRb is the private key of B.

Referring to Fig. 4, the cryptographic facility 30 of cryptographic system A contains a master key KMa and the cryptographic facility 30' of cryptographic system B contains a master key KMb. KMa and KMb are ordinarily different, being equal only by mere chance. At cryptographic system A, the public key PUa is encrypted with the Data Encryption algorithm (DEA) using variant key KMa.C1 to form the encrypted value eKMa.C1-(PUa), where KMa.C1 is formed as the Exclusive OR product of master key KMa and control vector C1. Likewise, at cryptographic system A, the private key PRa is encrypted with the DEA using variant key KMa.C2 to form the encrypted value eKMa.C2(PRa), where KMa.C2 is formed as the Exclusive OR product of master key KMa and control vector C2. The symbol "." denotes the Exclusive OR operation. The encrypted values eKMa.C1(PUa) and eKMa.C2(PRa) are stored in cryptographic key data set 32.

The control vector specifies whether the key is a public or private key and contains other key usage control information specifying how the key may be used. For example, when the encrypted key eKMa.C2(PRa) is decrypted for use within the cryptographic facility 30, control vector C2 indicates to the cryptographic facility how and in what way the key PRa may be used. Control vector C1 similarly controls the use of public key PUa. The use of the control vector to control key usage is described in U.S. Patents 4,850,017, 4,941,176, 4,918,176, 4,924,514, 4924,515, and 5,007,089 cited in the background art and in co-pending patent application serial number 07/602,989 also cited in the background art. Fig. 11 illustrates control vectors that define public and private keys, where the public and private keys are key management keys used by the cryptographic system to distribute DEA keys. The fields in each control vector consist of a CV TYPE, which specifies whether the control vector is a public or a private key and additionally whether the key pair is a key management key pair for use in distributing DEA keys or whether the key pair is some other kind of key pair. Other types of key pairs are possible, such as user keys which can be used for generation and verification of digital signatures but not for key distribution. Each control vector has a PR USAGE and PU USAGE field. For the public key control vector, the PU USAGE field controls the usage of the public key in cryptographic instructions whereas the PR USAGE field is only informational. For the private key control vector, the PR USAGE field controls the usage of the private key in cryptographic instructions whereas the PU USAGE field is only informational. The ALGORITHM field indicates the public key algorithm to which this key pair pertains. The HIST field records history information, e.g., the options used to import a public key (see co-pending patent application serial number 07/602,989 as cited in the background art, which describes the use of history information fields in the public key control vector). The reader will appreciate that the control vector may contain a variety of different control vector fields for the purpose of controlling the operation and use of the key within the cryptographic network and cryptographic systems within the network.

In an alternate embodiment, the public key PUa may be stored in an unencrypted form, since there is no intent to keep the value of this key secret. Encrypting PUa is done for sake of uniformity, so that all keys in the cryptographic key data set 32 are stored and recovered using one common method. Those skilled in the art will also recognize that the length of PUa and PRa will likely be different than the block size of the DEA, which is 64 bits, and hence PUa and PRa may need to be encrypted in separate 64-bit pieces. The particular method for encrypting PUa and PRa is unimportant to the invention. However, one way that this encryption can be carried out is to use the Cipher Block Chaining (CBC) mode of DEA encryption described in DES modes of operation, Federal Information Processing Standards Publication 81, National Bureau of Standards, US Department of Commerce, December 1980. In cases where KMa is a 128-bit key, the CBC mode of DEA encryption can be adapted to encrypt PUa under KMa. PUa is first encrypted with the leftmost 64 bits of KMa, then decrypted with the rightmost 64 bits of KMa, and then encrypted again with the leftmost 64 bits of KMa.

In like manner, at cryptographic system B, the public and private keys, PUb and PRb, are encrypted with master key KMb and control vectors C3 and C4, per the same method described for cryptographic system A. The encrypted values eKMb.C3(PUb) and eKMb.C4(PRb) are stored in cryptographic key data set 32'. Control vectors C3 and C4 control the usage of PUb and PRb, respectively.

Although encryption of the public and private keys has been described in terms of a DEA-based master key, those skilled in the art will appreciate that the DEA could be replaced by a public key algorithm and the master keys could be replaced by a PKA-based key pair used for this purpose. Moreover, the encryption of the public and private keys has been described in terms of encryption of the keys only. In some implementations it may be more practical to imbed these keys within key records that contain other key-related information besides the keys themselves.

In order for cryptographic systems A and B to carry out cryptographic operations using their respective implemented public key algorithms, they must share their public keys with each other. Thus, at cryptographic system A a function exists that permits the encrypted value eKMa.C1(PUa) to be accessed from cryptographic key data set 32 and decrypted so that the clear value of PUa may be exported to cryptographic system B at 300. At cryptographic system B a function exists that permits the clear value of PUa to be imported and encrypted under the variant key KMb.C1. The so-imported encrypted value eKMb.C1(PUa) is then stored in cryptographic key data set 32'. In like manner, functions exist at B and A that permit public key PUb to be decrypted at B, sent to A at 301, and re-encrypted at A for storage in A's cryptographic key data set.

Co-pending patent application by S. M. Matyas et al., serial number 07/602,989, "Method and Apparatus for Controlling the use of a Public Key, Based on the Level of Import Integrity for the Key," describes a method for generating public and private keys and for distributing public keys in order to initialize a public-key cryptographic system, and is incorporated by reference herein.

Key Distribution: Fig. 5 illustrates the process by which cryptographic system A may distribute a key to cryptographic system B using a public key algorithm (PKA). That is, it illustrates the process of key distribution using a PKA. In a hybrid key distribution scheme, the distributed key is a DEA key, e.g., an initial key-encrypting key to be used later with a DEA-based key distribution scheme to distribute all subsequent DEA keys. However, any key can be distributed using the so-described PKA-based key distribution scheme, including both DEA keys and PKA keys. The distributed DEA and PKA keys can be of any type or designated use. However, for purposes of illustration, Fig. 5 shall assume that the distributed key is a DEA key.

Referring to Fig. 5, the steps involved in distribution of a key from cryptographic system A to cryptographic system B are these. At cryptographic system A, a Generate Key Set PKA (GKSP) instruction is executed within the CF 30. Control information at 303 is provided to the GKSP instruction as input. In response, the GKSP instruction generates a key K and produces two encrypted copies of K, which are returned by the GKSP instruction at 305 and 306. The first encrypted copy of K is produced by encrypting K with the DEA using variant key KMa.C5 formed as the Exclusive OR product of master key KMa and control vector C5. C5 may be input to the GKSP instruction as part of the control information, at 303, or it may be produced within the CF 30 as part of the GKSP instruction, or it may be produced as a combination of both methods. The second encrypted copy of K is produced as follows. A key block (designated keyblk) is first formed. The key block includes the clear value of K, control information, and possibly other information unimportant to the present discussion, as illustrated in Fig. 6. The format of the keyblk is unimportant to the present discussion, and those skilled in the art will recognize that many possible arrangements of the keyblk information are possible. In all cases, the keyblk contains the necessary information to accomplish the task of key distribution. The length of the keyblk is assumed to be equal to the block size of the public key algorithm. For example, if the public key algorithm is the RSA algorithm, then the block size is just the modulus length. Also, it is assumed that the numeric value of the keyblk, say its binary value, is adjusted as necessary to permit it to be encrypted as a single block by the public key algorithm. For example, if the public key algorithm is the RSA algorithm, then the keyblk is adjusted so that its binary value is less than the binary value of the modulus. This can be done by forcing the high order (most significant) bit in the keyblk to zero. Once the keyblk has been formatted, it is encrypted with the public key PUb of cryptographic system B to form the encrypted value ePUb(keyblk), which is returned at 306. To permit this to be accomplished, the encrypted value eKMa.C3(PUb) and control vector C3 are supplied to the GKSP instruction at 304 as inputs and eKMa.C3(PUb) is decrypted under variant key KMa.C3. KMa.C3 is formed as the Exclusive OR product of master key KMa stored within the CF 30 and control vector C3.

The first encrypted output eKMa.C5(K) at 305 is stored in the cryptographic key data set 22 of cryptographic system A. Control vector C5 is also stored in the cryptographic key data set 22 together with the encrypted key eKMa.C5(K). In some implementations it may be convenient to store eKMa.C5(K) and C5 in an internal key token together with other key-related information. The internal key token is not relevant to the present discussion, and is therefore not shown in Fig. 5. If C5 is generated within the CF 30, it may also be provided as an output at 305 so that it may be store in CKDS 22.

The second encrypted output ePUb(keyblk) at 306 is formatted within an external key token 308. The external key token contains the encrypted key or encrypted key block ePUb(keyblk), control information, and other information unimportant to the present discussion, as shown in Fig. 7. The control information supplied as input to the GKSP instruction at 303 is also stored in external key token 308 at 307. However, the control information at 307 may include additional information available to the cryptographic facility access program (CFAP) which is not specified as an input to the GKSP instruction at 303. In other words, the source of the control information in the external key token 308 may be much broader than the control information supplied as input the the GKSP instruction at 303. One example, is the Environment Identifier (EID) value stored both in the CF 30 and in the CFAP. The EID value is an identifier that uniquely identifies each cryptographic facility or cryptographic system within a network. The EID value is loaded into the CF 30 during an initialization sequence prior to performing routine cryptographic operations within the cryptographic system. Another example of initialization is the loading of the master key KMa. The EID value need not be supplied to the CF since it is already stored in the CF. But the EID value may be stored within the external key token, in which case it is supplied as an input at 307. In like manner, the control information in the keyblk may include a control vector C6 specifying the usage of K at cryptographic system B. In that case, C6 may be supplied as part of the control information at 303, in which case it is also supplied as part of the control information at 307. If however C6 is generated within CF 30, then C6 is not supplied as part of the control information at 303, but is supplied as part of the control information at 307. Those skilled in the art will recognize that various alternatives exist for the specification or derivation of the necessary control information and that different combinations of inputs to the GKSP instruction and to the external key token are therefore possible.

The formatted external key token 308 is transmitted to cryptographic system B where it is processed. The CFAP at B first checks the control information in the external key token for consistency. For example, if the control information contains a control vector C6, then C6 is checked to ensure that it represents a key type and key usage approved by cryptographic system B. Likewise, if the control information contains an EID value, then the EID value is checked to ensure that the external key token and the key to be imported originated from cryptographic system A, i.e., it originated from the expected or anticipated cryptographic system that B 'thinks' it is in communication with and which it desires to establish a keying relationship. Once this has been accomplished, the received key is imported as follows. The encrypted keyblk, ePUb(keyblk) and part or all of the control information in the external key token are supplied as inputs to an Import DEA Key (IDK) instruction at 309, which is executed within CF 30' at cryptographic system B. In response, the IDK instruction decrypts ePUb(keyblk) under the private key PRb belonging to cryptographic system B. To permit this to be accomplished, the encrypted value eKMb.C4(PRb) and control vector C4 are supplied to the IDK instruction at 310 as inputs and eKMb.C4(PRb) is decrypted under variant key KMb.C4. KMb.C4 is formed as the Exclusive OR product of master key KMb stored within the CF 30' and control vector C4. Once ePUb(keyblk) has been decrypted and the clear value of keyblk has been recovered, the keyblk is processed as follows. The control information contained in the keyblk is checked for consistency against the control information, or reference control information, supplied as input at 309. If the consistency checking is satisfactory (okay), then the clear value of K is extracted from keyblk and it is encrypted with the variant key KMb.C6 to produce the encrypted key value eKMb.C6(K). KMb.C6 is formed as the Exclusive OR product of master key KMb stored within CF 30' and control vector C6. Control vector C6 may be obtained in different ways. C6 may be contained in the control information in keyblk, in which case it is extracted from keyblk. In other cases, C6 may be produced within CF 30'. For example, if there is only one key type and key usage permitted, then C6 can be a constant stored within the IDK instruction. The so-produced encrypted key value eKMb.C6(K) is provided as an output of the IDK instruction at 311, and is stored together with its control vector C6 within CKDS 22'. The value of C6 stored in CKDS 22' is obtained either from the control information input to the IDK instruction at 309 or, if C6 is not in the control information input to the IDK instruction at 309, then it is produced by the CFAP in the same way that it is produced by the IDK instruction and stored in CKDS 22'. Alternatively, C6 could be returned as an output of the IDK instruction. Those skilled in the art will realize that several alternatives exist for obtaining C6 depending on how and where it is produced within the cryptographic system and whether it is or is not included as part of the control information in the external key token.

In the preferred embodiment, the control information at 303 supplied to the GKSP instruction includes a specification of control vectors C5 and C6. This allows the GKSP instruction the freedom and flexibility to generate two encrypted copies of key K that have different key types and usages, as specified by C5 and C6. In that case, the GKSP instruction must incorporate some control vector checking to determine that C5 and C6 constitutes a valid pair. The various options for control vector design and checking pursued here are based on the control vector designs included in prior art, cited in the background art, and already discussed. Likewise, in the preferred embodiment, control vector C6 is included in the control information in the key block (keyblk) and also in the control information in the external key token. This permits the receiving cryptographic system to import keys of different types while still permitting the receiving system to verify that the imported key is one that it wants or expects. This is accomplished by the CFAP first checking the control vector in the external key token to make sure that it prescribes a key type and key usage that it expects or will allow to be imported. C6 is then supplied as an input in the control information at 309 to the CF 30'. At the time the IDK instruction recovers the clear value of keyblk, the value of C6 in the control information in keyblk is checked against the value of C6, or the reference value of C6, supplied as input. This permits the CF to verify that the value of C6 used to import the key K is the same control vector C6 in the external key token. Otherwise, if this check was ignored it would be possible for an adversary to substitute C6' for C6 in the external key token, causing a key to be imported that the CFAP may not permit.

In the preferred embodiment, each cryptographic facility stores a unique EID value, e.g., a 128-bit value set within the CF during an initialization sequence before routine operations are permitted. At the time a keyblk is prepared within the CF by a GKSP instruction, the EID value is obtained from the CF and included within the control information in the keyblk. In like manner, a duplicate copy of the EID value is stored outside the CF with integrity such that it is available to the CFAP. This EID value is obtained by the CFAP and is included within the control information in the external key token. Thus, the CFAP at the receiving cryptographic system can check the EID value in the control information of the received external key token to ensure that the external key token originated from the cryptographic system that is expected or anticipated. That is, B knows that the external key token came from A, which is what is expected. The EID value is also supplied as part of the control information at 309. Thus, when the IDK instruction obtains the clear keyblk, the EID value in the control information in the clear keyblk can be checked against the EID value, or reference EID value, supplied as an input. In this way, the CFAP is sure that the IDK instruction will import K only if the two EID values are equal. This prevents an adversary from changing the EID value in the external key token to a different value that might also be accepted by the receiving device. This might lead to a situation where B imports a key from A, thinking that it came from C.

The EID also serves another purpose, as now described. At the time the clear value of keyblk is obtained by the IDK instruction, a check is performed to ensure that the value of EID in the control information in keyblk is not equal to the value of EID stored in the CF at the receiving device. Thus, the encrypted value of ePUx(keyblk) produced at cryptographic system A, where PUx may be the public key of any cryptographic system in the network, including A itself, cannot be imported by A. This prevents an adversary at A, who specifies his own public key PUa to the GKSP instruction, from importing ePUa(keyblk) at A and thereby obtaining two encrypted copies eKMa.C5-(K) and eKMa.C6(K) of the same key with potentially different key types and key usage attributes. In some cases, bi-functional key pairs are undesirable and the key management design will specifically disallow such key pairs to be created using the key generation facilities provided by the key management services.

Key Distribution with Digital Signatures: The key distribution scheme described in Fig. 5 is not by itself the preferred embodiment of the invention. This is so because, as it stands, the scheme can be attacked by an adversary who knows the public value of B's key, PUb. In public key cryptographic systems, one naturally makes the assumption that PUb is known by anyone, even an adversary. The adversary can forge values of keyblk containing DEA keys of his choosing and freely encrypt these key blocks under PUb. Thus, at B, there is no way to know that an imported key originated with A or with an adversary posing as A. The importing function will import the forged values of keyblk, which results in known values of K being encrypted under the master key, of the form eKMb.C6(K), and stored in CKDS 22'. In that case, data or keys encrypted under K are easily deciphered by the adversary who knows K.

The preferred embodiment of the invention therefore includes a means by which the receiver, say cryptographic system B, can ensure that a received encrypted keyblk of the form ePUb-(keyblk) did in fact originate with the intended sender, say cryptographic system A. To accomplish this, the GKSP instruction at cryptographic system A produces a digital signature (designated DSIGa) on ePUb(keyblk) using its private key PRa. The so-produced digital signature is transmitted together with the external key token to cryptographic system B where the key is imported using an IDK instruction. In this case, the IDK instruction first verifies the digital signature DSIGa using the previously imported copy of PUa received from cryptographic system A. Only after DSIGa has been successfully verified will the IDK instruction continues as already described in Fig. 5 and import the key K.

Fig. 8 illustrates the scheme for DEA key distribution with digital signatures, which is the same as the scheme shown in Fig. 5 except as follows. Once the encrypted key value ePUb(keyblk) has been produced, the GKSP instruction additionally produces the digital signature DSIGa from ePUb(keyblk) and the private key PRa belonging to cryptographic system A. A common method for producing such a signature is to first calculate a hash value on ePUb(keyblk) using a one way cryptographic function, such as described in U. S. patent 4,908,861 by Brachtl et al., cited in the background art, which uses either two DEA encryptions or four DEA encryptions per each 64 bits of input text to be hashed, and then decrypt (or transform) the hash value using the private key PRa to produce a DSIGa of the form dPRa(hash value). The clear value of PRa is obtained by decrypting the encrypted value of eKMa.C2(PRa) supplied as an input to the GKSP instruction at 313 using the DEA and the variant key KMa.C2. KMa.C2 is formed as the Exclusive OR product of master key KMa stored in CF 30 and control vector C2 supplied as input to the GKSP instruction at 313. For example, if the public key algorithm is the RSA algorithm, a the digital signature may be calculated using the method as described in ISO Draft International Standard 9796 entitled "Information Technology -- Security Techniques -- Digital Signature Scheme Giving Message Recovery." The so-produced DSIGa 315 is returned as an output at 314. Both the external key token 308 and the DSIGa 315 are transmitted to cryptographic system B. At cryptographic system B, the IDK instruction is used to import the key K in similar fashion as described in Fig. 5 except that the IDK instruction first validates DSIGa using the public key PUa previously imported, encrypted, and stored in CKDS 22'. A DSIGa of the form dPRa(hash value) is validated by encrypting dPRa(hash value) with PUa, calculating a hash value on ePUb(keyblk) using the same one way cryptographic function, called the hash value of reference, and comparing the hash value of reference and the recovered clear hash value for equality. Only if this comparison check is successful does the IDK instruction continue and import the key K. The clear value of PUa is obtained by decrypting the encrypted value of eKMb.C1(PUa) supplied as an input to the IDK instruction at 316 using the DEA and the variant key KMb.C1. KMb.C1 is formed as the Exclusive OR product of master key KMb stored in CF 30' and control vector C1 supplied as input to the IDK instruction at 316. Thus, the GKSP instruction at cryptographic system A produces DSIGa and the IDK instruction at cryptographic system B verifies DSIGa. In an alternate embodiment, DSIGa can be calculated by the CF 30 using a separate instruction for generating digital signatures. In that case, after the GKSP instruction has been executed, the CFAP invokes the generate digital signature instruction causing DSIGa to be generated. In like manner, DSIGa can be verified by the CF 30' using a separate instruction for verifying digital signatures. In that case, before the IDK instruction is invoked, the CFAP invokes the verify digital signature instruction to ensure that DSIGa is valid.

Generate Key Set PKA (GKSP) Instruction: Fig. 9 illustrates the Generate Key Set PKA (GKSP) instruction. The GKSP instruction of Fig. 9 is identical to the GKSP instruction contained within the CF 30 of Fig. 8. The GKSP instruction generates a two encrypted copies of a generated DEA key K. The first copy is of the form eKM.C5(K) and is stored in the cryptographic key data set of the generating cryptographic device, say A. The second copy is of the form ePU(keyblk) and is transmitted to a designated receiving cryptographic device, say B, where the public key PU belonging to the receiving cryptographic device B. Also, the GKSP instruction produces a digital signature DSIG on ePU(keyblk) using the private key PR of the generating cryptographic device A. DSIG is also transmitted to cryptographic device B to serve as proof that ePU(keyblk) was produced at cryptographic device A, i.e., produce at a valid network cryptographic device.

Referring to Fig. 9, GKSP instruction 500 consists of control information retrieval means 504, PU recovery means 506, PR recovery means 507, key generation means 508, eKM.C5(K) production means 509, ePU(keyblk) production means 510, DSIG production means 511, and hash algorithms 512. GKSP instruction 500 is located in instruction processor 142 within cryptographic facility 30, as shown in Fig. 3. The inputs to the GKSP instruction are supplied to the GKSP instruction by CFAP 34, i.e., by the CFAP 34 to the CF 30 across the CFAP-to-CF interface. In similar manner, the outputs from the GKSP instruction are supplied to the CFAP 34, i.e., by the CF 30 to the CFAP 34 across the CFAP-to-CF interface.

The inputs to GKSP instruction 500 are (1) at 501, control information such as control vectors C5 and C6 that specify the key usage attributes of the two encrypted copies of the generated DEA key K, (2) at 502, control vector C3 and encrypted public key eKM.C3(PU), where C3 specifies the key usage attributes of public key PU belonging to the receiving cryptographic device, and (3) at 503, control vector C2 and encrypted private key eKM.C2(PR), where C2 specifies the key usage attributes of private key PR belonging to the sending or generating cryptographic device. The outputs from GKSP instruction 500 are (1) at 521, the encrypted key, eKM.C5(K), where K is encrypted under variant key KM.C5 formed as the Exclusive OR product of master key KM and control vector C5, (2) at 522, the encrypted key block, ePU-(keyblk), where keyblk is encrypted under public key PU belonging to the intended receiving cryptographic device and where keyblk is a key block containing the generated DEA key K, control information, and possibly other information as depicted in Fig. 6, and (3), at 523, a digital signature DSIG generated on ePU(keyblk) using the private key PR belonging to the sending or generating cryptographic device.

Control information retrieval means 504 accepts and parses control information supplied as input to the GKSP instruction at 501. Also, control information retrieval means 504 accesses control information stored within the secure boundary of the cryptographic facility, e.g., the Environment Identifier (EID) at 505. Control information retrieval means 504 may also perform consistency checking on the assembled control information. For example, control vectors C5 and C6 may be checked and cross checked for consistency, i.e., to ensure they are a valid control vector pair. GKSP instruction 500 is aborted if C5 and C6 are incorrect or do not specify the correct key usage required by the GKSP instruction. In an alternate embodiment, it may be possible for control information retrieval means 504 to generate or produce control vector C6 from control vector C5, or vice versa, in which case only one control vector is specified in the control information supplied at 501. In that case, cross checking of C5 and C6 is unnecessary. Control vector checking of C5 or C6 can be performed in control information retrieval means 504 or in eKM.C5(K) production means 509 if the control vector is C5 or in ePU(keyblk) production means 510 if the control vector is C6. The reader will appreciate that control vector checking may be accomplished in variety of ways within the different components parts of the GKSP instruction, and that these variations do not significantly depart of the general framework of the invention. In any event, control information retrieval means 504 makes the control information available to other component parts of the GKSP instruction. C5 is passed to eKM.C5(K) production means 509 and EID and C6 are passed to ePU(keyblk) production means 510. Optionally, control information may also be passed to DSIG production means 511 such as the identifier or name of a hashing algorithm to be used in the preparation of the digital signature. The GKSP instruction may support only one hashing algorithm in which case the identifier or name of a hashing algorithm need not be passed to DSIG production means. Those skilled in the art will recognize that many possible variations exist for inputting and accessing control information, for parsing, checking and making the control information available to different component parts of the GKSP instruction.

PU recovery means 506 decrypts input eKM.C3(PU) under variant key KM.C3 formed as the Exclusive OR product of master key KM stored in clear form within the cryptographic facility and directly accessible to GKSP instruction 500 and control vector C3 specified as an input to GKSP instruction 500. Prior to decrypting eKM.C3(PU), PU recovery means 506 performs control vector checking on C3. GKSP instruction 500 is aborted if C3 is incorrect or does not specify the correct key usage required by the GKSP instruction. Public key PU is stored in encrypted form so that PU Recovery means 506 will be, for all practical purposes, identical to PR Recovery means 507. Encryption of PU is also preferred since it permits control vector C3 to be cryptographically coupled with public key PU. Even though PU is public, and there is no need to protect the secrecy of PU, encryption of PU thus ensures that PU can be used only if C3 is correctly specified as an input to the GKSP instruction. This ensures that PU is used by the GKSP instruction only if it has been so designated for use. In an alternate embodiment, PU could be stored outside the cryptographic facility in clear form and PU Recovery means 506 could be omitted from GKSP instruction 500. In this case, the embodiment may choose to fix the usage attributes of PU so that there is no chance for an adversary to specify a control vector C3 that is incorrect, i.e., C3 is a fixed constant value. In any event, the recovered clear value of PU is supplied as an input to ePU(keyblk) production means 510.

PR recovery means 507 decrypts input eKM.C2(PR) under variant key KM.C2 formed as the Exclusive OR product of master key KM stored in clear form within the cryptographic facility and directly accessible to GKSP instruction 500. Prior to decrypting eKM.C2(PR), PR recovery means 507 performs control vector checking on C2. GKSP instruction 500 is aborted if C2 is incorrect or does not specify the correct key usage required by the GKSP instruction. The recovered clear value of PR is supplied as an input to DSIG production means 511.

Key generator means 508 is a pseudo random number generator for generating DEA keys. Alternatively, key generator means 508 could be a true random number generator. For sake of simplicity, key generator means 508 generates 64-bit random numbers which are adjusted for odd parity. That is, the eight bit of each byte in the generated random number is adjusted so that the value in each byte is odd. DEA keys may contain either 64 or 128 bits depending on their intended usage. Data-encrypting-keys used for encrypting data are 64-bit keys. Key-encrypting-keys used for encrypting keys are generally 128-bit keys, but may in some cases be 64-bit keys. To produce a 128-bit key, key generator means 508 is invoked twice. The so-generated DEA key is supplied as an input to both eKM.C5(K) production means 509 and ePU(keyblk) production means 510.

eKM.C5(K) production means 509 Exclusive ORs input KM and C5 to produce variant key KM.C5 and then encrypts input K with KM.C5 to form the encrypted output eKM.C5(K), which is returned to the CFAP at 521. If control vector C5 is not consistency checked in control information retrieval means, it may alternatively be checked here.

ePU(keyblk) production means 510 first prepares a key block, designated keyblk, from the inputs K, EID, and C6, and then encrypts keyblk with public key PU to form the encrypted output ePU(keyblk), which is returned to the CFAP at 522. If control vector C6 is not consistency checked in control information retrieval means, it may alternatively be checked here. The value ePU(keyblk) is also supplied as an input to DSIG production means 511 to allow the digital signature DSIG to be produced. The format of keyblk is shown in Fig. 6 and has been discussed previously. The procedure of preparing keyblk accomplishes two main goals. It ensures that all necessary information such as the key, control information, key-related information, keyblk parsing information, etc. is included within keyblk. Also, it ensure that keyblk is constructed in a way that keyblk can be encrypted with PU using the public key algorithm. For example, it may be necessary to pad keyblk so that its length and binary value are such that keyblk is encrypted properly and in conformance with restrictions imposed or that may be imposed by the public key algorithm.

DSIG production means 511 produces a digital signature on ePU(keyblk) using private key PR. To accomplish this, a hash value is first calculated on ePU(keyblk) using hash algorithm 512. Hash algorithm 512 may in fact be a set of hash algorithms. In that case, the hash algorithm is selected on the basis of a hash algorithm identifier or other appropriate encoded value passed by the control information retrieval means 504 to the DSIG production means 511. The so-produced hash value is then formatted in a suitable signature block and decrypted with private key PR to produce DSIG, which is returned to the CFAP at 523. The signature block can in the simplest case consist of the hash value and padding data, so as to construct a signature block whose length and value are in conformance with restrictions imposed or that may be imposed by the public key algorithm, as already discussed above. The DSIG production means 511 may also implement a digital signature method based on a national or international standard, such as International Standards Organization draft international standard (ISO DIS) 9796.

Import DEA Key (IDK) Instruction: Fig. 10 illustrates the Import DEA Key (IDK) instruction. The IDK instruction of Fig. 10 is identical to the IDK instruction contained within the CF 30 of Fig. 9 The IDK instruction permits a cryptographic device, say B, to import an encrypted DEA key of the form ePU(keyblk) that has been received from a sending cryptographic device, say A. The received digital signature DSIG is used by the IDK instruction to verify that ePU(keyblk) originated with cryptographic device A, i.e., at a valid network cryptographic device.

Referring to Fig. 10, IDK instruction 600 consists of PU recovery means 606, PR recovery means 607, control information retrieval means 608, hash algorithms 610, DSIG verification means 611, keyblk recovery means 612, eKM.C6(K) production means 613, and control information consistency checking means 614. IDK instruction 600 is located in instruction processor 142 within cryptographic facility 30, as shown in Fig. 3. The inputs to the IDK instruction are supplied to the IDK instruction by CFAP 34, i.e., by the CFAP 34 to the CF 30 across the CFAP-to-CF interface. In similar manner, the outputs from the IDK instruction are supplied to the CFAP 34, i.e., by the CF 30 to the CFAP 34 across the CFAP-to-CF interface.

The inputs to the IDK instruction 600 are (1) at 601, control vector C1 and encrypted public key eKM.C1(PU), where C1 specifies the key usage attributes of public key PU belonging to the sending cryptographic device, (2) at 602, digital signature DSIG, (3) at 603, encrypted key block ePU-(keyblk), where keyblk is encrypted under public key PU belonging to the the receiving cryptographic device and where keyblk is a key block containing the to-be-imported DEA key K, control information, and possibly other information as depicted in Fig. 6, (4) at 604, control vector C4 and encrypted private key eKM.C4(PR), where C4 specifies the key usage attributes of private key PR belonging to the receiving cryptographic device, and (5) at 605, control information, such as a reference control vector C6 and a reference EID value of the sending cryptographic device. The output of the IDK instruction 600 is the encrypted key eKM.C6(K), where K is the to-be-imported DEA key encrypted under variant key KM.C6 formed as the Exclusive OR product of master key KM and control vector C6.

PU recovery means 606 decrypts input eKM.C1(PU) under variant key KM.C1 formed as the Exclusive OR product of master key KM stored in clear form within the cryptographic facility and directly accessible to IDK instruction 600 and control vector C1 specified as an input to IDK instruction 600. Prior to decrypting eKM.C1(PU), PU recovery means 606 performs control vector checking on C1. IDK instruction 600 is aborted if C1 is incorrect or does not specify the correct key usage required by the IKK instruction. Public key PU is stored in encrypted form so that PU recovery means 606 well be, for all practical purposes, identical to PR recovery means 607 Encryption of PU is also preferred since it permits control vector C1 to be cryptographically coupled with public key PU, as argued previously under the description of the GKSP instruction. In an alternate embodiment, PU could be stored outside the cryptographic facility in clear form and PU recovery means 606 could be omitted from IDK instruction 600. In this case, the embodiment may choose to fix the usage attributes of PU so that there is no chance for an adversary to specify a control vector C1 that is incorrect, i.e., C1 is a fixed constant value. In any event, the recovered clear value of PU is supplied as an input to DISG verification means 611.

PR recovery means 607 decrypts input eKM.C4(PR) under variant key KM.C4 formed as the Exclusive OR product of master key KM stored in clear form within the cryptographic facility and directly accessible to IDK instruction 600. Prior to decrypting eKM.C4(PR), PR recovery means 607 performs control vector checking on C4. IDK instruction 600 is aborted if C4 is incorrect or does not specify the correct key usage required by the IDK instruction. The recovered clear value of PR is supplied as an input to keyblk recovery means 612.

Control information retrieval means 608 accepts and parses control information supplied as input to the IDK instruction at 605. Also, control information retrieval means 608 accesses control information stored within the secure boundary of the cryptographic facility, e.g., the Environment Identifier (EID) at 609. Control information retrieval means 608 supplies control information to control information consistency checking means 614 and possibly to other component parts of the IDK instruction, such as a hash algorithm identifier supplied to DSIG verification means 611 (not shown in Fig. 10).

DSIG verification means 611 uses public key PU belonging to the sending cryptographic device to verify the digital signature DSIG generated on ePU(keyblk) at the sending cryptographic device. To accomplish this, a hash value is first calculated on ePU(keyblk) using hash algorithm 512. Hash algorithm 512 may in fact be a set of hash algorithms. In that case, the hash algorithm is selected on the basis of a hash algorithm identifier or other appropriate encoded value passed by the control information retrieval means 608 to the DSIG verification means (not shown in Fig. 10). The clear public key PU obtained from PU recovery means 606 is then used to encrypt the value of DSIG specified as an input at 602. This recovers the original signature block in clear form, which is then parsed to recover the original hash value. The recovered hash value and the calculated hash value are then compared for equality. If this comparison is favorable, then DSIG is considered valid; otherwise, DSIG is not considered valid and IDK instruction 600 is aborted. The signature block recovery and processing of course will depend on the method of digital signature implemented. In the description of the GKSP instruction it was indicated that the signature block may consist of the hash value and padding data or it may be constructed on the basis of a national or international standard, such as International Standards Organization draft international standard (ISO DIS) 9796. Those skilled in the art will appreciate that many possible implementations of the digital signature are possible and that the precise method of digital signatures is unimportant to the invention. What is important is that a method of digital signature is used in the preferred embodiment to ensure that the receiving cryptographic device can authenticate that the to-be-imported DEA key did in fact originate from a valid network cryptographic device. As the reader will also see, the digital signature is made an integral part of the GKSP and IDK instructions themselves, which ensures that the process of signature production and signature verification occurs as part of the key export and key import processes and therefore the highest possible integrity over these processes is achieved. Although it is possible to perform signature production and signature verification as separate instructions, which achieves complete compatibility with the present descriptions of the GKSP and IDK instructions, one also sees that less integrity is achieved. This is so because the signature generate instruction has no way to ensure that a key of the form ePU(keyblk) was in fact produced by the GKSP instruction.

Keyblk recovery means 612 decrypts input ePU(keyblk), provided as an input to the IDK instruction at 603, under private key PR, provided as an output of PR recovery means 607. The recovered clear key block, keyblk, is provided as an output to both eKM.C6(K) production means 613 and control information consistency checking means 614.

Control information consistency checking means 614 checks the control information in the recovered keyblk output from keyblk recovery means 612 and the reference control information output from control information retrieval means 608 for consistency. A first check consists in checking control vector C6 in keyblk for consistency with the reference control vector C6 supplied as an input to the IDK instruction at 605. This ensures that the receiving cryptographic application imports a key from with the expected or intended key usage attributes. In this case, reference control vector C6 represents the expected control vector, whereas the recovered control vector C6 represents the actual control vector. The simplest form of consistency checking consists of checking these two control vectors for equality. However, a more refined procedure is possible wherein attributes in the reference control vector are allowed to override corresponding attributes in the recovered control vector. For example, the reference control vector could disable the ability to re-export the imported DEA key K, whereas the recovered control vector may or may not permit the imported DEA key K to be re-exported. More generally, the receiving device may disable any attribute granted within the received control vector. One will appreciate that taking away a right is not the same as granting a right, which only the sending cryptographic device is permitted to do. The IDK instruction can be designed to permit this kind of control vector override or it may not, depending on the desires of the designer of the IDK instruction. A second check consists of checking the EID value in keyblk for equality with the reference EID value supplied as an input to the IDK instruction at 605. This ensures that the receiving cryptographic application imports a key from the expected or intended sending cryptographic device. In this case, the reference EID value is the EID of the intended sending cryptographic device, which is checked against the EID value in keyblk which represents the EID value of the actual sending cryptographic device. A third check consists of checking the EID value in keyblk for inequality with the EID value stored in the cryptographic facility of the receiving device. This ensures that the imported DEA K originated at another cryptographic device, i.e., that A can't import a K produced at A, that B can't import a K produced at B, etc. The usefulness of this check has been discussed previously. In all cases, if the consistency checking fails, then the IDK instruction is aborted.

eKM.C6(K) production means 613 extracts the clear value of DEA key K and the control vector C6 from keyblk, obtained as an output from the keyblk recovery means 612. and K is then encrypted under variant key KM.C6 formed as the Exclusive OR product of master key KM and control vector C6 recovered from C6 to produce the encrypted key value eKM.C6(K). In an alternative embodiment where the reference control vector C6 can override the recovered control vector C6, the value of C6 used to form the variant key KM.C6 can be the reference control vector C6. In yet another alternative embodiment the IDK instruction itself can modify information in the control vector C6, so that K is encrypted with variant key KM.C6', where C6' is the IDK modified value of C6. In any event, the encrypted key eKM.C6(K) is returned to the CFAP as an instruction output at 615.

The reader will appreciate that the IDK instruction has been designed to perform consistency checking within the cryptographic facility in lieu of returning the recovered clear values of C6 and EID to the CFAP and performing this consistency checking outside of the cryptographic facility. In the preferred embodiment, this consistency checking is performed in the cryptographic facility hardware and the recovered clear values of C6 and EID are not exposed outside the CF. The reason for doing this is to ensure that the DEA key distribution channel does not also provide a covert privacy channel whereby secret data may be incorporated in the control information portion of the key block and transmitted from the sending cryptographic device to the receiving cryptographic device. In a good cryptographic design, the cryptographic instructions will perform only those cryptographic functions for which they were designed, and no more. Doing so, limits the ways in which an attacker can manipulate the cryptographic instructions for the purpose of subverting their intended security. For example, a system administrator in charge of security policy for the sending and recieving locations, may have a security policy which prohibits the transmission of private messages over the communications link, for example when the link is dedicated merely to the transmission of new keys. In an alternate security policy where the system administrator is to selectively allow privacy channels, there should be no "back door" method for subverting the system administrator's authority in enabling or prohibiting such privacy channels. The use of the control information transmitted over the separate channel to the receiver, is to enable the recipient to inspect the type of uses imposed on the receive key and allow the recipient the option of rejecting the keyblock. However, an alternate embodiment is possible wherein the recovered clear values of C6 and EID are returned to the CFAP and consistency checking is then performed by the CFAP.

Control Information: Fig. 12 further illustrates the unique role played by the encrypted key block, ePU(keyblk), and the external key token. Although Figs. 5, 7, and 8 depict external key token as containing an encrypted key block of the form ePU(keyblk) and reference control information in clear form, in the logical sense there are two information channels over which information flows: (1) and encrypted channel and (2) a clear channel. Referring now to Fig. 12, therein is shown two cryptographic systems, A and B, that communicate via a key distribution protocol through an encrypted channel 701 and a clear channel 702. Encrypted channel 701 is facilitated via the encrypted key block, ePU(keyblk). Control information in keyblk, which is subsequently encrypted with public key PU, is thus sent from A to B via an encrypted channel. Clear channel 702 is facilitated via the external key token. Control information in clear form stored in the external key token is, for all intents and purposes, passed from A to B via a clear channel.

Another distinguishing feature of the two channels is this. Encrypted channel 701 is a logical channel between the cryptographic facility 30 of cryptographic system A and cryptographic facility 30' of cryptographic system B. Clear channel 702 is a logical channel between application 36 in cryptographic system A and application 36' in cryptographic system B. and possibly a logical channel between CFAP 34 in cryptographic system A and CFAP 34' in cryptographic system B, depending on how the external key tokens are to be managed. In any event, the key distribution process is designed such that (1) in the case of encrypted channel 701, only the cryptographic facilities have access to the control information in keyblk, whereas (2) in the case of clear channel 702, the applications and possibly the CFAPs have access to the control information in the external key token as a routine part of the key distribution protocol. Since the CF is typically implemented within secure hardware, the CF is said to have a higher level of integrity than other parts of the the cryptographic system, such as the CFAP and applications operating within the cryptographic system. Thus, a higher degree of protection is achieved within the key distribution process by controlling that process, to the degree possible, from within the CF itself. To this end, control information is passed via encrypted channel 701, in keyblk, from A to B, thus enabling B to process the imported keyblk with a high degree of integrity. Of course, the assumption is made here that digital signatures are also a part of the key distribution process, as shown in Fig. 8, which forms another underpinning or layer of integrity that augments and enhances the overall integrity of information passed via encrypted channel 701.

Fig. 13 illustrates the process of reconciliation between control information transmitted via encrypted channel 701 and control information, called reference control information, transmitted via clear channel 702. The importance in having these two channels for passing control information can now be seen. Control information transmitted via encrypted channel 701 can be 'seen' by the cryptographic facility, but by no one outside the cryptographic facility. This ensures that the key distribution channel is not used as a covert privacy channel. Thus, the only way that the application program or the CFAP has of validating the control information transmitted via encrypted channel 701 is the specify reference control information to the CF in clear form. Since it is the application program or the CFAP that specifies the reference control information, the reference control information is consistency checked to determine its accuracy before being passed to the CF. Inside the protected boundary of the CF, the control information recovered from the decrypted keyblk is checked for consistency with the reference control information supplied in clear form by the application to the CFAP and thence by the CFAP to the CF. This permits all parties (CF, CFAP, and application) to be sure that the the control information associated with the to-be-imported key is correct and in accordance with expectations. In summary, all parties look at the reference control information and have a chance to agree or disagree with it, but only the CF sees the control information passed in keyblk and only the CF with highest integrity determines whether the control information received via encrypted channel 701 (i.e., in keyblk) is consistent with the reference control information received in the external key token by the application, or by the CFAP depending on whether key distribution is implemented at the application layer or at the cryptographic facility access program layer. Referring now to Fig. 13, reference control information (designated RCI) received via Clear Channel 702 is inspected by the receiving application program APPL 36. If APPL 36 finds the reference control information to be okay, i.e., it is accurate acceptable, and in accordance with the protocol, in all respects, then APPL 36 will issue a request to CFAP 34 to import the received DEA key, passing the reference control information in the received external key token to CFAP 34. If the CFAP 34 finds the reference control information to be okay, then CFAP 36 will issue an IDK instruction to CF 30 to import the received DEA key, passing the reference control information in the received external key token to CF 30. The control information (designated CI) received via Encrypted Channel 701 and the reference control information RCI received from the CFAP 34 are inspected by themselves for consistency and then they are compared or consistency checked, one against the other, to determine that CI is consistent with RCI. Only if this consistency checking succeeds, will the CF 30 import the DEA key.

Fig. 14 is a block diagram of the cryptographic facility 30 in the sending location A, as it is organized for performing the generate key set PKA (GKSP) instruction, illustrated in Fig. 9. Fig. 14 shows the cryptographic facility 30 at the sending location which includes the crypto variable retrieval means 40, shown in greater detail in Fig. 16. To prepare a crypto variable such as the key K for transmission from the cryptographic facility 30 to the cryptographic facility 30' at the receiving location, the key K is accessed from the crypto variable retrieval means 40 over the line 62 and applied to the concatenation means 42. In addition, control information such as a control vector and an environmental identification are accessed over line 60 from the crypto variable retrieval means 40 and are applied to the concatenation means 42. Concatenation means 42 will concatenate the key K with the control vector and the environmental identification and that will form the key block 80 which is applied to the public key algorithm encryption means 44. The public key is accessed over line 70 from the crypto variable retrieval means 40 and is applied to the key input of the encryption means 44. The key block 80 is encrypted forming the encrypted key block 85 which is then applied to the transmitting means 46. The encrypted key block 85 is then transmitted over the transmission link 12 to the cryptographic facility 30' at the receiving location shown in Fig. 15. The control information consisting of the control vector and the environmental identification which has been accessed over line 60 is also output as a separate information unit to the transmitting means 46 for transmission over the link 12 to the cryptographic facility 30' at the receiving location. The control information, which can be referred here as the reference control information, is separate from the encrypted key block 85. The reference control information can be transmitted over the same physical communications link 12 as the encrypted key block 85, in a different time slot or frequency slot in the case of frequency division multiplexing. Alternately, completely separate physical communication links can be employed to transmit the reference control information as distinguished from the transmission of the encrypted key block 85. The transmission of the reference control information can be in either clear text form, or alternately the reference control information can be encrypted and transmitted over a privacy channel if the sender and receiver share suitable keys.

In the receiving cryptographic facility 30' shown in Fig. 15, the reference control information is transferred from the communications link 12 to the overline 74 to the control information comparison means 59. The encrypted key block 85 is transferred from the receiving means 56 to the public key algorithm decryption means 54. A privacy key is accessed over line 72 from the crypto variable retrieval means 40' and is applied to the key input of the decryption means 54. The operation of the decryption means 54 generates the recovered key block 80 which is applied to the extraction means 52. The extraction means 52 extracts the control information 60 from the recovered key block 80 and applies the extracted control information to the control information comparison means 59. The control information comparison means 59 then compares the identity of the extracted control information from the key block 80 with the reference control information received from the communications link 12 over line 74. The control information comparison means 59 has an enabling output signal 90 which is produced if the comparison is satisfied. The enabling signal 90 is applied to an enabling input of the crypto variable storage means 50. The crypto variable, in this example the key K, is output from the extraction means 52 on line 62 and applied to the crypto variable storage means 50. The key K will be successfully stored in a crypto variable storage means 50 if the enabling signal 90 is applied from the comparison means 59. In addition, the control information which can include the control vector and the environmental ID of the sending location, can also be stored in the crypto variable storage means 50, if the enabling signal 90 is present.

Further in accordance with the invention, a comparison can be made between the environmental ID of the receiving station B and the environmental ID of the transmitting station A, in order to ensure that the environmental ID for the receiving station B is not identical with the environmental ID contained in the recovered key block 80. This comparison can also be performed in the comparison means 59 and the successful comparison can be made necessary to the generation of the enabling signal 90 as described above. The environmental ID in the reference control information should successfully compare with the environmental ID extracted by the extraction means 52 from the key block 80. In addition, the environmental ID extracted from the key block 80, which represents the environmental ID of the sending location A, should not be the same as the environmental ID of the receiving station B. When these two conditions exist and also when the control vector in the reference control information successfully compares with the control vector extracted by the extraction means 52 from the key block 80, then the comparison means 59 will output an enabling signal 90 to the storage means 50.

The crypto variable retrieval means 40 and 40' is shown in greater detail in Fig. 16. Input parameters 311 can be transferred over line 33 from the external storage 400 in the CFAP 34. These input parameters can then be applied to the crypto facility 30, over lines 31. Op codes 310 in the CFAP 34 can also be applied over lines 31 to the crypto facility 30. The crypto facility 30 includes the crypto variable retrieval means 40 which contains a random number generator 95, a data encryption algorithm decryption means 410 and an output selection means 420. A master key storage 99 is contained in the crypto facility 30, having an output connected to the key input of the decryption means 410. The random number generator 95 can generate a first type key K' to be applied to the output selection means 420. Alternately, a second type key K" in clear text form can be applied to the output selection means 420. Alternately, a third type key K'" can be applied to the output selection means 420, which is derived from the decryption by the decryption means 410 of an encrypted form of the key K'" which has been encrypted under the exclusive OR product of the master key KM and the control vector C5. The output of the selection means 420 is the key K which is the crypto variable which is discussed in relation to Figs. 14 and 15.

In the preferred embodiment of the invention, public key encryption is used as the encryption technique for transmitting the key block from the sending location to the receiving location, however, it is within the scope of the invention to use symmetric, private key techniques for enciphering and deciphering the key block. Also, in the preferred embodiment of the invention, where digital signatures are employed, as described above, the public key encryption technique for forming digital signatures is employed. However, in an alternate embodiment, conventional Message Authentication Code (MAC) techniques may be employed using a private key algorithm. In the preferred embodiment of the invention, Data Encryption Standard (DES) key is the crypto variable which is transmitted in the key block from the sending location to the receiving location, however, in alternate embodiments of the invention, the crypto variable can be a public key or a non-key-type expression.

Although a specific embodiment of the invention has been disclosed, it will be understood by those having skill in the art that changes can be made to the specific embodiment without departing from the spirit and the scope of the invention.

Claims

1. In a data processing system having a plurality of communicating nodes, at least a pair of nodes in the system exchanging cryptographic communications, an apparatus for enabling a first node of the pair to control a crypto variable after its transmission from the first node to a second node of the pair, comprising:

   a storage means at a transmitting node in the system for storing a crypto variable which is to be transmitted to a receiving node in the system;

   said storage means storing control information including a control vector to control said crypto variable after it is transmitted from said transmitting node;

   said storage means storing a first key expression;

   concatenating means at said transmitting node, coupled to said storage means, for concatenating said crypto variable with said control information, forming a key block;

   encryption means at said transmitting node, coupled to said storage means and said concatenating means, for encrypting said key block with said first key expression, forming an encrypted key block; and

   transmitting means at said transmitting node coupled to said encryption means and coupled over a communications link to a receiving means at said receiving node, for transmitting said encrypted key block to said receiving node.

2. In a data processing system having a plurality of communicating nodes, at least a pair of nodes in the system exchanging cryptographic communications, an apparatus for enabling a first node of the pair to control a crypto variable after its transmission from the first node to a second node of the pair, comprising:

   a first storage means at a transmitting node in the system for storing a crypto variable which is to be transmitted to a receiving node in the system;

   a second storage means at said transmitting node for storing control information to control said crypto variable after it is transmitted from said transmitting node said control information including a control vector to limit the uses of said crypto variable;

a third storage means at said transmitting node for storing a first key expression;

concatenating means at said transmitting node, coupled to said first and second storage means, for concatenating said crypto variable with said control information, forming a key block;

encryption means at said transmitting node, coupled to said third storage means and said concatenating means, for encrypting said key block with said first key expression, forming an encrypted key block;

transmitting means at said transmitting node coupled to said encryption means and coupled over a communications link to a receiving means at said receiving node, for transmitting said encrypted key block to said receiving node;

said transmitting means coupled to said second storage means, for transmitting a second copy of said control information to said receiving node;

fourth storage means at said receiving node, for storing a second key expression corresponding to said first key expression;

decryption means at said receiving node coupled to said receiving means and to said fourth storage means, for decrypting said encrypted key block using said second key expression, to obtain a recovered key block;

extraction means at said receiving node coupled to said decryption means, to extract said control information and said crypto variable from said recovered key block;

comparison means at said receiving node coupled to said extraction means and coupled to said receiving means for comparing said control information extracted from said recovered key block to said second copy of said control information, said comparison means having an enabling output for signalling when said comparison is satisfied;

control means coupled to said extraction means and having an enabling input coupled to said output of said comparison means, for controlling said crypto variable with said control information.

3. Apparatus for generating and distributing a Data Encryption Algorithm (DEA) key in a communications network, comprising:
   a) sending means for generating and producing at least two copies of a key-encrypting key (k-ek), and control information including a control vector for permitted uses of the k-ek;
   b) means included in the sending means for encrypting one copy of the k-ek under the public key of a receiving means and transmitting the public key encrypted k-ek to the receiving means in association with said control information;
   c) means further included in the sending means for encrypting another copy of the k-ek under a master key of the sending means;
   d) means further included in the sending means for storing the master key encrypted k-ek as a common distributing key for other encrypted keys used in the network, in association with said control information;
   e) control means included in the sending means, to limit uses of the k-ek to said permitted uses in response to said control information.

4. The apparatus of claim 1, 2, or 3, wherein:

said first key expression and said second key expression being symmetric keys, and/or wherein

said first key expression being a public key issued by said receiving node and said second key expression being a private key corresponding to said public key.

5. The apparatus of anyone of the preceeding claims, wherein said control means further comprises:

a reference storage means at said receiving node for storing a reference control vector characterizing required uses of said crypto variable at said receiving station;

a received control vector storage means at said receiving node, coupled to said extraction means, for storing said received control vector extracted from said recovered key block;

said comparison means at said receiving node coupled to said reference storage means and to said received control vector storage means, for comparing said reference control vector with said received control vector, and outputting an acceptance signal if the comparison succeeds;

crypto variable storage means at said receive node coupled to said extraction means and to said comparison means, for storing said crypto variable extracted by said extraction means if said acceptance signal is received from said compare means.

6. The apparatus of claim 5, wherein said crypto variable storage means further comprises:

a master key storage means at said receiving node for storing a master key;

an exclusive OR means at said receiving node coupled to said master key storage means and to said received or reference control vector storage means respectively, for forming an exclusive OR product of said master key and said received or reference control vector, respectively, forming a product key expression;

an encryption engine at said receiving node having a key input coupled to said exclusive OR means for inputting said product key expression, and having an operand input coupled to said crypto variable storage means, for encrypting said crypto variable under said master key, forming an encrypted crypto variable;

an encrypted crypto variable storage means at said receiving node, coupled to said encryption engine, for storing said encrypted crypto variable.

7. The apparatus of claim 6, which further comprises:

a control vector checking means at said receiving node coupled to a user input, for receiving a request from a user for using said crypto variable;

said control vector checking means being coupled to said received control vector storage means, for checking said received control vector to determine if said requested uses are permitted;

said control vector checking means outputting an enabling signal if said requested uses are permitted;

a processing means at said receiving node coupled to said control vector checking means, to said received control vector storage means and to said master key storage means, for receiving said enabling signal and in response thereto, forming an exclusive OR product of said master key and said received control vector, forming a product key expression;

a decryption engine at said receiving node having a key input coupled to said exclusive OR means for inputting said product key expression, and having an operand input coupled to said encrypted crypto variable storage means, for decrypting said encrypted crypto variable under said master key, recovering said crypto variable.

8. The apparatus of claim 6 or 7, which further comprises:

a control vector checking means at said receiving node coupled to a user input, for receiving a request from a user for using said crypto variable;

said control vector checking means being coupled to said reference control vector storage means, for checking said reference control vector to determine if said requested uses are permitted;

said control vector checking means outputting an enabling signal if said requested uses are permitted;

a processing means at said receiving node coupled to said control vector checking means, to said reference control vector storage means and to said master key storage means, for receiving said enabling signal and in response thereto, forming an exclusive OR product of said master key and said reference control vector, forming a product key expression;

a decryption engine at said receiving node having a key input coupled to said exclusive OR means for inputting said product key expression, and having an operand input coupled to said encrypted crypto variable storage means, for decrypting said encrypted crypto variable under said master key, recovering said crypto variable;

wherein said reference control vector is received preferably from said transmitting node.

9. The apparatus of claim 8, which further comprises:

said received control vector is a first hashed product of said reference control vector, received from said transmitting node;

hashing means in said receiving node coupled to said reference control vector storage means, for forming a second hash product of said reference control vector;

second comparison means coupled to said received control vector storage means and to said hashing means, for comparing said first hashed product with said second hashed product and outputting a second acceptance signal when the comparison is satisfied.

10. The apparatus of anyone of claims 1 to 4, which further comprises:

said control information includes a hashed control vector which represents limitations on uses of said crypto variable.

11. The apparatus of claim 10, wherein said control means further comprises:

a reference control vector storage means at said receiving node for receiving from said transmitting node and storing a reference control vector characterizing required uses of said crypto variable at said receiving station;

a hashed control vector storage means at said receiving node, coupled to said extraction means, for storing said hashed control vector extracted from said recovered key block;

hashing means in said receiving node coupled to said reference control vector storage means, for forming a hash product of said reference control vector;

compare means at said receiving node coupled to said hashing means and to said hashed control vector storage means, for comparing said hash product with said hashed control vector, and outputting an acceptance signal if the comparison succeeds;

crypto variable storage means at said receive node coupled to said extraction means and to said compare means, for storing said crypto variable extracted by said extraction means if said acceptance signal is hashed from said compare means.

12. The apparatus of claim 11, wherein said crypto variable storage means further comprises:

a master key storage means at said receiving node for storing a master key;

an exclusive OR means at said receiving node coupled to said master key storage means and to said hashed control vector storage means, for forming an exclusive OR product of said master key and said hashed control vector, forming a product key expression;

an encryption engine at said receiving node having a key input coupled to said exclusive OR means for inputting said product key expression, and having an operand input coupled to said crypto variable storage means, for encrypting said crypto variable under said master key, forming an encrypted crypto variable;

an encrypted crypto variable storage means at said receiving node, coupled to said encryption engine, for storing said encrypted crypto variable.

13. The apparatus of claim 12, which further comprises:

a control vector checking means at said receiving node coupled to a user input, for receiving a request from a user for using said crypto variable;

said control vector checking means being coupled to said reference control vector storage means, for checking said reference control vector to determine if said requested uses are permitted;

said control vector checking means outputting an enabling signal if said requested uses are permitted;

a processing means at said receiving node coupled to said control vector checking means, to said hashed control vector storage means and to said master key storage means, for receiving said enabling signal and in response thereto, forming an exclusive OR product of said master key and said hashed control vector, forming a product key expression;

a decryption engine at said receiving node having a key input coupled to said exclusive OR means for inputting said product key expression, and having an operand input coupled to said encrypted crypto variable storage means, for decrypting said encrypted crypto variable under said master key, recovering said crypto variable.

14. The apparatus of anyone of the preceeding claims, which further comprises:

said control information includes a transmitting node environment identification which characterizes the identity of said transmitting node.

15. The apparatus of claim 14, wherein said control means further comprises:

a receiving node environment identification storage means at said receiving node for storing a receiving node environment identification;

a received transmission node environment identification storage means at said receiving node, coupled to said extraction means, for storing said transmitting node environment identification extracted from said recovered key block;

compare means at said receiving node coupled to said receiving node environment identification storage means and to said received transmission node environment identification storage means, for comparing said receiving node environment identification and said transmitting node environment identification and outputting an acceptance signal if the comparison fails;

crypto variable storage means at said receive node coupled to said extraction means and to said compare means, for storing said crypto variable extracted by said extraction means if said acceptance signal is received from said compare means.

16. In a data processing system having a plurality of communicating nodes, at least a pair of nodes in the system exchanging cryptographic communications, a method for enabling a first node of the pair to control a crypto variable after its transmission from the first node to a second node of the pair, comprising:

storing a crypto variable which is to be transmitted to a receiving node in the system, at a transmitting node;

storing control information to control said crypto variable after it is transmitted from said transmitting node, at said transmitting node said control information including a control vector to limit the uses of said crypto variable;

storing a first key expression at said transmitting node;

concatenating said crypto variable with said control information, forming a key block, at said transmitting node;

encrypting said key block with said first key expression, forming an encrypted key block, at said transmitting node;

transmitting said encrypted key block to said receiving node;

transmitting a second copy of said control information to said receiving node;

storing a second key expression corresponding to said first key expression, at said receiving node;

decrypting said encrypted key block using said second key expression, to obtain a recovered key block, at said receiving node;

extracting said control information and said crypto variable from said recovered key block, at said receiving node;

comparing said control information extracted from siad recovered key block with said second copy of said control information and generating an enabling signal when the compare is satisfied;

controlling said crypto variable with said control information when said enabling signal has been generated.

17. In a data processing system having a plurality of communicating nodes, at least a pair of nodes in the system exchanging cryptographic communications, a method for enabling a first node of the pair to control a crypto variable after its transmission from the first node to a second node of the pair, comprising:

concatenating a crypto variable with control information including a control vector to control said crypto variable after it is transmitted from said transmitting node, forming a key block, at said transmitting node;

encrypting said key block with a first key expression, forming an encrypted key block, at said transmitting node;

transmitting said encrypted key block to said receiving node;

decrypting said encrypted key block using a second key expression, to obtain a recovered key block, at said receiving node;

extracting said control information and said crypto variable from said recovered key block, at said receiving node;

validating said control information extracted from said recovered key block and generating an enabling signal;

controlling said crypto variable with said control information when said enabling signal has been generated.

18. In a data processing system having a plurality of communicating nodes, at least a pair of nodes in the system exchanging cryptographic communications, a program for execution on the data processing system for enabling a first node of the pair to control a crypto variable after its transmission from the first node to a second node of the pair, comprising:

said program controlling the data processing system for storing a crypto variable which is to be transmitted to a receiving node in the system, at a transmitting node;

said program controlling the data processing system for storing control information to control said crypto variable after it is transmitted from said transmitting node, at said transmitting node said control information including a control vector to limit the uses of said crypto variable;

said program controlling the data processing system for storing a first key expression at said transmitting node;

said program controlling the data processing system for concatenating said crypto variable with said control information, forming a key block, at said transmitting node;

said program controlling the data processing system for encrypting said key block with said first key expression, forming an encrypted key block, at said transmitting node;

said program controlling the data processing system for transmitting said encrypted key block to said receiving node;

said program controlling the data processing system for transmitting a second copy of said control information to said receiving node;

said program controlling the data processing system for storing a second key expression corresponding to said first key expression, at said receiving node;

said program controlling the data processing system for decrypting said encrypted key block using said second key expression, to obtain a recovered key block, at said receiving node;

said program controlling the data processing system for extracting said control information and said crypto variable from said recovered key block, at said receiving node;

said program controlling the data processing system for comparing said control information extracted from said recovered key block with said second copy of said control information and generating an enabling signal when the compare is satisfied;

said program controlling the data processing system for controlling said crypto variable with said control information when said enabling signal has been generated.

19. The method of claim 16 or 17, or the program of claim 18, which further comprises:

said first key expression and said second key expression being symmetric keys, and/or

said first key expression being a public key issued by said receiving node and said second key expression being a private key corresponding to said public key.

20. The method of claim 16, 17, or the program of claim 18 or 19, which further comprises:

said control information includes a received control vector which defines limitations on uses of said crypto variable.

21. The program of claim 20, which further comprises:

said program controlling the data processing system for storing a reference control vector characterizing required uses of said crypto variable at said receiving node;

said program controlling the data processing system for storing said received control vector extracted from said recovered key block, at said receiving node;

said program controlling the data processing system for comparing said reference control vector with said received control vector, and outputting an acceptance signal if the comparison succeeds, at said receiving node;

said program controlling the data processing system for storing said crypto variable extracted by said extraction means if said acceptance signal is received from said compare means, at said receiving node.

22. The program of claim 21, which further comprises:

said program controlling the data processing system for storing a master key at said receiving node;

said program controlling the data processing system for forming an exclusive OR product of said master key and said received control vector, forming a product key expression, at said receiving node;

said program controlling the data processing system for encrypting said crypto variable under said master key, forming an encrypted crypto variable, at said receiving node;

said program controlling the data processing system for storing said encrypted crypto variable, at said receiving node.

23. The program of claim 22, which further comprises:

said program controlling the data processing system for receiving a request from a user for using said crypto variable, at said receiving node;

said program controlling the data processing system for checking said received control vector to determine if said requested uses are permitted, at said receiving node;

said program controlling the data processing system for outputting an enabling signal if said requested uses are permitted, at said receiving node;

said program controlling the data processing system for receiving said enabling signal and in response thereto, forming an exclusive OR product of said master key and said received control vector, forming a product key expression, at said receiving node;

said program controlling the data processing system for inputting said product key expression, and decrypting said encrypted crypto variable under said master key, recovering said crypto variable, at said receiving node.

24. The program of claim 21, which further comprises:

said program controlling the data processing system for storing a master key at said receiving node;

said program controlling the data processing system for forming an exclusive OR product of said master key and said reference control vector, forming a product key expression, at said receiving node;

said program controlling the data processing system for inputting said product key expression, and encrypting said crypto variable under said master key, forming an encrypted crypto variable, at said receiving node;

said program controlling the data processing system for storing said encrypted crypto variable, at said receiving node.

25. The program of claim 24, which further comprises:

said program controlling the data processing system for receiving a request from a user for using said crypto variable, at said receiving node;

said program controlling the data processing system for checking said reference control vector to determine if said requested uses are permitted, at said receiving node;

said program controlling the data processing system for outputting an enabling signal if said requested uses are permitted, at said receiving node;

said program controlling the data processing system for receiving said enabling signal and in response thereto, forming an exclusive OR product of said master key and said reference control vector, forming a product key expression, at said receiving node;

said program controlling the data processing system for decrypting said encrypted crypto variable under said master key, recovering said crypto variable, at said receiving node;

wherein said reference control vector is received preferably from said transmitting node.

26. The program of claim 25, which further comprises:

said received control vector is a first hashed product of said reference control vector, received from said transmitting node;

said program controlling the data processing system for forming a second hash product of said reference control vector, at said receiving node;

said program controlling the data processing system for comparing said first hashed product with said second hashed product and outputting a second acceptance signal when the comparison is satisfied, at said receiving node.

27. The program of claim 20, which further comprises:

said control information includes a hashed control vector which represents limitations on uses of said crypto variable.

28. The program of claim 27, wherein said control means further comprises:

said program controlling the data processing system for receiving from said transmitting node and storing a reference control vector characterizing required uses of said crypto variable at said receiving station, at said receiving node;

said program controlling the data processing system for storing said hashed control vector extracted from said recovered key block, at said receiving node;

said program controlling the data processing system for forming a hash product of said reference control vector, at said receiving node;

said program controlling the data processing system for comparing said hash product with said hashed control vector, and outputting an acceptance signal if the comparison succeeds, at said receiving node;

said program controlling the data processing system for storing said crypto variable extracted by said extraction means if said acceptance signal is hashed from said compare means, at said receiving node.

29. The program of claim 28, which further comprises:

said program controlling the data processing system for storing a master key at said receiving node;

said program controlling the data processing system for forming an exclusive OR product of said master key and said hashed control vector, forming a product key expression, at said receiving node;

said program controlling the data processing system for inputting said product key expression, and encrypting said crypto variable under said master key, forming an encrypted crypto variable, at said receiving node;

said program controlling the data processing system for storing said encrypted crypto variable, at said receiving node.

30. The program of claim 29, which further comprises:

said program controlling the data processing system for receiving a request from a user for using said crypto variable, at said receiving node;

said program controlling the data processing system for checking said reference control vector to determine if said requested uses are permitted, at said receiving node;

said program controlling the data processing system for outputting an enabling signal if said requested uses are permitted, at said receiving node;

said program controlling the data processing system for receiving said enabling signal and in response thereto, forming an exclusive OR product of said master key and said hashed control vector, forming a product key expression, at said receiving node;

said program controlling the data processing system for inputting said product key expression, and decrypting said encrypted crypto variable under said master key, recovering said crypto variable, at said receiving node.

31. The method or the program of anyone of the claims 16 to 30, which further comprises:

said control information includes a transmitting node environment identification which characterizes the identity of said transmitting node.

32. The program of claim 31, which further comprises:

said program controlling the data processing system for storing a receiving node environment identification, at said receiving node;

said program controlling the data processing system for storing said transmitting node environment identification extracted from said recovered key block, at said receiving node;

said program controlling the data processing system for comparing said receiving node environment identification and said transmitting node environment identification and outputting an acceptance signal if the comparison fails, at said receiving node;

said program controlling the data processing system for storing said crypto variable extracted by said extraction means if said acceptance signal is received from said compare means, at said receiving node.

FIG. 1

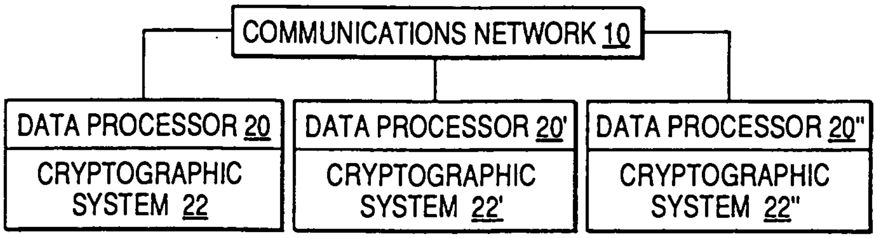

COMMUNICATIONS NETWORK 10
DATA PROCESSOR 20
CRYPTOGRAPHIC SYSTEM 22
DATA PROCESSOR 20'
CRYPTOGRAPHIC SYSTEM 22'
DATA PROCESSOR 20"
CRYPTOGRAPHIC SYSTEM 22"

FIG. 2

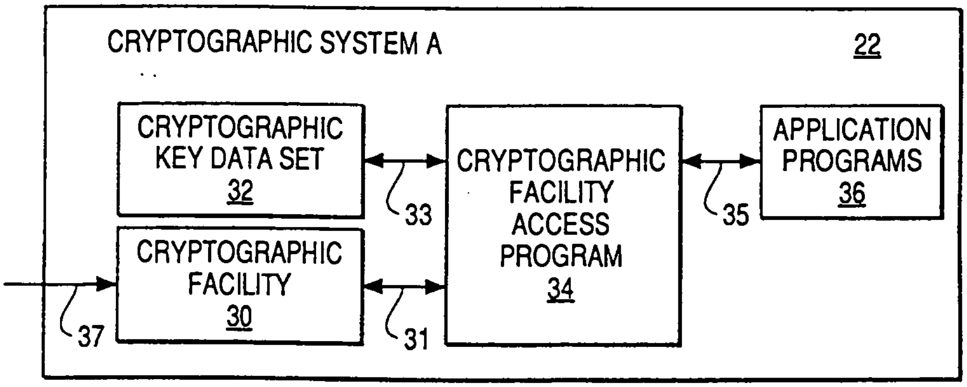

CRYPTOGRAPHIC SYSTEM A
22
CRYPTOGRAPHIC KEY DATA SET 32
33
CRYPTOGRAPHIC FACILITY ACCESS PROGRAM 34
35
APPLICATION PROGRAMS 36
CRYPTOGRAPHIC FACILITY 30
37
31

FIG. 3

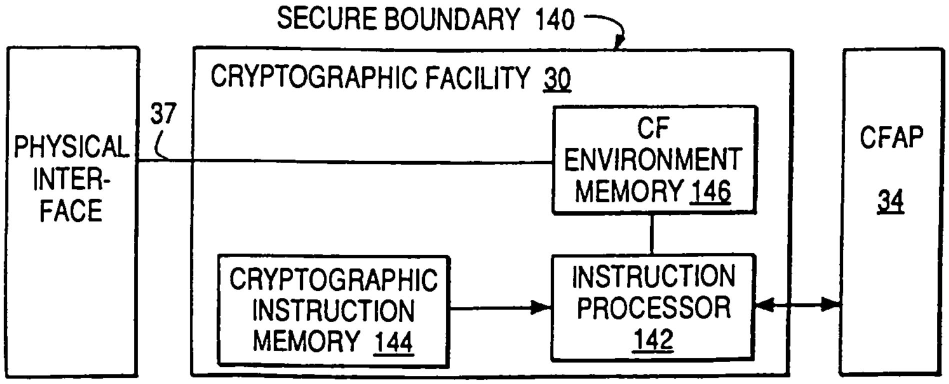

SECURE BOUNDARY 140
CRYPTOGRAPHIC FACILITY 30
PHYSICAL INTER-FACE
37
CF ENVIRONMENT MEMORY 146
CFAP 34
CRYPTOGRAPHIC INSTRUCTION MEMORY 144
INSTRUCTION PROCESSOR 142

FIG. 4

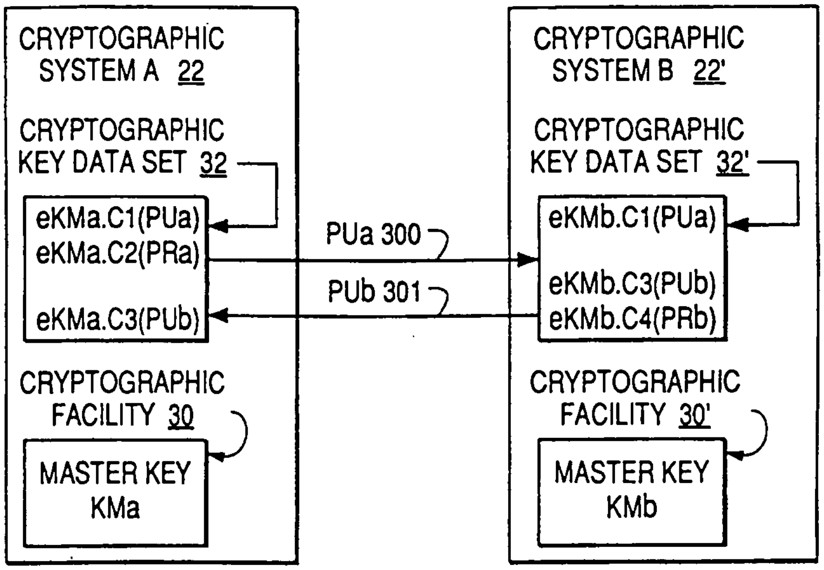

CRYPTOGRAPHIC SYSTEM A 22
CRYPTOGRAPHIC KEY DATA SET 32
eKMa.C1(PUa)
eKMa.C2(PRa)
eKMa.C3(PUb)
CRYPTOGRAPHIC FACILITY 30
MASTER KEY KMa
PUa 300
PUb 301
CRYPTOGRAPHIC SYSTEM B 22'
CRYPTOGRAPHIC KEY DATA SET 32'
eKMb.C1(PUa)
eKMb.C3(PUb)
eKMb.C4(PRb)
CRYPTOGRAPHIC FACILITY 30'
MASTER KEY KMb

FIG. 5

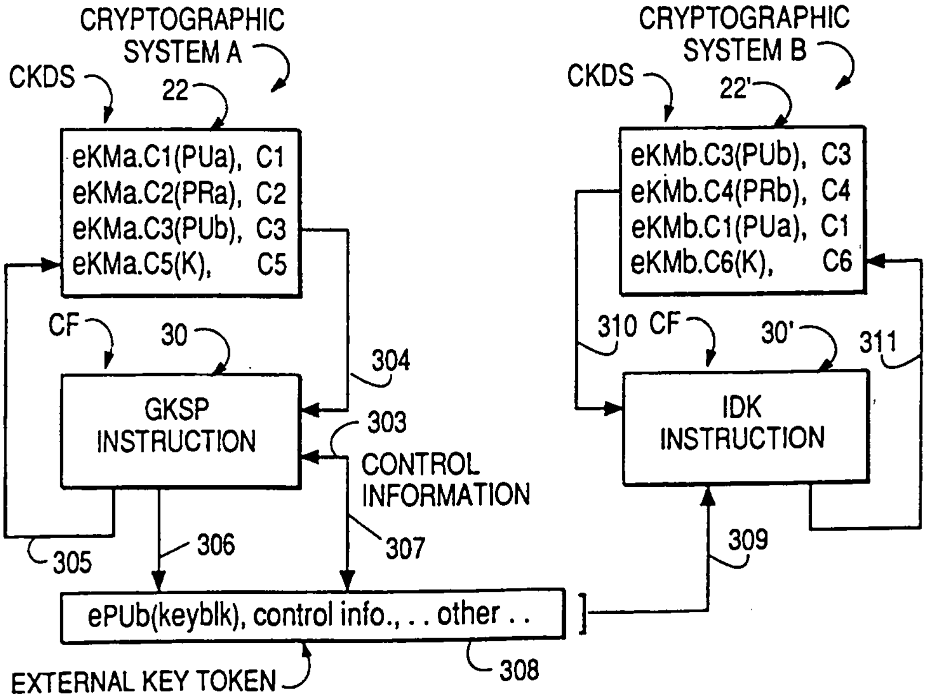

CRYPTOGRAPHIC SYSTEM A
22
CKDS
eKMa.C1(PUa), C1
eKMa.C2(PRa), C2
eKMa.C3(PUb), C3
eKMa.C5(K), C5
CF
30
304
GKSP INSTRUCTION
303
CONTROL INFORMATION
305
306
307
ePUb(keyblk), control info., . . other . .
EXTERNAL KEY TOKEN
308
CRYPTOGRAPHIC SYSTEM B
22'
CKDS
eKMb.C3(PUb), C3
eKMb.C4(PRb), C4
eKMb.C1(PUa), C1
eKMb.C6(K), C6
310
CF
30'
311
IDK INSTRUCTION
309

FIG. 6

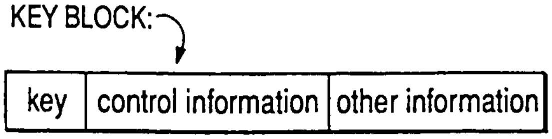

KEY BLOCK:
key
control information
other information

FIG. 7

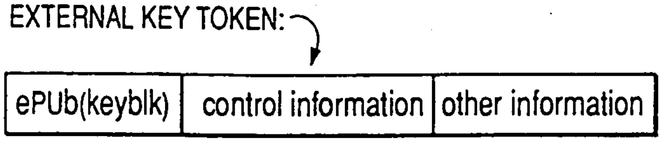

EXTERNAL KEY TOKEN:
ePUb(keyblk)
control information
other information

FIG. 8

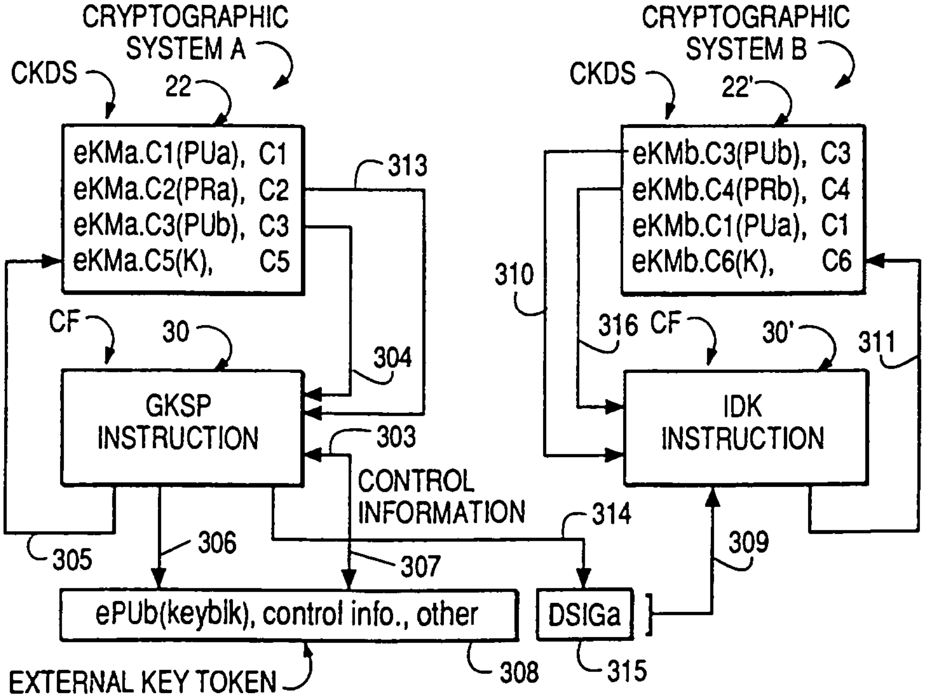

CRYPTOGRAPHIC SYSTEM A
CKDS
22
eKMa.C1(PUa), C1
eKMa.C2(PRa), C2
eKMa.C3(PUb), C3
eKMa.C5(K), C5
313
CF
30
304
GKSP INSTRUCTION
303
CONTROL INFORMATION
305
306
307
314
ePUb(keyblk), control info., other
DSIGa
EXTERNAL KEY TOKEN
308
315
CRYPTOGRAPHIC SYSTEM B
CKDS
22'
eKMb.C3(PUb), C3
eKMb.C4(PRb), C4
eKMb.C1(PUa), C1
eKMb.C6(K), C6
310
316
CF
30'
311
IDK INSTRUCTION
309

FIG. 9

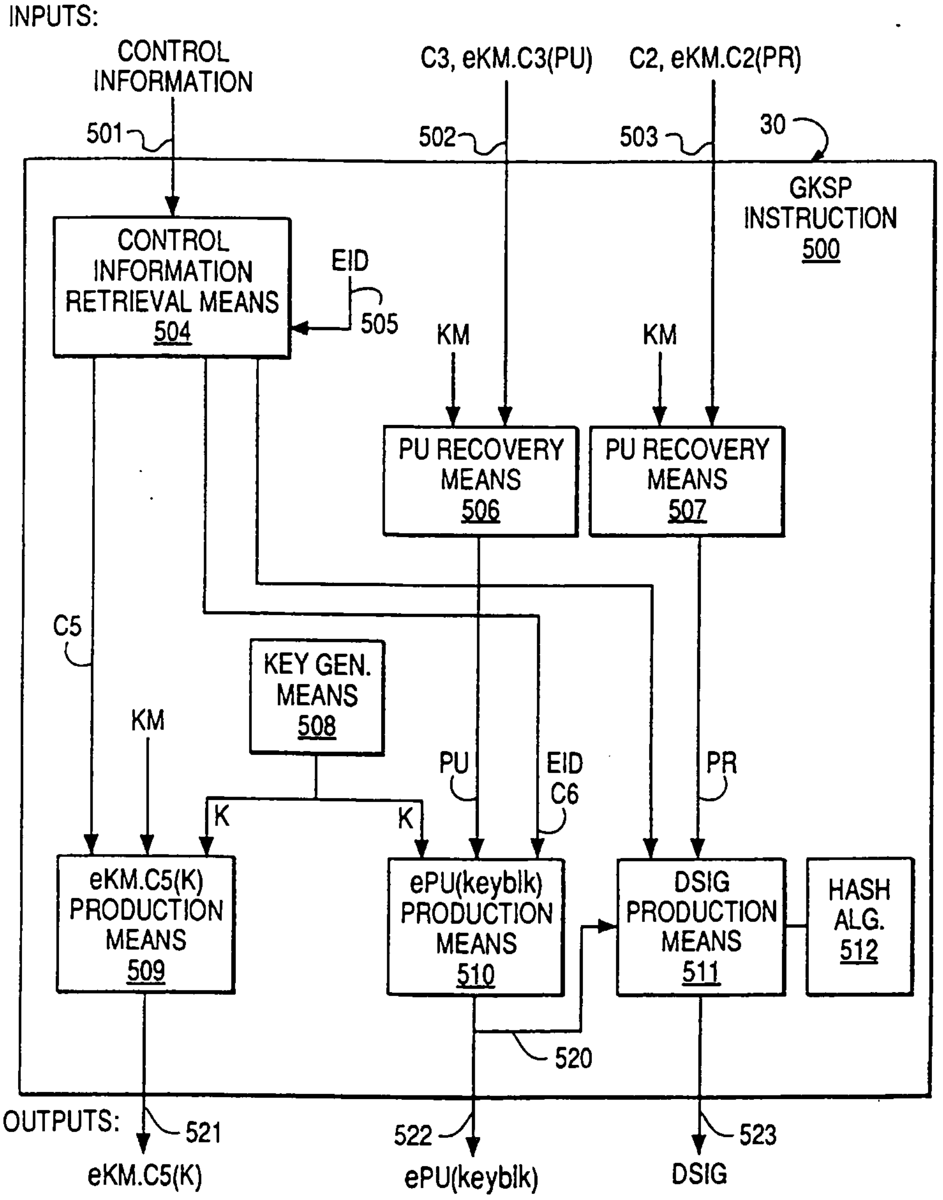

INPUTS:
CONTROL INFORMATION
C3, eKM.C3(PU)
C2, eKM.C2(PR)
501
502
503
30
GKSP INSTRUCTION 500
CONTROL INFORMATION RETRIEVAL MEANS 504
EID
505
KM
KM
PU RECOVERY MEANS 506
PU RECOVERY MEANS 507
C5
KM
KEY GEN. MEANS 508
K
K
PU
EID
C6
PR
eKM.C5(K) PRODUCTION MEANS 509
ePU(keyblk) PRODUCTION MEANS 510
DSIG PRODUCTION MEANS 511
HASH ALG. 512
520
OUTPUTS:
521
522
523
eKM.C5(K)
ePU(keyblk)
DSIG

FIG. 10

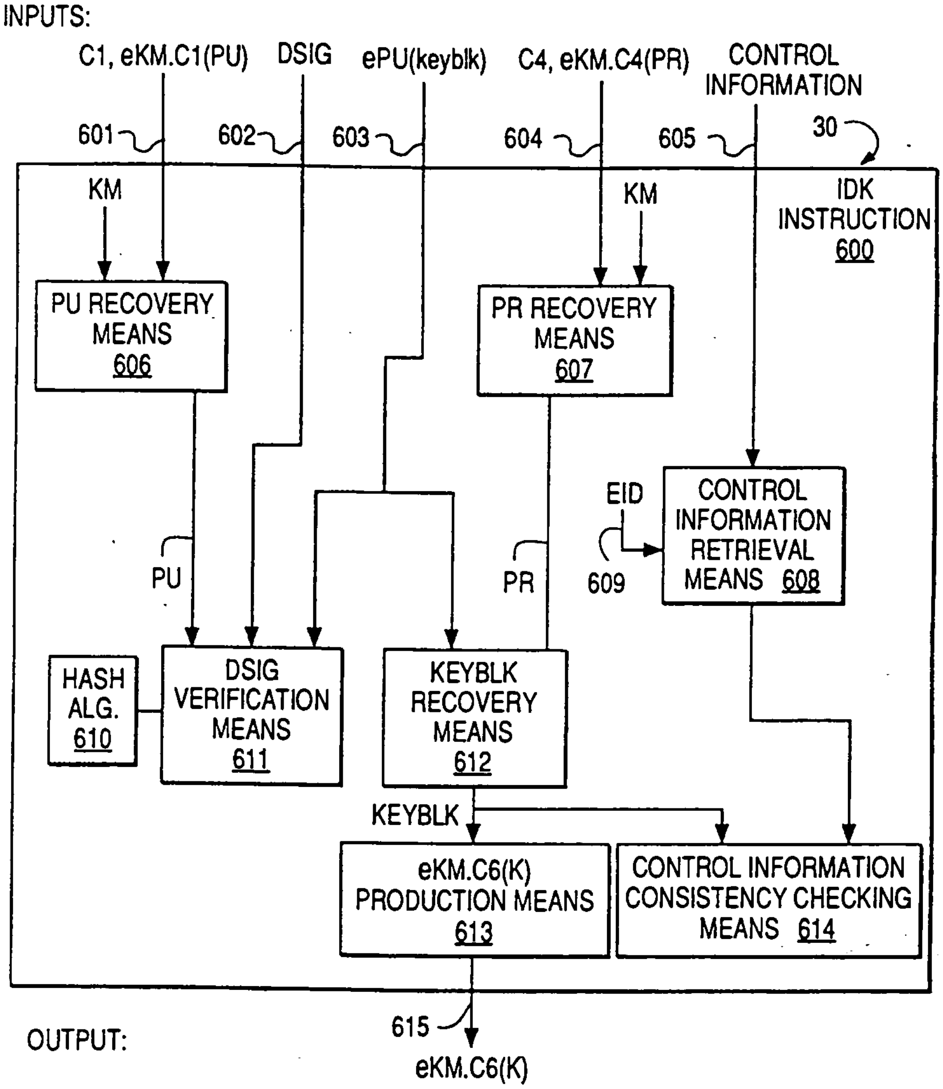
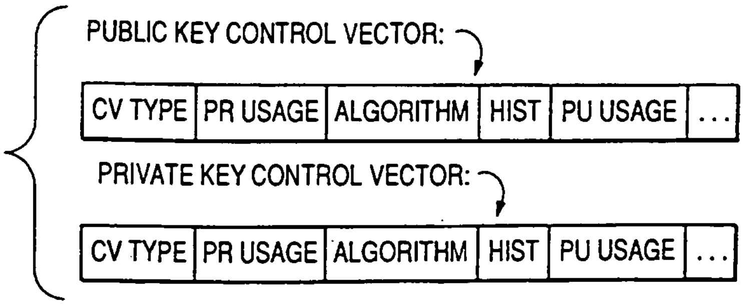

INPUTS:
C1, eKM.C1(PU)
DSIG
ePU(keyblk)
C4, eKM.C4(PR)
CONTROL INFORMATION
601
602
603
604
605
30
KM
KM
IDK INSTRUCTION 600
PU RECOVERY MEANS 606
PR RECOVERY MEANS 607
CONTROL INFORMATION RETRIEVAL MEANS 608
EID
609
PU
PR
HASH ALG. 610
DSIG VERIFICATION MEANS 611
KEYBLK RECOVERY MEANS 612
KEYBLK
eKM.C6(K) PRODUCTION MEANS 613
CONTROL INFORMATION CONSISTENCY CHECKING MEANS 614
OUTPUT:
615
eKM.C6(K)

FIG. 11



PUBLIC KEY CONTROL VECTOR:

CV TYPE	PR USAGE	ALGORITHM	HIST	PU USAGE	...

PRIVATE KEY CONTROL VECTOR:

CV TYPE	PR USAGE	ALGORITHM	HIST	PU USAGE	...

FIG. 12

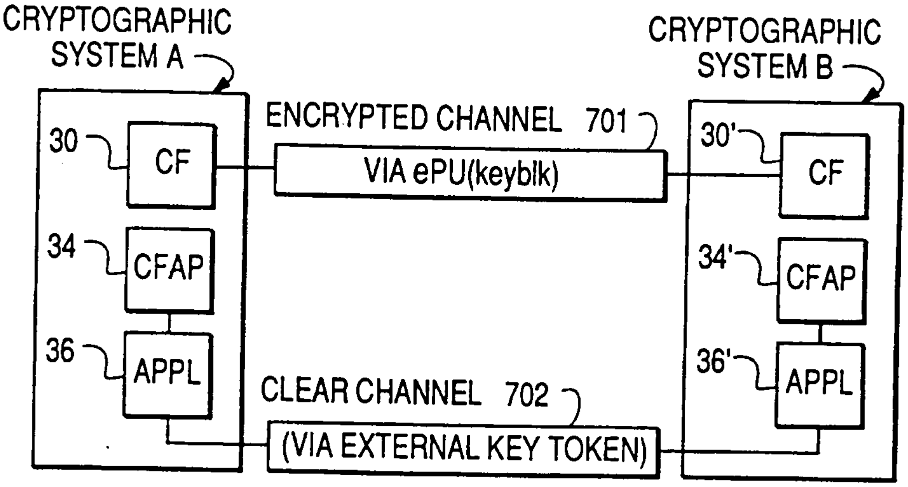

CRYPTOGRAPHIC SYSTEM A
CRYPTOGRAPHIC SYSTEM B
ENCRYPTED CHANNEL 701
VIA ePU(keyblk)
30
CF
34
CFAP
36
APPL
30'
CF
34'
CFAP
36'
APPL
CLEAR CHANNEL 702
(VIA EXTERNAL KEY TOKEN)

FIG. 13

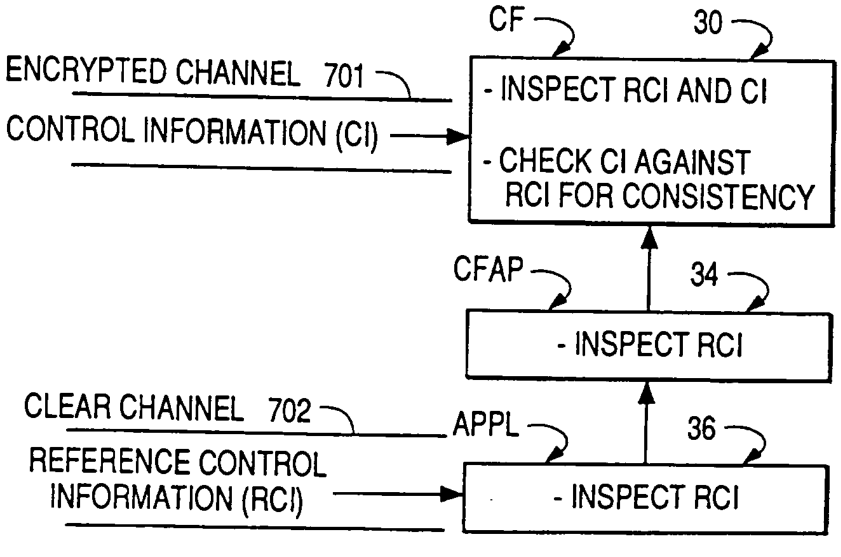

CF
30
ENCRYPTED CHANNEL 701
CONTROL INFORMATION (CI)
- INSPECT RCI AND CI
- CHECK CI AGAINST RCI FOR CONSISTENCY
CFAP
34
- INSPECT RCI
CLEAR CHANNEL 702
APPL
36
REFERENCE CONTROL INFORMATION (RCI)
- INSPECT RCI

FIG. 14

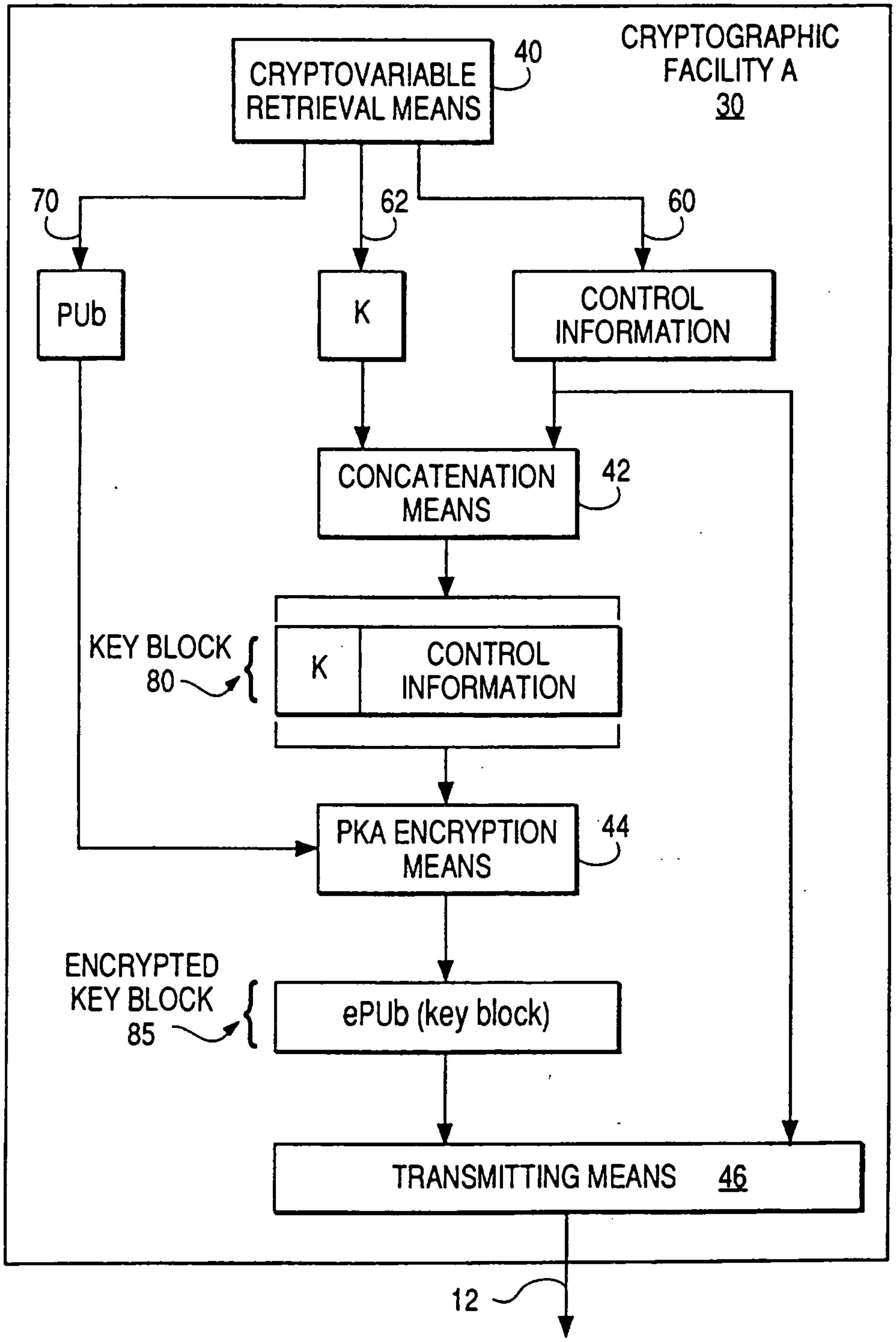

CRYPTOGRAPHIC FACILITY A
30
CRYPTOVARIABLE RETRIEVAL MEANS
40
70
62
60
PUb
K
CONTROL INFORMATION
CONCATENATION MEANS
42
KEY BLOCK
80
K
CONTROL INFORMATION
PKA ENCRYPTION MEANS
44
ENCRYPTED KEY BLOCK
85
ePUb (key block)
TRANSMITTING MEANS 46
12

FIG. 15

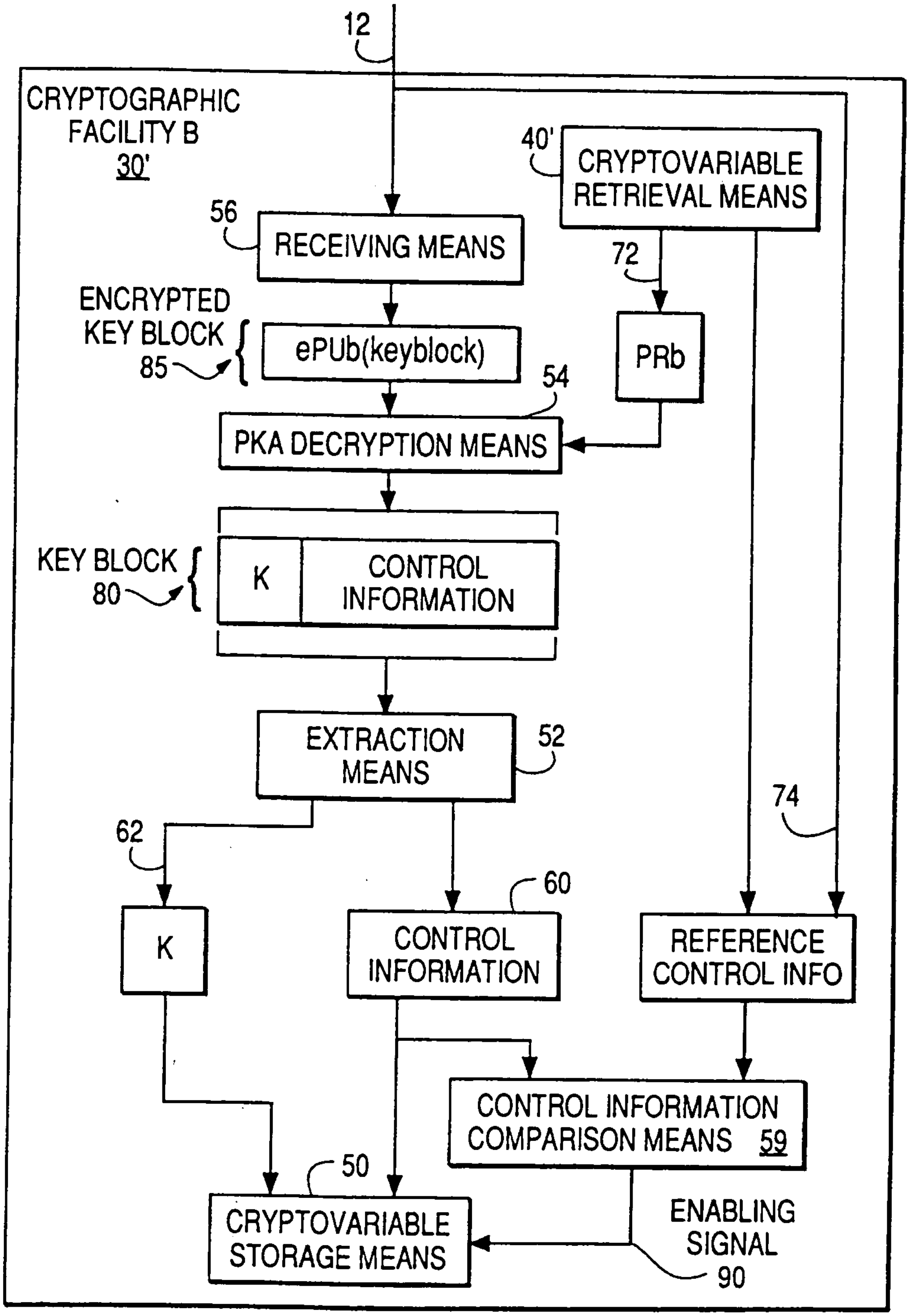
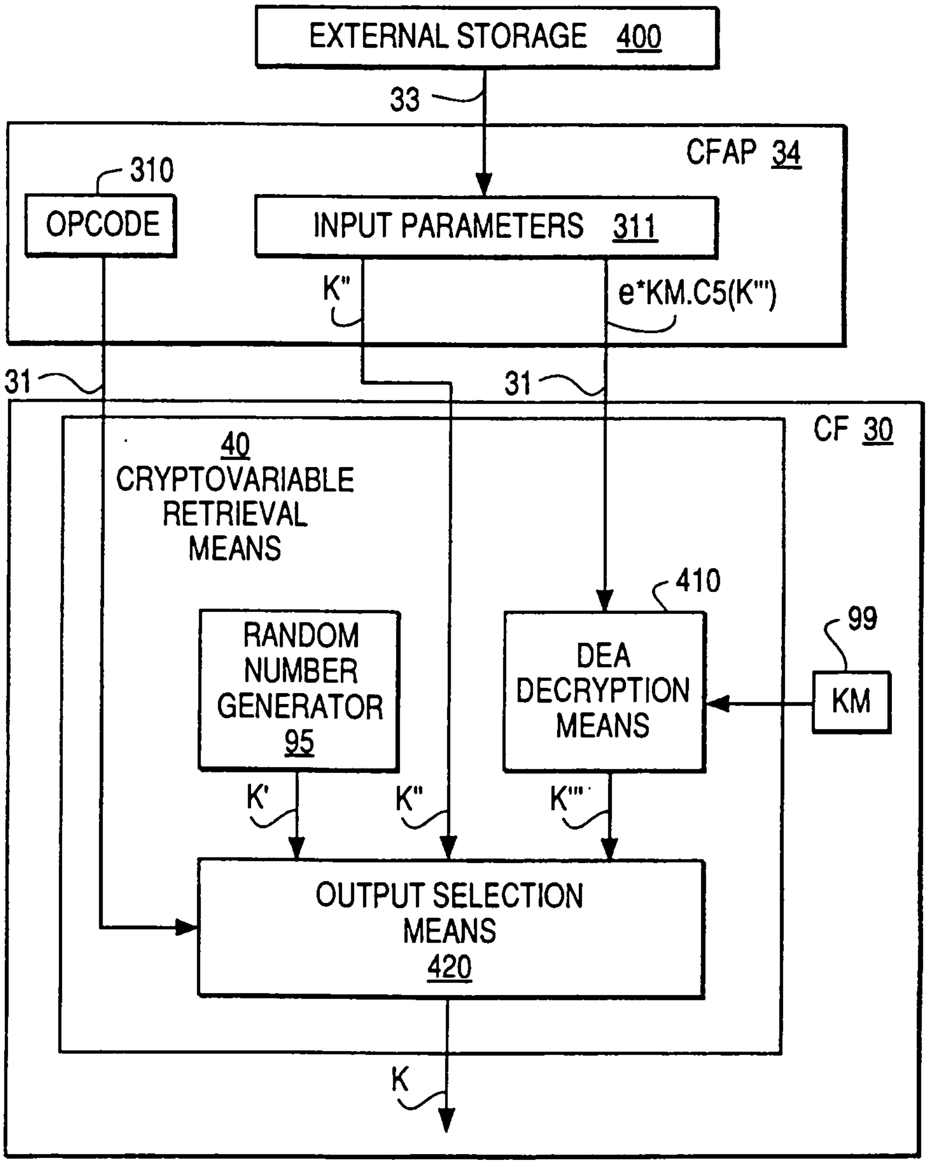

12
CRYPTOGRAPHIC FACILITY B
30'
40'
CRYPTOVARIABLE RETRIEVAL MEANS
56
RECEIVING MEANS
72
ENCRYPTED KEY BLOCK
85
ePUb(keyblock)
PRb
54
PKA DECRYPTION MEANS
KEY BLOCK
80
K
CONTROL INFORMATION
EXTRACTION MEANS
52
74
62
60
K
CONTROL INFORMATION
REFERENCE CONTROL INFO
CONTROL INFORMATION COMPARISON MEANS 59
50
CRYPTOVARIABLE STORAGE MEANS
ENABLING SIGNAL
90

FIG. 16


EXTERNAL STORAGE 400
33
CFAP 34
310
OPCODE
INPUT PARAMETERS 311
K"
e*KM.C5(K"')
31
31
CF 30
40
CRYPTOVARIABLE
RETRIEVAL
MEANS
410
99
RANDOM
NUMBER
GENERATOR
95
DEA
DECRYPTION
MEANS
KM
K'
K"
K"'
OUTPUT SELECTION
MEANS
420
K

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 April 2004 (22.04.2004)

PCT

(10) International Publication Number
WO 2004/034223 A2

(51) International Patent Classification^7: G06F

(21) International Application Number: PCT/US2003/032153

(22) International Filing Date: 8 October 2003 (08.10.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/417,913 9 October 2002 (09.10.2002) US

(71) Applicant (*for all designated States except US*): **LEGAL IGAMING, INC.** [US/US]; 200 Ultra Drive, Henderson, NV 89074 (US).

(72) Inventors; and
(75) Inventors/Applicants (*for US only*): **SAUNDERS, Michael, W.** [US/US]; 200 Ultra Drive, Henderson, NV 89074 (US). **MILLER, William, D., III** [US/US]; 8740 Country Pines Avenue, Las Vegas, NV 89129 (US). **CARLSON, Rolf, E.** [US/US]; 211 Dartmouth Avenue S.E., Albuquerque, NM 87106 (US).

(74) Agent: **MALLON, Joseph, J.**; Knobbe, Martens, Olson & Bear, LLP, 2040 Main Street, 14th Floor, Irvine, CA 92614 (US).

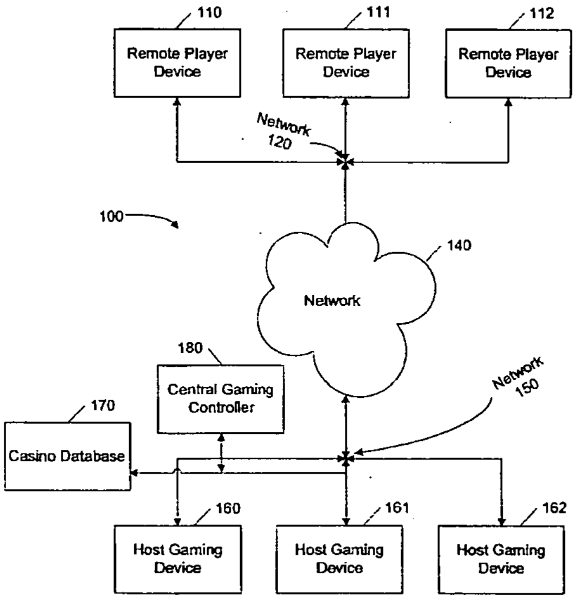
(81) Designated States (*national*): AE, AG, AL, AM, AT (utility model), AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ (utility model), CZ, DE (utility model), DE, DK (utility model), DK, DM, DZ, EC, EE (utility model), EE, EG, ES, FI (utility model), FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK (utility model), SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declaration under Rule 4.17:
— *of inventorship (Rule 4.17(iv)) for US only*

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR CONNECTING GAMING DEVICES TO A NETWORK FOR REMOTE PLAY



(57) **Abstract:** A system (100) and method for connecting remote player devices (110) to regulated host gaming devices (160) in a network to provide remote game play. A host gaming device (160) is configured to provide game information to a plurality of remote player devices (110) to allow remote play of the host game device (160). Whether each remote player device (110) is permitted to receive gaming data is based upon, at least in part, the geographic location of the remote player device (110).

WO 2004/034223 A2

Published:

- *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SYSTEM AND METHOD FOR CONNECTING GAMING DEVICES TO A NETWORK FOR REMOTE PLAY

Background of the Invention

Field of the Invention

[0001] The present invention generally relates to electronic devices. In particular, the invention relates to methods and systems of interactive gaming.

Description of the Related Technology

[0002] Traditionally, the way for a gaming operator to increase revenue from gaming devices is to increase the number of gaming devices available for play. In order for casinos to increase the number of gaming devices available for play, casino floor space must be added to house the additional gaming devices. The floor space allocated to house additional gaming devices must meet specific criteria as defined by the gaming authority for the jurisdiction in which the gaming devices are to be located. Providing additional floor space is an expensive process for casino operators and often requires constructing new casino properties. Also, adding gaming devices typically requires payment of additional licensing fees for each additional game.

[0003] A trend in the gaming industry has been to provide Internet gaming. Internet gaming allows players to make wagers on the outcome of casino style games similar to that described above, except that the player does not have to be physically located in a casino to do so. Internet players make wagers and play casino games using a personal computer and wager on games running on computers connected to the Internet.

[0004] More broadly, interactive gaming is the conduct of gambling games through the use of electronic devices. The popularity of Internet gambling sites has indicated a strong market for remotely accessible gaming, or other interactive gaming. Regulated casino operators strongly desire to provide interactive gaming while capitalizing on existing infrastructure. Thus there is a need for improved electronic devices that support regulated remote gaming.

Summary of the Invention

[0005] The system of the present invention has several aspects, no single one of which is solely responsible for its desirable attributes. Without limiting the scope of this invention as expressed by the claims which follow, its more prominent features will now be discussed briefly. After considering this discussion, and particularly after reading the section entitled "Detailed Description of the Invention" one will understand how the features of this invention provide advantages which include providing remote gaming in regulated environment.

[0006] A gaming system and method of using the same to allow a host gaming device to be played from remote player devices to allow casino operators to obtain maximum advantage from their gaming licenses.

[0007] More particularly, in one embodiment gaming system may comprise a data network, a host gaming device connected to the data network, the gaming device configured to execute at least one game and a plurality of remote player devices connected to the data network. Each of the remote player devices is configured to receive game information provided by the host gaming device. Whether each remote player device is permitted to receive gaming data may be based upon, at least in part, the geographic location of the remote player device.

[0008] The host gaming device may be configured to allow no more than a predetermined number of remote player devices to concurrently receive game information provided by the host gaming device during the gaming session. This predetermined number may be determined by a gaming agency.

[0009] In another embodiment of a gaming system, at least one of the plurality of remote player devices may be permitted to receive game data based upon, at least in part, the geographic location of the remote player device, an age of a user of the remote player device.

[0010] A gaming system according to the invention may also include a central gaming controller configured to record gaming transactions on the host gaming device and on each remote gaming device.

[0011] The data network may be, in part, the Internet, and be comprised of one or more logical segment, which may include closed-loop networks. The host gaming device may be configured to identify the geographic location of a remote player device based, at least in part, on a logical segment corresponding to the remote player device. A mobile communications network, or a GPS device may also allow identification of the geographic location of the remote player device.

[0012] The host gaming device may be in a location approved by a gaming agency and include at least one game control configured to provide local use. This game control may be disabled when the host gaming device is providing game information to a remote player device. A host gaming device may also be configured to save an encrypted game state allowing a game to be resumed following a device or network failure.

[0013] A remote player device may be coupled to a credential device configured to receive information relating to a user of the remote player device. The information relating to a user may include the age of the user, or a password that is input by the user. The credential device is a smart card reader, a biometric device such as a fingerprint reader, or any type of input device. The credentials may be verified against information, such as age, password, or fingerprint in a database configured to provide information associated with each of a plurality of users of the gaming system.

[0014] In another embodiment, a gaming system may be comprised of a means for executing at least one game, the game providing game information during its execution, a local access means provides local access to the game information for a user in a location approved by a gaming agency, player means for receiving game information, presenting the game information to a user and providing at least one game control, a means for providing the game information over a data network to a predetermined number of receiving means, means for determining the location of the receiving means, and means for disabling the local access means. Other similar embodiments may also be comprised of means for creating an auditable record of gaming transactions on the playing means and on the gaming means.

[0015] Another embodiment of a gaming system, in addition to the features of the embodiments discussed above, may also include customized promotional messages to players of gaming devices.

[0016] On a remote player device, an embodiment of a method of remotely accessing a host gaming device may include: establishing access to the host gaming device through a data network, receiving gaming related information from the host gaming device through the data network, presenting the gaming related information to a player, receiving at least one control signal from the player, sending the control signal to the host gaming device through the data network, and disabling local use of the host gaming device. In one embodiment, the method may also include recording each gaming transaction occurring on the remote player device. Another embodiment of the method may include providing a geographic location of the remote player device. In another embodiment of the method, the age of the user of the remote player device is also provided.

[0017] On a host gaming device, an embodiment of a method of providing remote access, including: verifying the geographic location of a remote player device, establishing a gaming session on a host gaming device from a remote player device through a data network, receiving at least one control signal from the remote player device through the data network, and sending gaming related information from the gaming device through the data network. One embodiment of a method may also include recording each gaming transaction occurring on the host gaming device,

[0018] In order to provide tolerance for failures of system components, a method of resuming an interrupted gaming session on a gaming device is provided. One embodiment of a method may include generating a gaming state of the gaming session on the first gaming device, encrypting the gaming state, transporting the encrypted gaming state from the gaming device. The method may also include the converse: transporting the encrypted gaming state from the first gaming device to a second gaming device, decrypting the gaming state on the second gaming device; and loading the game state into a second gaming device to resume the gaming session.

[0019] An embodiment of a gaming system which provides for resuming interrupted gaming sessions across a data network. The system may include a first host gaming device connected to the data network, the gaming device configured to execute at least one game, generate a gaming state based on execution of at least one game, encrypt the gaming state, and send the encrypted gaming state over the data network. A second host gaming device may be connected to the data network, the second gaming device configured to receive the encrypted gaming state over the data network, decrypt the gaming state, and resume executing at least one game from the gaming state. A plurality of remote player devices, configured to receive game information provided by the host gaming device, may be connected to the data network. The gaming state may include user payment or credit information, and game jackpot or payout information.

[0020] Another embodiment of a gaming system providing resumption of interrupted gaming sessions may include means for executing at least one game, means for generating a gaming state based on execution of at least one game, means for encrypting the gaming state, and means for sending the encrypted gaming state. The system may also include means for receiving the encrypted gaming state, means for decrypting the gaming state and means for resuming executing at least one game from the gaming state.

[0021] To enable gaming regulatory compliance, methods authenticating gaming system users are also provide. An embodiment of a method of authenticating a user of a host gaming device may include receiving a security certificate from the smart card, sending the security certificate from the gaming device to an authenticator device, receiving an authentication reply from the authenticator, and playing a game in response to the authentication reply.

[0022] An embodiment of the method may also include presenting the security certificate from the gaming device to a certificate authority for authentication over a data network.

[0023] An embodiment of a method of authenticating a user of a remote player device for playing a host gaming device may include receiving an indicia of identity for a user, sending the indicia of identity to an authenticator device, receiving an authentication reply from the authenticator device, and authorizing use of a host gaming device based on the indicia of identity. The indicia of identity for a user may be provided by a biometric device, a smart card, or a password provided by the user.

[0024] Another embodiment of a gaming system provides authentication of users. The system may include a data network, a host gaming device interfaced to the data network, a plurality of remote player devices interfaced to the data network, and a security device configured to provide player credentials to at least one remote player device. The each of the remote player devices may be configured to receive game information provided by the host gaming device. The host gaming device may provide game information to a predetermined number of permitted remote player devices. Whether a remote player device is permitted to receive gaming information may be based upon, at least in part, on player credentials provided by the security device.

[0025] In one embodiment, a method of remotely accessing a gaming device provides for creating records of gaming transactions on both host gaming devices and remote player devices sufficient to provide an auditable record for a gaming authority in the jurisdiction. The method may include establishing a gaming session on a gaming device for a remote player device through a data network, sending gaming related information from the gaming device through the data network, receiving at least one control signal from the remote player device through the data network, creating an auditable gaming session record representing each gaming transaction of a gaming session on the host gaming device and on the remote gaming device. In addition, the record may be sent to a third party, such as a gaming authority, through the data network.

[0026] In another embodiment of a gaming system, the gaming system includes a network comprised of a plurality of logical segments. A security policy controls the flow of data between logical segments. A host gaming device may be connected to the data network, the gaming device configured to execute at least one game. A plurality of remote player devices may be connected to the data network. The plurality of remote player devices are each configured to receive game information provided by the host gaming device, and to control a gaming session established on the gaming device, subject to the security policy. The security policy may be based, at least in part, on the geographic location of a logical segment.

[0027] One embodiment of the gaming system may include a promotional message server to deliver customized promotional messages to users of the gaming system. In this embodiment, a gaming system may include a data network, a promotional message server configured to provide customized promotional messages. Each message may be customized with information associated with a user of the gaming system. In addition, a gaming system may include a host gaming device interfaced to the data network, and a plurality of remote player devices interfaced to the data network. The plurality of remote player devices are each configured to receive game information provided by the host gaming device and to receive and present promotional messages.

[0028] In another embodiment, a gaming system may include a means for data communication, means for executing at least one game, means for providing game information over the data network to a predetermined number of receiving means, a plurality of means for receiving game information over the data communication means. Each means for receiving game information may be coupled to a means for receiving customized promotional messages. A gaming system may also include a means for presenting promotional messages in conjunction with gaming data.

[0029] A related method of displaying information on a remote player device is also provided. The method may include receiving a promotional message on a remote player device, presenting the promotional message in conjunction with gaming information for an amount of time; and removing the promotional message from the remote player device. Information in the promotional message may be used to calculate the amount of time to present the promotional message.

[0030] A remote player interface of a gaming system may have a number of embodiments. In one embodiment of a gaming system, the gaming system includes data network, a host gaming device interfaced to the data network, and at least one remote player device interfaced to the data network. The remote player device is configured to receive game information provided by the host gaming device. The remote player interface of the gaming system may include a video display device in communication with the remote player device and a remote control device in communication with the remote player device. The remote control device is configured to control operation of a game.

[0031] An embodiment of method of remotely accessing a gaming device may include establishing a gaming session on the host gaming device from a remote player device through a data network, receiving gaming related information from the host gaming device through the data network, presenting gaming related information to a player via a video display device, receiving at least one control signal generated by a remote control device for controlling the gaming session, and sending the control signal to the host gaming device through the data network.

Brief Description of the Drawings

[0032] **FIG. 1** depicts a simplified block diagram of a gaming system according to one embodiment of the invention.

[0033] **FIG. 2** depicts a simplified block diagram of system elements relating to a host gaming device of **FIG. 1** according to one embodiment of the invention.

[0034] **FIG. 3** depicts a simplified block diagram of system elements relating to a remote player device of **FIG. 1** according to one embodiment of the invention.

[0035] **FIG. 4** is a flowchart depicting the sequence of events for acknowledging command messages in a gaming system as embodied in **FIG. 1.**

[0036] **FIG. 5** is a flowchart depicting the sequence of events for establishing a remote gaming session, playing a game, and terminating the remote gaming session in a gaming system as embodied in **FIG. 1.**

[0037] **FIG. 6** is a flowchart depicting the sequence of events for transferring funds from a player's source of funds in the gaming system of **FIG. 1.**

[0038] **FIG. 7** is a flowchart depicting the sequence of events for a host gaming device of **FIG. 2** to connect to a network using security certificates and a certificate authority.

[0039] **FIG. 8** is a flowchart depicting the sequence of events for a gaming device of **FIG. 2** to build and deliver an encrypted block of data representing the complete state of the gaming device.

[0040] **FIG. 9** is a flowchart depicting the sequence of events for retrieving a block of data representing the state of a gaming device from a database and loading the block into a gaming device as performed by a gaming system embodiment as in **FIG. 1**.

[0041] **FIG. 10** is a more detailed block diagram of a gaming system as depicted in **FIG. 1**.

[0042] **FIG. 11** is a detailed block network diagram of a portion of a gaming system as depicted in **FIG. 10**.

Detailed Description of the Preferred Embodiment

[0043] The following detailed description is directed to certain specific embodiments of the invention. However, the invention can be embodied in a multitude of different ways as defined and covered by the claims. In this description, reference is made to the drawings wherein like parts are designated with like numerals throughout.

[0044] In a traditional casino environment, gaming devices are generally located on a gaming floor. Gaming devices are subject to regulation by gaming regulatory agencies. Regulations may limit the locations where gaming devices may be placed and by limit users of gaming devices to those of legal age to gamble in the respective jurisdiction. Regulatory agencies for a given jurisdiction may also limit the number of licensed gaming devices provided to a licensee. Where gaming devices are physically located on a casino gaming floor, verification of whether a device is being used in its licensed location within the jurisdiction may be determined by physical inspection of the gaming floor. Further, monitoring of the gaming floor in casinos ensures that players are of legal age as set by the jurisdiction.

[0045] An embodiment of a gaming system according to the present invention allows a licensed host gaming device to be used by one or more remote player devices geographically separated from the host gaming device, but still located within the jurisdiction of a gaming authority. **FIG. 1** depicts a simplified block diagram of an embodiment of a gaming system **100** according to the invention. One or more host gaming devices **160, 161, 162** are licensed gaming devices. Although three host gaming devices are shown on **FIG. 1**, the gaming system **100** may employ any number of host gaming devices ranging from one to thousands. For convenience of discussion, set forth below is a description of certain aspects of the host gaming device **160**. It is to be appreciated that the other gaming devices may contain the following or different aspects.

[0046] A host gaming device may be any device, comprised of electronic, mechanical, or a combination of electronic and mechanical components, which is used for gaming and which affects the result of a wager by determining win or loss. A host gaming device **160** is connected to a data network **150**. In the embodiment depicted in **FIG. 1**, the data network of gaming system **100** is comprised of three logical segments. Gaming network **150** connects each host gaming device **160** and related elements such as the database **170** and central gaming controller **180**. Remote network **120** connects remote player devices **110**, **111**, **112** to the system. Backbone network **140** provides interconnection between the gaming network **150** and the remote network **120**.

[0047] The database **170** may be computer server running database software, or any other commercially available database solution. In one embodiment, as depicted, the database **170**, is a casino database. In other embodiments, the database may also contain other data related, or unrelated to the casino operation.

[0048] Remote network **120** connects remote player devices **110**, **111**, **112** to the system. Each remote player device **110** allows a user to play a game executing on a host gaming device **160**. For convenience of discussion, set forth below is a description of certain aspects of the remote player device **110**. It is to be appreciated that the other remote player devices may contain the following or different aspects. Although three remote player devices are shown on **FIG. 1**, the gaming system **100** may employ any number of remote player devices ranging from one to thousands.

[0049] The remote network **120** may be any form of computer network, as discussed below. In one particular embodiment, the remote network **120** is part of a network provided by a cable television system. **FIG. 10** depicts an embodiment of a gaming system where the remote network **120** is provided through a digital home communications terminal (DHCT) **1000**, such as a set-top box.

[0050] Each host gaming device **160** may be located in any location approved by a gaming agency, such as a casino gaming floor. A host gaming device **160** provides a legally regulated random number generator. Once generation of random number has been performed, a game result is determined. Any further interaction through the game's user interface is for the benefit of a user. For example, in one embodiment of a gaming system, the host gaming device may be a slot machine. After payment is made, through a coin, token, credit device, etc, the player pulls a lever arm to execute play. In a mechanical game, for example, a slot machine, a game result may be determined by the interaction of spinning wheels. In a host gaming device **160** of an embodiment of the present invention, however, pulling the arm triggers generation of a random number which determines the game result. Thus any spinning wheels or its electronic equivalent is purely for entertainment of the user. A host gaming device **160** plays at least one game of chance, including, but not limited to, Slots, Blackjack, Poker, Keno, Bingo, or Lotteries.

[0051] **FIG. 2** depicts a more detailed block diagram of an embodiment of a gaming system **100** showing additional gaming system elements coupled to the host gaming device **160**. The host gaming device **160** may include local controls **220** such as an arm. The host gaming device **160** may have a display **210** to present the results of a game to a user. Further, the gaming device **160** may have a smart card reader **280**. Functions of the smart card reader **280** may include receiving payment for a game, or identifying a user for promotional or loyalty programs. A biometric identity device **290**, such as a fingerprint scanner, may be used for similar functions by the gaming system.

[0052] Networks **120, 140, 150** may include any type of electronically connected group of computers including, for instance, the following networks: Internet, Intranet, Local Area Networks (LAN) or Wide Area Networks (WAN). In addition, the connectivity to the network may be, for example, remote modem, Ethernet (IEEE 802.3), Token Ring (IEEE 802.5), Fiber Distributed Datalink Interface (FDDI) Asynchronous Transfer Mode (ATM), Wireless Ethernet (IEEE 802.11), or Bluetooth (IEEE 802.15.1). Note that computing devices may be desktop, server, portable, hand-held, set-top, or any other desired type of configuration. As used herein, the network includes network variations such as the public Internet, a private network within the Internet, a secure network within the Internet, a private network, a public network, a value-added network, an intranet, and the like. In embodiments of the present invention where the Internet is the backbone network **140**, gaming network **150** and remote network **120** may form a virtual private network (VPN) transported over the Internet.

[0053] In preferred embodiments, the remote network **120** may be a closed-loop network, such as the cable network depicted in **FIG. 10**. A closed-loop network **120** may have a limited geographic scope which allows the geographic location of a remote player device **110** to be identified. For example, a given cable network may be limited to a specific hotel. Each hotel room may be provided with a remote player device **110** which may then be identified with that location. In other embodiments, the remote network **120** may be a mobile telephone network which is capable of identifying a caller's geographic location.

[0054] As depicted in the simplified block diagram of **FIG. 3**, a remote player interface **300** may comprise a remote player device **110**, a display **310** for presenting game information and a control **320** to provide user game control for the remote player device **160**. In one embodiment, a remote player interface **110** may also comprise a remote control **395** to provide game controls. In preferred embodiments of the remote control, the connection **394** between the remote control **395** and the remote player device **160** may be any type of wireless connection, including infra-red based protocols, or a RF wireless protocol such as Bluetooth (802.15.1). The remote control **395** may also be connected to the remote player device **160** through a wired connection such as Universal Serial Bus (USB), serial, or equivalent connection. The remote control **395** may also include controls customized for gaming. A handheld computer may also comprise a remote control **395**.

[0055] The display **310** may be a television, a personal computer, or a handheld computer device. A fixed or wireless telephone handset may comprise a display **310** and controls **320** of a remote player interface. In some embodiments the controls **320** may be integrated with display **310**, as for instance, in a touch screen.

[0056] In one embodiment, the game information may be a random number which represents the result of the game, information related to gaming device jackpots, or player credits. In another embodiment, the gaming information may be multimedia, sound and images, including, in one embodiment, video, representing the execution of a game. In another embodiment, game information may also be software for execution on a remote player device **110** or on any element of a remote player interface **300**, such as a remote control **395**, which interactively presents the game through the remote player interface **300**.

[0057] To enable regulatory conformance of the gaming system, gaming device users must be geographically within an approved jurisdiction and of legal age in the jurisdiction. In a regulated gaming environment, such as a gaming floor, physical control of the premises allows enforcement of this requirement. For remote player devices **110** not operated in the regulated gaming environment of a gaming floor, the age of the user of a remote player device **110** must be verified before game information is provided by a host gaming device **160**. Credentials may be received from a user using a variety of security devices and compared to records, such as in a database **170** to confirm identity and thus age of the user.

[0058] To ensure compliance with regulatory requirements, a gaming system **100** may identify the geographic location of a remote player device **110**. As discussed above, a network **120** may be a closed-loop network **120** whose devices are thereby identified in geographic location by the location of that network. Other embodiments may employ a GPS system on the remote player device **110** to provide the geographic location of the device **110**. In other embodiments, the remote network **120** may be a mobile communications network which provides the geographic location of network clients, such as a remote player device **110**.

[0059] In one embodiment, a security device may be a smart card reader **380** that is coupled to the remote player device **110**. In embodiments using a smart card reader, a user inserts a smart card into the reader which provides credentials sufficient to verify the age of the user. In one such embodiment, indicia present on the smart card reader are compared to records in a casino database **170** to verify the age of the user.

[0060] In other embodiments, a remote player device **110** may be coupled to a biometric identity device **390**, such as a fingerprint scanner. In one embodiment, information received from the biometric identity device **390** may be compared to records in a casino database **170** to verify the age of the user. In other embodiments a biometric identity device **390** may be retinal scanner or facial recognition device.

[0061] In some embodiments, the controls **320** may include an input device (not pictured in **FIG.** 3) coupled to a remote player device **110** to receive a password or PIN as a security device. The password or PIN may be compared to information, such as records in a casino database **170** to verify the identity, and thus the age, of the remote player device user. For example, the input device may be a keyboard, rollerball, pen and stylus, mouse, or voice recognition system. The input device may also be a touch screen associated with an output device. The user may respond to prompts on the display by touching the screen. The user may enter textual or graphic information through the input device. The controls **320** may be coupled to a display **310** in the form of a personal computer, a television, a television with a set-top box, a handheld computer, or a telephone, fixed or mobile, handset.

[0062] Embodiments of a remote player device **110** may be a television, a cable interactive set-top box, a remote control, a personal computer, or a mobile or fixed telephone handset. Another embodiment may comprise a handheld computer coupled to a fixed or preferably wireless network. Also, a host gaming device **160** may also be a remote player device **110**.

[0063] In one embodiment, a remote gaming device **110** may be in a location approved by a gaming agency with controls **320** and display **310** which match the appearance of a stand-alone gaming device. For example, a remote gaming device **110** may be appear to be a slot machine with an arm control **320**, a mechanical or electronic "slots" display **310**. In other embodiments, remote gaming devices **110**, regardless of location, may have controls and displays which match the appearance of a host gaming device **160**. This may include control devices coupled to personal computers or set-top boxes which may be customized for one or more games.

[0064] Indicia of identity and age received from a smart card reader **380**, biometric identity device **390**, or user entry of a password may also be compared to records stored on the remote player device **110**. For example, a remote player device **110** in a hotel room may be programmed by hotel staff to store identification information for eligible guests in the room containing the gaming device without the identification information being included in the casino database **170**. In these embodiments, access to the remote player device thus may itself be an indicium of legal age to the central gaming controller **180** or host gaming device **160**.

[0065] A central gaming controller **180** may manage the interaction of remote player devices and host gaming devices. The central gaming controller **180** may comprise one or more server computers or may be integrated with a host gaming device. In the embodiment depicted in **FIG. 10**, the application server **1027** and request processing servers **1023** comprise the central gaming controller **180**.

[0066] One embodiment of a gaming system **100** comprises a single remote player on a remote player device **110** establishing a gaming session on a host gaming device **160** with no local player using the host gaming device **160**. In this embodiment, the local controls **220** of a host gaming device **160** become disabled for local play during the remote gaming session. Correspondingly, a host gaming device **160** in this embodiment also becomes unavailable for remote play while a player uses the local controls **220** to use the host gaming device **160**.

[0067] Another embodiment comprises a single player using the local controls **220** of a host gaming device **160** and a single remote player on remote player device **110** concurrently. Thus in this embodiment, the local game controls **220** on the host gaming device **160** are not disabled during the remote gaming session.

[0068] Another embodiment of the gaming system **100** comprises a single local player of the host gaming device **160** and multiple remote players on a plurality of remote player devices **110** having concurrent gaming sessions. A similar embodiment comprises multiple concurrent remote players and no local players on the host gaming device **160** because the local controls **220** may be disabled during the remote gaming sessions.

[0069] Another embodiment of a gaming system **100** comprises one or more remote player devices **110** which are physically located in a location approved by a gaming agency and networked to a host gaming device **160** that hosts both local and remote player sessions. Players physically located in the casino may occupy a remote player device **110** and play the games provided by the host gaming device **160**. Concurrently, gaming sessions to one or more remote player devices **110** physically located outside the casino may be provided. Thus, in this embodiment, players may concurrently play using the host gaming device **160**, a physically remote player device **110**, or a remote player device **110** in a location approved by a gaming agency.

[0070] Another embodiment of the invention comprises one or more remote player devices **110**, physically located in a location approved by a gaming agency and at least one host gaming device **160**. In this embodiment, player sessions may only be established on a host gaming device **160** from a remote player device **110** if that remote player device **110** is physically located in a location approved by a gaming agency, such as a casino gaming floor. Players may also play the host gaming device **160** using local controls **220** concurrently with remote player sessions.

Thus, in this embodiment, players may concurrently play using the host gaming device **160**, or a remote player device **110** that is located in a location approved by a gaming agency.

[0071] In each of the above disclosed embodiments, the remote player devices **110** that may concurrently receive game information from a host gaming device **160** may be limited to a predetermined number that is determined by a regulatory gaming agency for the jurisdiction.

[0072] A remote player device **110** that is physically located in the casino in a location approved by a gaming agency, such as a casino gaming floor, may differ from a remote player device physically located outside the casino floor. In one embodiment, a remote player device **110** located in a location approved by a gaming agency resembles the appearance of a stand-alone gaming device and may thus be similar in appearance and operation to the host gaming device **160**.

[0073] In one embodiment, a remote player device **110** requests game data from the host gaming device **160** by sending a request for a game to a central gaming controller **180**. The central gaming controller **180** then transmits the request for a game to the host gaming device **160**. The host gaming device **160** receives the request and provides game data to the central gaming controller **180** that passes to the remote player device **110**. That information is then translated into a game by the remote player device **110** and displayed or performed to the player. The remote player device **110** may contain on-board hardware and software that may be required to present a game. The regulated portion of hardware and software required to execute a game, such as a random number generator, is on the host gaming device **160** and the information transmitted to the remote player device **110** each time a game is requested.

[0074] Gaming devices according to an embodiment of the invention may use mixed-protocol delivery systems for game content and game results. Game information and results comprising image and sound data may be delivered by packet based network protocols such as IP datagrams, by connection-oriented network protocols, or by a combination of both. Streaming media protocols may also be employed. During a given gaming session, these communication methods may be used interchangeably or concurrently.

[0075] In one embodiment, communication over the data networks **120**, **140**, or **150**, may use IP datagrams to package image and sound data comprising a host gaming device interface and display, encrypts it, and delivers it to the remote player device.

[0076] Internet Protocol (IP) is a network layer protocol used by many corporations, governments, and the Internet worldwide. IP is a connectionless network layer protocol that performs addressing, routing and control functions for transmitting and receiving datagrams over a network. The network layer routes packets from source to destination. An IP datagram is a data packet comprising a header part and a data part. The header part includes a fixed-length header segment and a variable-length optional segment. The data part includes the information being transmitted over the network. As a connectionless protocol, IP does not require a predefined path associated with a logical network connection. Hence, IP does not control data path usage. If a network device or line becomes unavailable, IP provides the mechanism needed to route datagrams around the affected area.

[0077] The remote player interacts with a game through a remote player interface **300**. A remote player device **110** may send commands back to the central gaming controller **180** as, in one embodiment, IP datagrams. The IP datagrams are interpreted by the central gaming controller **180** and used to proxy user interface interaction between the gaming device and the remote player. Game results may also be packaged as IP datagrams and delivered to the remote player through this method.

[0078] Alternative embodiments may use connection-oriented protocols such as TCP, or a combination of connection oriented protocols and connectionless packet protocols such as IP. Transmission Control Protocol (TCP) is a transport layer protocol used to provide a reliable, connection-oriented, transport layer link among computer systems. The network layer provides services to the transport layer. Using a two-way handshaking scheme, TCP provides the mechanism for establishing, maintaining, and terminating logical connections among computer systems. TCP transport layer uses IP as its network layer protocol. Additionally, TCP provides protocol ports to distinguish multiple programs executing on a single device by including the destination and source port number with each message. TCP performs functions such as transmission of byte streams, data flow definitions, data acknowledgments, lost or corrupt data re-transmissions, and multiplexing multiple connections through a single network connection. Finally, TCP is responsible for encapsulating information into a datagram structure.

[0079] Static content comprising the game interface or other elements of the game may be delivered to the remote player device **110** and stored on the remote player device. This delivery of content may use a mixed-protocol as described above. A static image may be a fixed image or an animation activated by the remote control device. Such images may further be overlaid with additional game content such as images and sound that is delivered dynamically during game play.

[0080] In an embodiment of the invention, a central gaming controller **180** converts image and sound data comprising the gaming device interface and display from the remote machine into a data stream (for example but not limited to MPEG-2), encrypts it, and delivers it to the remote player device **110**. The remote player interacts with the game using the remote player interface **300** to send commands back to the central gaming controller as IP datagrams. The IP datagrams may be interpreted by the central gaming controller **180** and used to proxy user interface interaction between the gaming device **160** and the remote player device **110**. Game results may also be packaged as a data stream and delivered to the remote player through this method.

[0081] FIG. 4 is a flowchart depicting a method employed when a command message is acknowledged by a central gaming controller **180** according to one embodiment of a gaming system **100**. Depending on the embodiment, additional steps may be added, others removed, steps merged, or the order of the steps rearranged. Note that in some embodiments, not all messages received by the central gaming controller **180** need be acknowledged. Starting at step **401**, a command message is sent to the central gaming controller **180** by a host on the network. The host may be remote player device **110** used for remote play, or other authorized network devices. Next, at step **405**, a qualified request message is received by the central gaming controller **180**. Moving to step **410**, the message is then recorded in a database. The database may be a casino database **170**. Proceeding to step **415**, the message is processed and a response prepared. Next at step **420**, the response is recorded in the database. Moving to step **425**, the response is sent back to the requesting device. At step **430**, a test to determine whether an acknowledgment of the message has been received is made. Continuing at step **435**, if the timeout value has passed control continues to step **440**, if the timeout period has not expired control returns to step **430**. Moving to step **440**, whether the message has not been acknowledged by the originating host is tested. If acknowledgement has been received, control proceeds to **445**, if not control proceeds to step **455**. At step **445**, the message status is recorded as "RECEIVED" and the process moves to the end state. Returning to step **455**, where the process flow continues following an unacknowledged message, the system sends a status request message to the sending host. Next, at step **460**, if the originating device responds to the message then flow continues to step **465**, otherwise control moves to step **480**. Moving to step **465**, a diagnostic message is sent to query whether the originating device is ready to receive the original message. Next at step **470**, if the originating host responds that it is ready to receive the original message, then control transfers to step **425** but if the originating host fails to respond then control moves to step **480**. Moving to step **480**, the status of the originating host is set to offline until such time as the originating host can respond or reinitializes, and the process moves to the end state.

[0082] FIG. 5 is a flowchart depicting a method used when a request for a remote gaming session is received, when playing a game, and when terminating the remote gaming session. Depending on the embodiment, additional steps may be added, others removed, steps merged, or the order of the steps rearranged. Starting at **510**, a request for a remote gaming session is received as a request for a secured encrypted connection to the central gaming controller **180**. Included in the request are the remote players security credentials in the form of a security certificate, for example, X.509 certificate. Next at **515**, the security credentials are authenticated.

This authentication may be performed by submitting the security certificate to a certificate authority for authentication. Moving to **520** if the player is not authenticated, control reverts to **515**. Continuing to step **525**, the central gaming controller **180** establishes a secure encrypted connection with the remote player device **110**. Next, at step **530**, if required the player transfers funds to use during the remote gaming session. Continuing to step **535**, the player then chooses a host gaming device **160** to play. Next, at step **540**, in one embodiment, when a host gaming device **160** is chosen for remote access play the local controls of the host gaming device **160** is disabled to prevent local play. Moving on to step **545**, a remote play session is opened on the host gaming device **160**. Continuing at step **550**, after a remote gaming session is established on the host gaming device, the central gaming controller **180** sends a message to the host gaming device **160** instructing it to displace representations of its user controls, graphics and sounds to the remote player interface **300**. The central gaming controller **180** directs the host gaming device **160** controls over the secured encrypted connection and manages the remote gaming session. Next at step **555**, the remote player may transfer funds from a player account to the host gaming device **160** for wagering on the host gaming device **160**. Moving to step **560**, a wager is made. Next at, **656** a game is played. Continuing to step **570**, the central gaming controller **180** delivers the results of the game to the remote player interface **300**. Next at step **571**, the remote player may repeat the sequence from step **560**. Next at step **575**, if there are any credits on the host gaming device **160** when the player terminates the remote gaming session, the central gaming controller **180** automatically transfers those credits back to the players account. Moving to step **580**, the central gaming controller **180** terminates the remote gaming session with the host gaming device **160**. Continuing to step **585**, the central gaming controller **180**, enables local play on the host gaming device **160**, control is then transferred to the end state.

[0083] **FIG. 7** is a flowchart depicting a method for a host gaming device **160** to become connected to a network using security certificates and a certificate authority. Depending on the embodiment, additional steps may be added, others removed, steps merged, or the order of the steps rearranged. Starting at **705**, a host gaming device **160** starts the process of connecting to a network as part of its initialization mode. Continuing to step **720**, at a point during initialization, the host gaming device **160** submits a security certificate to a certificate authority for authentication. Moving to step **725**, the certificate authority authenticates the certificate. Next at step **730**, if the certificate is authenticated control moves to step **740**, otherwise control moves to step **735**. Continuing on to step **740**, the host gaming device **160** is permitted onto the network and the process moves to its end state. Returning to step **735**, if the certificate is not authenticated then a log entry is generated and the host gaming device **160** is not permitted onto the network.

[0084] Embodiments according to the invention may also use instant messaging and/or email messaging systems. Typical instant messaging systems permit computer users to type text messages and add file attachments into a host program and have the host program automatically deliver the text through a virtual direct connection to a target computer. Public email systems are those available for general use, as over the internet. Examples of public instant messaging systems in use today include but are not limited to chat programs like IRC, MSN Messenger, AOL Instant Messaging and a host of others. Private systems are restricted to a casino or gaming system. Typical email messaging systems permit messages and file attachments to be entered into a host program and addressed to a specific recipient on a network. These messages may not be delivered directly to the addressee, but are sent to a storage area where the recipient may retrieve the message at a time of their own choosing.

[0085] Gaming devices **160** and remote player devices **110** routinely exchange information with a central gaming controller **180** for, typically, but not limited to, account and game tracking functions. In one embodiment of the invention, devices may send and receive data over public and/or private email-type messaging systems. The message body of any particular message may vary, using a proprietary or non-proprietary format, and may be encrypted or in human-readable format. Messages may be sent at a time determined by the message originator, typically, but not exclusively, in response to an event. The recipient of the message may be any device capable of consuming the message. The message recipient may be responsible for checking the prescribed message storage area for messages addressed to it. The message recipient may reply to a received message or may generate a new message to a specific recipient, a group of recipients, or all recipients connected to the system. Remote player devices **110** may periodically check for new messages in the system and process them.

[0086] According to one embodiment of the invention, gaming devices **160** may send and receive data over public and/or private instant messaging systems. The message body of any particular message may vary, using a proprietary or non-proprietary format, and may be encrypted or in human-readable format. Messages may be sent at a time determined by the message originator, typically, but not exclusively, in response to an event. The recipient of the message may be any device capable of consuming the message. Both the gaming device **160** and the message recipient may queue incoming and outgoing messages. Queuing messages permits devices involved in instant message communications to accept new messages while processing received messages and to generate outgoing messages for delivery as system resources permit.

[0087] In another embodiment according to the invention, devices may send and receive data over public and/or private email-type messaging systems. The message body of any particular message may vary, using a proprietary or non-proprietary format, and may be encrypted or in human-readable format. Messages may be sent at a time determined by the message originator, typically, but not exclusively, in response to an event. The recipient of the message may be any device capable of consuming the message. The message recipient may be responsible for checking the prescribed message storage area for messages addressed to it. The message recipient may reply to a received message or may generate a new message to a specific recipient, a group of recipients, or all recipients connected to the system. Gaming system devices **110** and **160** may periodically check for new messages in the system and process them.

[0088] Embodiments according to the invention may present promotional messages during remote play sessions. Messages sent may comprise instant messages for promotional information, notification of events, or other pieces of information that can be communicated electronically. Promotional messages may also include jackpot and bonus information. A promotional message server may be used to construct and send promotional messages. In one embodiment, a computer server, comprising a central gaming controller **180,** may also comprise the promotional message server.

[0089] A user interface may be provided to construct message templates. These templates are then used to construct a deliverable message. Embodiments of a message template may comprise a timeout value that indicates how long the message is to be displayed, the frequency with which the message displays in relationship to other scheduled messages, a limitation value that prevents the message from being displayed too often and an expiration date after which the message is no longer used in the system. Custom graphics and display modes may also be specified for a message template, such as icons, animations, and various scrolling methods.

[0090] A remote player device **110** may present a promotional message for an amount of time determined from the contents of the promotional message. The promotional message may be presented to a user in conjunction with gaming information. The presentation may contain icons, animations, and various scrolling methods. In addition multimedia such as sound and video may be utilized.

[0091] The promotional message server may also provide a dynamic data insertion function to insert player information such as the player's name or birthday into a message prior to delivery. Dynamic data insertion may be accomplished through the use of specialized tags within the message body. When encountered, the tag characters within the message are replaced with data from a related data source. The specific tag's character sequence is associated with a specific subset of the data in the data source, such as a player's name in a data source of player information. Processing comprises reading the data source and its subsets, parsing the specialized tags from the message template, indexing the data source and replacing the tag characters with data from the data source to create a deliverable message for each item in the data source. This sequence continues until all the data in the data source has been included in messages. The messages may be delivered as they are created or queued until all items in the data source have been used to create messages, then all messages may be sent at the same time.

[0092] In one embodiment, a gaming system **100** may comprise a card reader installed in a gaming device **280** or remote player device **380**. Promotional messages may be based on information obtained about a player that is either stored on a card inserted into the card reader or by using identifying information from the card to access the casino's proprietary database systems **170**.

[0093] One embodiment of the promotional message server may also provide a dynamic grouping function in which a subset of players currently gaming is selected and collected into a group. Casino operators may address a message template to this dynamic subset of current players and send a specific message or messages exclusively to that subset. These messages may be constructed using the dynamic data function. The dynamic grouping function may use criteria specified by the casino and available in the casino's proprietary database systems **170** and criteria generated by live gaming activity to establish a profile that players must meet to be selected. The criteria may comprise loyalty points the player has earned, a player's birthday, length of current gaming session, or other data that is collected by the casino on players and gaming activity.

[0094] The dynamic grouping function may be scheduled to run at time intervals determined by the casino. Each time the interval is reached the promotional gaming server searches for current players that meet the established criteria and builds a dynamic group then sends the assigned message to that group of players exclusively. The gaming devices **160**, remote player device **110**, card readers installed in gaming devices **280** and remote player device **380**, and casino proprietary database systems **170** may provide data to search for players that meet the specified criteria and assemble them into a dynamic group.

[0095] In one embodiment of the invention, the casino may advertise a casino sponsored event. The casino may use a user interface display to construct the message and schedule its delivery start time, duration of the message e.g. number of hours, days, weeks, or months that the message will run, and specific values that weight the message's delivery interval and frequency amongst other promotional messages scheduled in the system. The style of message may also be specified, including but not limited to flashing, scrolling, scroll direction, and the use of custom graphics. The casino operator may also specify the criteria players must meet to receive the message. Once the casino operator accepts the promotional message configuration, the promotional message server may deliver the message across a network to remote player devices **110** or host gaming systems **160**.

[0096] An embodiment of a gaming system **100** may provide for the electronic transfer of funds to a gaming device for the purpose of making wagers. When a player chooses a gaming device **160** to play remotely, funds are electronically transferred to the gaming device and appear as credits on the gaming device **160**. The player then uses those credits to make wagers on game outcome. When the player is finished, the system transfers any remaining credits on the gaming device back to the source of funds or to an alternate storage. Limitations on the amount of funds transferred may be set for a minimum or maximum amount transferred, a minimum or maximum amount transferred within a given time period, or a minimum or maximum amount transferred for the life of the account, or a combination of any of these. The limitation may also vary between accounts, permitting one account to have a different limitation on transfers than another. When the limitation set is reached, further transactions are prevented until the limitation is resolved. The limitation may be set voluntarily by the player, by the casino, or by a gaming authority. Limitations may be set for all players within a specific jurisdiction or for selected players only. The source of funds used by a player for remote access play may be maintained in a database located on a computer that is directly or indirectly connected to the casino network **150**.

[0097] **FIG. 6** is a flowchart depicting an embodiment of the invention whereby a player transfers funds from a bank account to a player account for the purpose of wagering on games. Depending on the embodiment, additional steps may be added, others removed, steps merged, or the order of the steps rearranged. Starting at step **601**, a remote player device **110** initiates an electronic funds transfer. Continuing to step **605**, the central gaming controller **180** verifies the remote players banking information. Next at step **610**, if the banking information is valid, control transfers to step **620**, otherwise control moves to step **615**. Continuing at step **620**, the remote player device **110** prompts the player to enter the amount of the transfer. Moving to step **615**, the central gaming controller **180** verifies fund availability. Next at step **630**, if funds are not available control moves to step **615**. Otherwise, control moves to step **635**, where, in a one embodiment, the central gaming controller **180** may consult a casino database **170** and determine whether the remote players total gaming activity exceed limits placed on that activity. Next at step **640**, if the limit is reached control moves to step **615**. Otherwise, continuing at step **645**, the transfer is completed. Returning to step **615**, if the players banking information is not correct, funds are not available or a transfer limit is reached, then the transaction is canceled and control transferred to the end state.

[0098] An embodiment of a gaming system **100** may record the interaction between remote players and host gaming devices **160** during remote gaming sessions for the purpose of resuming games in-progress after a communications failure. If at anytime the connection between the remote player and a gaming device becomes unavailable, the system has a sufficient record of player positions to restart the game as at the time just prior to the failure. Thus an embodiment of a gaming system may record, transfer, and reinstate on a like device an encrypted block of data representing the precise state of a particular gaming device **160** at the time that the data block is requested. The encrypted block of data is generated by the gaming device **160** and transferred using a communication protocol. The encrypted block of data may be used to continue a game in-progress that was interrupted by a gaming device **160** failure or other system failure. In addition, the payer's wager and credit data along with gaming payout data may be included in the data block. The data may also be transported to another gaming device **160** for the purpose of completing an interrupted game or resuming a gaming session. The destination gaming device **160** receives the encrypted block of data, decrypts it, and loads the game state into its own systems, allowing a game in-progress to complete or a game session to continue.

[0099] **FIG. 8** is a flowchart depicting a method for a gaming device **160** to build and deliver an encrypted block of data representing the complete state of the gaming device. Depending on the embodiment, additional steps may be added, others removed, steps merged, or the order of the steps rearranged. Starting at **805**, a central gaming controller **180** sends a message to a host gaming device **160** to initiate the build of the encrypted data block. Continuing to step **10**, the gaming device responds with an acknowledgement. Next, at step **815**, the gaming device **160** begins the build process. When finished with the build and encryption process, at step **820**, the gaming device saves the data block to non-volatile memory in the gaming device. Continuing to step **825**, the gaming device **160** sets an indication that may be queried by the central gaming controller **180** as to the status of the build/encryption process. Moving to step **830**, the central gaming controller **180** checks the gaming device's status. Next at step **835**, if the build/encryption process is complete, control continues to step **840**, otherwise control returns to step **830**. Moving to step **840**, the central gaming controller **180** retrieves the data block from the gaming device **160**. Next, at step **845**, when the central gaming controller **180** has retrieved the data block it saves the data block to a database. Continuing to step **850**, the central gaming controller then checks the validity of the saved data block. If the data block is not verified then the central gaming controller initiates another retrieval by returning control to step **840**.

[0100] **FIG. 9** is a flowchart depicting a method for retrieving an encrypted block of data representing the state of a gaming device from a database and loading the encrypted block into a gaming device. Depending on the embodiment, additional steps may be added, others removed, steps merged, or the order of the steps rearranged. Starting at step **905**, the central gaming controller **180** retrieves a saved encrypted data block from the database. Next at **910**, the controller **180** verifies the integrity of the data block. Continuing to **915**, if the data block is verified, control continues to step **925**, if not control moves to step **920**. Returning to the flow of control at **925**, the central gaming controller **180** notifies a target gaming device **160** of an intent to upload the data block. Next, at step **930**, the target gaming device **160** responds with a message indicating whether it is available for the upload. Moving to step **935**, if the target device is ready control moves to step **940**, if not control is diverted to step **920**. Returning back to step **940**, the encrypted data block is uploaded to the target gaming device **160**. Next at step **945**, the target gaming device **160** verifies the encrypted data block. Moving on to step **950**, if the data block was verified, the gaming device moves on to step **955**, if not verified, control moves to step **920**. Continuing on to step **955**, the gaming device **160** initializes its state to the new state defined by the received data block and the process moves to the end state. Returning back to step **920**, which is reached on error conditions, an error log entry is generated and the requesting process notified.

[0101] **FIG. 10** is a block diagram depicting one embodiment of a gaming system according to the present invention wherein the host gaming devices **160** are available for remote play over a network that connects to a cable modem termination system. The cable modem termination system **1005** is located at the head-end of a cable television provider who makes broadband network connectivity available as a service to its customers. Cable television customers who subscribe to broadband or digital television services access the remote network **120** through a digital home communications terminal (DHCT) **1000**. The remote player device **110** may be a stand-alone cable modem or a set-top box that includes a cable modem and a digital television broadcast decoder. The DHCT **1000** may, in some embodiments include the remote player device **110**. The remote player interface **300** may be any device or combination of devices that remote players operate to interact with the remote player device **110**, for example, a television with remote control or a personal computer. To connect to the central gaming controller **180**, a remote player uses the remote player device **110** to send messages, using, in one embodiment, IP datagrams, through the DHCT and the cable modem termination system **1005**. The cable modem termination system **1005** uses a network router **1004** to route the IP datagrams over a network connection **140** to the central gaming controller **180**. The backbone network connection **140** can be any type of network connection such as a dedicated T1 or fiber optic over which network traffic can be exchanged. In preferred embodiments the backbone network **140** is part of a closed loop network. However, in other embodiments, a public network such as the Internet may form at least a portion of the backbone network. Encryption of the data may be performed, either at the endpoints such as remote player device **110**, at a host gaming device **160**, at a central gaming controller **180**, over network **120**, or only over network **140**.

[0102] Network traffic from the remote network **120** and backbone network **140** travels over a number of virtual local area networks (VLAN) configured using a multilayer network switch **1022**. Segmenting the internal network into VLANs creates security zones whereby only permitted network traffic appears on a given VLAN.

[0103] IP datagrams are received over the backbone network **140** through network router **1020** and firewall **1021**. Network router **1020** filters IP datagrams that are not coded with the configured port for access to the gaming network **150**. If an IP datagram passes the network router **1020** it then must pass the firewall **1021** in order for the IP datagram to be processed by the request processing server(s) **1023** which comprise a portion of a central gaming controller **180** in this embodiment.

[0104] The firewall **1021** has two network interfaces **1050**, **1051**; the external-facing network interface **1050** is connected to the router **1020** and the internal-facing network interface **1051** is connected to the multilayer network switch **1022**. In this configuration the firewall **1021** acts as a type of network switch that may perform additional security checks on the IP datagram, then move the datagram to the internal-facing network interface **1051** where the multilayer network switch **1022** moves the datagram to the VLAN where request processing server(s) **1023** are located.

[0105] Each request processing server **1023** has two network interfaces **1052**, **1053**, both connected to the multilayer network switch **1022**. Each network interface **1052**, **1053** may be configured on a different VLAN of the multilayer network switch **1022**. The multilayer network switch **1022** moves IP datagrams between the firewalls **1021** internal-facing network interface **1051** and the request processing server(s) **1023** external-facing network interface **1052**. This embodiment provides a layer of protection for the host gaming devices **160** in the event that the request processing server(s) **1023** are compromised.

[0106] When an IP datagram arrives at a request processing servers **1023** external-facing network interface **1052**, the request processing server **1023** interprets the IP datagram and issues commands over its internal-facing network interface **1053** to the application server **1027**. The request processing server **1023** may reject invalid commands or make other determinations as to the appropriateness of a request that prevent the request from being passed on to the application server **1027**. Likewise, the request processing server **1023** may request data from the application server for use in building its own response to the request, which may or may not require an acknowledgement from the remote player device **110** as described below.

[0107] Command messages received by the application server **1027** may be recorded in a database using the database server **1025**. The application server **1027** then executes the command, which may include any function relevant to the operation of the host gaming device **160** and may or may not return data to the request processing server **1023** for delivery to the remote access player. In one embodiment, the database server **1025** may comprise the casino database **170**. In other embodiments the database server **1025** and the application server **1027** may comprise the casino database **170**.

[0108] Some commands may require the remote player device **110** to acknowledge the receipt of information sent from the central gaming controller **180**. For commands that require acknowledgement, the central gaming controller **180** queues the status of the messages that are sent to the remote player device **110**. The status of messages sent but not acknowledged is stored in a database as "open" using the database server **1025**. When the remote player device **110** receives the message it sends an acknowledgment message back to the central gaming controller, which in turn marks the message in the database as "closed"; indicating that the message has reached its destination and has been acknowledged. If the message is not acknowledged within a specified timeout, the message is resent. **FIG. 4** depicts the sequence of events for the receipt, queuing and response loop for qualifying messages.

[0109] Recording of messages between the remote player device **110** and a host gaming device **160** by the central gaming controller **180** allows each game or transaction, on both the host gaming device **160** and remote player device **110**, to be recorded. This allows each host gaming device or remote player device to be individually auditable using standard accounting practices in the gaming jurisdiction where the game is located. In one embodiment, a third party, such as a gaming authority may be sent the records of games and transactions online by the gaming system **100**.

[0110] When the application server **1027** receives a command request that requires communication with gaming devices **160, 161, 162** it connects to those devices using terminal server **1035**. Terminal server **1035** provides Ethernet connectivity to the RS232 serial interface **1054** of the game. Through that interface the remote player device **110** communicates to the gaming devices **160, 161, 162** using a communications protocol supplied by the gaming machine manufacturer. The protocol includes commands that permit the remote operation of the gaming devices **160, 161, 162** and the reporting of game results so that the application server **1027** can control remote play.

[0111] **FIG. 11** depicts a more detailed network diagram of one embodiment of network **150** and elements of a gaming system **100** connected to network **150**. This includes a host gaming device **160**, and a database **160**. As in the embodiment of **FIG. 10**, a central gaming controller **180** may be comprised of request processing servers **1027** and an application server **1023** connected to one or more VLANs of network **150**.

[0112] While the above detailed description has shown, described, and pointed out novel features of the invention as applied to various embodiments, it will be understood that various omissions, substitutions, and changes in the form and details of the device or process illustrated may be made by those skilled in the art without departing from the spirit of the invention. The scope of the invention is indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

WHAT IS CLAIMED IS:

1. A gaming system comprising:

a data network, wherein the data network is comprised of at least one logical segment, wherein at least one logical segment is a closed-loop network;

a host gaming device connected to the data network, the gaming device configured to execute at least one game wherein the host gaming device in a location approved by a gaming agency;

a plurality of remote player devices connected to the closed-loop network; and

a central gaming controller configured to create an auditable record of gaming transactions on the host gaming device and on each of the plurality of remote player devices,

wherein the plurality of remote player devices are each configured to receive game information provided by the host gaming device, and

wherein the host gaming device is configured to provide game information to a predetermined number of permitted remote player devices.

2. A gaming system comprising:

a data network;

a host gaming device connected to the data network, the gaming device configured to execute at least one game; and

a plurality of remote player devices connected to the data network,

wherein the plurality of remote player devices are each configured to receive game information provided by the host gaming device,

wherein the host gaming device is configured to provide game information to a predetermined number of permitted remote player devices, and

wherein at least one of the plurality of remote player devices is permitted based upon, at least in part, the geographic location of the remote player device.

3. The system of Claim 2, wherein the predetermined number is determined by a gaming agency.

4. The system of Claim 2, wherein at least one of the plurality of remote player devices is permitted based upon, at least in part, an age of a user of the remote player device.

5. The system of Claim 2, wherein the data network is, at least in part, the Internet.

6. The system of Claim 2, wherein the data network is comprised of at least one logical segment.

7. The system of Claim 6, wherein at least one logical segment is a closed-loop network.

8. The system of Claim 6, wherein the host gaming device is configured to identify the geographic location of a remote player device based, at least in part, on a logical segment corresponding to the remote player device.

9. The system of Claim 2, wherein the host gaming device is configured to identify the geographic location of a remote player device based, at least in part, on information provided by a mobile communications network.

10. The system of Claim 2, wherein the host gaming device is configured to identify the geographic location of a remote player device based, at least in part, on information provided by a GPS device.

11. The system of Claim 2, wherein the data network is, at least in part, the casino intranet.

12. The system of Claim 2, wherein the data network is, at least in part, the hotel intranet.

13. The system of Claim 2, wherein the data network is, at least in part, a wireless network.

14. The system of Claim 2, wherein the host gaming device is in a location approved by a gaming agency.

15. The system of Claim 2, wherein the host gaming device includes at least one game control configured to provide local use.

16. The system of Claim 15, wherein the host gaming device is configured to disable local use when the host gaming device is providing game information to a remote player device.

17. The system of Claim 2, wherein each of the remote player devices is in a location approved by a gaming agency.

18. The system of Claim 2, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on the host gaming device.

19. The system of Claim 2, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on each of the plurality of remote player devices.

20. The system of Claim 2, wherein the gaming information is, at least in part, software.

21. The system of Claim 2, wherein at least one remote player device is coupled to a credential device configured to receive information relating to a user of the remote player device.

22. The system of Claim 21, wherein the information relating to the user is an age of the user.

23. The system of Claim 21, wherein the information relating to a user is a password that is input by the user.

24. The system of Claim 21, wherein the credential device is an input device configured to receive a password from the user.

25. The system of Claim 21, wherein the credential device is a smart card reader.

26. The system of Claim 21, wherein the credential device is a biometric device.

27. The system of Claim 28, wherein the biometric device is a fingerprint reader.

28. The system of Claim 21, further comprising: a database configured to provide information associated with each of a plurality of users of the gaming system.

29. The system of Claim 28, wherein the information associated with a user includes a password.

30. The system of Claim 28, wherein the information associated with a user includes an age of the user.

31. The system of Claim 28, wherein the information associated with a user includes information relating to a fingerprint of the user.

32. The system of Claim 2, wherein the host gaming device is configured to encrypt the game information.

33. The system of Claim 2, wherein the game information is provided via a public email system.

34. The system of Claim 2, wherein the game information is provided via a private email system.

35. The system of Claim 2, wherein the game information is provided through a public messaging system.

36. The system of Claim 2, wherein the game information is provided through a private messaging system.

37. A gaming system comprising:

a data network;

a host gaming device in a location approved by a gaming agency connected to the data network, the gaming device configured to execute at least one game; and

a plurality of remote player devices connected to the data network.

wherein the plurality of remote player devices are each configured to receive game information provided by the host gaming device, and

wherein the host gaming device is configured to disable local use of the gaming device when providing game information to the remote player devices.

38. The system of Claim 37, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on the host gaming device.

39. The system of Claim 37, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on each of the plurality of remote player devices.

40. The system of Claim 37, wherein the host gaming device is configured to allow no more than a predetermined number of remote player devices to concurrently receive game information provided by the host gaming device.

41. A gaming system comprising:

gaming means for executing at least one game, the game providing game information during execution;

local access means for providing local access to the game information for a user in a location approved by a gaming agency;

player means for receiving game information, presenting game information and providing at least one game control;

means for providing the game information over a data network to a predetermined number of receiving means;

means for determining the location of the receiving means; and

means for disabling the local access means.

42. The system of Claim 41, further comprising:

a means for creating an auditable record of gaming transactions on the gaming means.

43. The system of Claim 41, further comprising:

a means for creating an auditable record of gaming transactions on the playing means.

44. The system of Claim 41, wherein the predetermined number is determined by a gaming agency.

45. The system of Claim 41, further comprising:

means for receiving information associated with a user of the gaming system.

46. The system of Claim 45, wherein the information associated with the user includes the age of the user.

47. The system of Claim 45, wherein the means for receiving information associated with a user is a smart card reader.

48. The system of Claim 45, wherein the means for receiving information associated with a user is a biometric identity device.

49. The system of Claim 45, wherein the means for receiving information associated with a user is a keyboard configured to receive a password.

50. The system of Claim 45, wherein the user information includes, at least, a credential for authentication of the user.

51. The system of Claim 50, further comprising:

means for authenticating the credential coupled to means for limiting access to the gaming system.

52. A method of remotely accessing a host gaming device on a remote player device comprising:

establishing access to the host gaming device from the remote player device through a data network;

receiving gaming related information from the host gaming device through the data network;

presenting the gaming related information to a player;

receiving at least one control signal from the player;

sending the control signal to the host gaming device through the data network; and

disabling local use of the host gaming device.

53. The method of Claim 52, further comprising:

recording each gaming transaction occurring on the remote player device.

54. The method of Claim 52, further comprising:

providing a geographic location of the remote player device.

55. The method of Claim 52, further comprising:

providing information relating to a user of the remote player device to the gaming device.

56. The method of Claim 55, wherein the information relating to a user includes, at least, the age of the user.

57. The method of Claim 52, further comprising:

allowing no more than a predetermined number of remote player devices to concurrently establish a gaming session on the gaming device.

58. A method of providing remote access to a host gaming device comprising:

verifying a geographic location of a remote player device;

establishing a gaming session on a host gaming device from a remote player device through a data network;

receiving at least one control signal from the remote player device through the data network;

sending gaming related information from the gaming device through the data network;

59. The method of Claim 58, further comprising:

recording each gaming transaction occurring on the host gaming device.

60. The method of Claim 58, further comprising:

receiving information relating to a user of the remote player device on the gaming device.

61. The method of Claim 60, wherein the information relating to a user includes, at least, the age of the user.

62. The method of Claim 58, further comprising:

disabling local access to the gaming device.

63. The method of Claim 58, further comprising:

allowing no more than a predetermined number of remote player devices to concurrently establish a gaming session on the gaming device.

64. A method of resuming an interrupted gaming session on a first host gaming device comprising:

generating a gaming state of the gaming session on the first gaming device;

encrypting the gaming state;

transporting the encrypted gaming state from the first gaming device;

transporting the encrypted gaming state to a second gaming device;

decrypting the gaming state on the second gaming device; and

loading the game state into a second gaming device to resume the gaming session.

65. A gaming system comprising:

a data network;

a first host gaming device connected to the data network, the gaming device configured to:

execute at least one game,

generate a gaming state based on execution of at least one game;

encrypt the gaming state; and

send the encrypted gaming state over the data network;

a second host gaming device connected to the data network, the gaming device configured to:

receive the encrypted gaming state over the data network;

decrypt the gaming state;

resume executing at least one game from the gaming state; and

a plurality of remote player devices connected to the data network, wherein the plurality of remote player devices are each configured to receive game information provided by the host gaming device.

66. The system of Claim 65, wherein the remote player devices are each configured to receive an encrypted gaming state from a first gaming device over the data network and to send the encrypted gaming state to the second gaming device.

67. The system of Claim 66, wherein the first gaming device is the second gaming device.

68. The system of Claim 65, wherein the second gaming device is configured to receive an encrypted gaming state from a first gaming device over the data network.

69. The system of Claim 65, wherein the gaming state includes user payment information.

70. The system of Claim 65, wherein the gaming state includes gaming machine payout information.

71. The system of Claim 65, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on the host gaming device.

72. The system of Claim 65, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on each of the plurality of remote player devices.

73. A gaming system comprising:

means for executing at least one game;

means for generating a gaming state based on execution of at least one game;

means for encrypting the gaming state;

means for sending the encrypted gaming state;

means for receiving the encrypted gaming state;

means for decrypting the gaming state; and

means for resuming executing at least one game from the gaming state.

74. The system of Claim 73, wherein the gaming state includes user payment information.

75. The system of Claim 73, wherein the gaming state includes gaming machine payout information.

76. The system of Claim 73, further comprising:

a means for creating an auditable record of gaming transactions on the host gaming device.

77. The system of Claim 73, further comprising:

a means for creating an auditable record of gaming transactions on each of the plurality of remote player devices.

78. A method of authenticating a user of a host gaming device comprising:

receiving a security certificate from the smart card;

sending the security certificate to a certificate authority for authentication;

receiving an authentication reply from the authority; and

playing a game in response to the authentication reply.

79. A method of authenticating a user of a remote player device comprising:

receiving an indicia of identity for a user;

sending the indicia of identity to an authenticator device;

receiving an authentication reply from the authenticator device; and

authorizing use of a host gaming device based on the indicia of identity

80. The method of Claim 79, wherein the indicia of identity for a user is provided by a biometric identity device.

81. The method of Claim 79, wherein the indicia of identity for a user is provided by a password input by the user.

82. The method of Claim 79, wherein the indicia of identity for a user is provided by a smart card.

83. A gaming system comprising:

a data network;

a host gaming device interfaced to the data network;

a plurality of remote player devices interfaced to the data network; and

a security device configured to provide player credentials to at least one remote player device,

wherein the plurality of remote player devices are each configured to receive game information provided by the host gaming device,

wherein the host gaming device is configured to provide game information to a predetermined number of permitted remote player devices, and

wherein at least one of the plurality of remote player devices is permitted based upon, at least in part, on player credentials provided by the security device.

84. The system of Claim 83, wherein the security device is a smart card reader.

85. The system of Claim 83, wherein the security device is a biometric device.

86. The system of Claim 83, wherein the security device is an input device.

87. The system of Claim 86, wherein the player credentials are, at least in part, a password.

88. The system of Claim 83, wherein the remote player device is authorized to receive game information provided by the host gaming device based, in part, on the player credentials.

89. The system of Claim 83, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on the host gaming device.

90. The system of Claim 83, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on each of the plurality of remote player devices.

91. A method of remotely accessing a gaming device comprising:

establishing a gaming session on a gaming device for a remote player device through a data network;

sending gaming related information from the gaming device through the data network;

receiving at least one control signal from the remote player device through the data network.

creating an auditable gaming session record representing each gaming transaction of a gaming session on the host gaming device;

creating an auditable gaming session record representing each gaming transaction of a gaming session on the remote gaming device; and

sending the record to a third party through the data network.

92. The method of Claim 91 wherein the third party is a gaming authority.

93. A gaming system comprising:

a data network comprised of a plurality of logical segments wherein a security policy controls the flow of data between logical segments;

a host gaming device connected to the data network, the gaming device configured to execute at least one game; and

a plurality of remote player devices connected to the data network,

wherein the plurality of remote player devices are each configured to receive game information provided by the host gaming device, and

wherein the plurality of remote player devices are each configured to control a gaming session established on the gaming device subject to the security policy wherein the security policy is based, at least in part, on the geographic location of a logical segment.

94. The system of Claim 93, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on the host gaming device.

95. The system of Claim 93, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on each of the plurality of remote player devices.

96. A gaming system comprising:

a data network;

a promotional message server configured to provide customized promotional messages wherein each message is customized with information associated with a user of the gaming system;

a host gaming device interfaced to the data network; and

a plurality of remote player devices interfaced to the data network,

wherein the plurality of remote player devices are each configured to receive game information provided by the host gaming device and to receive and present promotional messages.

97. The system of Claim 96, wherein the remote player devices are in a location approved by a gaming agency.

98. The system of Claim 96, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on the host gaming device.

99. The system of Claim 96, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on each of the plurality of remote player devices.

100. The system of Claim 96, wherein promotional message are comprised of bonus information.

101. The system of Claim 96, wherein promotional message are comprised of jackpot information.

102. The system of Claim 96, further comprising: at least one database configured to provide information associated with a plurality of users of the gaming system.

103. The system of Claim 96, wherein each of the plurality of remote game devices is associated with a user.

104. The system of Claim 96, further comprising a smart card reader configured to provide information associated with a user of the gaming system.

105. The system of Claim 102, wherein the database is configured to provide information which forms, at least in part, the content of the promotional message.

106. The system of Claim 96, wherein each of the plurality of remote player devices is configured to receive and present the promotional message in conjunction with game information provided by the host gaming device.

107. The system of Claim 106, wherein each of the plurality of remote player devices is configured to present the promotional message for an amount of time.

108. The system of Claim 106, wherein the amount of time is based, at least, in part on information associated with the promotional message.

109. The system of Claim 102, wherein the database is configured to provide information which comprises, at least in part, the content of the promotional message.

110. The system of Claim 96, wherein the promotional messages are transported via an instant messaging system.

111. The system of Claim 96, wherein the promotional messages are transported via an email system.

112. A method of displaying information on a remote player device comprising:

receiving a promotional message on a remote player device;

presenting the promotional message in conjunction with gaming information for an amount of time; and

removing the promotional message from the remote player device.

113. The method of Claim 112, further comprising

calculating the amount of time based, at least in part, on information associated with the promotional message.

114. A gaming system comprising:

means for data communication;

means for executing at least one game;

means for providing game information over the data network to a predetermined number of receiving means; and

a plurality of means for receiving game information over the data communication means, each coupled to a means for receiving customized promotional messages.

115. The method of Claim 114, further comprising:

means for presenting customized promotional messages in conjunction with game information.

116. The method of Claim 114, further comprising:

means for sending promotional messages.

117. The method of Claim 114, further comprising:

means for providing data used to select which players receive customized promotional messages.

118. The method of Claim 114, further comprising:

means for providing data which forms, at least in part, the content of promotional messages.

119. The system of Claim 114, further comprising:

a means for creating an auditable record of gaming transactions on the host gaming device.

120. The system of Claim 114, further comprising:

a means for creating an auditable record of gaming transactions on each of the plurality of remote player devices.

121. A gaming system comprising:

a data network;

a host gaming device interfaced to the data network;

at least one remote player device interfaced to the data network;

a video display device in communication with the remote player device; and

a remote control device in communication with the remote player device,

wherein the remote player device is configured to receive game information provided by the host gaming device and the remote control device is configured to control operation of a game.

122. The system of Claim 121, wherein the video display device is a television.

123. The system of Claim 121, wherein the video display device is a computer.

124. The system of Claim 121, wherein the video display device is a control device.

125. The system of Claim 121, wherein the remote player device is coupled to a cable television system.

126. The system of Claim 121, wherein the data network is, at least in part, the Internet.

127. The system of Claim 121, wherein the data network is, at least in part, the casino intranet.

128. The system of Claim 121, wherein the data network is, at least in part, the hotel intranet.

129. The system of Claim 121, wherein the data network is, at least in part, a wireless network.

130. The system of Claim 121, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on the host gaming device.

131. The system of Claim 121, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on each of the plurality of remote player devices.

132. A method of remotely accessing a host gaming device comprising:

establishing a gaming session on the host gaming device from a remote player device through a data network;

receiving gaming related information from the host gaming device through the data network;

presenting gaming related information to a player via a video display device;

receiving at least one control signal generated by a remote control device for controlling the gaming session; and

sending the control signal to the host gaming device through the data network.

133. The method of Claim 132, further comprising:

recording each gaming transaction occurring on the remote player device.

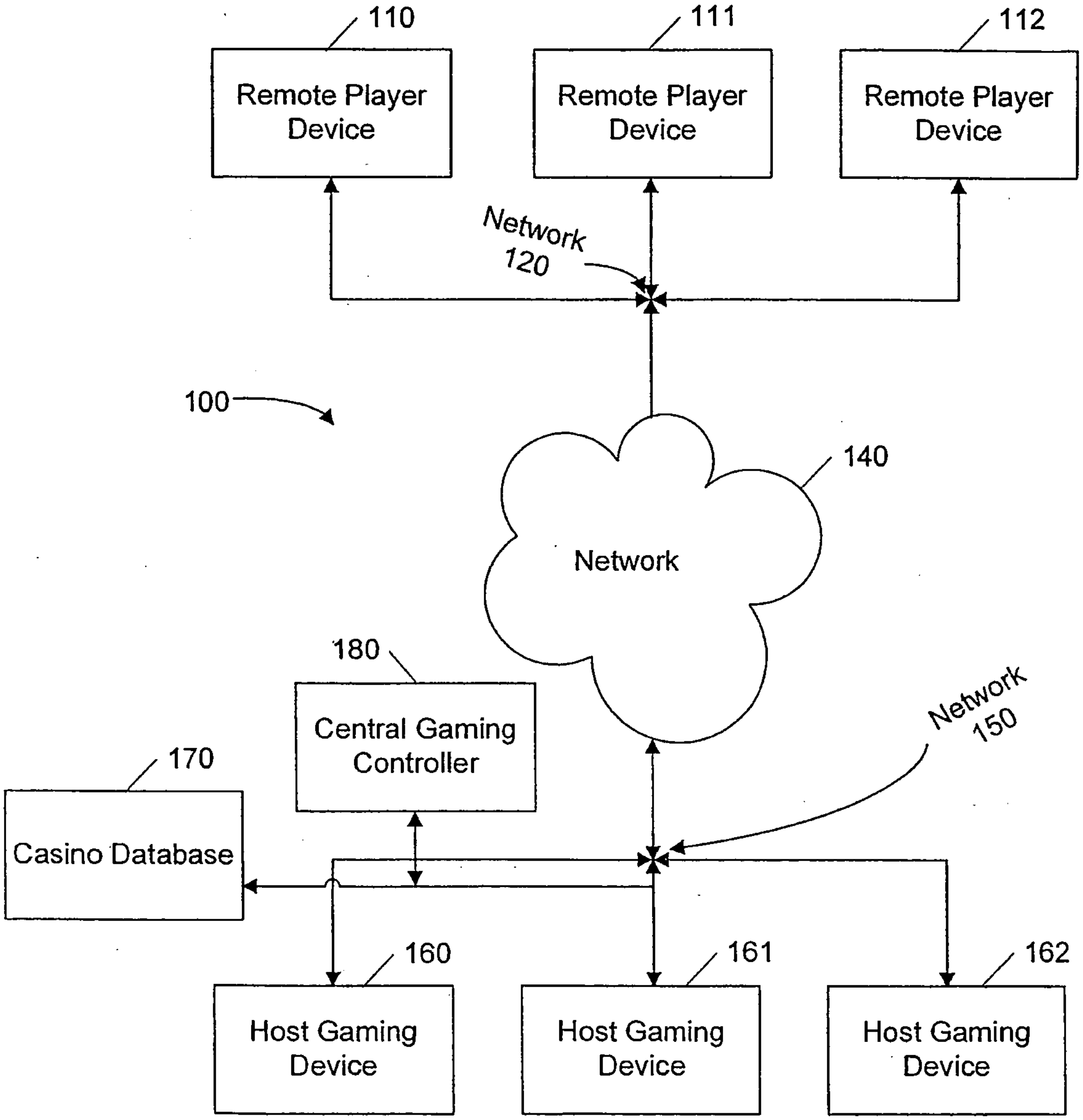

110
Remote Player Device
111
Remote Player Device
112
Remote Player Device
Network 120
100
140
Network
180
Central Gaming Controller
170
Casino Database
Network 150
160
Host Gaming Device
161
Host Gaming Device
162
Host Gaming Device

FIG. 1

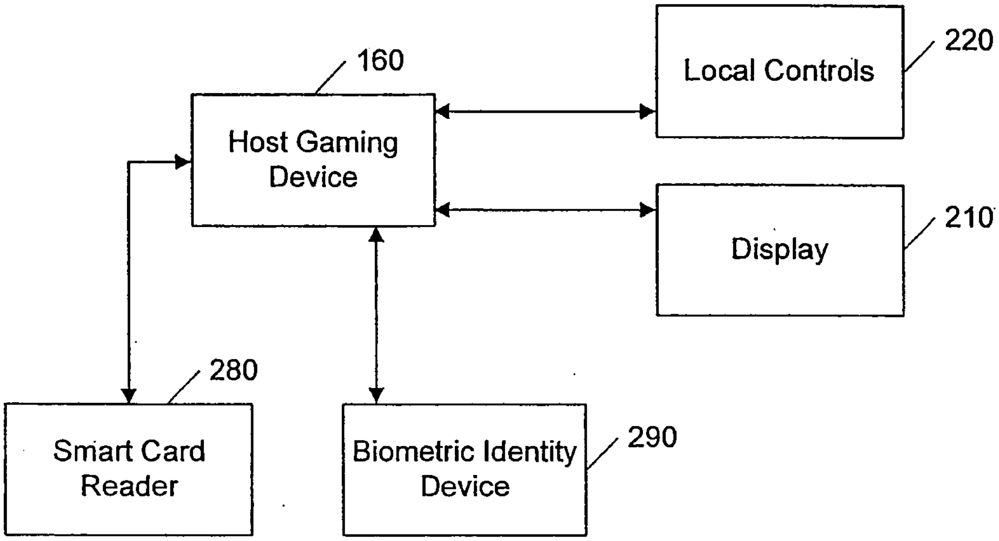

160
Host Gaming Device
Local Controls
220
Display
210
280
Smart Card Reader
Biometric Identity Device
290

FIG. 2

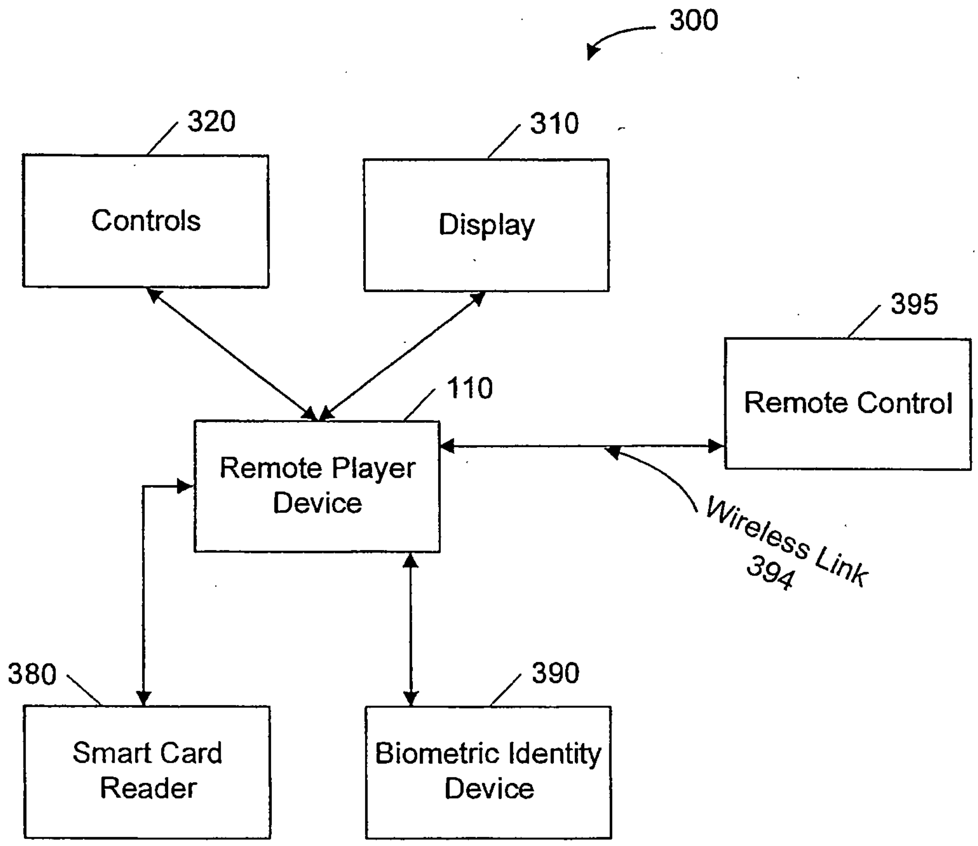

300
320
Controls
310
Display
395
Remote Control
110
Remote Player Device
Wireless Link 394
380
Smart Card Reader
390
Biometric Identity Device

FIG. 3

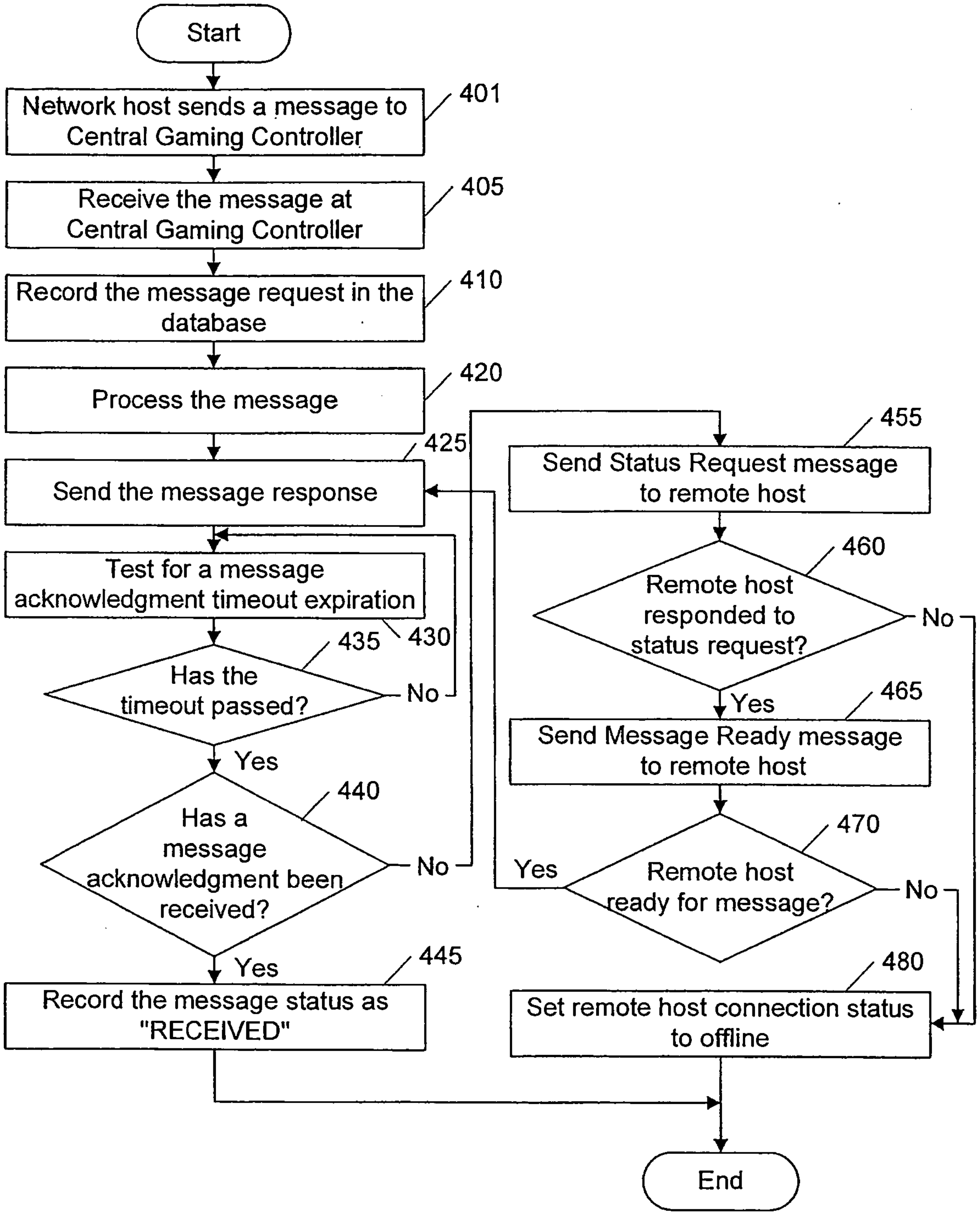

Start
Network host sends a message to Central Gaming Controller
401
Receive the message at Central Gaming Controller
405
Record the message request in the database
410
Process the message
420
Send the message response
425
Test for a message acknowledgment timeout expiration
430
Has the timeout passed?
435
No
Yes
Has a message acknowledgment been received?
440
No
Yes
Record the message status as "RECEIVED"
445
Send Status Request message to remote host
455
Remote host responded to status request?
460
No
Yes
Send Message Ready message to remote host
465
Remote host ready for message?
470
Yes
No
Set remote host connection status to offline
480
End

FIG. 4

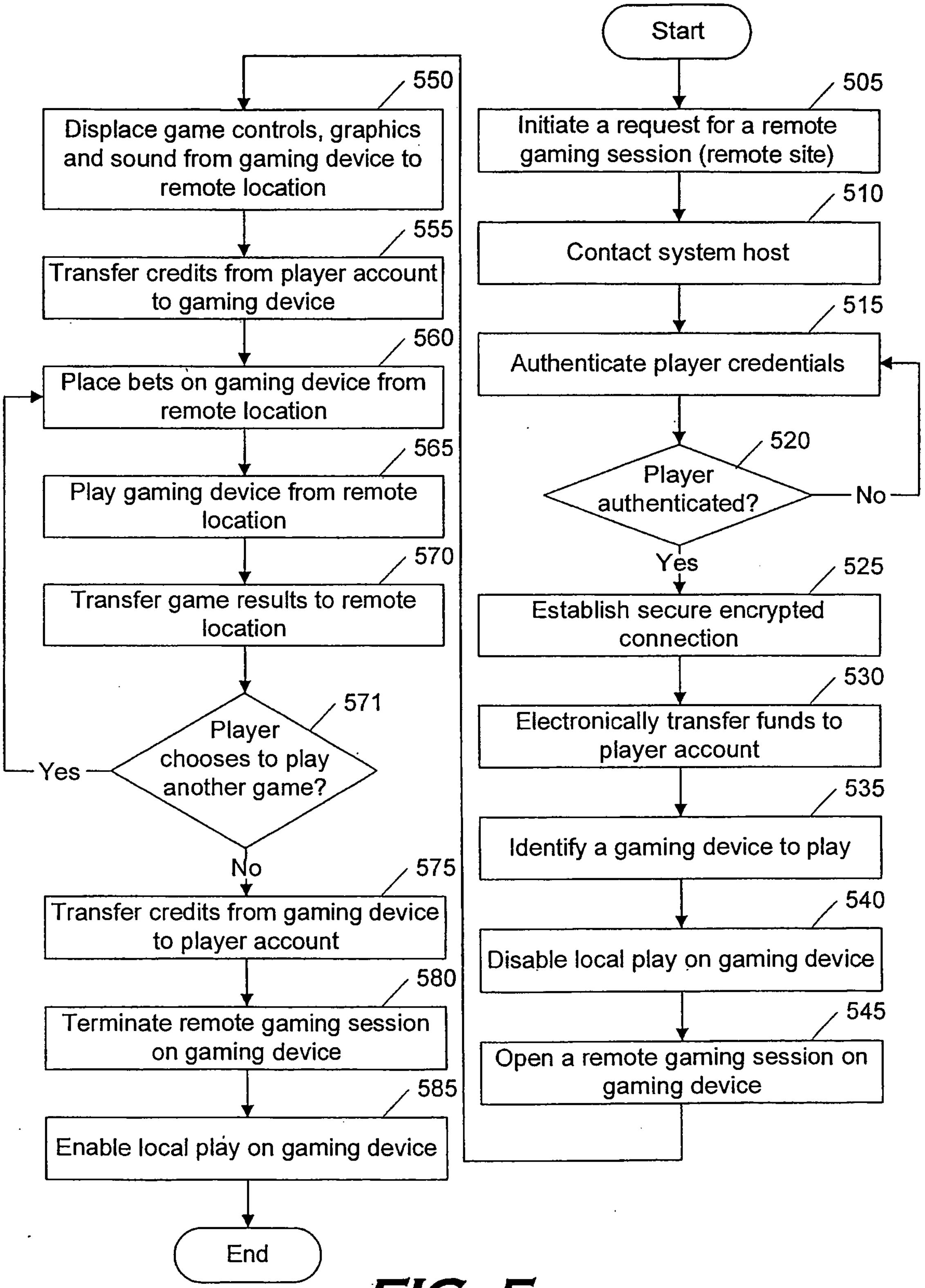

Start
505
Initiate a request for a remote gaming session (remote site)
510
Contact system host
515
Authenticate player credentials
520
Player authenticated?
No
Yes
525
Establish secure encrypted connection
530
Electronically transfer funds to player account
535
Identify a gaming device to play
540
Disable local play on gaming device
545
Open a remote gaming session on gaming device
550
Displace game controls, graphics and sound from gaming device to remote location
555
Transfer credits from player account to gaming device
560
Place bets on gaming device from remote location
565
Play gaming device from remote location
570
Transfer game results to remote location
571
Player chooses to play another game?
Yes
No
575
Transfer credits from gaming device to player account
580
Terminate remote gaming session on gaming device
585
Enable local play on gaming device
End

FIG. 5

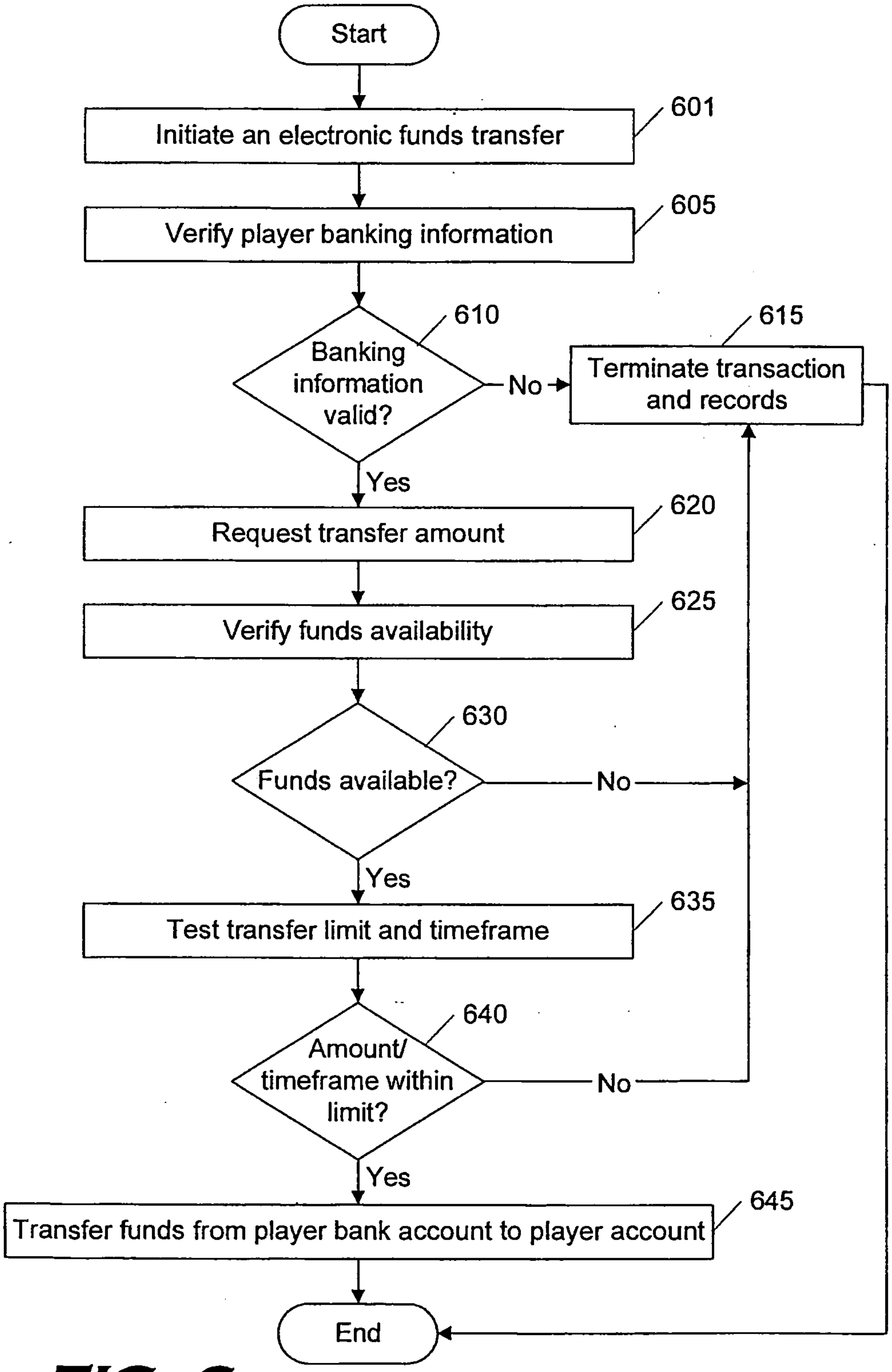

Start
Initiate an electronic funds transfer
601
Verify player banking information
605
Banking information valid?
610
No
Terminate transaction and records
615
Yes
Request transfer amount
620
Verify funds availability
625
Funds available?
630
No
Yes
Test transfer limit and timeframe
635
Amount/ timeframe within limit?
640
No
Yes
Transfer funds from player bank account to player account
645
End

FIG. 6

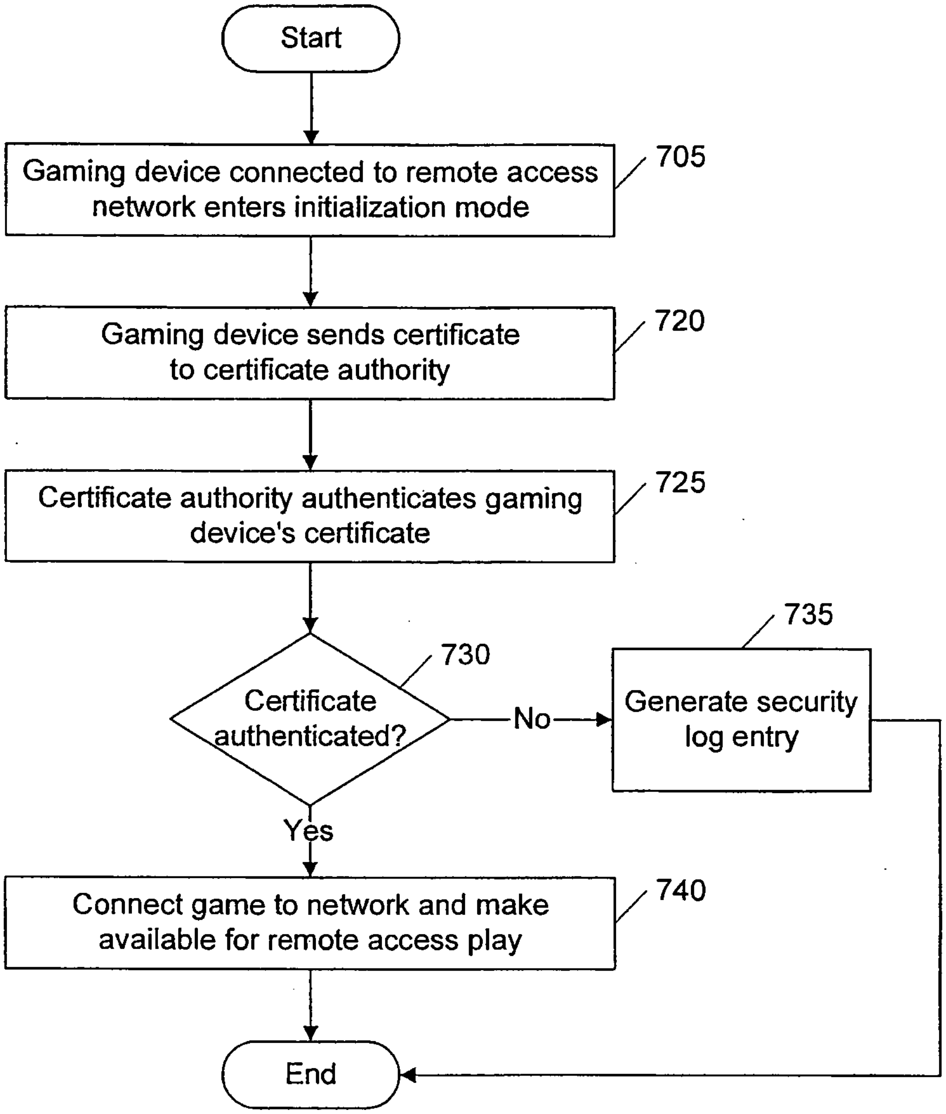

Start
Gaming device connected to remote access network enters initialization mode
705
Gaming device sends certificate to certificate authority
720
Certificate authority authenticates gaming device's certificate
725
Certificate authenticated?
730
No
Generate security log entry
735
Yes
Connect game to network and make available for remote access play
740
End

FIG. 7

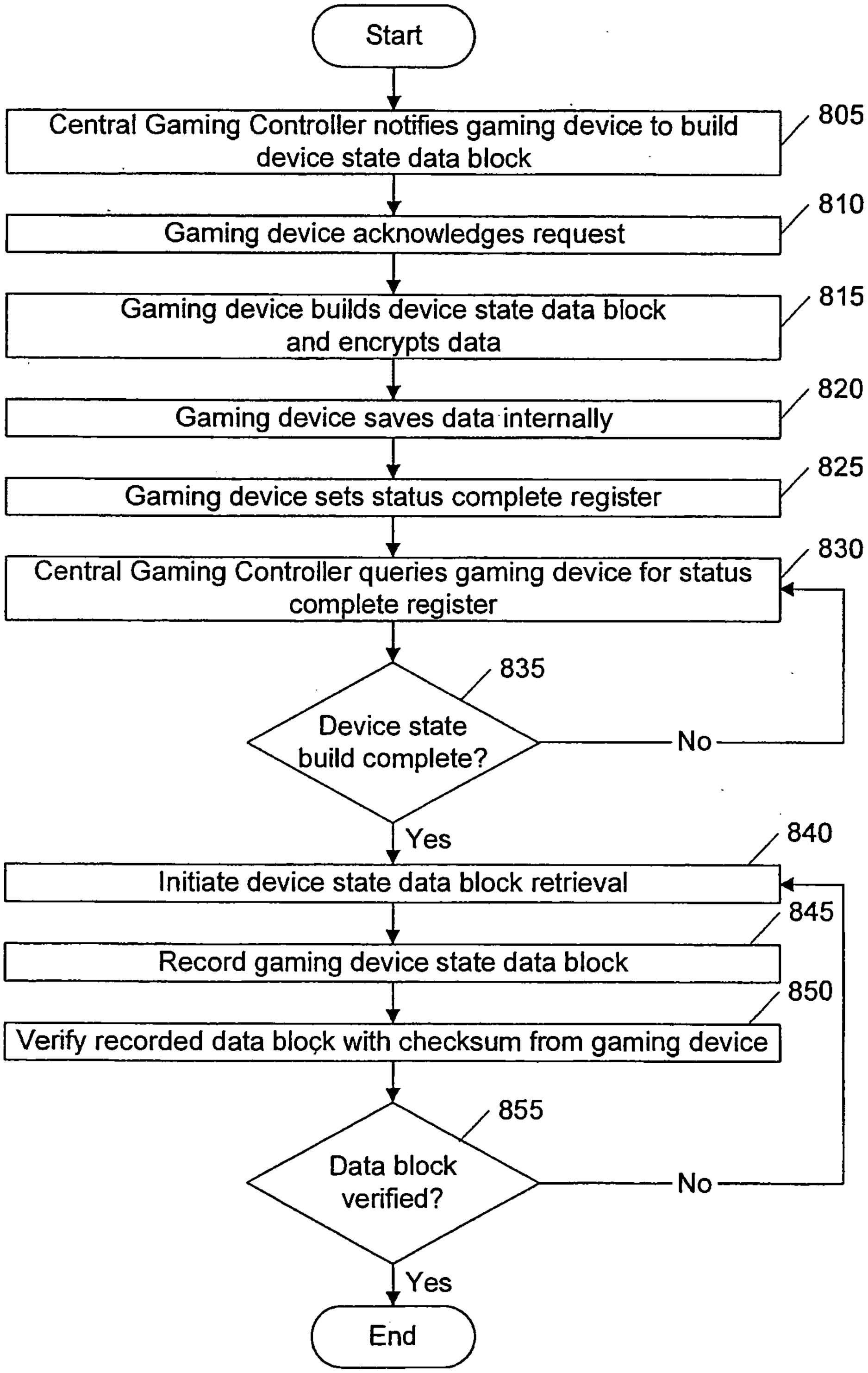

Start
Central Gaming Controller notifies gaming device to build device state data block
805
Gaming device acknowledges request
810
Gaming device builds device state data block and encrypts data
815
Gaming device saves data internally
820
Gaming device sets status complete register
825
Central Gaming Controller queries gaming device for status complete register
830
Device state build complete?
835
No
Yes
Initiate device state data block retrieval
840
Record gaming device state data block
845
Verify recorded data block with checksum from gaming device
850
Data block verified?
855
No
Yes
End

FIG. 8

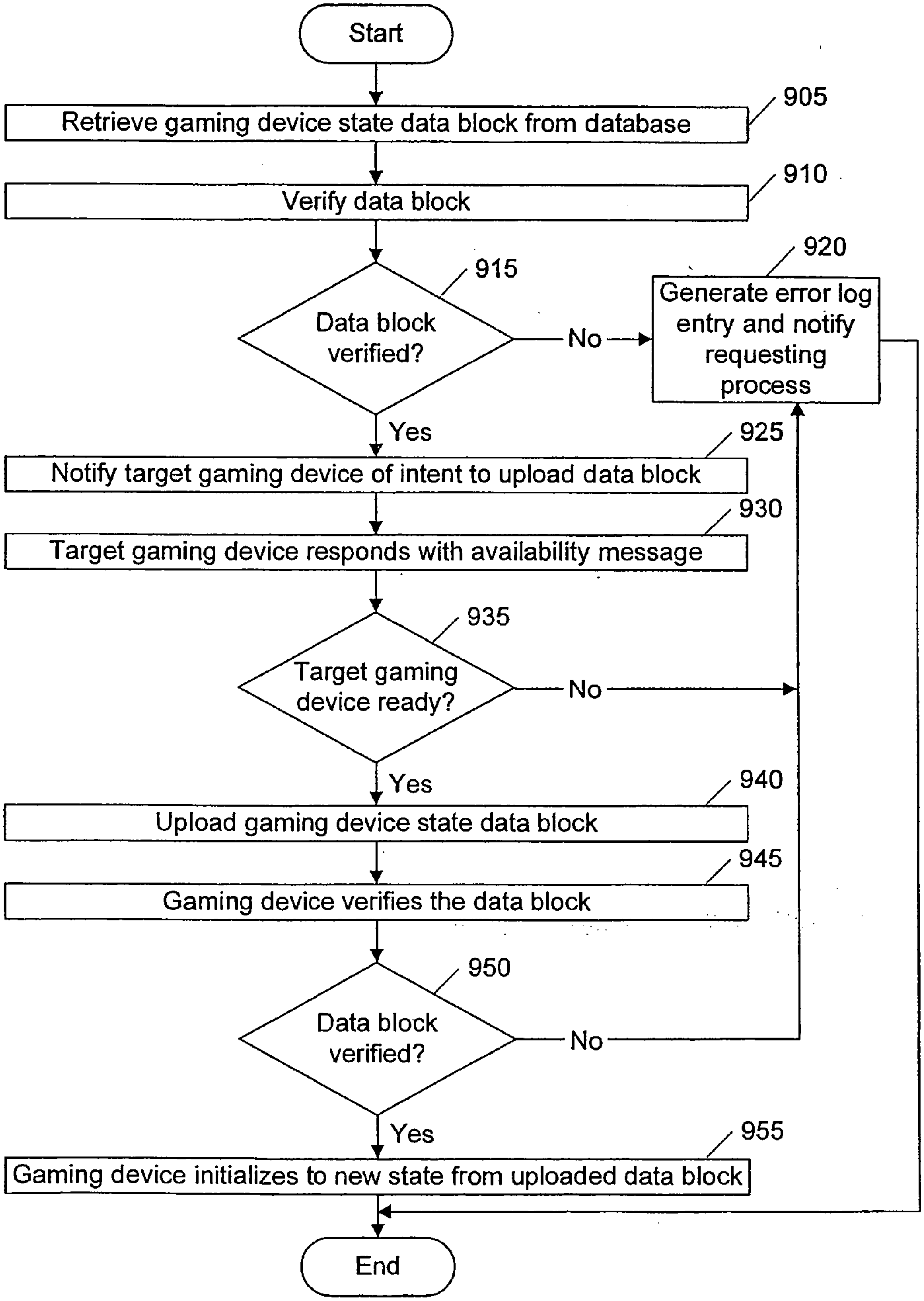

Start
Retrieve gaming device state data block from database
905
Verify data block
910
Data block verified?
915
No
Generate error log entry and notify requesting process
920
Yes
Notify target gaming device of intent to upload data block
925
Target gaming device responds with availability message
930
Target gaming device ready?
935
No
Yes
Upload gaming device state data block
940
Gaming device verifies the data block
945
Data block verified?
950
No
Yes
Gaming device initializes to new state from uploaded data block
955
End

FIG. 9

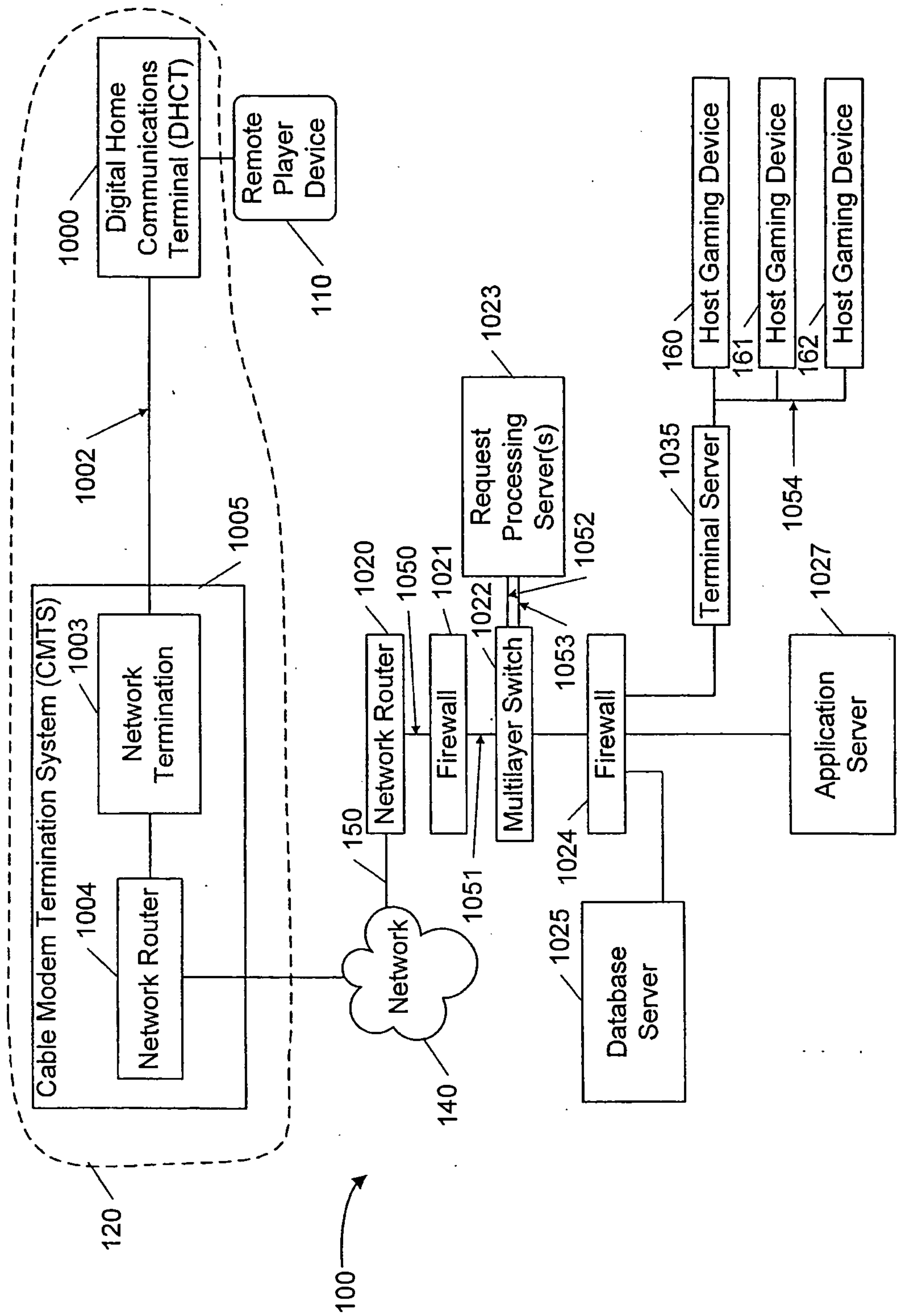

Digital Home Communications Terminal (DHCT)
Remote Player Device
Cable Modem Termination System (CMTS)
Network Termination
Network Router
Network
Network Router
Firewall
Multilayer Switch
Request Processing Server(s)
Firewall
Database Server
Terminal Server
Host Gaming Device
Host Gaming Device
Host Gaming Device
Application Server

FIG. 10

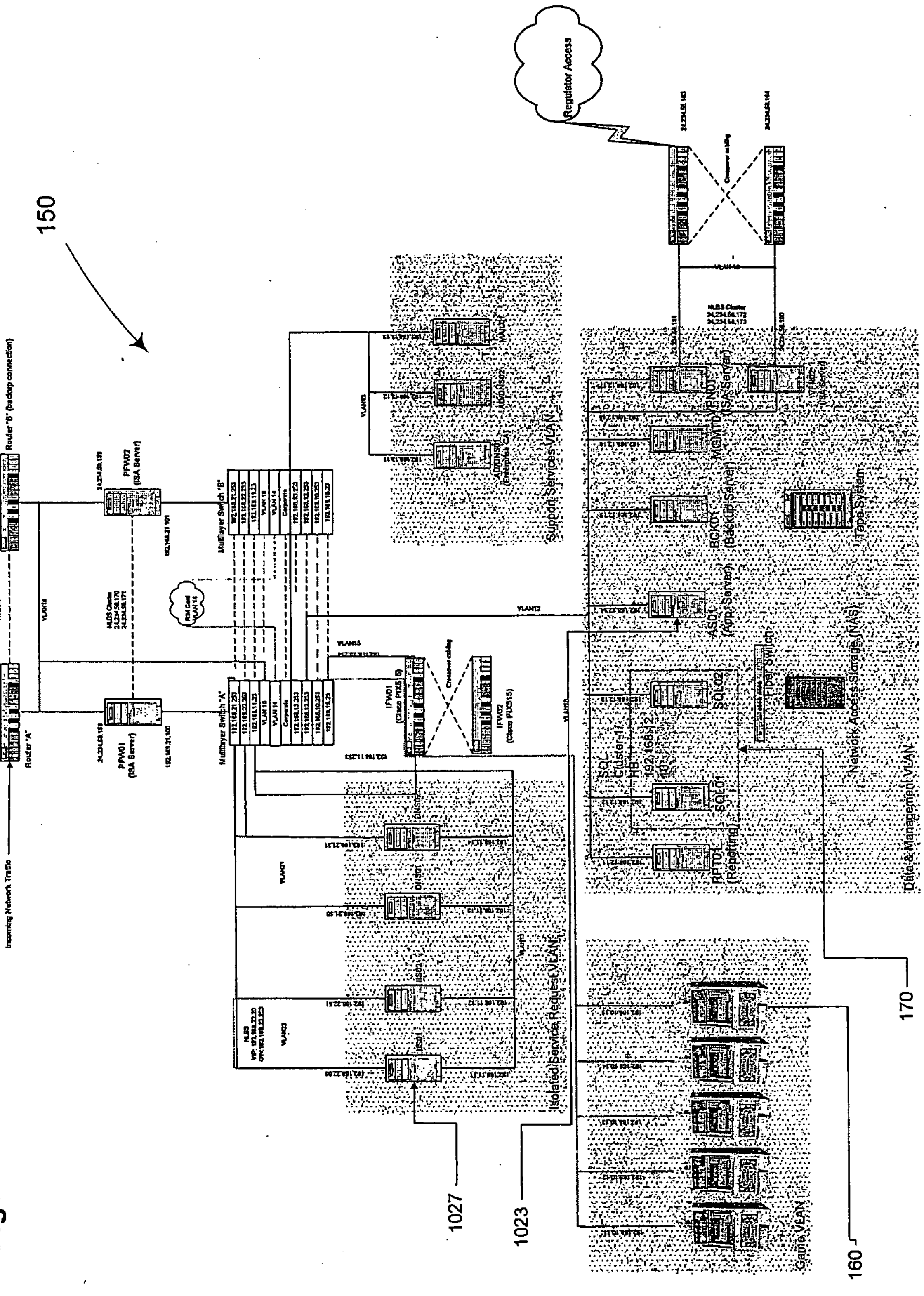

150
Incoming Network Traffic
Router "A"
Router "B" (backup connection)
PFW01 (ISA Server)
PFW02 (ISA Server)
Multilayer Switch "A"
Multilayer Switch "B"
1FW01 (Cisco PIX515)
1FW02 (Cisco PIX515)
Crossover cabling
Support Services VLAN
Isolated Service Request VLAN
Game VLAN
Data & Management VLAN
Regulator Access
NLBS Cluster
AS01 (App Server)
BCK01 (Backup Server)
MGMT01 (ISA Server)
SQL01
SQL02
SQL Cluster
RPT01 (Reporting)
Fiber Switch
Network Access Storage (NAS)
Tape System
1027
1023
160
170

Figure 11

⑲ **Europäisches Patentamt**
European Patent Office
Office européen des brevets

⑪ Publication number: **0 257 585**
A2

⑫ EUROPEAN PATENT APPLICATION

㉑ Application number: **87112158.8**

㉒ Date of filing: **21.08.87**

(51) Int. Cl.⁴: **H04L 9/00**

㉚ Priority: **22.08.86 JP 197610/86**
22.08.86 JP 197611/86

(43) Date of publication of application:
02.03.88 Bulletin 88/09

(84) Designated Contracting States:
BE DE FR GB

(71) Applicant: **NEC CORPORATION**
33-1, Shiba 5-chome, Minato-ku
Tokyo 108(JP)

(72) Inventor: **Okamoto, Eiji c/o NEC Corporation**
33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(74) Representative: **Vossius & Partner**
Siebertstrasse 4 P.O. Box 86 07 67
D-8000 München 86(DE)

(54) Key distribution method.

(57) The invention relates to a method of distributing a key for enciphering un unenciphered or plaintext message and for deciphering the enciphered message.

The method comprises the following steps: generating a first random number in a first system (101); generating first key distribution information in the first system (101) by applying a predetermined first transformation to the first random number on the basis of first secret information known only by the first system (101); transmitting the first key distribution information to a second system (102) via a communication channel (103); receiving the first key distribution information in the second system (102); generating a second random number in the second system (102); generating second key distribution information by applying the predetermined first transformation to the second random number on the basis of second secret information known only by the second system (102); transmitting the second key distribution information to the first system (101) via the channel (103); receiving the second key distribution information in the first system (101); and generating an enciphering key in the first system (101) by applying a predetermined second transformation to the second key distribution information on the basis of the first random number and identification information of the second system (102) which is not secret.

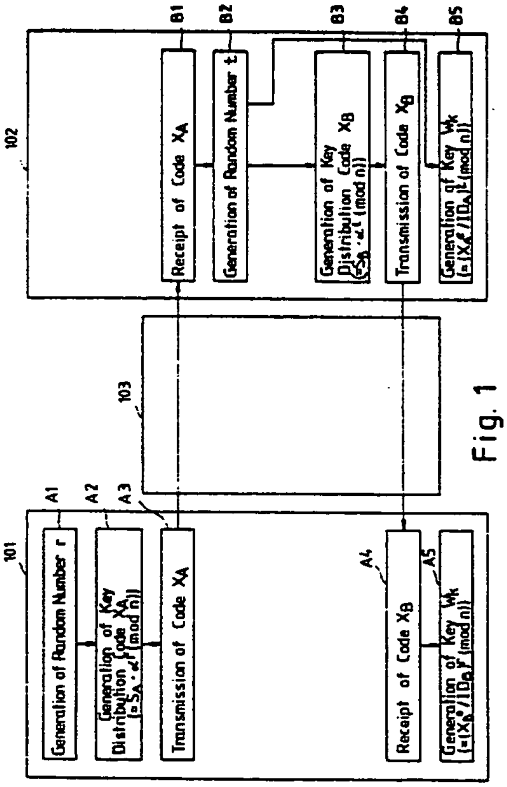

B1
B2
B3
B4
B5
102
Receipt of Code X_A
Generation of Random Number t
Generation of Key Distribution Code X_B (=$S_B \cdot \alpha^t$ (mod n))
Transmission of Code X_B
Generation of Key W_K (=$(X_A^e/ID_A)^t$ (mod n))
103
101
A1
A2
A3
A4
A5
Generation of Random Number r
Generation of Key Distribution Code X_A (=$S_A \cdot \alpha^r$ (mod n))
Transmission of Code X_A
Receipt of Code X_B
Generation of Key W_K (=$(X_B^e/ID_B)^r$ (mod n))

Fig. 1

EP 0 257 585 A2

KEY DISTRIBUTION METHOD

BACKGROUND OF THE INVENTION

The invention relates to a method of distributing a key for enciphering an unenciphered or plaintext message and for deciphering the enciphered message.

A public key distribution method used in a public key cryptosystem as a well-known key distribution method is disclosed in a paper entitled "New Directions in Cryptography" by W. Diffie and M.E. Hellman, published in the IEEE Transactions on Information Theory, Vol. IT-22, No. 6, pp. 644 to 654, November issue, 1976. The key distribution method disclosed in the paper memorizes public information for each of conversers. In the system, before a converser A sends an enciphered message to a converser B, the converser A prepares an enciphering key (which represents a number obtained by calculating $Y_B{}^{X_A}$ (mod p)) generated from public information Y_B of the converser B and secret information X_A which is kept secret by the converser A. The number p is a large prime number of about 256 bits in binary representation, which is publicly known. a (mod b) means a remainder of division of the number a by the number b. The converser B also prepares the key wk in accordance to $Y_A{}^{X_B}$ (mod p) in a similar manner. Y_A and Y_B are selected so as to be equal to α^{X_A} (mod p) and α^{X_B} (mod p), respectively. As a result, $Y_B{}^{X_A}$ (mod p) becomes equal to $Y_A{}^{X_B}$ (mod p). It is known that even if Y_A, a and p are known, it is infeasible for anybody except the converser A to obtain X_A which satisfies $Y_A = \alpha^{X_A}$ (mod p).

The prior art key distribution system of the type described, however, has disadvantages in that since the system needs a large amount of public information corresponding to respective conversers, the amount of the public information increases as the number of conversers increases. Further, strict control of such information becomes necessary to prevent the information from being tampered.

SUMMARY OF THE INVENTION

An object of the invention is, therefore, to provide a key distribution method free from the above-mentioned disadvantages of the prior art system.

According to an aspect of the invention, there is provided a method which comprises the following steps: generating a first random number in a first system; generating first key distribution information in the first system by applying a predetermined first transformation to the first random number on the basis of first secret information known only by the first system; transmitting the first key distribution information to a second system via a communication channel; receiving the first key distribution information in the second system; generating a second random number in the second system; generating second key distribution information by applying the predetermined first transformation to the second random number on the basis of second secret information known only by the second system; transmitting the second key distribution information to the first system via the channel; receiving the second key distribution information in the first system; and generating an enciphering key in the first system by applying a predetermined second transformation to the second key distribution information on the basis of the first random number and identification information of the second system which is not secret.

According to another aspect of the invention, there is provided a method which comprises the following steps: generating a first random number in the first system; generating first key distribution information by applying a predetermined first transformation to the first random number on the basis of public information in the first system and generating first identification information by applying a predetermined second transformation to the first random number on the basis of first secret information known only by the first system; transmitting the first key distribution information and the first identification information to a second system via a communication channel; receiving the first key distribution information and the first identification information in the second system; examining whether or not the result obtained by applying a predetermined third transformation to the first key distribution information on the basis of the first identification information satisfies a first predetermined condition, and, if it does not satisfy, suspending key distribution processing; generating a second random number if said condition is satisfied in the preceding step; generating second key distribution information by applying the predetermined first transformation to the second random number on the basis of the public information, and generating second identification information by applying the predetermined second transformation to the second random number on the basis of second secret information known only by the second system; transmitting the second key distribution information and the second identification information to the first system via the communication channel; and examining whether or not the result obtained by applying a third predetermined transformation to the second key distribution information on the basis of the second identification information in the first system satisfies a predetermined second condition, and if the result does not satisfy the second condition, suspending the key distribution processing, or if it satisfies the second condition, generating an enciphering key by applying a fourth predetermined transformation to the first random number on the basis of the second key distribution information.

BRIEF DESCRIPTION OF THE DRAWINGS

Other features and advantages of the invention will become more apparent from the following detailed description when taken in conjunction with the accompanying drawings in which:

FIG. 1 is a block diagram of a first embodiment of the invention;

FIG. 2 is a block diagram of a second embodiment of the invention; and

FIG. 3 is a block diagram of an example of systems 101, 102, 201 and 202.

In the drawings, the same reference numerals represent the same structural elements.

PREFERRED EMBODIMENTS

Referring now to FIG. 1, a first embodiment of the invention comprises a first system 101, a second system 102 and an insecure communication channel 103 such as a telephone line which transmits communication signals between the systems 101 and 102. It is assumed herein that the systems 101 and 102 are used by users or conversers A and B, respectively. The user A has or knows a secret integer number S_A and public integer numbers $\underline{e}$, $\underline{c}$, $\underline{\alpha}$ and $\underline{n}$ which are not necessarily secret while the user B has or knows a secret integer number $S_B$and the public integer numbers. These integer numbers are designated and distributed in advance by a reliable person or organization. The method to designate the integer numbers will be described later.

An operation of the embodiment will next be described on a case in which the user A starts communication. The system 101 of the user A generates a random number $\underline{\gamma}$ (Step A1 in FIG. 1) and sends a first key distribution code X_A representative of a number obtained by computing $S_A \bullet \alpha^{\gamma}$ (mod $\underline{n}$) (Step A2) to the system 102 of the user B (step A3). Next, when the system 102 receives the code X_A(Step B1), it generates a random number $\underline{t}$ (Step B2), calculates $(X_A{}^e/ID_A)^t$ (mod $\underline{n}$) (Step B5), and keeps the resulting number as a enciphering key wk for enciphering a message into storage means (not shown). The identification code ID_A represents herein a number obtained by considering as a numeric value a code obtained by encoding the address, the name and so on of the user A. The encoding is, for instance, performed on the basis of the American National Standard Code for Information Interchange. Then, the system 102 transmits to the system 101 of the user A a second key distribution code X_B representative of a number obtained by calculating $S_B \bullet \alpha^t$ (mod $\underline{n}$) (Steps B3 and B4).

The system 101, on the other hand, receives the code X_B (Step A4), calculates $(X_B{}^e/ID_B)^{\gamma}$ (mod $\underline{n}$) (Step A5), and keeps the resulting number as the key wk for enciphering a message. The identification code ID_B represents the numbers obtained by considering as a numeric value a code obtained by encoding the name, address, and so on of the user B.

Subsequently, communication between the users A and B will be conducted by transmitting messages enciphered with the enciphering key wk via the channel 103.

The integer numbers S_A, S_B, $\underline{e}$, $\underline{c}$, $\underline{\alpha}$ and $\underline{n}$ are determined as follows. $\underline{n}$ is assumed to be a product of two sufficiently large prime numbers $\underline{p}$ and $\underline{q}$. For instance, $\underline{p}$ and $\underline{q}$ may be 2^{256} or so. $\underline{e}$ and $\underline{c}$ are prime numbers which are equal to or less than $\underline{n}$, while $\underline{\alpha}$ is a positive integer number which is equal to or less than $\underline{n}$. Further, $\underline{d}$ is defined as an integer number which satisfies e.d (mod (p-1)•(q-1)) = 1. S_A and S_B are defined as numbers obtainable from $ID_A{}^d$ (mod $\underline{n}$) and $ID_B{}^d$ (mod $\underline{n}$), respectively.

If S_A, S_B, $\underline{e}$, $\underline{c}$, $\underline{\alpha}$, and $\underline{n}$ are defined as above, ID_A and ID_B become equal to $S_A{}^e$ (mod n) and $S_B{}^e$(mod $\underline{n}$), respectively. This can be proved from a paper entitled "A Method for Obtaining Digital Signatures and Publick-Key Cryptosystems" by R.L. Rivest et al., published in the Communication of the ACM, Vol. 21, No. 2, pp. 120 to 126. Since the key obtained by $(X_B{}^e/ID_B)^r$ (mod $\underline{n}$) on the side of the user A becomes equal to α^{ert} (mod $\underline{n}$) and the key obtained by $(X_A{}^e/ID_A)^t$ (mod $\underline{n}$) on the side of the user B becomes equal to α^{ert} (mod $\underline{n}$), they can prepare the same enciphering key. Even if a third party tries to assume the identity of the user A, he cannot prepare the key wk since he cannot find out $\underline{z}$ which meets $ID_A = Z^e$ (mod $\underline{n}$).

Referring now to FIG. 2, a second embodiment of the invention comprises a first system 201, a second system 202 and an insecure communication channel 203. It is assumed herein that the systems 201 and 202 are used by users A and B, respectively. The user A has or knows a secret integer number S_A and public integer numbers $\underline{e}$, $\underline{c}$, $\underline{\alpha}$, and $\underline{n}$, which are not necessarily secret while the user B has or knows a secret integer number $S_B$and the public integer numbers. These integer numbers are designated and distributed by a reliable person or organization in advance. The method to designate the integer numbers will be described later.

An operation of the embodiment will next be described on a case where the user A starts communication. The system 201 of the user A generates a random number $\underline{\gamma}$ (Step AA1 in FIG. 2) and determines a first key distribution code X_A representative of a number obtained by computing $\alpha^{e \cdot r}$ (mod $\underline{n}$) as well as a first identification code $Y_A$indicative of a number obtained by computing $S_A \bullet \alpha^{c \cdot r}$(mod $\underline{n}$) (AA2). The system 201 then transmits a first pair of X_A and Y_A to the system 202 of the user B (Step AA3). Thereafter, the system 202 receives the first pair (X_A , Y_A) (Step BB1), calculates $Y_A{}^e / X_A{}^c$ (mod $\underline{n}$, and examines whether or not the number obtained by the calculation is identical to the number indicated by an identification code ID_A obtained by the address, the name and so on of the user A in a similar manner to in the first embodiment (Step BB2). If they are not identical to each other, the system suspends processing of the key distribution (Step BB7). On the other hand, if they are identical to each other, the system 202 generates a random number $\underline{t}$ (Step BB3) and determines a second key distribution code X_B representative of a number obtained by calculating $\alpha^{e \cdot t}$ (mod $\underline{n}$) and a second identification code Y_B obtained by calculating $S_B \bullet \alpha^{c \cdot t}$ (mod $\underline{n}$) (Step BB4). The system 202 then transmits a second pair of X_B and Y_B to the system 201 of the user A (Step BB5). The system 202 calculates $X_A{}^t$ (mod $\underline{n}$) and keeps the number thus obtained as a enciphering key wk (Step BB6).

The system 201, on the other hand, receives the second pair (X_B, Y_B) (Step AA4), calculates $Y_B{}^e / X_B{}^c$ (mod $\underline{n}$), and examines whether or not the number thus obtained is identical to the number indicated by an identification code ID_B obtained by the address, the name and so on of the user B in a similar manner to in the first embodiment (Step AA5). If they are not identical to each other, the system suspends the key distribution processing (Step AA7). If they are identical to each other, the system 201 calculates $X_B{}^r$ mod $\underline{n}$), and stores the number thus obtained as a enciphering key wk (Step AA6). Although the codes ID_A and ID_B are widely known, they may be informed by the user A to the user B.

The integer numbers S_A, S_B, $\underline{e}$, $\underline{c}$, $\underline{\alpha}$ and $\underline{n}$ are determined in the same manner as in the first embodiment. As a result, ID_A and ID_B becomes equal to $Y_A{}^e / X_A{}^c$ (mod $\underline{n}$) (= $S_A^e \bullet \alpha^{erc} / \alpha^{erc}$ (mod $\underline{n}$)) and $Y_B{}^e / X_B{}^c$ (mod $\underline{n}$) (= $S_B^e \bullet \alpha^{etc} / \alpha^{etc}$ (mod $\underline{n}$)), respectively. If we presuppose that the above-mentioned reliable person or organization who prepared S_A and S_B do not act illegally, since S_A is possessed only by the user A while S_B is possessed only by the user B, the first pair (x_A, y_A) which satisfies $y_A{}^e / x_A{}^c$ (mod $\underline{n}$) = ID_A can be prepared only by the user A while the second pair (x_B , y_B) which satisfies $y_B{}^e / x_B{}^c$ (mod $\underline{n}$) = ID_B can be prepared only by the user B. It is impossible to find out a number $\underline{x}$ which satisfies $\underline{x}^f$ (mod $\underline{n}$) = $\underline{b}$ on the basis of $\underline{f}$, $\underline{b}$and $\underline{n}$ since finding out $\overline{X}$ is equivalent to breaking the RSA public key cryptogram system disclosed in the above-mentioned the Communication of the ACM. It is described in the above-referenced IEEE Transactions on Information Theory that the key wk cannot be calculated from the codes x_A or x_B and $\underline{n}$. The key distribution may be implemented similarly by making the integer number $\underline{C}$ variable and sending it from a user to another.

An example of the systems 101, 102, 201 and 202 to be used in the first and second embodiments will next be described referring to FIG. 3.

Referring now to FIG. 3, a system comprises a terminal unit (TMU) 301 such as a personal computer equipped with communication processing functions, a read only memory unit (ROM) 302, a random access memory unit (RAM) 303, a random number generator (RNG) 304, a signal processor (SP) 306, and a common bus 305 which interconnects the TMU 301, the ROM 302, the RAM 303, the RNG 304 and the SP 306.

The RNG 304 may be a key source 25 disclosed in U.S. Patent No. 4,200,700. The SP 306 may be a processor available from CYLINK Corporation under the trade name CY 1024 KEY MANAGEMENT PROCESSOR.

The RNG 304 generates random numbers $\underline{r}$ or $\underline{t}$ by a command given from the SP 306. The ROM 407 stores the public integer numbers $\underline{e}$, $\underline{c}$, $\underline{\alpha}$, $\underline{n}$ and the secret integer number S_A (if the ROM 407 is used in the system 101 or 201) or the secret integer number S_B (if the ROM 407 is used in the system 102 or 202). The numbers S_A and S_B may be stored in the RAM 303 from the TMU 301 everytime users communicates. According to a program stored in the ROM 407, the SP 306 executes the above-mentioned steps A2, A5, AA2, AA5, AA6 and AA7 (if the SP 306 is used in the system 101 or 201), or the steps B3, B5, BB2, BB4, BB6 and BB7 (if the SP 306 is used in the system 102 or 202). The RAM 303 is used to temporarily store calculation results in these steps.

Each of the systems 101, 102, 201 and 202 may be a data processing unit such as a general purpose computer and an IC (integrated circuit) card.

As described in detail hereinabove, this invention enables users to effectively implement key distribution simply with a secret piece of information and several public pieces of information.

While this invention has thus been described in conjunction with the preferred embodiments thereof, it will now readily be possible for those skilled in the art to put this invention into practice in various other manners.

Claims

1. A key distribution method comprising the following steps:

a) generating a first random number in a first system;

b) generating first key distribution information in said first system by applying a predetermined first transformation to said first random number on the basis of first secret information known only by said first system;

c) transmitting said first key distribution information to a second system via a communication channel;

d) receiving said first key distribution information in said second system;

e) generating a second random number in said second system;

f) generating second key distribution information by applying said predetermined first transformation to said second random number on the basis of second secret information known only by said second system;

g) transmitting said second key distribution information to said first system via said channel;

h) receiving said second key distribution information in said first system; and

i) generating an enciphering key in said first system by applying a predetermined second transformation to said second key distribution information on the basis of said first random number and identification information of said second system which is not secret.

2. A key distribution method as claimed in Claim 1, in which said first system includes first data processing means for executing said steps a), b) and i), and first communication processing means for executing said steps c) and h).

3. A key distribution method as claimed in Claim 1 or 2, in which said second system includes second data processing means for executing said steps e) and f), and second communication processing means for executing said steps d) and g).

4. A key distribution method comprising the following steps:

a) generating a first random number in a first system;

b) generating first key distribution information in said first system by applying a predetermined first transformation to said first random number on the basis of public information and generating first identification information by applying a predetermined second transformation to said first random number on the basis of first secret information known only by said first system;

c) transmitting said first key distribution information and said first identification information to a second system via a communication channel;

d) receiving said first key distribution information and said first identification information in said second system;

e) examining whether or not the result obtained by applying a predetermined third transformation to said first key distribution information on the basis of said first identification information satisfies a predetermined first condition and, if it does not satisfy, suspending key distribution processing;

f) generating a second random number if said first condition is satisfied at said step e);

g) generating second key distribution information by applying said predetermined first transformation to said second random number on the basis of said public information, and generating second identification information by applying said predetermined second transformation to said second random number on the basis of second secret information known only by said second system;

h) transmitting said second key distribution information and said second identification information to said first system via said communication channel; and

i) examining in said first system whether or not the result obtained by applying a predetermined third transformation to said second key distribution information on the basis of said second identification information satisfies a predetermined second condition and, if the result does not satisfy said second condition, suspending said key distribution processing or, if it satisfies said second condition, generating said enciphering key by applying a predetermined fourth transformation to said first random number on the basis of said second key distribution information.

5. A key distribution method as claimed in Claim 4, in which said first system includes first data processing means for executing said steps a), b) and i), and first communication processing means for executing said step c).

6. A key distribution method as claimed in Claim 4 or 5, in which said second system includes second data processing means for executing said steps e), f) and g), and second communication processing means for executing said steps d) and h).

22. Dez. 1987

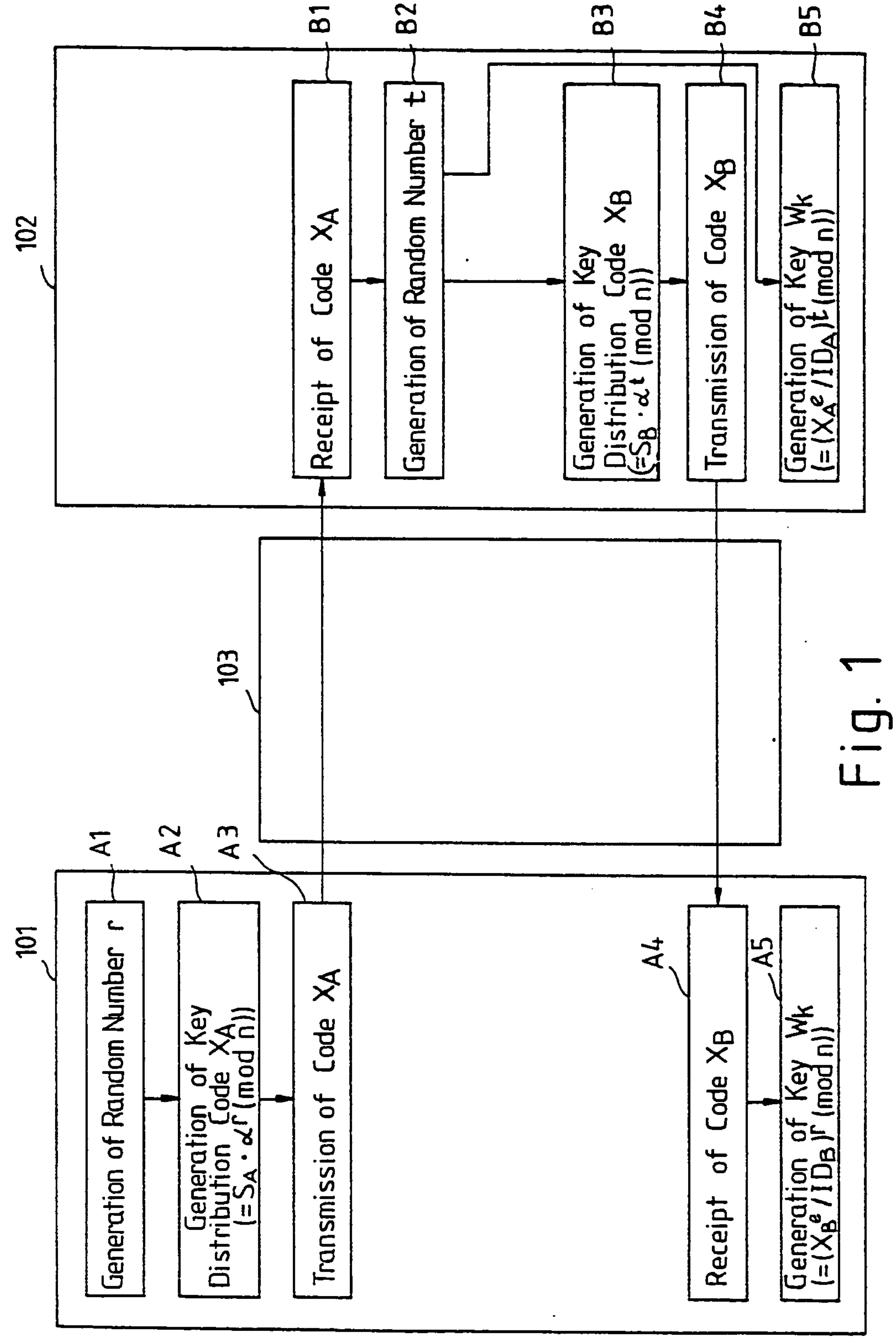

101
A1
Generation of Random Number r
A2
Generation of Key Distribution Code X_A $(=S_A \cdot \alpha^r \pmod n)$
A3
Transmission of Code X_A
A4
Receipt of Code X_B
A5
Generation of Key W_K $(=(X_B^e/ID_B)^r \pmod n)$
103
102
B1
Receipt of Code X_A
B2
Generation of Random Number t
B3
Generation of Key Distribution Code X_B $(=S_B \cdot \alpha^t \pmod n)$
B4
Transmission of Code X_B
B5
Generation of Key W_K $(=(X_A^e/ID_A)^t \pmod n)$

Fig. 1

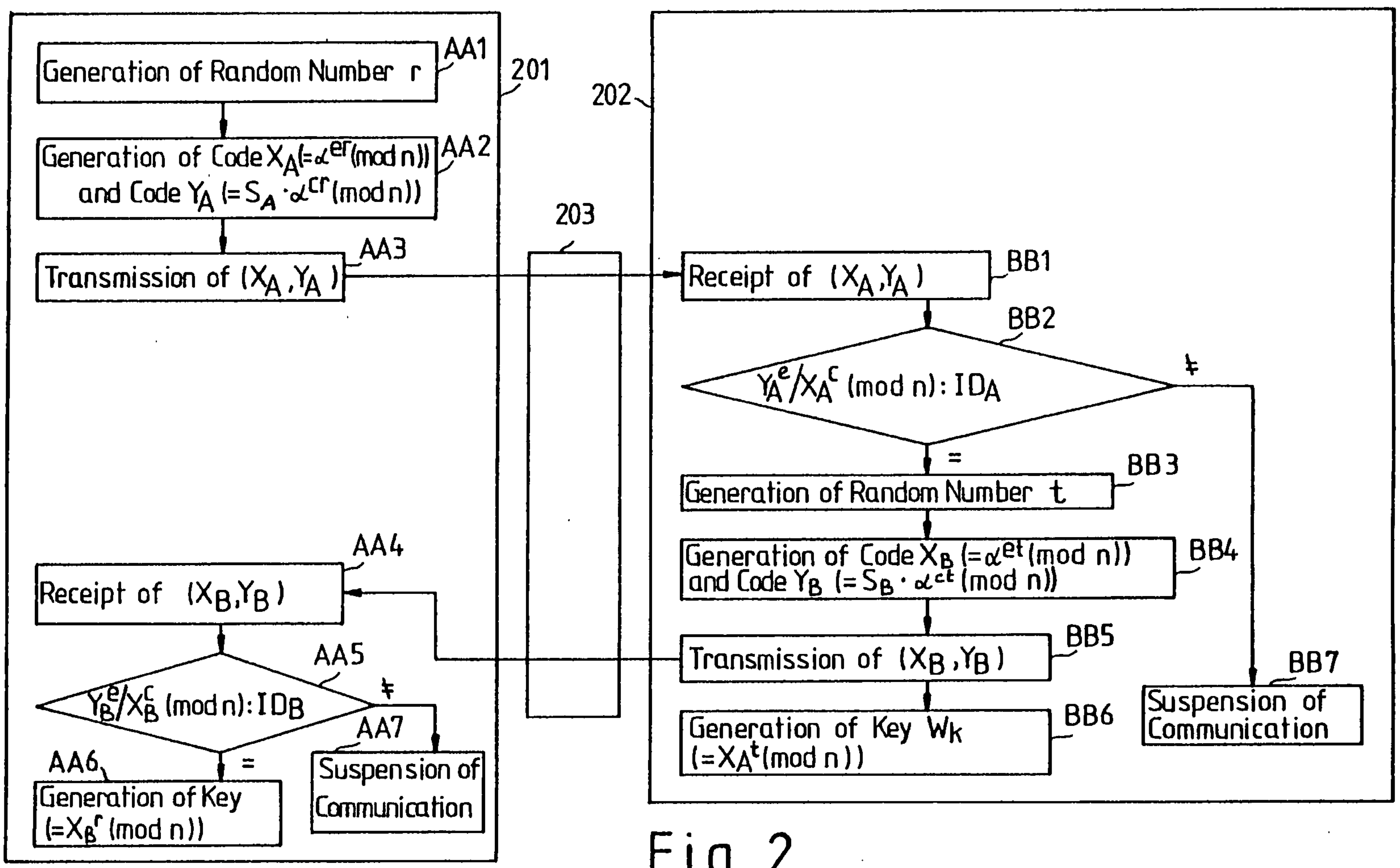

Generation of Random Number r
AA1
201
Generation of Code X_A (= α^{er} (mod n)) and Code Y_A (= $S_A \cdot \alpha^{cr}$ (mod n))
AA2
Transmission of (X_A, Y_A)
AA3
203
202
Receipt of (X_A, Y_A)
BB1
BB2
Y_A^e / X_A^c (mod n) : ID_A
≠
=
Generation of Random Number t
BB3
Generation of Code X_B (= α^{et} (mod n)) and Code Y_B (= $S_B \cdot \alpha^{ct}$ (mod n))
BB4
Transmission of (X_B, Y_B)
BB5
Generation of Key W_K (= X_A^t (mod n))
BB6
BB7
Suspension of Communication
AA4
Receipt of (X_B, Y_B)
AA5
Y_B^e / X_B^c (mod n) : ID_B
≠
AA7
Suspension of Communication
AA6
=
Generation of Key (= X_B^r (mod n))

Fig. 2

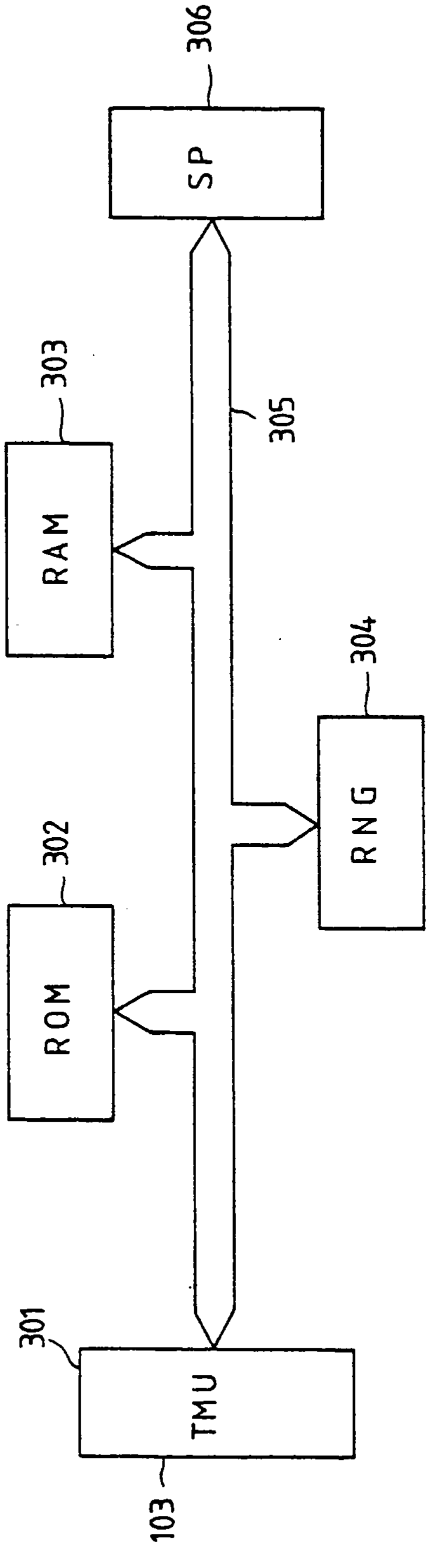

SP
306
RAM
303
305
RNG
304
ROM
302
TMU
301
103

Fig. 3

⑲ Europäisches Patentamt
European Patent Office
Office européen des brevets

⑪ Publication number: 0 332 304 A2

⑫ **EUROPEAN PATENT APPLICATION**

㉑ Application number: 89301510.7

(51) Int. Cl.4: G06F 1/00

㉒ Date of filing: 16.02.89

㉚ Priority: 07.03.88 US 164944

㊸ Date of publication of application:
13.09.89 Bulletin 89/37

(84) Designated Contracting States:
DE FR GB

(71) Applicant: DIGITAL EQUIPMENT CORPORATION
111 Powdermill Road
Maynard Massachusetts 01754-1418(US)

(72) Inventor: Robert, Gregory
12 Carson Circle
Nashua New Hampshire 03062(US)
Inventor: Chase, David
28 Bay View Road
Wellesley Massachusetts 02181(US)
Inventor: Schaefer, Ronald
7 Gioconda Avenue
Acton Massachusetts 01720(US)

(74) Representative: Goodman, Christopher et al
Eric Potter & Clarkson 14 Oxford Street
Nottingham NG1 5BP(GB)

(54) Software licensing management system.

(57) A license management system which includes a license management facility that determines whether usage of a licensed program is within the scope of the license. The license management system maintains a license unit value for each licensed program and a pointer to a table identifying an allocation unit value associated with each use of the licensed program. In response to a request to use a licensed program, the license management system responds with an indication as to whether the license unit value exceeds the allocation unit value associated with the use. Upon receiving the response, the operation of the licensed program depends upon policies established by the licensor.

EP 0 332 304 A2

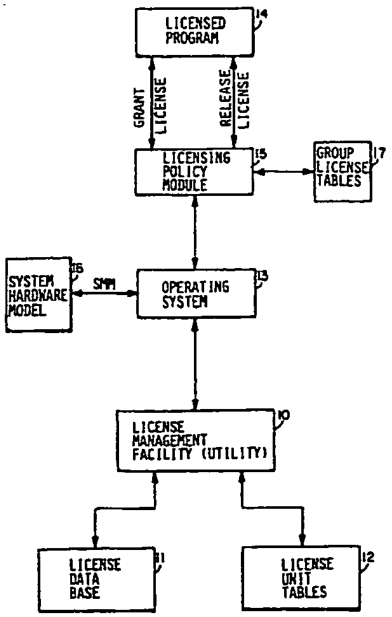

LICENSED PROGRAM
14
GRANT LICENSE
RELEASE LICENSE
LICENSING POLICY MODULE
15
GROUP LICENSE TABLES
17
SYSTEM HARDWARE MODEL
16
SMM
OPERATING SYSTEM
13
LICENSE MANAGEMENT FACILITY (UTILITY)
10
LICENSE DATA BASE
11
LICENSE UNIT TABLES
12

FIG. 1

Xerox Copy Centre

SOFTWARE LICENSING MANAGEMENT SYSTEM

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates generally to the field of digital data processing systems, and more specifically to a system for managing licensing for, and usage of, the various software programs which may be processed by the systems to ensure that the software programs are used within the terms of the software licenses.

2. Description of the Prior Art

A digital data processing system includes three basic elements, namely, a processor element, a memory element and an input/output element. The memory element stores information in addressable storage locations. This information includes data and instructions for processing the data. The processor element fetches information from the memory element, interprets the information as either an instruction or data, processes the data in accordance with the instructions, and returns the processed data to the memory element for storage therein. The input/output element, under control of the processor element, also communicates with the memory element to transfer information, including instructions and data to be processed, to the memory, and to obtain processed data from the memory.

Typically, an input/output element includes a number of diverse types of units, including video display terminals, printers, interfaces to the public telecommunications network, and secondary storage subsystems, including disk and tape storage devices. A video display terminal permits a user to run programs and input data and view processed data. A printer permits a user to obtain a processed data on paper. An interface to the public telecommunications network permits transfer of information over the public telecommunications network.

The instructions processed by the processor element are typically organized into software programs. Recently, generation and sales of software programs have become significant businesses both for companies which are primarily vendors of hardware, as well as for companies which vend software alone. Software is typically sold under license, that is, vendors transfer copies of software to users under a license which governs how the users may use the software. Typically, software costs are predicated on some belief as to the amount of usage which the software program may provide and the economic benefits, such as cost saving which may otherwise be incurred, which the software may provide to the users. Thus, license fees may be based on the power of the processor or the number of processors in the system, or the number of individual nodes in a network, since these factors provide measures of the number of users which may use the software at any given time.

In many cases, however, it may also be desirable, for example, to have licenses and license fees more closely relate to the actual numbers of users which can use the program at any given time or on the actual use to which a program may be put. Furthermore, it may be desirable to limit the use of the program to specified time periods. A problem arises particularly in digital data processing systems which have multiple users and/or multiple processors, namely, managing use of licensed software to ensure that the use is within the terms of the license, that is, to ensure that the software is only used on identified processors or by the numbers of users permitted by the license.

SUMMARY OF THE INVENTION

The invention provides a new and improved licensing management system for managing the use of licensed software in a digital data processing system.

In brief summary, the license management system includes a license management facility and a licensing policy module that jointly determine whether a licensed program may be operated. The license management facility maintains a license unit value for each licensed program and a pointer to a table identifying a license usage allocation unit value associated with usage of the licensed program. In response to a request to use a licensed program, the license management facility determines whether the remaining license unit value exceeds the license usage allocation unit value associated with the use. If the license unit value does not exceed the license usage allocation unit value, the license management facility permits usage of the licensed program and adjusts the license unit value by a function of the license usage allocation unit value to reflect the usage. On the other hand, if the license unit value associated with use of the license program does exceed the license usage allocation unit value, the licensing policy module determines whether to allow the licensed program to be used in response to other licensing policy factors.

BRIEF DESCRIPTION OF THE DRAWINGS

This invention is pointed out with particularity in the appended claims. The above and further advantages of this invention may be better understood by referring to the following description taken in conjunction with the accompanying drawings, in which:

Fig. 1 is a general block diagram of a new system in accordance with the invention;

Figs. 2 and 3 are diagrams of data structures useful in understanding the detailed operation of the system depicted in Fig. 1; and

Figs. 4A-1 through 4B-2 are flow diagrams which are useful in understanding the detailed operations of the system depicted in Fig. 1.

DETAILED DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

Fig. 1 depicts a general block diagram of a system in accordance with the invention for use in connection with a digital data processing system which assists in managing software use in accordance with software licenses. With reference to Fig. 1, the new system includes a license management facility 10 which operates in conjunction with a license data base 11 and license unit tables 12, and under control of an operating system 13 and licensing policy module 15 to control use of licensed programs, such as licensed program 14, so that the use is in accordance with the terms of the software license which controls the use of the software program on a system 16 identified by a system marketing model (SMM) code in a digital data processing system.

As is conventional, the digital data processing system including the licensing management system may include one or more systems 16, each including one or more processors, memories and input/output units, interconnected in a number of ways. For example, the digital data processing system may comprise one processor, which may include a central processor unit which controls the system and one or more auxiliary processors which assist the central processor unit. Alternatively, the digital data processing system may comprise multiple processing systems, in which multiple central processor units are tightly coupled, or clustered or networked systems in which multiple central processor units are loosely coupled, generally operating relatively autonomously, interacting by means of messages transmitted over a cluster or network connection. In a tightly coupled multiple processing system, for example, it may be desirable to control the number of users which may use a particular software program at one time. A similar restriction may be obtained in a cluster or network environment by controlling the number of particular nodes, that is, connections to the communications link in the cluster or network over which messages are transferred. In addition, since the diverse processors which may be included in a digital data processing system may have diverse processing speeds and powers, represented by differing system marketing model (SMM) codes, it may be desirable to include a factor for speeds and power in determining the number of processors on which a program may be used concurrently.

As will be explained in greater detail below, the license data base 11 contains a plurality of entries 20 (described below in connection with Fig. 2) each containing information relating to the terms of the license for a particular licensed program 14. In one embodiment such information may include a termination date, if the license is for a particular time period or expires on a particular date, and a number of licensing units if the license is limited by usage of the license program. In that embodiment, the entry also includes identification of a license unit table 40 (described below in connection with Fig. 3) in the license unit tables 12 that identifies the number of allocation units for usage of the licensed program on the types of systems 16 which may be used in the digital data processing system as represented by the system marketing model (SMM) codes.

When a user wishes to use a licensed program 14, a GRANT LICENSE request message is generated which requests information as to the licensing status of the licensed program 14. The GRANT LICENSE request message is transmitted to the licensing policy module 15, which notifies the operating system of the request. The operating system 13, in turn, passes the request, along with the system marketing model of the specific system 16 being used by the user, to the license management facility 10 which determines whether use of the program is permitted under the license.

In response to the receipt of the GRANT LICENSE request from the user and the system marketing model (SMM) code of the system 16 being used by the user on which the licensed program will be processed, the license management facility 10 obtains from the license data base the entry 20 associated with the licensed program

14 and determines whether the use of the licensed program 14 is within the terms of the license as indicated by the information in the license data base 11 and the license unit tables 12.

In particular, the license management facility 10 retrieves the contents of the entry 20 associated with the licensed program. If the entry 20 indicates a termination data, the license management facility 10 compares the system data, which is maintained by the digital data processing system in a conventional manner, with the termination date identified in the entry. If the system date is after the termination date identified in the entry 20, the license has expired and the license management facility 10 generates a usage disapproved message, which it transmits to the operating system 13. On the other hand, if the termination date indicated in the entry 20 is after the system date, the license has not expired and the license management facility 10 proceeds to determine whether the usage of the licensed program 14 is permitted under other terms of the license which may be embodied in the entry 20.

In particular, the license management facility 10 then determines whether the usage of the licensed program is permitted under usage limitations. In that operation, the license management facility obtains the number of license units remaining, which indicates usage of the licensed program 14 not including the usage requested by the user, as well the identification of the table 40 in license unit tables 12 associated with the licensed program 14. The license management facility 10 then compares the number of license units which would be allocated for use of the licensed program 14, which it obtains from the table 40 identified by entry 20 in the license data base 11, and the number of remaining units to determine whether sufficient license units remain to permit usage of the licensed program 14.

If the number of remaining license units indicated by entry 20 in the license data base 11 exceeds the number, from license unit tables 12, of license units which would be allocated for use of the licensed progam 14, the usage of the licensed program is permitted under the license. Accordingly, the license management facility transmits a usage approved response to the operating system 13. In addition, the license management facility 10 adjusts the number of remaining license units in entry 20 by a function of the license units allocated to use of the licensed program to reflect the usage.

On the other hand, if the number of remaining license units indicated by entry 20 in the license data base is less than the number of license units which would be allocated for use of the licensed program 14, the usage of the licensed program 14 is not permitted by the license. In that case, the license management facility 10 transmits a usage disapproved response to the operating system 13. In addition, the license management facility 10 may also log the usage disapproved response; this information may be used by a system operator to determine whether usage of the licensed program 14 is such as to warrant obtaining an enlarged license.

Upon receipt of either a usage approved response or a usage disapproved response to the GRANT LICENSE request, the operating system 13 passes the response to the licensing policy module 15. If a usage approved response is received, the licensing policy module normally allows usage of the licensed program 14. If a usage disapproved response is received, the licensing policy module determines whether the usage of the licensed program may be permitted for other reasons. For example, usage of the licensed program 14 may be permitted under a group license, whose terms are embodied in entries in group license tables 17. Under a group license, usage may be permitted of any of a group of licensed programs. The operations to determine to whether usage is permitted may be performed in the same manner as described above in connection with license management facility 10. In addition, if the usage of the licensed program 14 is not permitted under a group license, usage may nonetheless be permitted under the licensor's licensing practices, which may be embodied in the licensing policy module 15. If the licensing policy module determines that usage of the program should be permitted, notwithstanding a usage disapproved response from the license management facility 10, because the usage is permitted under a group license or the licensor's licensing practices, the licensing policy module 15 permits usage of the licensed program. Otherwise, the licensing policy module does not permit usage of the licensed program in response to the GRANT LICENSE request.

When a user no longer requires use of a licensed program 14, it transmits a RELEASE LICENSE request to the licensing policy module 15. The operations performed by the licensing policy module depend on the basis for permitting usage of the licensed program. If usage was permitted as a result of a group license, if the group license is limited by usage, the licensing policy module 15, if necessary, adjusts the records in the group license tables 17 related to the group license to reflect the fact that the licensed program 14 related to the group license is not being used. If the usage was permitted as a result of a group license which is not limited by usage, but instead is limited in duration, or if the usage was permitted in response to the licensor's licensing policies, the licensing policy module 15 need do nothing. If the licensing policy module 15 maintains a log of usage outside the scope of a group or program license, it may make an entry in the log of the RELEASE request.

Finally, if usage was permitted as a result of the license management facility 10 providing an approve usage response to the GRANT LICENSE request, the licensing policy module 15 transmits the RELEASE LICENSE request to the operating system 13. In response, the operating system 13 transfers the RELEASE LICENSE request. to the license management facility 10, along with an identification of the system 16 using the licensed program 14. The license management facility 10 then obtains from the license data base the identification of the appropriate license usage allocation unit value table in license unit tables 12, and determines the number of allocation units associated with this use of the licensed program 14 based on the identified allocation table and the processor. The license management facility 10 then adjusts the number of license units for the licensed program 14 in the license data base 11 to reflect the release.

It will be appreciated by those skilled in the art that, the license management facility 10 may, in response to a GRANT LICENSE request, instead of deducting allocation units from the entries in the license data base 11 associated with the licensed programs 14, determine the number of allocation units which would be in use if usage of the licensed program 14 is permitted, and respond based on that determination. If the license mangement facility 10 operates in that manner, it may be advantageous for the entries in license data base 11 relating to each licensed program 14 to maintain a running record of the number of allocation units associated with its usage. The licensing policy module 15 may operate similarly in connection with group licenses that are limited by usage.

It will also be appreciated that the new license management system thus permits the digital data processing system to control use of a licensed program 14 based on licensing criteria in the license data base 11, the license unit tables 12, the group licensing tables 17 and the licensor's general licensing policies rather than requiring an operator to limit or restrict use of a licensed program or charging for the license based on some function of the capacity of all of the processors in the digital data processing system. The new license management system allows for very flexible pricing of licenses and licensing policies, since the digital data processing system itself enforces the licensing terms controlling use of the licensed programs 14 in the system.

Fig. 2 depicts the detailed structure of the license data base 12 (Fig. 1) used in the license management system depicted in Fig. 1. With reference to Fig. 2, the license data base includes a plurality of entries generally identified by reference numberal 20, with each entry being associated with one licensed program 14. Each entry 20 includes a number of fields, including an issuer name field 21 identifying the issuer of the license, an authorization number field 22 which contains an authorization number, a producer name field 23 which identifies the name of the vendor of the licensed program, and a product name field 24 which contains the name of the licensed program. The contents of these fields may be used, for example, in connection with other license management operations, such as determining the source of licensed programs in the event of detection of errors in programs, and in locating duplicate entries in the license data base or entries which may be combined as a result of licenses being obtained and entered by, perhaps different operators or at different times.

Each entry 20 in the licensing data base 11 also includes a license number field 25 whose contents identify the number of licensing units remaining. A license of a licensed program 14 identifies a number of licensing units, which may be a function of the price paid for the license. An availability table field 26 and an activity table field 27 identify license usage allocation unit value tables in the license unit tables 12 (described in connection with Fig. 3) to be used in connection with the GRANT LICENSE and RELEASE LICENSE requests.

By way of background, a license may be in accordance with a licensing paradigm which requires concurrent use of the licensed program 14 on several processors to be a function of the processor power and capacity, and the availability table field 26 identifies a license usage allocation unit table to be used in connection with that. In an alternative, a license may be in accordance with a licensing paradigm which requires concurrent use of the licensed program to be a function of the number of users using the program, and the activity table field 27 identifies a license usage allocation unit valve table in the license unit tables 12 to be used in connection with that. If either licensing paradigm is used to the exclusion of the other, one field contains a non-zero value and the other field contains a zero value. In addition, a license may be in accordance with both licensing paradigms, that is, concurrent use of a program may be limited by both processor power and capacity and by the number of concurrent users, and in that case both fields 26 and 27 have non-zero values.

In one embodiment of the licensing management system, fields 21 through 27 of an entry 20 in the licensing data base 11 are required. In that embodiment, an entry 20 in the licensing data may also have several optional fields. In particular, an entry 20 may include a date/version number field 30 whose contents comprise either a date or version number to identify the licensed program. If a license is to terminate on a specific date, the entry 20 may include a licensor termination date field 31 or a licensee termination date field 32 whose contents specify the termination date assigned by the licensor or licensee. This may be particularly useful, for example, as a mechanism for permitting licensees to demonstrate or try a program before committing to a long or open term license.

Finally, an entry 20 in the license data base includes a checksum field 33, which includes a checksum of the contents of the other fields 21 through 27 and 30 through 32 in the entry 20, which may be established by means of a mathematical algorithm applied to the contents of the various fields. The general mechanism for establishing checksums is well known in the art, and will not be described further herein. The contents of all fields 21 through 27 and 30 through 33 of a new entry 20 are entered by an operator. Prior to establishment of an entry in the license data base 11, the license management facility 10 may verify correct entry of the information in the various fields by calculating a checksum and comparing it to the checksum provided by the operator. If the checksum provided by the operator and the checksum determined by the license management facility are the same, the entry 20 is established in the license data base 11. On the other hand, if the checksum provided by the operator and the checksum determined by the license management facility differ, the license management facility 10 determines that the information is erroneous or the license is invalid and does not establish the entry 20 in the license data base 11. It will be appreciated that, if the checksum-generation algorithm is hidden from an operator, the checksum provides a mechanism for verifying, not only that the information has been properly loaded into the entry, but also that the license upon which the entry is based is authorized by the licensor.

The structure of group license tables 17 may be similar to the structure of the license data base 11, with the addition that the entries for each license reflected in the group license tables 17 will need to identify all of the licensed programs covered thereby.

As described above, the licensing unit tables 12 (Fig. 1) contain information as to the allocation units for use in determining the number of licensing units associated with use of a licensed program. The structure of a licensing unit table 40 is depicted in Fig. 3. With reference to Fig. 3, the licensing unit table includes a plurality of entries 41(1) through 41(N) (generally identified by reference numeral 41) each identified by a particular type of processor. One entry 41 in the table 40 is provided for each type of processor which can be included in the digital data processing system which can use the licensed programs 14 which reference the license unit table 40. The processor associated with each entry is identified by a processor identification field 42. The successive fields in the entries 41 (which form the various columns in the table 40 depicted in Fig. 3) form license usage allocation unit value tables 43(1) through 43-(M) (generally identified by reference numeral 43). The contents of the availability table field 26 and the activity table field 27 identify a license usage allocation unit value table 43. If there are non-zero contents in both availability field 26 and activity field 27, the contents which identify be the same license usage allocation unit value table 43 or different license usage allocation unit value tables 43. As described above, the contents of the license usage allocation unit value table identify the number of licensing units associated with use of the licensed programs which identify the particular license usage allocation unit value table, for each of the identified processors.

The operation of the licensing management system is depicted in detail in Figs. 4A-1 through 4-B. Figs. 4A-1 through 4A-4 depict, in a number of steps the details of operation of the licensing management system in connection with the GRANT LICENSE request from a licensed program 14. Figs. 4B-1 and 4B-2 depict, in a number of steps, the details of operation in connection with the RELEASE LICENSE request from a licensed program 14. In the Figs., the particular steps performed by the licensing policy module 15, the license management facility 10 and the operating system 13 are indicated in the respective steps. Since the operations depicted in Figs. 4A-1 through 4B-2 are substantially as described above in connection with Fig. 1, they will not be described further herein.

The foregoing description has been limited to a specific embodiment of this invention. It will be apparent, however, that variations and modifications may be made to the invention, with the attainment of some or all of the advantages of the invention. Therefore, it is the object of the appended claims to cover all such variations and modifications as come within the true spirit and scope of the invention.

Claims

1. A license management system for managing usage of a licensed software program comprising:
licensing storage means for storing a licensing unit value identifying a number of licensing units asso-

ciated with the licensed software program;
usage allocation value storage means for storing a usage allocation value identifying a number of licensing units associated with a use of the licensed software program; and
licensing verification means responsive to a usage request to use said licensed software program for determining, based on the contents of said licensing storage means and said usage allocation value storage means, whether usage of said licensed software program is permitted and, if usage is permitted, for adjusting the contents of said licensing storage means by a value to the contents of said usage allocation value storage means.

2. A license management system as defined in claim 1 for use in a digital data processing system which generates a system date value, said licensing storage means includes a plurality of fields including a licensing unit storage field for storing said licensing unit number identifying value and a field identifying a termination date, said licensing verification means further determining whether usage of said licensed software program is permitted in response to a comparison of said system date and said termination date.

3. A license management system as defined in claim 1 for managing usage of plurality of licensed software programs, wherein said licensing storage means includes a plurality of entries each containing a program identification field identifying a licensed software program and a licensing unit storage field for storing said licensing unit value, said licensing verification means including:
request receiving means for receiving a usage request identifying a licensed software program;
licensing unit retrieval means responsive to said request receiving means receipt of a usage request for retrieving the contents of said licensing unit storage field from the entry of said licensing storage means whose program identification field identifies the licensed software program identified in said usage request; and
licensing unit processing means for determining, based on the contents of retrieved licensing unit storage field and said usage allocation value storage means, whether usage of said licensed software program is permitted and, is usage is permitted, for adjusting the contents of said licensing storage means by a value related to the contents of said usage allocation value storage means.

4. A license management system as defined in claim 3 for use in a digital data processing system which generates a system date value, each entry in said licensing storage means further including a termination date field identifying a termination date, said licensing unit processing means further determining whether usage of said licensed software program is permitted in response to a comparison of said system date and said termination date.

5. A license management system as defined in claim 3 wherein said usage allocation value storage means includes a plurality of usage allocation tables each storing a value identifying a number of licensing units, each entry in said licensing storage means further including a usage allocation table identification field identifying a usage allocation table, said licensing verification means further including usage allocation table retrieval means responsive to said request receiving means receipt of a usage request for retrieving the contents of the usage allocation table identified by the contents of said usage allocation table identification field of said retrieved entry, said licensing unit processing means using said retrieved usage allocation table in its determination.

6. A license management system as defined in claim 5 wherein a request message further includes licensing usage allocation value selection criteria and each usage allocation table includes a plurality of entries each identifying a usage allocation value associated with a licensing usage allocation value selection criterion, said licensing verification means including means for retrieving, from the usage allocation table identified by said entry in said licensing storage means, the usage allocation value associated with the licensing usage allocation value selection criterion in said request message and using said retrieved usage allocation value in its determination.

7. A license management system as defined in claim 3 wherein a request message further includes licensing usage allocation value selection criteria and said usage allocation table includes a plurality of entries each identifying a usage allocation value associated with a licensing usage allocation selection criterion, said licensing verification means including means for retrieving the usage allocation value associated with the licensing usage allocation selection criterion in said request message and using said retrieved usage allocation value in its determination.

8. A license management system as defined in claim 1 wherein said licensing verification means further operates in response to a release request message for adjusting the contents of said licensing storage means by a value related to the contents of said usage allocation value storage means.

9. A license management system as defined in claim 8 for managing usage of a plurality of licensed software programs, wherein said licensing storage means includes a plurality of entries each containing a program identification field identifying a licensed software program and a licensing unit storage field for storing said licensing unit value,

said licensing verification means including:
request receiving means for receiving a release request identifying a licensed software program;
licensing unit processing means for adjusting the contents of said licensing storage means by a value related to the contents of said usage allocation value storage means.

10. A license management system as defined in claim 9 wherein said usage allocation value storage means includes a plurality of usage allocation tables each storing a value identifying a number of licensing units, each entry in said licensing storage means further including a usage allocation table identification field identifying a usage allocation table, said licensing verification means further including usage allocation table retrieval means responsive to said request receiving means receipt of a usage request for retrieving the contents of said usage allocation table identification field of said retrieved entry, said licensing unit processing means using retrieved usage allocation table in its adjusting.

11. A license management system as defined in claim 10 wherein a release message further includes licensing usage allocation value selection criteria and each usage allocation table includes a plurality of entries each identifying a usage allocation value associated with a licensing usage allocation value selection criterion, said licensing verification means including means for retrieving, from the usage allocation table identified by said entry in said licensing storage means, the usage allocation value associated with the licensing usage allocation value selection criterion in said request message and using said retrieved usage allocation value in its adjusting.

12. A license management system as defined in claim 8 wherein a release message further includes licensing usage allocation value selection criteria and each usage allocation table includes a plurality of entries each identifying a usage allocation value associated with a licensing usage allocation value selection criterion, said licensing verification means including means for retrieving, from the usage allocation value table identified by said entry in said licensing storage means, the usage allocation value associated with the licensing usage allocation value selection criterion in said request message and using said retrieved usage allocation value in its adjusting.

13. A license management system for use in a digital data processing system including a system date generating means for generating a system date value, said license management system comprising:
licensing storage means including a plurality of entries each associated with a licensed software program, each entry containing a licensing units field for storing a licensing unit value identifying a number of licensing units associated with the license software program, a usage allocation table, and a termination date;
usage allocation table storage means for storing a plurality of usage allocation tables, each usage allocation table having a plurality of usage allocation entries each usage allocation entry being associated with a licensing usage allocation value selection criterion and storing a usage allocation value identifying a number of licensing units; and
licensing verification means including:
usage grant means including:
usage request message receiving means for receiving a usage request message from a licensed software program, said usage request message identifying said licensed software program and usage grant criteria;
entry retrieval means responsive to the receipt of a usage request message for retrieving from said licensing storage means the licensing table entry associated with said licensed software program;
usage allocation table retrieval means for retrieving from said usage allocation table storage means a usage allocation entry identified by said retrieved licensing table entry and the licensing usage allocation value selection criterion identified by the received usage request message;
licensing request processing means including:
usage determination means including licensing unit comparing means for comparing the contents of said licensing units field and said usage allocation units field and date comparison means for comparing the system date value with the contents of said termination date field to determine whether usage of said licensed software program is permitted.
response generation means for generating a message in response to the determination by said usage determination means; and
licensing unit adjusting means for adjusting the contents of said licensing units field in response to a positive determination by said usage determination means;
usage release means including:
usage release message receiving means for receiving a usage request message from a licensed software program; said usage request message identifying said licensed software program and usage grant criteria;
entry retrieval means responsive to the receipt of a usage request message for retrieving from said licensing storage means the licensing table entry associated with said licensed software program;
usage allocation table retrieval means for retrieving from said usage allocation table storage means a usage allocation entry identified by said retrieved licensing table entry and the licensing usage allocation value selection criterion identified by the

received usage request message;
licensing release processing means for adjusting the contents of said licensing units field in relation to the value of said usage allocation entry.

Nouvellement déposé

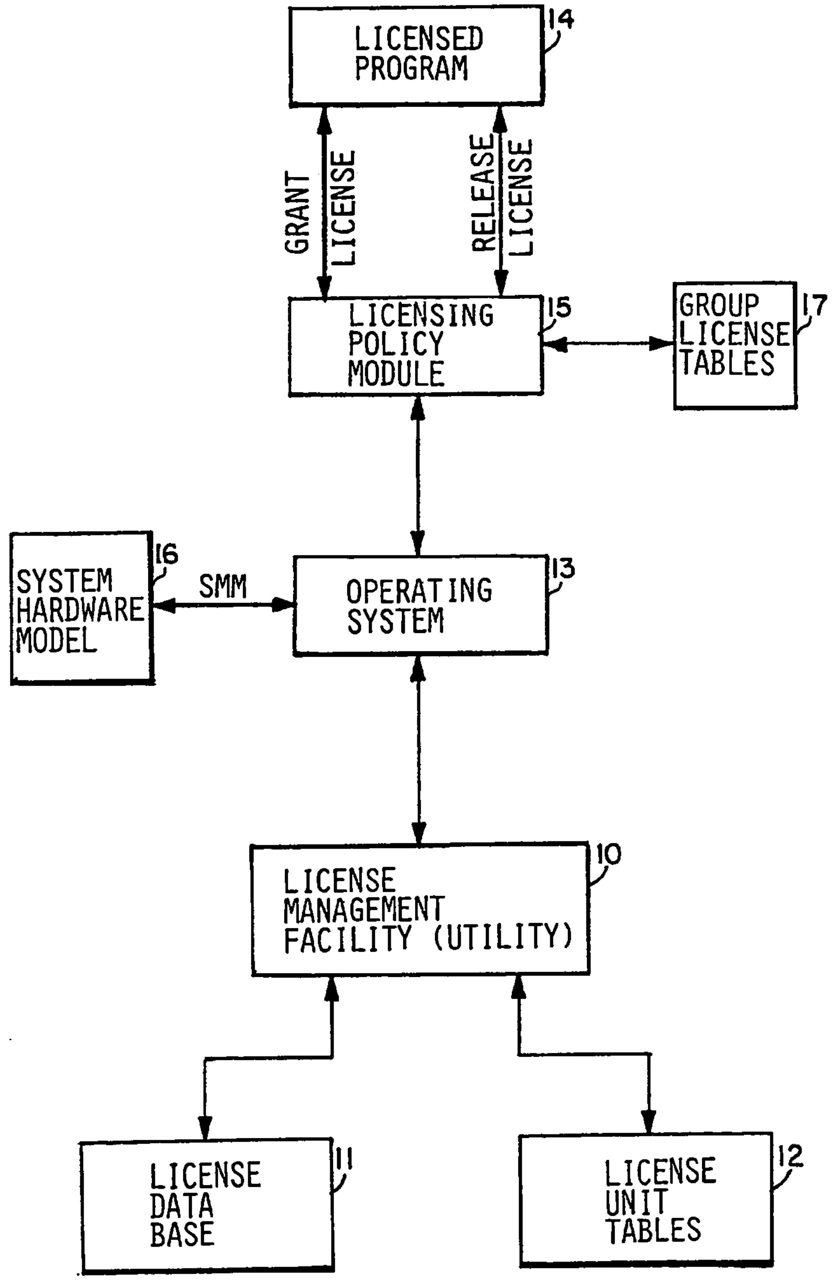

LICENSED PROGRAM
14
GRANT LICENSE
RELEASE LICENSE
LICENSING POLICY MODULE
15
GROUP LICENSE TABLES
17
SYSTEM HARDWARE MODEL
16
SMM
OPERATING SYSTEM
13
LICENSE MANAGEMENT FACILITY (UTILITY)
10
LICENSE DATA BASE
11
LICENSE UNIT TABLES
12

FIG. 1

u eingereicht / Newly filed
Nouvellement déposé

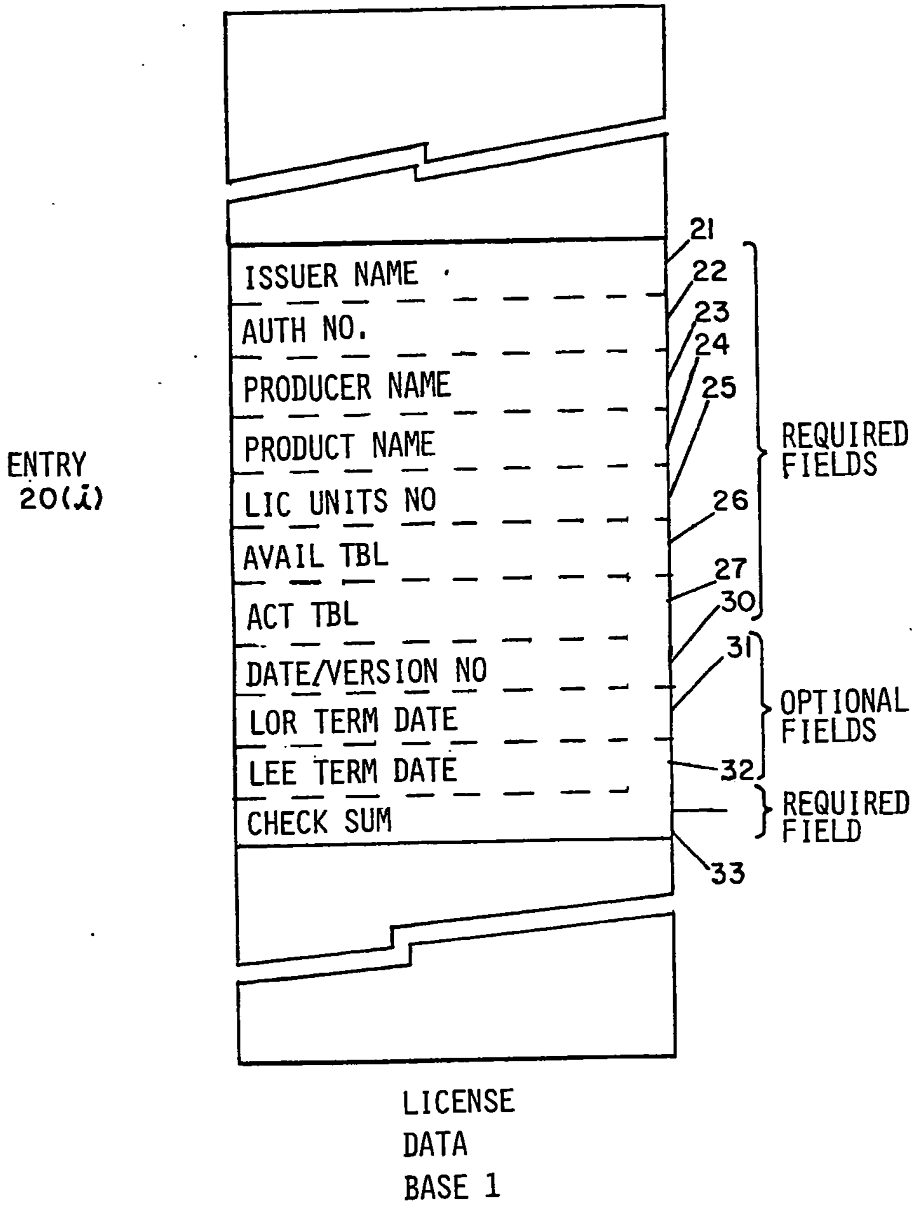
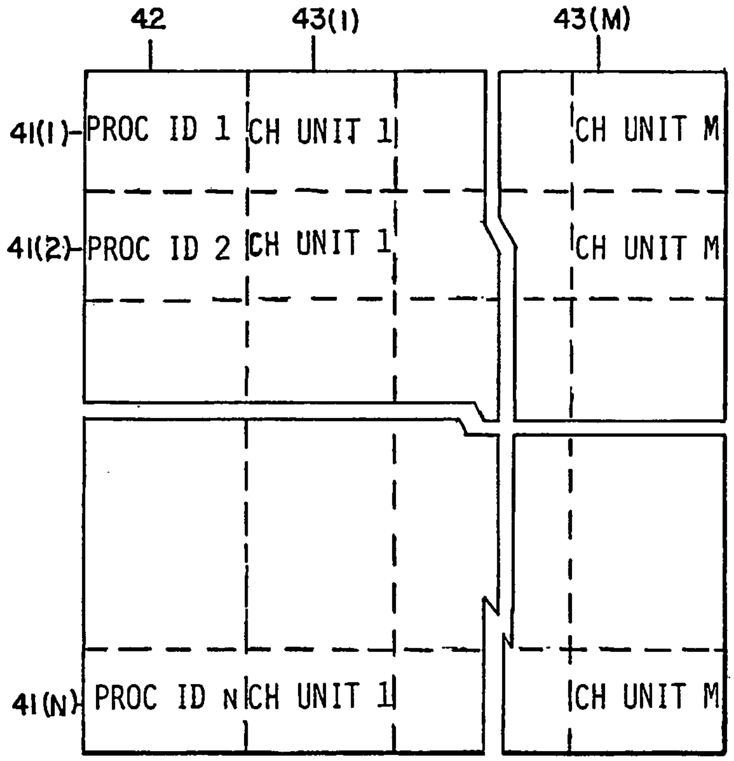

ENTRY
20(i)
21
ISSUER NAME
22
AUTH NO.
23
24
PRODUCER NAME
25
PRODUCT NAME
LIC UNITS NO
26
AVAIL TBL
27
ACT TBL
30
DATE/VERSION NO
31
LOR TERM DATE
LEE TERM DATE
32
CHECK SUM
33
REQUIRED FIELDS
OPTIONAL FIELDS
REQUIRED FIELD
LICENSE
DATA
BASE 1

FIG. 2

Neu eingereicht / Newly filed
Nouvellement déposé


42
43(1)
43(M)
41(1)
PROC ID 1
CH UNIT 1
CH UNIT M
41(2)
PROC ID 2
CH UNIT 1
CH UNIT M
41(N)
PROC ID N
CH UNIT 1
CH UNIT M

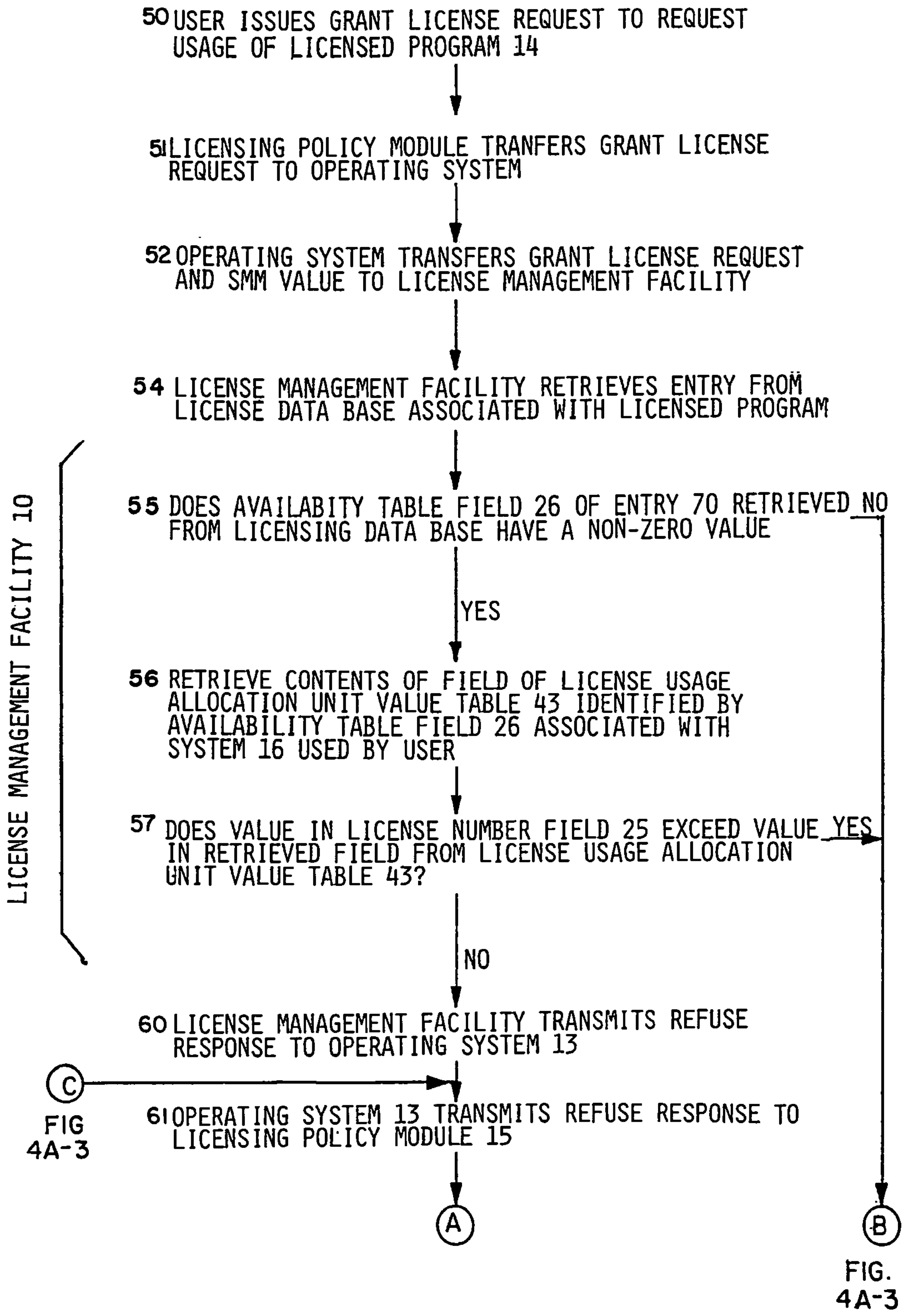
LICENSE UNIT TABLE 40

FIG. 3

Nouvellement déposé

FIG. 4A-1

GRANT LICENSE


50 USER ISSUES GRANT LICENSE REQUEST TO REQUEST USAGE OF LICENSED PROGRAM 14
51 LICENSING POLICY MODULE TRANFERS GRANT LICENSE REQUEST TO OPERATING SYSTEM
52 OPERATING SYSTEM TRANSFERS GRANT LICENSE REQUEST AND SMM VALUE TO LICENSE MANAGEMENT FACILITY
54 LICENSE MANAGEMENT FACILITY RETRIEVES ENTRY FROM LICENSE DATA BASE ASSOCIATED WITH LICENSED PROGRAM
LICENSE MANAGEMENT FACILITY 10
55 DOES AVAILABITY TABLE FIELD 26 OF ENTRY 70 RETRIEVED FROM LICENSING DATA BASE HAVE A NON-ZERO VALUE
NO
YES
56 RETRIEVE CONTENTS OF FIELD OF LICENSE USAGE ALLOCATION UNIT VALUE TABLE 43 IDENTIFIED BY AVAILABILITY TABLE FIELD 26 ASSOCIATED WITH SYSTEM 16 USED BY USER
57 DOES VALUE IN LICENSE NUMBER FIELD 25 EXCEED VALUE IN RETRIEVED FIELD FROM LICENSE USAGE ALLOCATION UNIT VALUE TABLE 43?
YES
NO
60 LICENSE MANAGEMENT FACILITY TRANSMITS REFUSE RESPONSE TO OPERATING SYSTEM 13
C
FIG 4A-3
61 OPERATING SYSTEM 13 TRANSMITS REFUSE RESPONSE TO LICENSING POLICY MODULE 15
A
B
FIG. 4A-3

Neu eingereicht / Newly filed
Nouvellement déposé

FIG. 4A-2

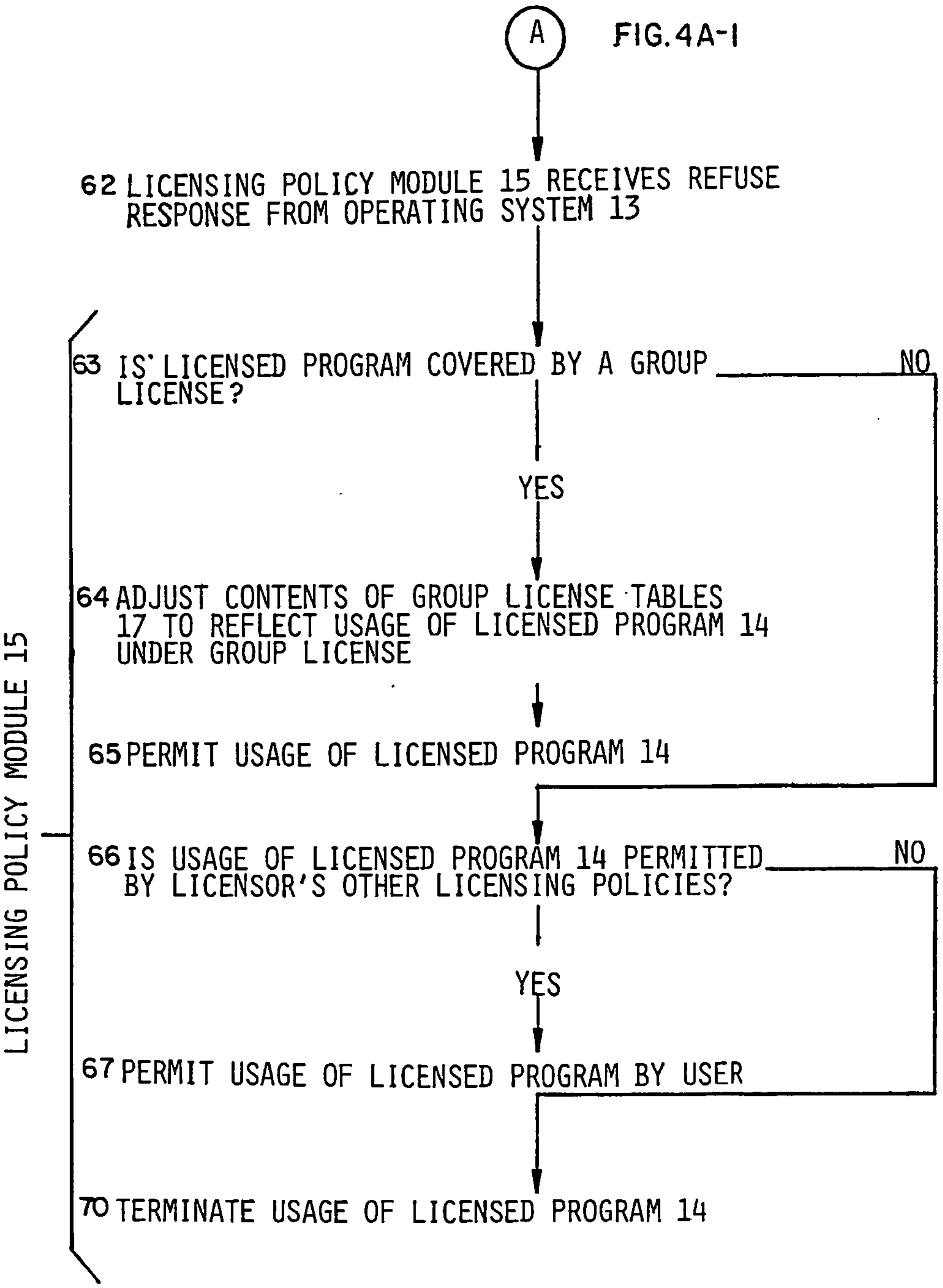

A
FIG.4A-1
62 LICENSING POLICY MODULE 15 RECEIVES REFUSE RESPONSE FROM OPERATING SYSTEM 13
63 IS LICENSED PROGRAM COVERED BY A GROUP LICENSE?
NO
YES
64 ADJUST CONTENTS OF GROUP LICENSE TABLES 17 TO REFLECT USAGE OF LICENSED PROGRAM 14 UNDER GROUP LICENSE
65 PERMIT USAGE OF LICENSED PROGRAM 14
66 IS USAGE OF LICENSED PROGRAM 14 PERMITTED BY LICENSOR'S OTHER LICENSING POLICIES?
NO
YES
67 PERMIT USAGE OF LICENSED PROGRAM BY USER
70 TERMINATE USAGE OF LICENSED PROGRAM 14
LICENSING POLICY MODULE 15

FIG. 4A-3

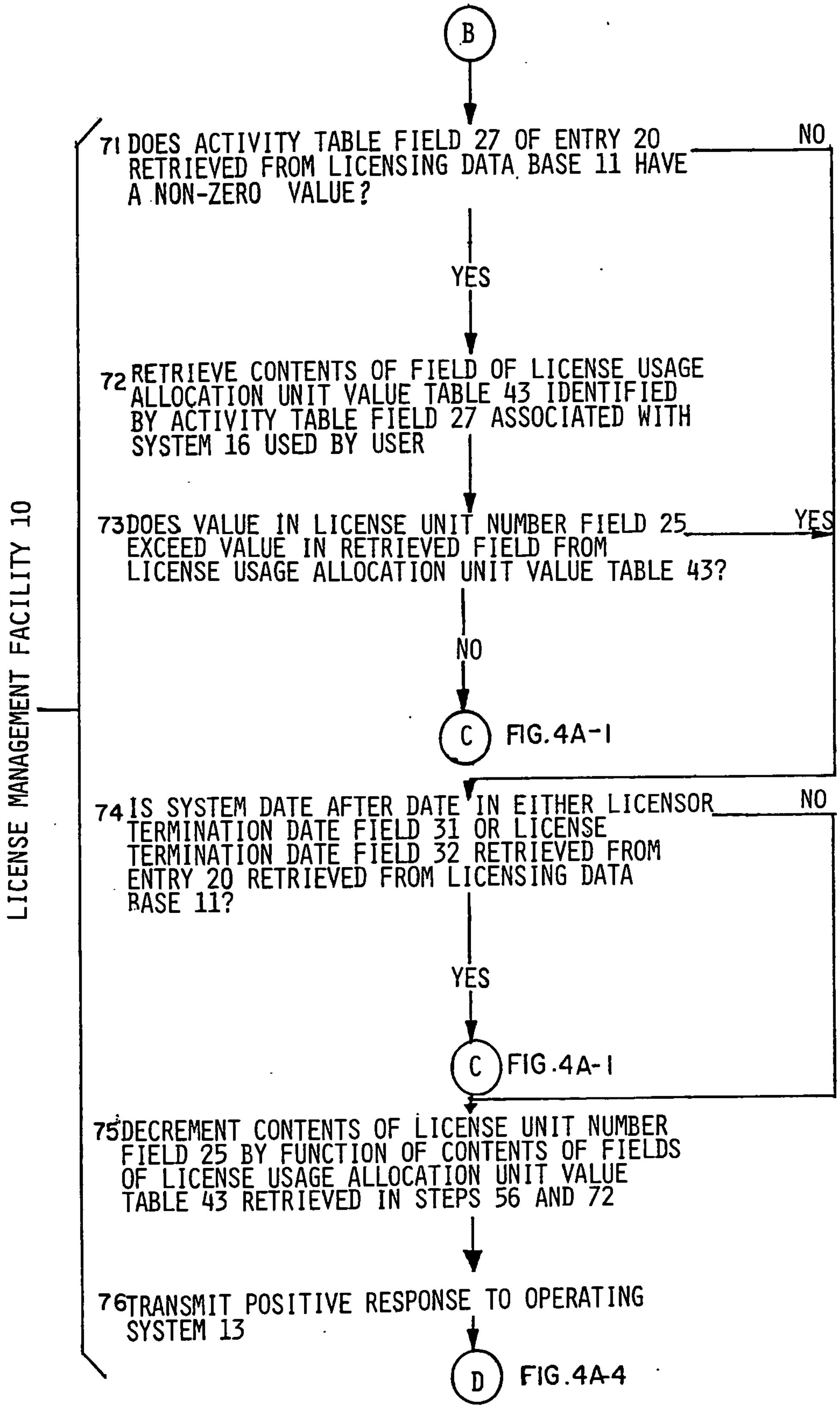

B
71 DOES ACTIVITY TABLE FIELD 27 OF ENTRY 20 RETRIEVED FROM LICENSING DATA BASE 11 HAVE A NON-ZERO VALUE?
NO
YES
72 RETRIEVE CONTENTS OF FIELD OF LICENSE USAGE ALLOCATION UNIT VALUE TABLE 43 IDENTIFIED BY ACTIVITY TABLE FIELD 27 ASSOCIATED WITH SYSTEM 16 USED BY USER
73 DOES VALUE IN LICENSE UNIT NUMBER FIELD 25 EXCEED VALUE IN RETRIEVED FIELD FROM LICENSE USAGE ALLOCATION UNIT VALUE TABLE 43?
YES
NO
C
FIG.4A-1
LICENSE MANAGEMENT FACILITY 10
74 IS SYSTEM DATE AFTER DATE IN EITHER LICENSOR TERMINATION DATE FIELD 31 OR LICENSE TERMINATION DATE FIELD 32 RETRIEVED FROM ENTRY 20 RETRIEVED FROM LICENSING DATA BASE 11?
NO
YES
C
FIG.4A-1
75 DECREMENT CONTENTS OF LICENSE UNIT NUMBER FIELD 25 BY FUNCTION OF CONTENTS OF FIELDS OF LICENSE USAGE ALLOCATION UNIT VALUE TABLE 43 RETRIEVED IN STEPS 56 AND 72
76 TRANSMIT POSITIVE RESPONSE TO OPERATING SYSTEM 13
D
FIG.4A-4

FIG. 4A-4

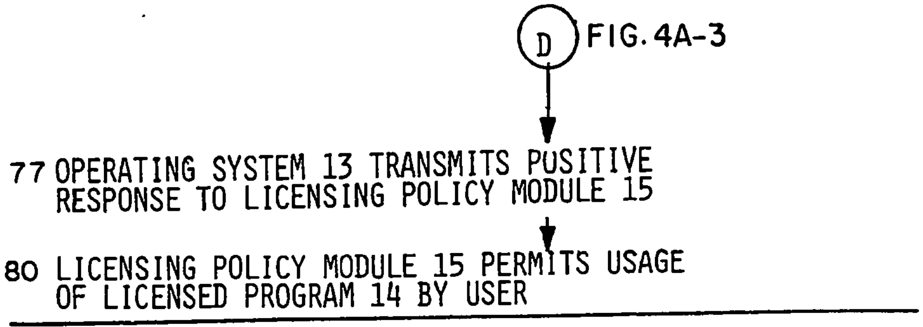

D
FIG. 4A-3
77 OPERATING SYSTEM 13 TRANSMITS PUSITIVE RESPONSE TO LICENSING POLICY MODULE 15
80 LICENSING POLICY MODULE 15 PERMITS USAGE OF LICENSED PROGRAM 14 BY USER

FIG.4B-1

RELEASE LICENSE

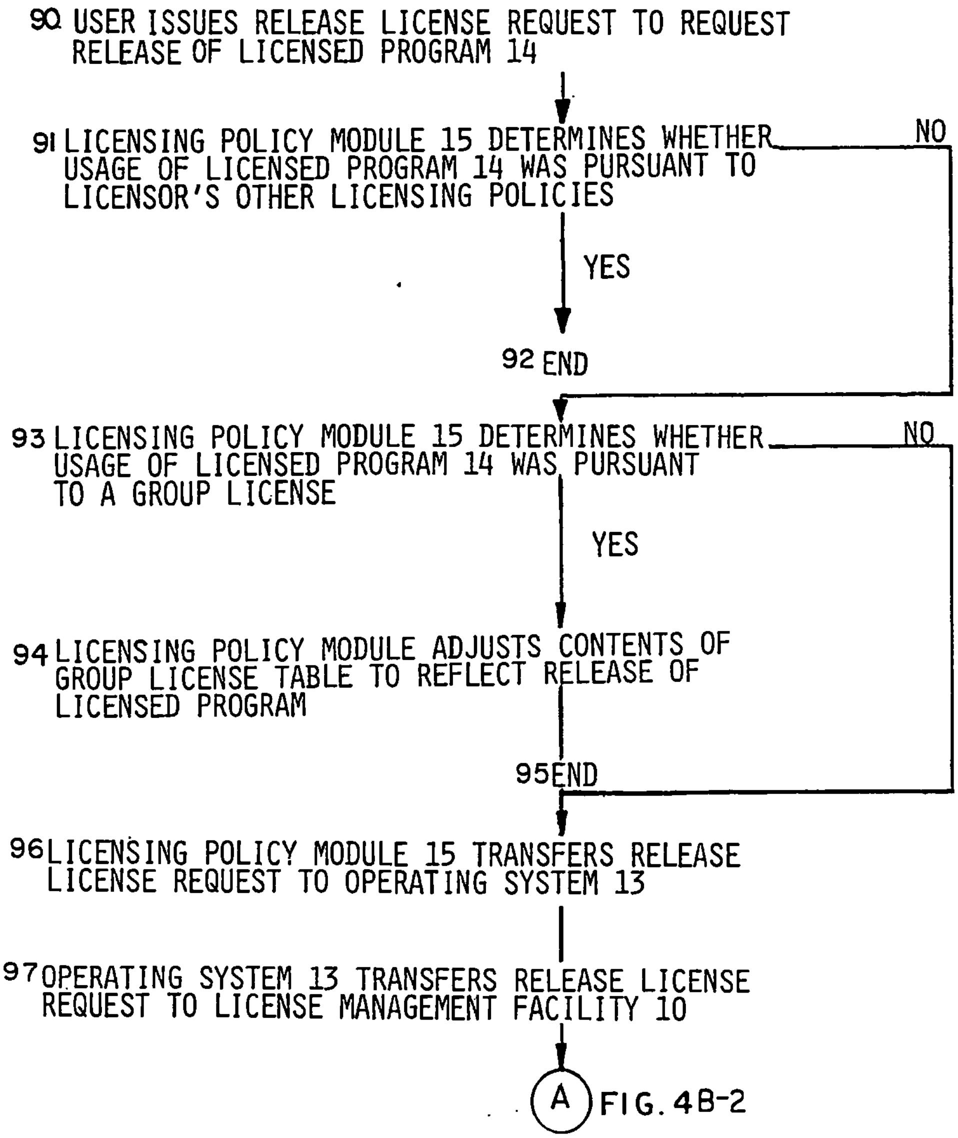
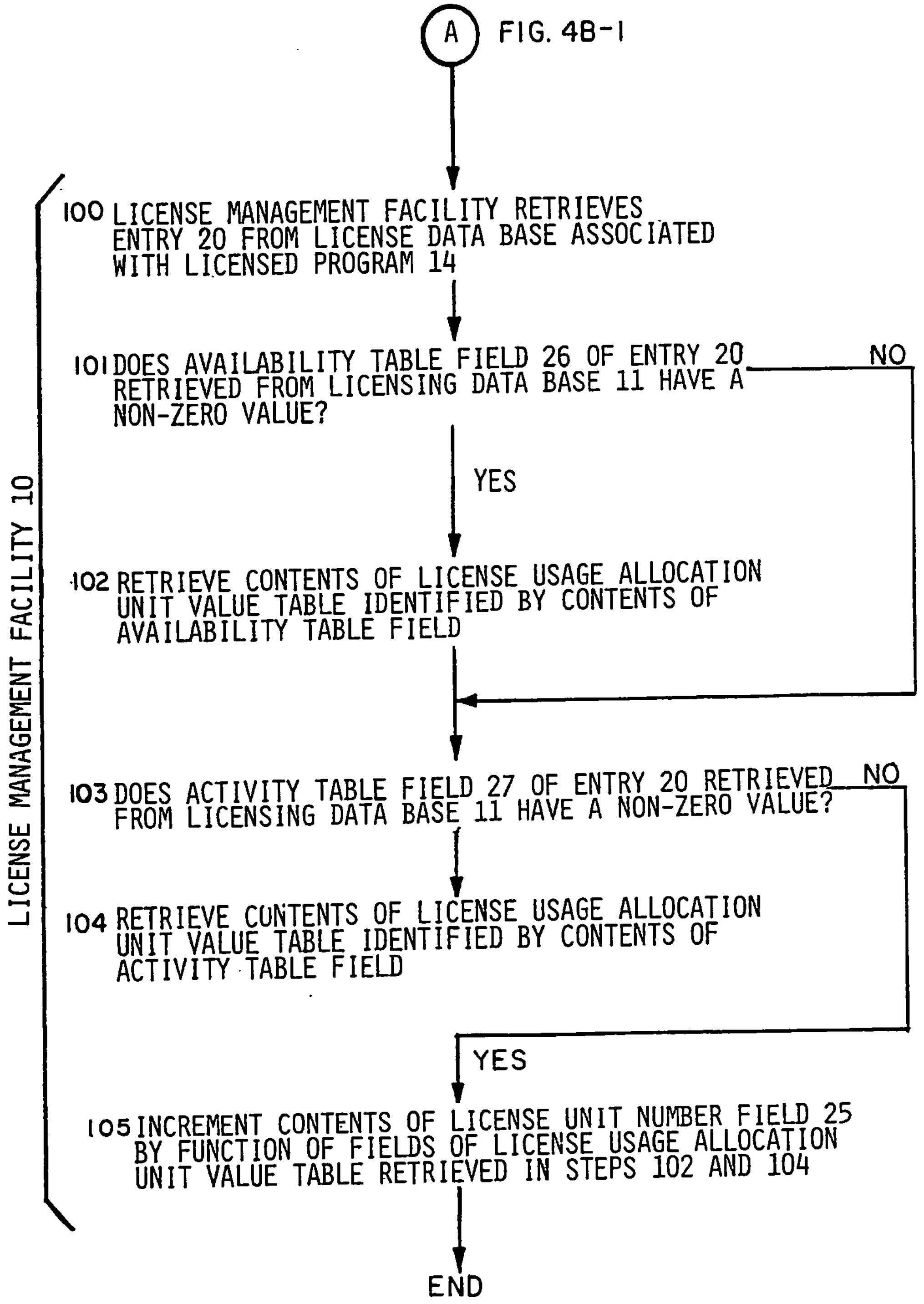

90 USER ISSUES RELEASE LICENSE REQUEST TO REQUEST RELEASE OF LICENSED PROGRAM 14
91 LICENSING POLICY MODULE 15 DETERMINES WHETHER USAGE OF LICENSED PROGRAM 14 WAS PURSUANT TO LICENSOR'S OTHER LICENSING POLICIES
NO
YES
92 END
93 LICENSING POLICY MODULE 15 DETERMINES WHETHER USAGE OF LICENSED PROGRAM 14 WAS PURSUANT TO A GROUP LICENSE
NO
YES
94 LICENSING POLICY MODULE ADJUSTS CONTENTS OF GROUP LICENSE TABLE TO REFLECT RELEASE OF LICENSED PROGRAM
95 END
96 LICENSING POLICY MODULE 15 TRANSFERS RELEASE LICENSE REQUEST TO OPERATING SYSTEM 13
97 OPERATING SYSTEM 13 TRANSFERS RELEASE LICENSE REQUEST TO LICENSE MANAGEMENT FACILITY 10
A
FIG. 4B-2

FIG. 4B-2


A
FIG. 4B-1
LICENSE MANAGEMENT FACILITY 10
100 LICENSE MANAGEMENT FACILITY RETRIEVES ENTRY 20 FROM LICENSE DATA BASE ASSOCIATED WITH LICENSED PROGRAM 14
101 DOES AVAILABILITY TABLE FIELD 26 OF ENTRY 20 RETRIEVED FROM LICENSING DATA BASE 11 HAVE A NON-ZERO VALUE?
NO
YES
102 RETRIEVE CONTENTS OF LICENSE USAGE ALLOCATION UNIT VALUE TABLE IDENTIFIED BY CONTENTS OF AVAILABILITY TABLE FIELD
103 DOES ACTIVITY TABLE FIELD 27 OF ENTRY 20 RETRIEVED FROM LICENSING DATA BASE 11 HAVE A NON-ZERO VALUE?
NO
104 RETRIEVE CONTENTS OF LICENSE USAGE ALLOCATION UNIT VALUE TABLE IDENTIFIED BY CONTENTS OF ACTIVITY TABLE FIELD
YES
105 INCREMENT CONTENTS OF LICENSE UNIT NUMBER FIELD 25 BY FUNCTION OF FIELDS OF LICENSE USAGE ALLOCATION UNIT VALUE TABLE RETRIEVED IN STEPS 102 AND 104
END

⑲ **Europäisches Patentamt**
European Patent Office
Office européen des brevets

⑪ Publication number: **0 393 806 A2**

⑫ EUROPEAN PATENT APPLICATION

㉑ Application number: **90300115.4**

㉒ Date of filing: **05.01.90**

(51) Int. Cl.^{5}: **H04L 9/32, H04L 9/08**

㉚ Priority: **17.04.89 US 339555**

㊸ Date of publication of application:
24.10.90 Bulletin 90/43

(84) Designated Contracting States:
DE FR GB IT

(71) Applicant: **TRW INC.**
1900 Richmond Road
Cleveland Ohio 44124(US)

(72) Inventor: **Goss, Kenneth C.**
1470 Island Court
Oceano California 93445-9464(US)

(74) Representative: **Allden, Thomas Stanley et al**
A.A. THORNTON & CO. Northumberland
House 303-306 High Holborn
London WC1V 7LE(GB)

(54) Cryptographic method and apparatus for public key exchange with authentication.

(57) A technique for use in a public key exchange cryptographic system, in which two user devices establish a common session key by exchanging information over an insecure communication channel, and in which each user can authenticate the identity of the other, without the need for a key distribution center. Each device has a previously stored unique random number Xi, and a previously stored composite quantity that is formed by transforming Xi to Yi using a transformation of which the inverse in computationally infeasible; then concatenating Yi with a publicly known device identifier, and digitally signing the quantity. Before a communication session is established, two user devices exchange their signed composite quantities, transform them to unsigned form, and authenticate the identity of the other user. Then each device generates the same session key by transforming the received Y value with its own X value. For further security, each device also generates another random number X'i, which is transformed to a corresponding number Y'i. These Y'i values are also exchanged, and the session key is generated in each device, using a transformation that involves the device's own Xi and X'i numbers and the Yi and Y'i numbers received from the other device.

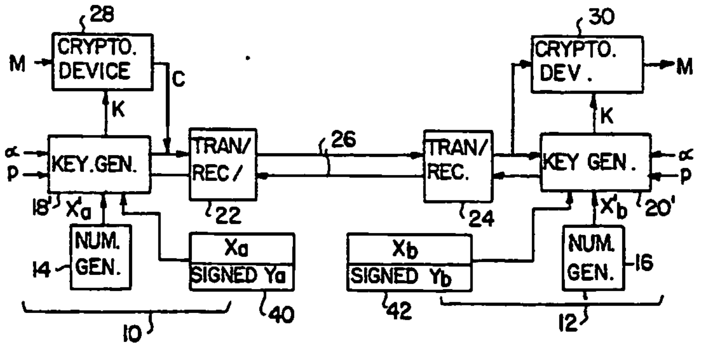


FIG. 3

EP 0 393 806 A2

Xerox Copy Centre

BACKGROUND OF THE INVENTION

This invention relates generally to cryptographic systems and, more particularly, to cryptographic systems in which an exchange of information on an unsecured communications channel is used to establish a common cipher key for encryption and decryption of subsequently transmitted messages. Cryptographic systems are used in a variety of applications requiring the secure transmission of information from one point to another in a communications network. Secure transmission may be needed between computers, telephones, facsimile machines, or other devices. The principal goal of encryption is the same in each case: to render the communicated data secure from unauthorized eavesdropping.

By way of definition, "plaintext" is used to refer to a message before processing by a cryptographic system. "Ciphertext" is the form that the message takes during transmission over a communications channel. "Encryption" or "encipherment" is the process of transformation from plaintext to ciphertext. "Decryption" or "decipherment" is the process of transformation from ciphertext to plaintext. Both encryption and decryption are controlled by a "cipher key" or keys. Without knowledge of the encryption key, a message cannot be encrypted, even with knowledge of the encrypting process. Similarly, without knowledge of the decryption key, the message cannot be decrypted, even with knowledge of the decrypting process.

More specifically, a cryptographic system can be thought of as having an enciphering transformation E_k, which is defined by an enciphering algorithm E that is used in all enciphering operations, and a key K that distinguishes E_k from other operations using the algorithm E. The transformation E_k encrypts a plaintext message M into an encrypted message, or ciphertext C. Similarly, the decryption is performed by a transformation D_k defined by a decryption algorithm D and a key K.

Dorothy E.R. Denning, in "Cryptography and Data Security," Addison-Wesley Publishing Co. 1983, suggests that, for complete secrecy of the transmitted message, two requirements have to be met. The first is that it should be computationally infeasible for anyone to systematically determine the deciphering transformation D_k from intercepted ciphertext C, even if the corresponding plaintext M is known. The second is that it should be computationally infeasible to systematically determine plaintext M from intercepted ciphertext C. Another goal of cryptography systems is that of data authenticity. This requires that someone should not be able to substitute false ciphertext C' for ciphertext C without detection.

By way of further background, cryptographic systems may be classified as either "symmetric" or "asymmetric." In symmetric systems, the enciphering and deciphering keys are either the same easily determined from each other. When two parties wish to communicate through a symmetric cryptographic system, they must first agree on a key, and the key must be transferred from one party to the other by some secure means. This usually requires that keys be agreed upon in advance, perhaps to be changed on an agreed timetable, and transmitted by courier or some other secured method. Once the keys are known to the parties, the exchange of messages can proceed through the cryptographic system.

An asymmetric cryptosystem is one in which the enciphering and deciphering keys differ in such a way that at least one key is computationally infeasible to determine from the other. Thus, one of the transformations E_k or D_k can be revealed without endangering the other.

In 1976, the concept of a "public key" encryption system was introduced by W. Diffie and M. Hellman, "New Directions in Cryptography," IEEE Trans. on Info. Theory, Vol. IT-22(6), pp. 644-54 (Nov. 1976). In a public key system, each user has a public key and private key, and two users can communicate knowing only each other's public keys. This permits the establishment of a secured communication channel between two users without having to exchange "secret" keys before the communication can begin. As pointed out in the previously cited text by Denning, a public key system can be operated to provide secrecy by using a private key for decryption; authenticity by using a private key for encryption; or both, by using two sets of encryptions and decryptions.

In general, asymmetric cryptographic systems require more computational "energy" for encryption and decryption than symmetric systems. Therefore, a common development has been a hybrid system in which an asymmetric system, such as a public key system, is first used to establish a "session key" for use between two parties wishing to communicate. Then this common session key is used in a conventional symmetric cryptographic system to transmit messages from one user to the other. Diffie and Hellman have proposed such a public key system for the exchange of keys on an unsecured communications channel. However, as will be described, the Diffie-Hellman public key system is subject to active eavesdropping. That is to say, it provides no foolproof authentication of its messages. With knowledge of the public keys, an eavesdropper can decrypt received ciphertext, and then re-encrypt the resulting plaintext for transmission to the intended receiver, who has no way of knowing that the message has been intercepted. The present invention relates to a significant improvement in techniques for public key exchange or public key management.

One possible solution to the authentication problem in public key management, is to establish a key distribution center, which issues secret keys to authorized users. The center provides the basis for identity authentication of transmitted messages. In one typical technique, a user wishing to transmit to another user sends his and the other user's identities to the center; e.g. (A,B). The center sends to A the ciphertext message $E_A(B,K,T,C)$, where E_A is the enciphering transformation derived from A's private key, K is the session key, T is the current date and time, and $C = E_B(A,K,T)$, where E_B is the enciphering transformation derived from B's private key. Then A sends to B the message C. Thus A can send to B the session key K encrypted with B's private key; yet A has no knowledge of B's private key. Moreover, B can verify that the message truly came from A, and both parties have the time code for further message identity authentication. The difficulty, of course, is that a central facility must be established as a repository of private keys, and it must be administered by some entity that is trusted by all users. This difficulty is almost impossible to overcome in some applications, and there is, therefore, a significant need for an alternative approach to public key management. The present invention fulfills this need.

Although the present invention has general application in many areas of communication employing public key management and exchange, the invention was first developed to satisfy a specific need in communication by facsimile (FAX) machines. As is now well known, FAX machines transmit and receive graphic images over ordinary telephone networks, by first reducing the images to digital codes, which are then transmitted, after appropriate modulation, over the telephone lines. FAX machines are being used at a rapidly increasing rate for the transmission of business information, much of which is of a confidential nature, over lines that are unsecured. There is a substantial risk of loss of the confidentiality of this imformation, either by deliberate eavesdropping, or by accidental transmission to an incorrectly dialed telephone number.

Ideally, what is needed is an encrypting/decrypting box connectable between the FAX machine and the telephone line, such that secured communications can take place between two similarly equipped users, with complete secrecy of data, and identity authentication between the users. For most users, a prior exchange of secret keys would be so inconvenient that they could just as well exchange the message itself by the same secret technique. A public key exchange system is by far the most convenient solution but each available variation of these systems has its own problems, as discussed above. The Diffie-Hellman approach lacks the means to properly authenticate a message, and although a key distribution center would solve this problem, as a practical matter no such center exists for FAX machine users, and none is likely to be established in the near future. Accordingly, one aspect of the present invention is a key management technique that is directly applicable to data transmission using FAX machines.

SUMMARY OF THE INVENTION

The present invention resides in a public key cryptographic system that accomplishes both secrecy and identity authentication, without the need for a key distribution center or other public facility, and without the need for double encryption and double decryption of messages. Basically, the invention achieves these goals by using a digitally signed composite quantity that is pre-stored in each user communication device. In contrast with the conventional Diffie-Hellman technique, in which random numbers Xi are selected for each communication session, the present invention requires that a unique number Xi be preselected and pre-stored in each device that is manufactured. Also stored in the device is the signed composite of a Yi value and a publicly known device identifier. The Yi value is obtained by a transformation from the Xi value, using a transformation that is practically irreversible.

Before secure communications are established, two devices exchange these digitally signed quantities, which may then be easily transformed into unsigned form. The resulting identifier information is used to authenticate the other user's identity, and the resulting Yi value from the other device is used in a transformation with Xi to establish a session key. Thus the session key is established without fear of passive or active eavesdropping, and each user is assured of the other's identity before proceeding with the transfer of a message encrypted with the session key that has been established.

One way of defining the invention is in terms of a session key generator, comprising storage means for storing a number of a first type selected prior to placing the key generator in service, and a digitally signed composite quantity containing both a unique and publicly known identifier of the session key generator and a number of a second type obtained by a practically irreversible transformation of the number of the first type. The session key generator has a first input connected to receive the number of the first type, and a second input connected to receive an input quantity transmitted over an insecure communications channel from another session key generator, the input quantity being digitally signed and containing both a publicly known identifier of the other session key generator and a number of the second type generated by a practically irreversible transformation of a number of the first type stored in the other session key generator. The session key generator also has a first output for transmitting the stored, digitally signed composite quantity over the insecure communications channel to the other session key generator, a second output, means for decoding the signed input quantity received at the second input, to obtain the identifier of the other session key generator and the received number of the second type, and means for generating a session key at the second output, by performing a practically irreversible transformation of the number of the second type received through the second input, using the number of the first type received through the first input.

For further security of the session key, the session key generator further includes a third input, connected to receive another number of the first type, generated randomly, and means for generating at the first output, for transmission with the digitally signed composite quantity, a number of the second type obtained by a practically irreversible transformation of the number of the first type received through the third input. The session key generator also includes means for receiving from the second input another number of the second type generated in and transmitted from the other session key generator. The means for generating a session key performs a practically irreversible transformation involving both numbers of the first type, received at the first and third inputs, and both numbers of the second type received at the second input, whereby a different session key may be generated for each message transmission session.

More specifically, the number of the second type stored in digitally signed form in the storage means is obtained by the transformation $Ya = \alpha^{Xa} \bmod p$, where Xa is the number of the first type stored in the storage means, and α and p are publicly known transformation parameters. The number of the second type received in the digitally signed composite quantity from the other session key generator is designated Yb, and the means for generating the session key performs the transformation $K = Yb^{Xa} \bmod p$.

When additional numbers X'a and X'b are also generated prior to transmission, the means for generating the session key performs the transformation
$K = (Y'b)^{Xa} \bmod p \oplus (Yb)^{X'a} \bmod p$,
where X'a is the number of the first type that is randomly generated, Y'b is the additional number of the second type received from the other session key generator, and the $\oplus$ symbol means an exclusive OR operation.

In terms of a novel method, the invention comprises the steps of transmitting from each device a digitally signed composite quantity to the other device, the composite quantity including a publicly known device identifier IDa and a number Ya derived by a practically irreversible transformation of a secret number Xa that it is unique to the device, receiving a similarly structured digitally signed composite quantity from the other device, and transforming the received digitally signed composite quantity into an unsigned composite quantity containing a device identifier IDb of the other device and a number Yb that was derived by transformation from a secret number Xb that is unique to the other device. Then the method performs the steps of verifying the identity of the other device from the device identifier IDb, and generating a session key by performing a practically irreversible transformation involving the numbers Xa and Yb.

Ideally, the method also includes the steps of generating another number X'a randomly prior to generation of a session key, transforming the number X'a to a number Y'a using a practically irreversible transformation, transmitting the number Y'a to the other device, and receiving a number Y'b from the other device. In this case, the step of generating a session key includes a practically irreversible transformation involving the numbers Xa, X'a, Y'b and Y'b.

In particular, the transformations from X numbers to Y numbers is of the type $Y = \alpha^{X} \bmod p$, where α and p are chosen to maximize irreversibility of the transformations, and the step of generating a session key includes the transformation
$K = (Y'b)^{Xa} \bmod p \oplus (Yb)^{X'a} \bmod p$,
where $\oplus$ denotes an exclusive OR operation.

It will be appreciated from this brief summary that the present invention represents a significant advance in the field of cryptography. In particular, the invention provides for both secrecy and identity authenticity when exchanging transmissions with another user to establish a common session key. Other aspects and advantages of the invention will become apparent from the following more detailed description, taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 is a block diagram showing a public key cryptographic system of the prior art;

FIG. 2 is a block diagram similar to FIG. 1, and showing how active eavesdropping may be used to attack the system;

FIG. 3 is a block diagram of a public key cryptographic system in accordance with the present invention;

FIG. 4 is a block diagram of a secure facsimile system embodying the present invention; and

FIG. 5 is a block diagram showing more detail of the cryptographic processor of FIG. 4.

DESCRIPTION OF THE PREFERRED EMBODIMENT

As shown in the accompanying drawings for purposes of illustration, the present invention is concerned with a public key cryptographic system. As discussed at length in the preceding background section of this specification, public key systems have, prior to this invention, been unable to provide both secrecy and identity authentication of a message without either a costly double transformation at each end of the communications channel, or the use of key distribution center.

U.S. Patent No. 4,200,770 to Hellman et al. discloses a cryptographic apparatus and method in which two parties can converse by first both generating the same session key as a result of an exchange of messages over an insecure channel. Since the technique disclosed in the Hellman et al. '770 patent attempts to provide both secrecy and authentication in a public key cryptographic system, the principles of their technique will be summarized here. This should provide a better basis for an understanding of the present invention.

In accordance with the Hellman et al. technique, two numbers α and p are selected for use by all users of the system, and may be made public. For increased security, p is a large prime number, and α has a predefined mathematical relationship to p, but these restrictions are not important for purposes of this explanation. Before starting communication, two users, A and B, indicated in FIG. 1 at 10 and 12, perform an exchange of messages that results in their both computing the same cipher key, or session key K, to be used in transmitting data back and forth between them. The first step in establishing the session key is that each user generates a secret number in a random number generator 14, 16. The numbers are designated Xa, Xb, respectively, and are selected from a set of positive integers up to p-1. Each user also has a session key generator 18, 20, one function of which is to generate other numbers Y from the numbers X, α and p, using the transformations:

Ya = α^{Xa} mod p,

Yb = α^{Xb} mod p.

The values Ya, Yb are then processed through a conventional transmitter/receiver 22, 24, and exchanged over an insecure communications channel 26.

The term "mod p' means modulo p, or using modulo p arithmetic. Transforming an expression to modulo p can be made by dividing the expression by p and retaining only the remainder. For example, 34 mod 17 = 0, 35 mod 17 = 1, and so forth. Similarly, the expression for Ya may be computed by first computing the exponential expression α^{Xb}, then dividing the result by p and retaining only the remainder.

If α and p are appropriately chosen, it is computationally infeasible to compute Xa from Ya. That is to say, the cost of performing such a task, in terms of memory or computing time needed, is large enough to deter eavesdroppers. In any event, new X and Y values can be chosen for each message, which is short enough to preclude the possibility of any X value being computed from a corresponding Y value.

After the exchange of the values Ya, Yb, each user computes a session key K in its session key generator 18, 20, by raising the other user's Y value to the power represented by the user's own X value, all modulo p. For user A, the computation is:

K = Yb^{Xa} mod p.

Substituting for Yb,

K = $(\alpha^{Xb})^{Xa}$ mod p = α^{XaXb} mod p.

For user B, the computation is:

K = Ya^{Xb} mod p.

Substituting for Ya,

K = $(\alpha^{Xa})^{Xb}$ mod p = α^{XaXb} mod p.

The two users A, B now have the same session key K, which is input to a conventional cryptographic device 28, 30. A transmitting cryptographic device, e.g. 28, transforms a plaintext message M into ciphertext C for transmission on the communications channel 26, and a receiving cryptographic device 30 makes the inverse transformation back to the plaintext M.

The Hellman et al. 770 patent points out that the generation of a session key is secure from eavesdropping, because the information exchanged on the insecure channel includes only the Y values, from which the corresponding X values cannot be easily computed. However, this form of key exchange system still has two significant problems. One is that the system is vulnerable to attack from active eavesdropping, rather than the passive eavesdropping described in the patent. The other is that identity authentication can be provided only by means of a public key directory.

Active eavesdropping takes place when an unauthorized person places a substitute message on the communications channel. FIG. 2 depicts an example of active eavesdropping using the same components as FIG. 1. The active eavesdropper E has broken the continuity of the unsecured line 26, and is receiving messages from A and relaying them to B, while sending appropriate responses to A as well. In effect, E is pretending to be B, with device Eb, and is also pretending to be A, with device Ea. E has two cryptographic devices 34a, 34b, two session key generators 36a, 36b, and two number generators 38a, 38b. When device Eb receives Ya from A, it generates Xb' from number generator 38b, computes Yb' from Xb' and transmits Yb' to A. Device Eb and user A compute the same session key and can begin communication of data. Similarly, device Ea and user B exchange Y numbers and both generate a session key, ifferent from the one used by A and Eb. Eavesdropper E is able to decrypt the ciphertext C into plaintext M, then encipher again for transmission to B. A and B are unaware that they are not communicating directly with each other.

In accordance with the present invention, each user is provided with proof of identity of the party with whom he is conversing, and both active and passive eavesdropping are rendered practically impossible. FIG. 3 shows the key management approach of the present invention, using the same reference numerals as FIGS. 1 and 2, except that the session key generators are referred to in FIG. 3 as $18'$ and $20'$, to indicate that the key generation function is different in the present invention. The user devices also include a number storage area 40, 42. Storage area 40 contains a preselected number Xa, stored at the time of manufacture of the A device, and another number referred to as "signed Ya," also stored at the time of manufacture. Xa was chosen at random, and is unique to the device. Ya was computed from Xa using the transformation

$$Ya = \alpha^{Xa} \bmod p.$$

Then the Ya value was concatenated with a number IDa uniquely identifying the user A device, such as a manufacturer's serial number, and then encoded in such a way that it was digitally "signed" by the manufacturer for purposes of authenticity. The techniques for digitally signing data are known in the cryptography art, and some will be discussed below. For the present, one need only consider that the number designated "signed (Ya,IDa)" contains the value Ya and another value IDa uniquely identifying the A device, all coded as a "signature" confirming that the number originated from the manufacturer and from no-one else. User B's device 12 has stored in its storage area 42 the values Xb and signed (Yb,IDb).

Users A and B exchange the signed (Ya,IDa) and signed (Yb,IDb) values, and each session key generator 18, 20 then "unsigns" the received values and verifies that it is conversing with the correct user device. The user identifiers IDa and IDb are known publicly, so user device A verifies that the number IDb is contained in the signed (Yb,IDb) number that was received. Likewise, user device B verifies that the value signed (Ya,IDa) contains the known value IDa. By performing the process of "unsigning" the received messages, the user devices also confirm that the signed data originated from the manufacturer and not from some other entity.

Since the Xa, Xb values are secret values, and it is infeasible to obtain them from the transmitted signed (Ya,IDa) and signed (Yb,IDb) values, the users may both compute identical session keys in a manner similar to that disclosed in the Hellman et al. '770 patent. If an eavesdropper E were to attempt to substitute fake messages for the exchanged ones, he would be unable to satisfy the authentication requirements. E could intercept a signed (Ya,IDa) transmission, could unsign the message and obtain the values Ya and IDa. E could similarly obtain the values Yb and IDb. However, in order for E and A to use the same session key, E would have to generate a value Xe, compute Ye and concatenate it with IDb, which is known, and then digitally "sign" the composite number in the same manner as the manufacturer. As will be explained, digital signing involves a transformation that is very easy to effect in one direction, the unsigning direction, but is computationally infeasible in the other, the signing direction. Therefore, eavesdropper E would be unable to establish a common session key with either A or B because he would be unable to generate messages that would satisfy the authentication requirements.

As described thus far, the technique of the invention establishes a session key that is derived from X and Y values stored in the devices at the time of manufacture. Ideally, a new session key should be established for each exchange of message traffic. An additional unsecured exchange is needed to accomplish this.

The number generator 14 in the A device 10 generates a random number $X'a$ and the number generator 16 in the B device 12 generates a random number $X'b$. These are supplied to the session key generators $18'$, $20'$, respectively, which generate values $Y'a$ and $Y'b$ in accordance with the transformations:

$$Y'a = \alpha^{X'a} \bmod p,$$

$$Y'b = \alpha^{X'b} \bmod p.$$

These values are also exchanged between the A and B devices, at the same time that the values of signed (Ya,IDa) and signed (Yb,IDb) are exchanged. After the authenticity of the message has been confirmed, as described above, the session key generators perform the following transformations to derive a session key. At the A device, the session key is computed as

$Ka = (Y'b)^{Xa} \bmod p \oplus (Yb)^{X'a} \bmod p,$

and at the B device, the session key is computed as

$Kb = (Y'a)^{Xb} \bmod p \oplus (Ya)^{X'b} \bmod p,$

where "⊕" means an exclusive OR operation.

Thus the session key is computed at each device using one fixed number, i.e. fixed at manufacturing time, and one variable number, i.e. chosen at session time. The numbers are exclusive ORed together on a bit-by-bit basis. It can be shown that Ka = Kb by substituting for the Y values. Thus:

$Ka = (\alpha^{X'b})^{Xa} \bmod p \oplus (\alpha^{Xb})^{X'a} \bmod p$
$= (\alpha^{Xa})^{X'b} \bmod p \oplus (\alpha^{X'a})^{Xb} \bmod p$
$= (Ya)^{X'b} \bmod p \oplus (Y'a)^{Xb} \bmod p$
$= (Y'a)^{Xb} \bmod p \oplus (Ya)^{X'b} \bmod p$
$= Kb.$

This common session key satisfies secrecy and authentication requirements, and does not require double encryption-decryption or the use of a public key directory or key distribution center. The only requirement is that of a manufacturer who will undertake to supply devices that have unique device ID's and selected X values encoded into them. For a large corporation or other organization, this obligation could be assumed by the organization itself rather than the manufacturer. For example, a corporation might purchase a large number of communications devices and complete the manufacturing process by installing unique ID's, X values, and signed Y values in the units before distributing them to the users. This would relieve the manufacturer from the obligation.

The process described above uses parameters that must meet certain numerical restrictions. The length restrictions are to ensure sufficient security, and the other requirements are to ensure that each transformation using modulo arithmetic produces a unique transformed counterpart. First, the modulus p must be a strong prime number 512 bits long. A strong prime number is a prime number p that meets the additional requirement that (p-1)/2 has at least one large prime factor or is preferably itself a prime number. The base number must be a 512-bit random number that satisfies the relationships:

$\alpha^{(p-1)/2} \bmod p = p-1$, and
$1 < \alpha < p-1.$

Finally, the values X and X' are chosen as 512-bit random numbers such that

$1 < X, X' < p-1.$

As indicated above, the process of authentication in the invention depends on the ability of the manufacturer, or the owner of multiple devices, to supply a signed Y value with each device that is distributed. A digital signature is a property of a message that is private to its originator. Basically, the signing process is effected by a transformation that is extremely difficult to perform, but the inverse transformation, the "unsigning," can be performed easily by every user. The present invention is not limited to the use of a particular digital signature technique.

One approach is to use an RSA public key signature technique. The RSA technique takes its name from the initial letters of its originators, Rivest, Shamir and Adleman, and is one of a class of encryption schemes known as exponentiation ciphers. An exponentiation cipher makes the transformation $C = P^e \bmod n$, where e and n constitute the enciphering key. The inverse transformation is accomplished by $P = C^d \bmod n$. With appropriate selection of n, d and e, the values of n and d can be made public without giving away the exponent e used in the encryption transformation. Therefore, a digital signature can be applied to data by performing the exponentiation transformation with a secret exponent e, and providing a public decryption exponent d, which, of course, will be effective to decrpyt only properly "signed" messages.

In the preferred embodiment of the present invention, another approach is used for digital signature, namely a modular square-root transformation. In the expression $x = m^2 \bmod n$, the number m is said to be the square root of x mod n, or the modular square root of x. If n is appropriately selected, the transformation is very difficult to perform in one direction. That is to say, it is very difficult to compute m from x, although easy to compute x from m. If the modulus n is selected to be the product of two large prime numbers, the inverse or square-root transformation can only be made if the factors of the modulus are known. Therefore, the modulus n is chosen as the product of two prime numbers, and the product is 1,024 bits long. Further, the factors must be different in length by a few bits. In the devices using the present invention, the value "signed (Ya,IDa)" is computed by first assembling or concatenating the codes to be signed. These are:

1. A numerical code IDa uniquely identifying the A device. In the present embodiment of the invention, this is a ten-digit (decimal) number encoded in ASCII format, but it could be in any desired format.
2. A number of ASCII numerical codes indicating a version number of the device. This may be used for device testing or analyzing problems relating to device incompatability.
3. The value Ya computed from the chosen value of Xa, encoded in binary form.
4. A random value added to the least-significant end of the composite message, and used to ensure that the composite message is a perfect

modular square.

The last element of the message is needed because of inherent properties of the modular squaring process. If one were to list all possible values of a modular square x, from 1 to n-1, and all corresponding values of the modular square root m, some of the values of x would have multiple possible values of m, but others of the values of x would have no corresponding values of m. The value added to the end of the message ensures that the number for which a modular square root is to be computed, is one that actually has a modular square root. A simple example should help make this clear.

Suppose the modulus n is 7849. It can be verified by calculator that a value x of 98 has four possible values of m in the range 1 to n-1: 7424, 1412, 6437 and 425, such that m^2 mod 7849 = 98. However, the x value 99 has no possible modular square root values m. If the composite message to be signed had a numerical value of 99, it would be necessary to add to it a value such as 1, making a new x value of 100, which has four possible square root values in the range 1 to n-1, namely 1326, 7839, 10 and 6523. In most instances, it does not matter which of these is picked by the modular square root process employed, since the squaring or "unsigning" process will always yield the composite message value 100 again. However, there are a few values of m that should be avoided for maximum security. If the x value is a perfect square in ordinary arithmetic (such as the number 100 in the example), two values of m that should be avoided are the square root of x by ordinary arithmetic (the number 10 in the example), and the number that is the difference between the modulus n and the ordinary-arithmetic square root of x (i.e. 7839 in the example). If a number fitting this definition is used as a signed message, the signature is subject to being "forged" without knowledge of the factors of n. Therefore, such numbers are avoided in assigning signatures, and each device can be easily designed to abort an exchange when the signed message takes the form of one of these avoided numbers.

When the modular square root process is used for digitally signing the composite data stored in each device, the "unsigning" process upon receipt of a signed composite message is simply the squaring of the message, modulo n. The value n is not made public, although it could be determined by close examination of one of the devices. Even with knowledge of the modulus n, however, the computation of the modular square root is computationally infeasible without knowledge of the factorization of n.

With a knowledge of the factorization of the modulus n, the computation of the modular square root becomes a feasible, although laborious task, which may be performed by any known computational method. It will be recalled that this process is performed prior to distribution of the devices embodying the invention, so computation time is not a critical factor.

It will be understood that the cryptographic technique of the invention may be implemented in any form that is convenient for a particular application. Modular arithmetic is now well understood by those working in the field, and may be implemented in hardware form in the manner described in the '770 Hellman et al. patent. More conveniently, off-the-shelf modular arithmetic devices are available for connection to conventional microprocessor hardware. For example, part number CY1024 manufactured by CYLINK, of Sunnyvale, California 94087, performs modular addition, multiplication and exponentiation.

For application to facsimile communications, the technique of the invention may be made completely "transparent" to the user. FIG. 4 shows the architecture of a device for connection between a conventional FAX machine 50 and a telephone line 52. The device includes a first conventional modem 54 (modulator/demodulator) for connection to the FAX machine 50 and a second modem 56 for connection to the telephone line 52. The modems 54, 56 function to demodulate all messages entering the device from either the FAX machine or the telephone line, and to modulate messages for transmission to the FAX machine or onto the telephone line. The device further includes a communications processor 58 connected between the two modems 54, 56, and a cryptographic processor 60 connected to the communications processor 58. The communications processor 58 manages message traffic flow to and from the modems 54, 56 and to and from the cryptographic processor 60, and ensures that the necessary communications protocols are complied with. In one preferred embodiment of the invention, the communications processor is a microprocessor specified by part number MC68000, manufactured by Motorola Corporation.

As shown in FIG. 5, the cryptographic processor 60 includes a conventional microprocessor 62 having a data bus 64 and a data bus 66, to which various other modules are connected. The microprocessor 62 may be, for example, a National Semiconductor Company device specified by part number NSC800. The connected modules include a random access memory (RAM) 68, a read-only memory (ROM) 70, which serves as a storage area for the X value and the signed Y value, an integrated-circuit chip 72 for implementation of the Data Encryption Standard (DES), a modular arithmetic device 74 such as the CYLINK CY1024, and an interface module 76 in the form of a dual-port RAM, for connection to the communications processor 58.

For transparent operation of the device shown in FIGS. 4 and 5, a user supplies not only the telephone number of a destination FAX machine, but also the ID of the intended destination FAX encoding/decoding device. When the digitally signed Y values are exchanged, the sending user device automatically "unsigns" the transmission by performing a modular squaring function; then compares the intended destination ID with the user ID returned with the Y value, and aborts the session if there is not a match. The key management steps previously described proceed automatically under control of the cryptographic processor 60, and when a session key has been derived, this is automatically applied in a conventional cryptographic process, such as the DES, to encrypt and decrypt a facsimile transmission.

It will be appreciated from the foregoing that the present invention represents a significant advance in cryptographic systems. In particular, the invention provides a technique for establishing a common session key for two users by means of an exchange of messages over an insecure communications channel. What distinguishes the invention from prior approaches to public key exchange systems is that the technique of the invention provides for identity authentication of the users without the need for a key distribution center or a public key register. Further, the technique is resistant to both passive and active eavesdropping. It will also be appreciated that, although an embodiment of the invention has been described in detail for purposes of illustration, various modifications may be made without departing from the spirit and scope of the invention. Accordingly, the invention is not to be limited except as by the appended claims.

Claims

1. A secure key generator, comprising:
storage means for storing a number of a first type selected prior to placing the key generator in service, and a digitally signed composite quantity containing both a unique and publicly known identifier of the key generator and a number of a second type obtained by a practically irreversible transformation of the number of the first type;
a first input connected to receive the number of the first type;
a second input connected to receive an input quantity transmitted over an insecure communications channel from another key generator, the input quantity being digitally signed and containing both a publicly known identifier of the other key generator and a number of the second type generated by a practically irreversible transformation of a number of the first type stored in the other key generator;
a first output for transmitting the stored, digitally signed composite quantity over the insecure communications channel to the other key generator;
a second output;
means for decoding the signed input quantity received at the second input, to obtain the identifier of the other key generator and the received number of the second type; and
means for generating a session key at the second output, by performing a practically irreversible transformation of the number of the second type received through the second input, using the number of the first type received through the first input.

2. A secure key generator as defined in claim 1, wherein the key generator further comprises:
a third input, connected to receive another number of the first type, generated randomly;
means for generating at the first output, for transmission with the digitally signed composite quantity, a number of the second type obtained by a practically irreversible transformation of the number of the first type received through the third input; and
means for receiving from the second input another number of the second type generated in and transmitted from the other key generator;
and wherein the means for generating a session key performs a practically irreversible transformation involving both numbers of the first type, received at the first and third inputs, and both numbers of the second type received at the second input, whereby a different session key may be generated for each message transmission session.

3. A secure key generator as defined in claim 1, wherein:
the number of the second type stored in digitally signed form in the storage means is obtained by the transformation $Ya = \alpha^{Xa} \bmod p$, where Xa is the number of the first type stored in the storage means, and α and p are publicly known transformation parameters;
the number of the second type received in the digitally signed composite quantity from the other key generator is designated Yb; and
the means for generating the session key performs the transformation $K = Yb^{Xa} \bmod p$.

4. A secure key generator as defined in claim 2, wherein:
the number of the second type stored in digitally signed form in the storage means is obtained by the transformation $Ya = \alpha^{Xa} \bmod p$, where Xa is the number of the first type stored in the storage means, and α and p are publicly known transformation parameters;

the number of the second type received in the digitally signed composite quantity from the other key generator is designated Yb; and

the means for generating the session key performs the transformation

$K = (Y'b)^{X'a} \bmod p \oplus (Yb)^{X'a} \bmod p,$

where X'a is the number of the first type that is randomly generated, Y'b is the additional number of the second type received from the other key generator, and the $\oplus$ symbol denotes an exclusive OR operation.

5. A method of generating a secure session key between two user devices connected by an insecure communications channel, comprising the following steps performed at both devices:

transmitting a digitally signed composite quantity to the other device, the composite quantity including a publicly known device identifier IDa and a number Ya derived by a practically irreversible transformation of a secret number Xa that it is unique to the device;

receiving a similarly structured digitally signed composite quantity from the other device;

transforming the received digitally signed composite quantity into an unsigned composite quantity containing a device identifier IDb of the other device and a number Yb that was derived by transformation from a secret number Xb that is unique to the other device;

verifying the identity of the other device from the device identifier IDb; and

generating a session key by performing a practically irreversible transformation involving the numbers Xa and Yb.

6. A method as defined in claim 5, and further including the steps of:

generating another number X'a randomly prior to generation of a session key;

transforming the number X'a to a number Y'a using a practically irreversible transformation;

transmitting the number Y'a to the other device; and

receiving a number Y'b from the other device;

wherein the step of generating a session key includes a practically irreversible transformation involving the numbers Xa, X'a, Yb and Y'b.

7. A method as defined in claim 6, wherein:

the transformations from X numbers to Y numbers is of the type $Y = \alpha^X \bmod p$, where α and p are chosen to maximize irreversibility of the transformations; and

the step of generating a session key includes the transformation

$K = (Y'b)^{Xa} \bmod p \oplus (Yb)^{X'a} \bmod p,$

where $\oplus$ denotes an exclusive OR operation.

8. A method of authentication in a public key cryptographic system, the method comprising the steps of:

selecting a unique random number Xi for each cryptographic device to be distributed;

transforming the number Xi to a new number Yi using a practically irreversible transformation;

forming a composite quantity by combining the number Yi with a publicly known device identifier IDi;

digitally signing the composite quantity containing Yi and IDi;

storing the signed composite quantity and the number Xi permanently in each device;

exchanging, between two devices, a and b, desiring to establish secured communication, the signed composite quantities stored in each;

authenticating, in each of the two devices, the identity of the other device; and

generating, in each of the two devices, a session key to be used for secured communication.

9. A method as defined in claim 8, wherein the step of authenticating includes:

transforming the digitally signed composite quantity received from the other device into unsigned form; and

comparing the value of IDb in the unsigned quantity with the known IDb of the other device.

10. A method as defined in claim 9, wherein:

the step of generating the session key includes performing a transformation that involves a value Yb received from the other device and the value Xa of this device.

11. A method as defined in claim 10, wherein:

the step of digitally signing includes computing a modular square root of the composite quantity; and

the step of transforming the digitally signed composite quantity to unsigned form includes computing a modular square of the signed quantity.

12. A method as defined in claim 11, wherein:

the steps of computing a modular square root and computing a modular square both employ a modulus that is the product of two prime numbers.

13. A method as defined in claim 8, and further comprising the steps of:

transforming, in each of the two devices, the digitally signed composite quantity received from the other device into unsigned form; and

generating, in each of the two devices, a, b, a random number X'a, X'b;

transforming the numbers X'a, X'b into numbers Y'a, Y'b by a transformation that is practically irreversible; and

exchanging the numbers Y'a, Y'b between the two devices;

and wherein the step of generating the session key includes performing a practically irreversible transformation involving the numbers Xa, X'a, Yb, and Y'b in device a, and the numbers Xb, X'b, Ya, and Y'a in device b.

14. A method as defined in claim 13, wherein:

the transformations from X numbers to Y numbers is of the type $Y = \alpha^{X} \bmod p$, where α and p are chosen to maximize irreversibility of the transformations; and
the step of generating a session key includes the transformations
$K = (Y'b)^{Xa} \bmod p \oplus (Yb)^{X'a} \bmod p$,
for device a, and
$K = (Y'a)^{Xb} \bmod p \oplus (Ya)^{X'b} \bmod p$,
for device b, where $\oplus$ denotes an exclusive OR operation.

15. A method as defined in claim 13, wherein:
the step of digitally signing includes computing a modular square root of the composite quantity; and
the step of transforming the digitally signed composite quantity to unsigned form includes computing a modular square of the signed quantity.

16. A method as defined in claim 15, wherein:
the steps of computing a modular square root and computing a modular square both employ a modulus that is the product of two prime numbers.

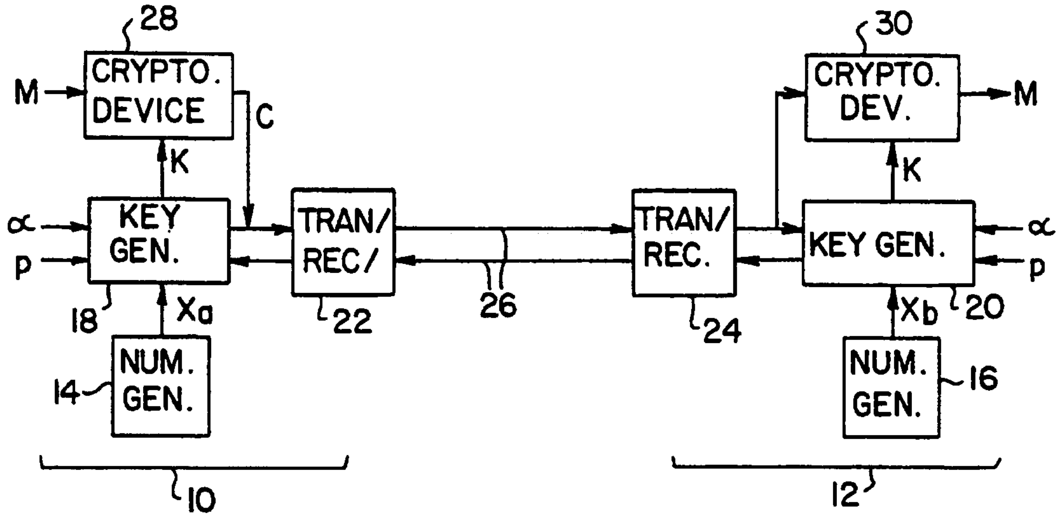

28
CRYPTO. DEVICE
M
C
K
KEY GEN.
α
p
18
Xa
NUM. GEN.
14
TRAN/ REC/
22
26
TRAN/ REC.
24
30
CRYPTO. DEV.
M
K
KEY GEN.
α
p
Xb
20
NUM. GEN.
16
10
12

FIG. 1 (PRIOR ART)

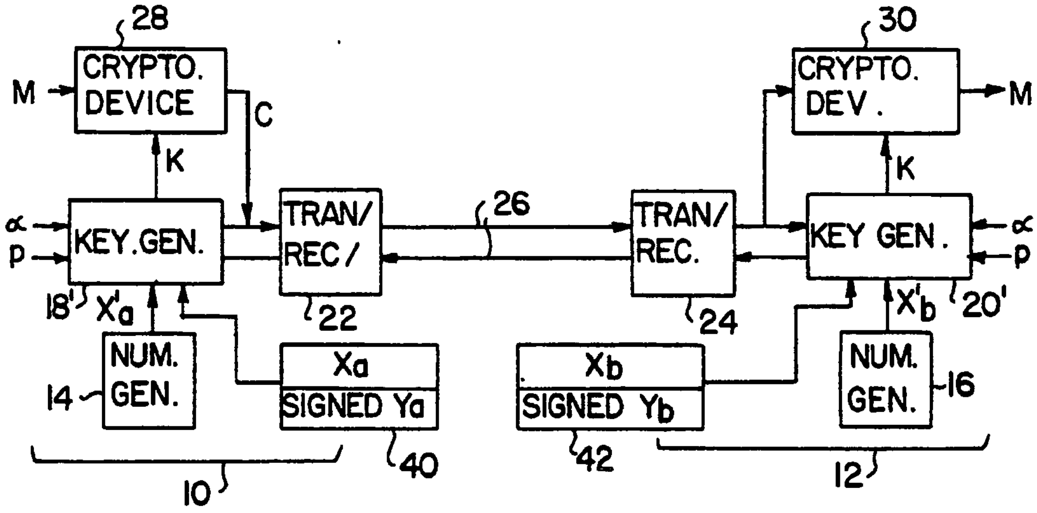

28
CRYPTO. DEVICE
M
C
K
KEY.GEN.
α
p
18'
X'a
NUM. GEN.
14
Xa
SIGNED Ya
40
TRAN/ REC/
22
26
TRAN/ REC.
24
Xb
SIGNED Yb
42
30
CRYPTO. DEV.
M
K
KEY GEN.
α
p
X'b
20'
NUM. GEN.
16
10
12

FIG. 3

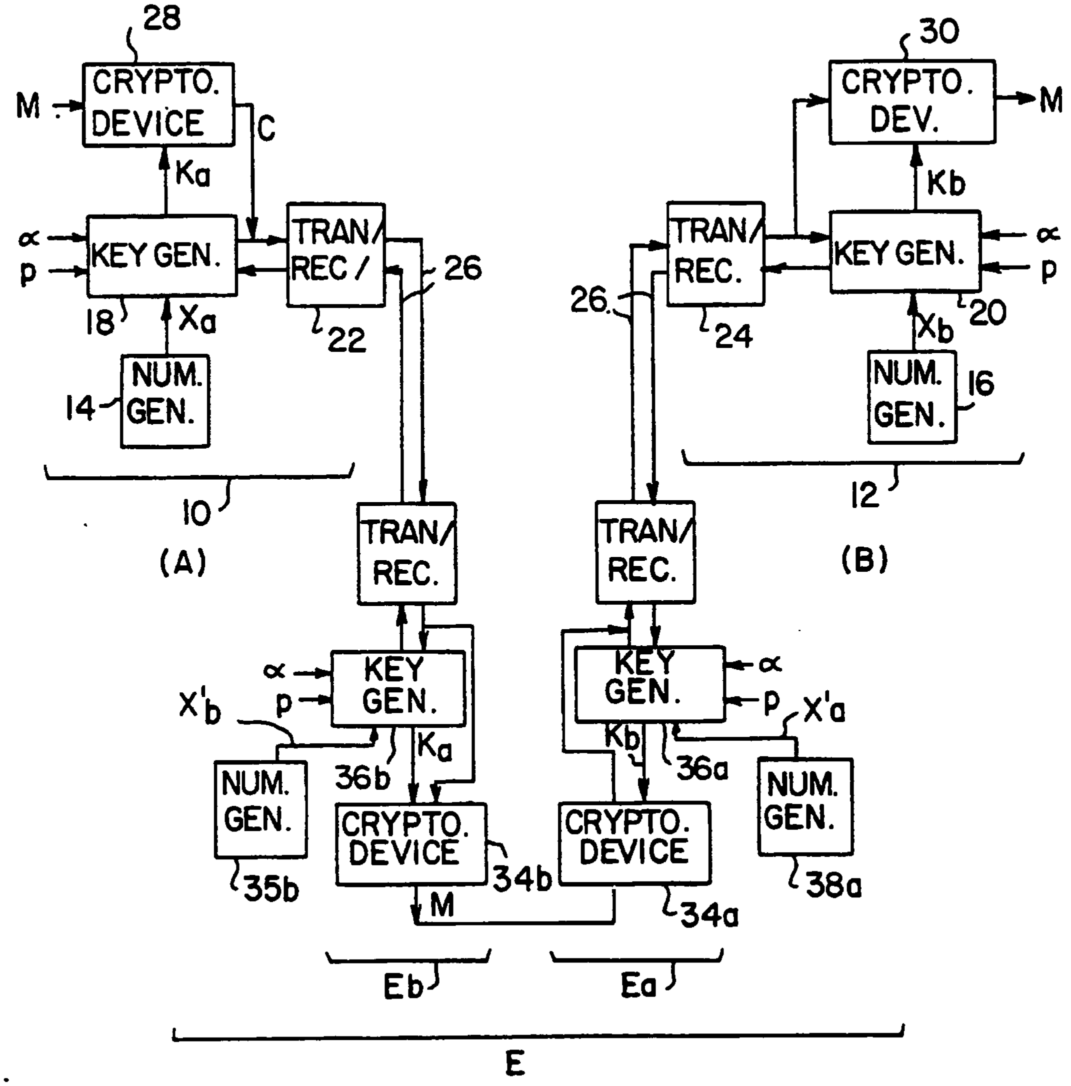
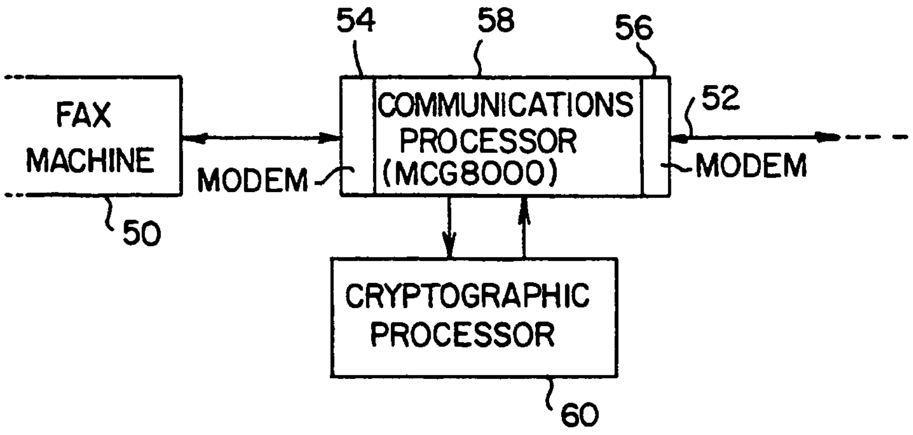

28
CRYPTO. DEVICE
M
C
Ka
α
p
KEY GEN.
18
Xa
TRAN/ REC/
26
22
NUM. GEN.
14
10
(A)
30
CRYPTO. DEV.
M
Kb
TRAN/ REC.
KEY GEN.
α
p
26.
24
20
Xb
NUM. GEN.
16
12
(B)
TRAN/ REC.
TRAN/ REC.
α
p
KEY GEN.
X'b
Ka
36b
NUM. GEN.
35b
CRYPTO. DEVICE
34b
M
KEY GEN.
α
p
X'a
Kb
36a
CRYPTO. DEVICE
34a
NUM. GEN.
38a
Eb
Ea
E

FIG. 2 (PRIOR ART)

FIG. 4


54
58
56
52
FAX MACHINE
COMMUNICATIONS PROCESSOR (MCG8000)
MODEM
MODEM
50
CRYPTOGRAPHIC PROCESSOR
60

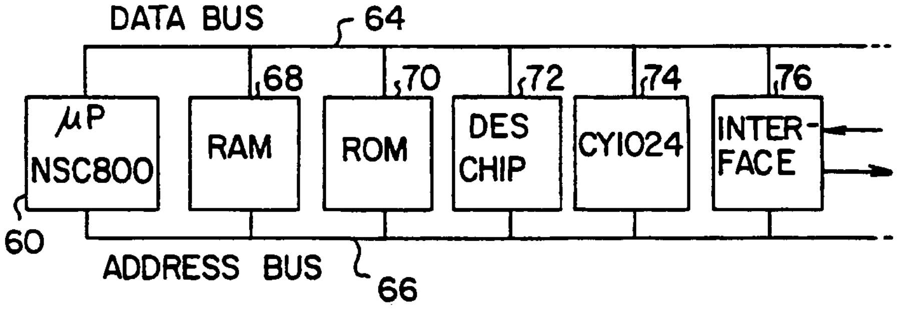

DATA BUS
64
68
70
72
74
76
µP NSC800
RAM
ROM
DES CHIP
CY1024
INTER-FACE
60
ADDRESS BUS
66

FIG. 5

⑲ Europäisches Patentamt
European Patent Office
Office européen des brevets

⑪ Publication number : **0 450 841 A2**

⑫ EUROPEAN PATENT APPLICATION

㉑ Application number : 91302657.1

㉒ Date of filing : 25.03.91

(51) Int. Cl.^{5} : **H04N 7/16**

㉚ Priority : **29.03.90 US 501620**
29.03.90 US 501682
29.03.90 US 501683
29.03.90 US 561684
29.03.90 US 501685
29.03.90 US 501658

(43) Date of publication of application :
09.10.91 Bulletin 91/41

(84) Designated Contracting States :
BE DE FR GB IT

(71) Applicant : **GTE LABORATORIES INCORPORATED**
1209 Orange Street
Wilmington Delaware 01901 (US)

(72) Inventor : **Walker, Stephen S.**
117 Kelleher Road
Marlborough, MA 01752 (US)
Inventor : **Sidlo, Clarence M.**
5 Lowry Road
Framlingham, MA 01701 (US)
Inventor : **Teare, Melvin J.**
21 Woodleigh Road
Framlingham, MA 01701 (US)

(74) Representative : **Bubb, Antony John Allen et al**
GEE & CO. Chancery House Chancery Lane
London WC2A 1QU (GB)

(54) Video control system.

(57) A video control system includes a central facility (11) and a terminal (10). Video program means provided the terminal with a video program including a series of television fields including a first field containing both a random digital code encrypted according to a code encryption key and program identification data, and a second field containing an unintelligible video signal previously transformed from an intelligible video signal according to the random digital code. The terminal (10) includes means (22) for sending the program identification data to the central facility (11). The central facility includes a data base (19) for storing and retrieving at least one code encryption key corresponding to the program identification data and means (20) for sending the code encryption key from the central facility (11) to the terminal (10). The terminal (10) further includes means (22) for receiving the code encryption key from the central facility, decrypting means (23) for decrypting the encrypted digital code of the first frame in accordance with the code encryption key and means (24) for transforming the unintelligible video signal of the second frame to the intelligible video signal using the decrypted random digital code. The video program means may transmit the program to said terminal (10) or be located at the terminal (10) for playing a video recording medium storing the program.

EP 0 450 841 A2

Jouve, 18, rue Saint-Denis, 75001 PARIS

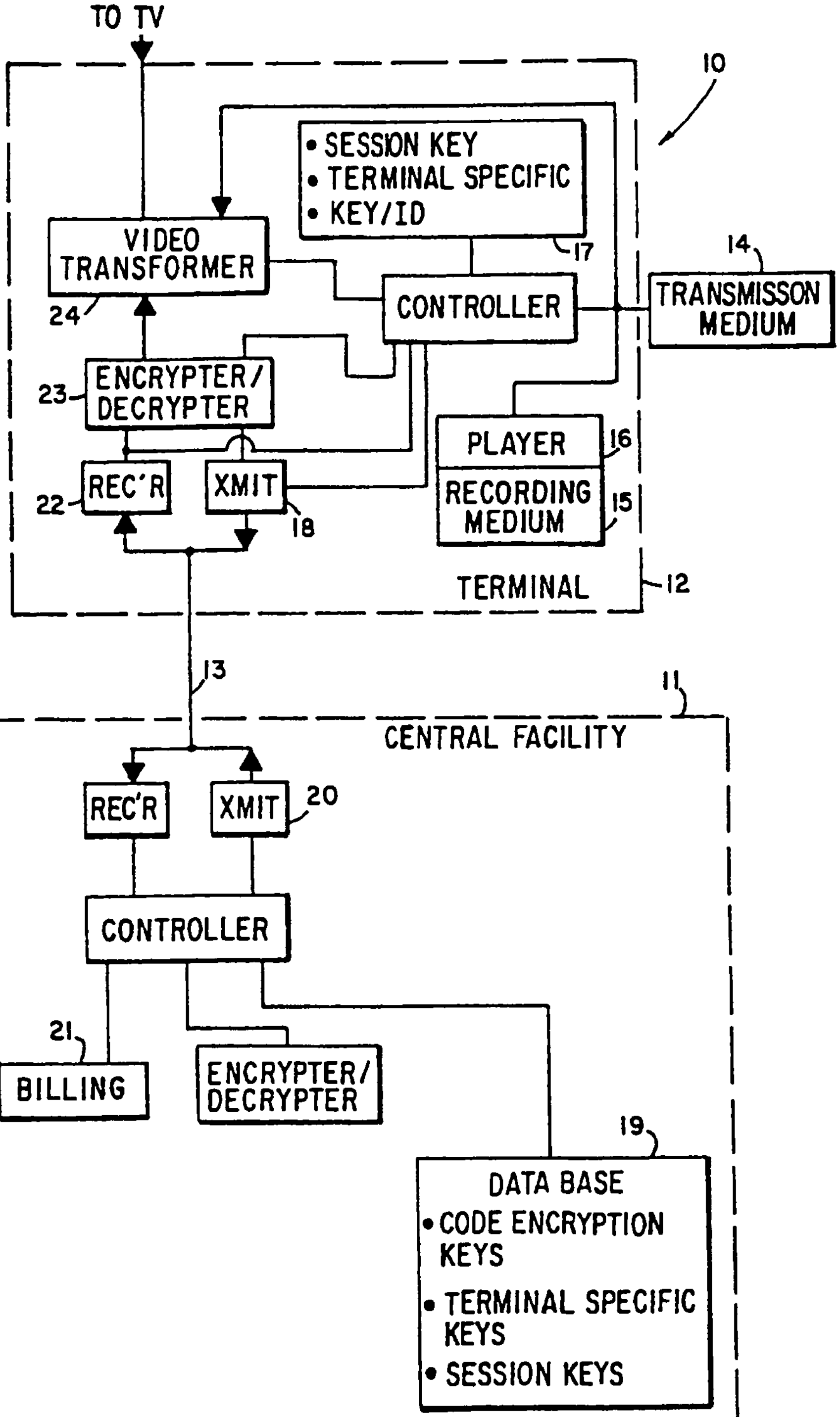

TO TV
10
• SESSION KEY
• TERMINAL SPECIFIC
• KEY/ID
VIDEO TRANSFORMER
17
14
CONTROLLER
TRANSMISSON MEDIUM
24
ENCRYPTER/ DECRYPTER
23
PLAYER
16
REC'R
XMIT
RECORDING MEDIUM
15
22
18
TERMINAL
12
13
11
CENTRAL FACILITY
REC'R
XMIT
20
CONTROLLER
21
BILLING
ENCRYPTER/ DECRYPTER
19
DATA BASE
• CODE ENCRYPTION KEYS
• TERMINAL SPECIFIC KEYS
• SESSION KEYS

FIG. 1

This invention is concerned with video control systems. It is desirable to provide a video control system which decrypts encrypted broadcasts or recorded copies of video material such that the subsequent viewing is controlled. This allows the owner to either forbid viewing, or collect revenue at his or her discretion.

In the prior art, a software distribution system is known wherein a computer program is downloaded once, followed by an access key to allow use of it on each subsequent use. This system uses a dynamic key that constantly changes, and is directly related to a user's decoder box, both by ID and an internal dynamic counter.

Also known is a video system that autonomously controls the viewing of a recording for either 24 hours or once only. It does not have the power of control desired.

Accordingly the present invention provides a video system comprising: a central facility; a terminal; and video program means for providing to said terminal a video program including a series of television fields including a first field containing both a random digital code encrypted according to a code encryption key and program identification data, and a second field containing an unintelligible video signal previously transformed from an intelligible video signal according to said random digital code; said terminal including means for sending said program identification data to said central facility; said central facility including a data base for storing and retrieving at least one code encryption key corresponding to the program identification data and means for sending said code encryption key from said central facility to said terminal; said terminal further including means for receiving the code encryption key from said central facility, decrypting means for decrypting the encrypted digital code of said first frame in accordance with said code encryption key and means for transforming said unintelligible video signal of said second frame to said intelligible video signal using the decrypted random digital code.

One embodiment of the invention will now be described, by way of example, with reference to the accompanying drawings in which:

Figure 1 is a block diagram of a video system embodying the invention; and

Figure 2 shows an encryption arrangement according to the invention.

Reference is made to Figure 1 which is a block diagram of a video system 10 embodying the invention. The video system comprises a central facility 11, a terminal 12, and a duplex communication link 13 between central facility 11 and terminal 12. An overview of the system is first given.

Terminal 12 is provided with a video program including a series of television fields including a first field containing both a random digital code encrypted according to a code encryption key and program identification data, and a second field containing an unintelligible video signal previously transformed from an intelligible video signal according to the random digital code.

The video program may be transmitted by broadcast, cable, satellite, fiber, or any other transmission medium 14. Alternative the video program may be stored on a video recording medium 15 such as magnetic tape or video disk and played by player 16. The unintelligible video signal may be either analog or digital.

A second field has a vertical blanking interval containing both a random digital code encrypted according to a code encryption key and program identification data, is followed by a third field containing an unintelligible video signal previously transformed from an intelligible video signal according to the random digital code of the second field.

Terminal 12 includes means 17 to store terminal identification data and means to send to the central facility 11 the terminal identification data and the program identification data over link 13.

Central facility 11 includes a data base 19 for storing and retrieving at least one code encryption key corresponding to the program identification data, means 20 for sending the code encryption key from the central facility 11 to the terminal 12, and means 21 for generating billing data based on both terminal identification data and program identification data.

Terminal 12 further including means 22 for receiving the code encryption key from central facility 11, decrypting means 23 for decrypting the encrypted random digital code of the first frame in accordance with the code encryption key, and means 24 for transforming the unintelligible video signal of the second frame to the intelligible video signal using the decrypted random digital code.

Each terminal 12 may have a terminal specific encryption key and means 18 to send to the central facility the program identification data and the terminal 11 identification data encrypted according to the terminal specific encryption key. The central facility 11 has means for storing a duplicate of the terminal specific encryption key, means for encrypting the code encryption key according to the terminal specific encryption key; and means for sending the encrypted code encryption key from central facility 11 to terminal 12.

Terminal 12 further includes means 22 for receiving the encrypted code encryption key from central facility 11, decryption means 23 for decrypting the code encryption key according to the terminal specific encryption key, and decrypting the encrypted random digital code of the first frame in accordance with the code encryption key, and means 24 for transforming the unintelligible video signal of the second frame to the intelligible video signal using the decrypted random digital code.

Terminal 12 includes means to encrypt the terminal identification data according to the terminal specific encryption key, means to send unencrypted terminal identification data and encrypted terminal identification data to the central facility, which in turn includes means to compare unencrypted and encrypted terminal identification data to verify terminal identity.

A plurality of code encryption keys may be used for one program wherein a desired code encryption key is selected from the plurality of code encryption keys in accordance with code encryption key identification data corresponding to the random digital code.

Various features of the system are now discussed in more detail.

System 10 controls the viewing of video programs, by which is meant any video material, either transmitted or recorded, in television format consisting of a series of fields of lines. Two interlaced fields make up a television frame.

Video programs are rendered unintelligible, e.g. scrambled, by any analog or digital method, and are made intelligible, e.g. descrambled, using random digital codes located in fields. The random digital keys are themselves encrypted, and decrypted by a one or more key obtained from a database located at the central facility, along with user-specific information at the time of viewing. The system does not stop copying, it controls viewing, while protecting revenues. As such, it can encourage copying, which could ease the distribution issue by controlling the playback such that revenue can be collected each time.

Preferably duplex communication link 13 is a continuous data channel between a terminal and a central facility such as an ISDN D-channel or by modem over a regular phone line.

The video program is encrypted, and needs a decrypter in the terminal for viewing. The decrypter uses data embedded in the video program along with a data access to correctly perform the decryption, so the process is completely controlled. The embedded data and key transfer from the remote database may be protected with public domain encryption techniques, providing high level security before first viewing.

The video program may be recorded as is, but it is still unviewable. To view it, the decrypter is used, along with the encrypted embedded data, and an access to a secure database, to perform the decryption. Recordings may be freely copied, but remain unviewable unless used with the decrypter.

To view the programs requires access to the database using encrypted data transfer. This process yields the control of the video program, whether recording or trans mission. The decrypter requires one or more keys that arrives from the database. To get the key, information from the video program as well as terminal identification is sent to the database.

A direct Electronic funds Transfer (EFT) debit can be performed using the information. It the program is a video store copy, the EFT could include the store fee and the copyright fee. Note that the video distribution to video stores becomes trivial, as they are encouraged to take a direct recording with a video store key, along with their authorized converter box, and make as many copies as they like. The revenue control takes place at viewing time. This encourages a shareware type of distribution.

A passkey can be sent to the database, to allow viewing of questionable taste films by adults, controlling access by minors.

On the first access, the database will capture a signature derived from the user's equipment and the recording, and store it for subsequent tracking. As there is a compelled database access in this process, data on usage may be collected. This same process may be used for revenue collection.

The system preferably uses at least one downloadable key, an encrypted video program that uses the key for decryption, and data stored in a field of the video program. It may be implemented in an all digital, analog, or mixed analog/digital environment.

The video programs are encrypted, with data relating to the programs, e.g. where and when, who transmitted it. The data may also contain part of the decryption key. This information would be extracted from the signal, and used to access a database, maintained by the program's owners, to obtain an encrypted key for the decrypter. After a subscriber and/or a credit check is successfully completed, the one or more keys would be transmitted. At this time the owner has obtained usage data, with a specific user's ID, and has the option of billing him. If it is a free program, at least the viewer data is available.

If a user records a transmission or another recording, he captures the encrypted signal, along with embedded data, as described above. This accomplishes the signature part of the process. A recording created by this method may be on a regular VCR, but is encrypted and individually marked. Copying a recording does not affect the system, as the rerecording is only usable with the correct keys. Potentially, the first few minutes of a program might be viewable without the need of a key, to allow the user to see what the contents of the program are, as well as to allow time for the database access and key synchronization process.

To play a recording back, it is necessary to re-obtain the one or more keys. The combination of data stored in a field is used to access the database. Before the keys are made available, there is a check that the terminal identification and the embedded data match.

In the case wherein a recording is rented from a video store, a code may identify the store. The database recognizes the recording as a rental copy, and charge either the user or the video store a fee. If the recording is viewed a second time, the charge is repeated. In the event a copy is made, when it is played, the database will identify the originating video store, but not the actual copier. However, if validation is performed at rental time, there would be some measure of control. If the entire charging process were to be reversed, such that the viewer carries all the liability for charges, then copying is encouraged, as per shareware, and the distribution problem is minimized, while revenues are maintained on a usage basis.

The program's owner has the responsibility to get a secured copy to whoever deals with the distribution of the programs. The programs are encrypted, and require a database update to enable viewers to make use of the program. The viewer has a terminal including a decrypter, linked to the central facility's database via an automatic dial-up, that, when enabled, decrypts the video program. As appropriate, there can be credit checks and billing from the database, as well as statistics collection.

The encryption has two levels, one for protection of video decryption codes on the program, and one for protection of messages between the terminal and the central facility. Both may use the NBS Data Encryption Standard (DES).

DES encryption and decryption may be implemented with a commercial Motorola 6859 Data Security Device or similar product at the terminal and at the central facility.

The decryption code itself is protected by being DES-encrypted. The decryption key is not on the video program but is retained in the database at the central facility. A program identification number and a decryption key number allow the central facility to recover the decryption key itself and send it to the terminal for decrypting the decryption codes.

A different DES decryption key is not required for every field. One key can span several fields. DES key requests and acknowledgements from the terminal may also act as keep-alive messages to the central facility.

DES decryption keys are transmitted from the central facility to the terminal protected by a higher-level DES "session" key. terminal requests for new keys as the tape progresses are also protected by the DES session key. This key is generated by the central facility at the beginning of the session and remains valid for the duration of the session. The terminal begins the session using a terminal-unique DES key stored in a ROM.

Frame contents are transferred from the Analog Subsystem to the DCSS and the decrypted decryption code from the DCSS to the Analog Subsystem over the analog interface shown in the Figure. Transfer of data between the subsystems may be coordinated by means of the vertical and horizontal blanking signals and their derivative interrupts.

All messages between terminal and central facility use Cyclic Redundancy Code (CRC) checking to verify message integrity. The CRC-CCITT generating polynomial generates two block check characters (BCC) for each message. If the terminal receives a message that is not verified by the BCC, it sends a request (ARQ) to the central facility to retransmit the last message. The central facility does not attempt to ARQ garbled messages. It discards them and waits for a terminal to send again.

Message exchange in the VCS is by a positive acknowledgent scheme in which a response of some kind is expected for every message sent. For example, a terminal expects a DES decryption key message after it sends a request for the same; the central facility expects a key receipt acknowledge after it sends the key message.

When a user begins to play a protected program, the terminal initiates a session by sending a "session start" message (STS) to the central facility containing user and program identifications. The message contains message type, user number and CRC code in the clear, but the balance of the message is DES-encrypted with the initial DES session key stored in the terminal ROM. (The user identification is also stored in ROM.) The central facility uses the unencrypted data to access its database and find the user DES value for decrypting the remainder of the message.

The central facility authenticates the message by comparing clear and decrypted user numbers. If the user numbers are identical, the central facility then confirms that the program serial number is valid. The central facility may also check user credit. If all is well, the central facility accepts the session and generates a new (and random) DES key that is unique for that session. It encrypts this using the initial user value in the database and sends it to the terminal, which decrypts the message and stores the new value in its database (MCU RAM) as the session key for the remainder of the session.

The central facility then uses the tape and decryption key number in the STS message to recover a set of DES decryption keys for the program from the database. These are encrypted with the session key and sent to the terminal at the start of a session or during the course of a session.

The terminal generates session start, key acknowledgement, and ARQ messages. The central facility responds in kind. Both the central facility and the terminal generate and verify block check characters.

The preferred embodiment and best mode of practicing the invention have been described. Alternatives now will be apparent to those skilled in the art in light of these teachings. Accordingly the invention is to be defined by the following claims and not by the particular examples given.

Claims

1. A video system comprising:
   a central facility;
   a terminal; and
   video program means for providing to said terminal a video program including a series of television fields including a first field containing both a random digital code encrypted according to a code encryption key and program identification data, and a second field containing an unintelligible video signal previously transformed from an intelligible video-signal according to said random digital code;
   said terminal including means for sending said program identification data to said central facility;
   said central facility including a data base for storing and retrieving at least one code encryption key corresponding to the program identification data and means for sending said code encryption key from said central facility to said terminal;
   said terminal further including means for receiving the code encryption key from said central facility, decrypting means for decrypting the encrypted digital code of said first frame in accordance with said code encryption key and means for transforming said unintelligible video signal of said second frame to said intelligible video signal using the decrypted random digital code.

2. The system of claim 1 wherein a plurality of code encryption keys are used for one program, and wherein a desired code encryption key is selected from said plurality of code encryption keys in accordance with code encryption key identification data corresponding to the random digital code encrypted with said desired code encryption key.

3. The system of claim 1 or 2 wherein said video program means is means for transmitting said program to said terminal.

4. The system of claim 3 wherein said means for transmitting is a CATV system.

5. The system of any one of claims 1-4 wherein:
   said terminal further includes means to store terminal identification data and a terminal specific encryption key; and means to send to said central facility said terminal identification data with said program identification data;
   said central facility further includes means for storing a duplicate of said terminal specific encryption key; means for encrypting said code encryption key according to said terminal specific encryption key; and means for sending the encrypted code encryption key from said central facility to said terminal; and
   said terminal further further includes means for receiving the encrypted code encryption key from said central facility; and decryption means for decrypting said code encryption key according to said terminal specific encryption key.

6. The video system of any one of claims 1-4 wherein:
   said terminal further includes means to store terminal identification data and a terminal specific encryption key; and means to send to said central facility said program identification data and said terminal identification data,
   said central facility further includes means for providing a session encryption key; means for encrypting said session encryption key according to said terminal specific encryption key; means for sending the encrypted session encryption key from said central facility to said terminal;
   means for encrypting said code encryption key according to said encrypted session encryption key; and means for sending the encrypted code encryption key from said central facility to said terminal; and
   said terminal further includes means for receiving the encrypted session encryption key from said central facility; decryption means for decrypting said session encryption key according to said terminal specific encryption key, means for receiving the encrypted code encryption key from said central facility; and decryption means for decrypting said code encryption key according to said session encryption encryption key.

7. The system of claim 5 or 6 wherein said terminal includes means to encrypt said terminal identification data according to said terminal specific encryption key, and means to send unencrypted terminal identification data and encrypted terminal identification data to said central facility, and said central facility includes means to compare unencrypted and encrypted terminal identification data to authenticate terminal identity.

8. The system of any one of claims 5-7 wherein said central facility further includes means for generating billing data based on said terminal identification data and said program identification data.

9. The video system of any one of claims 1-8 wherein said video program means is a means located at said terminal for playing a video recording medium storing said program.

10. A video recording medium storing a video program including a series of television fields including a first field containing both a random digital code encrypted according to a code encryption key and program identification data, and a second field containing an unintelligible video signal previously transformed from an intelligible video signal according to said random digital code.

11. The medium of claim 10 wherein a plurality of code encryption keys are used for one program, and wherein a desired code encryption key is selected from said plurality of code encryption keys in accordance with code encryption key identification data corresponding to the random digital code encrypted with said desired code encryption key.

12. The medium of claim 10 or 11 wherein said second field has a vertical blanking interval containing both a random digital code encrypted according to a code encryption key and program identification data, and is followed by a third field containing an unintelligible video signal previously transformed from an intelligible video signal according to said random digital code of the second field.

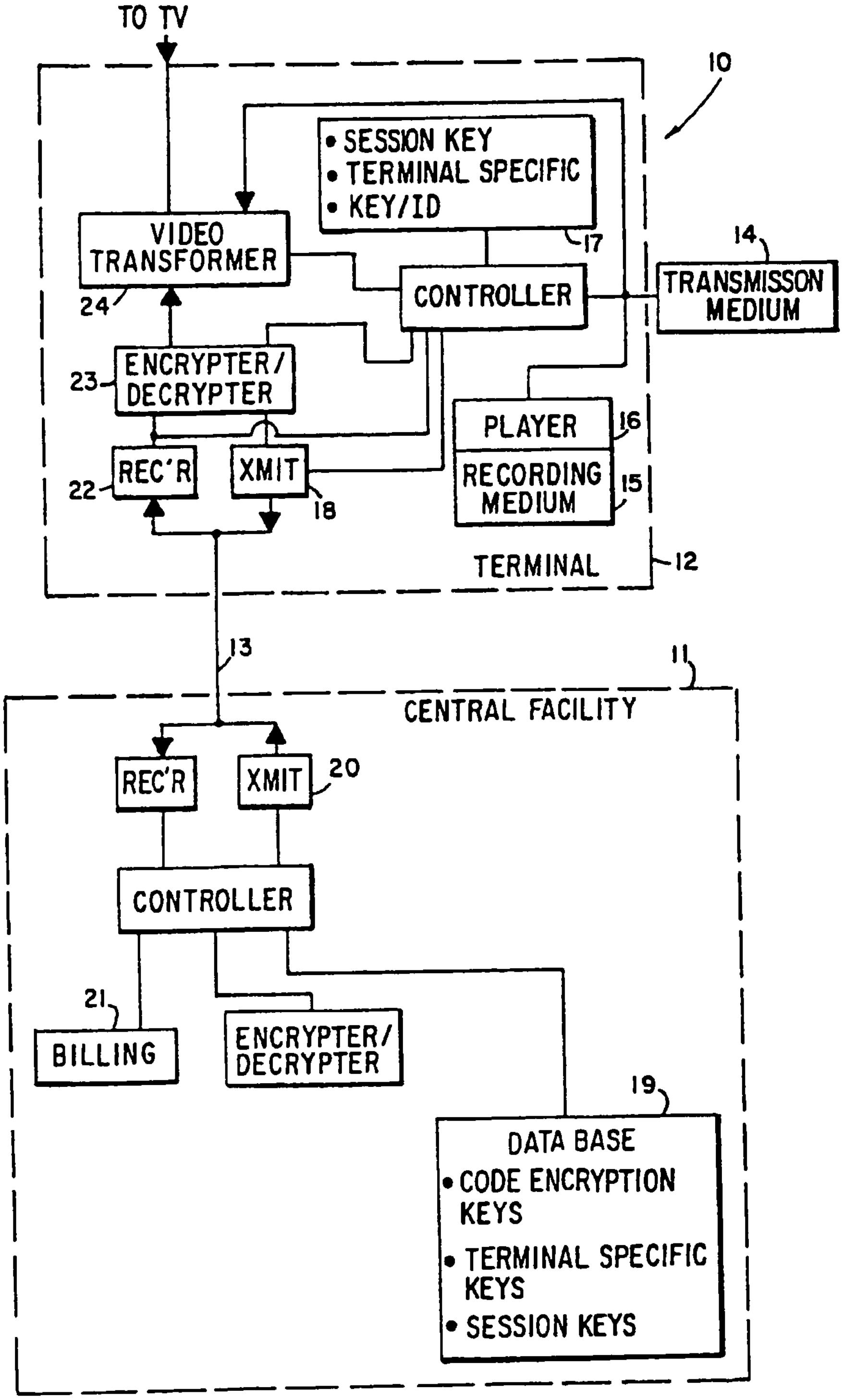

TO TV
10
• SESSION KEY
• TERMINAL SPECIFIC
• KEY/ID
17
VIDEO TRANSFORMER
24
CONTROLLER
14
TRANSMISSON MEDIUM
ENCRYPTER/ DECRYPTER
23
PLAYER
16
RECORDING MEDIUM
15
REC'R
22
XMIT
18
TERMINAL
12
13
11
CENTRAL FACILITY
REC'R
XMIT
20
CONTROLLER
21
BILLING
ENCRYPTER/ DECRYPTER
19
DATA BASE
• CODE ENCRYPTION KEYS
• TERMINAL SPECIFIC KEYS
• SESSION KEYS

FIG.1

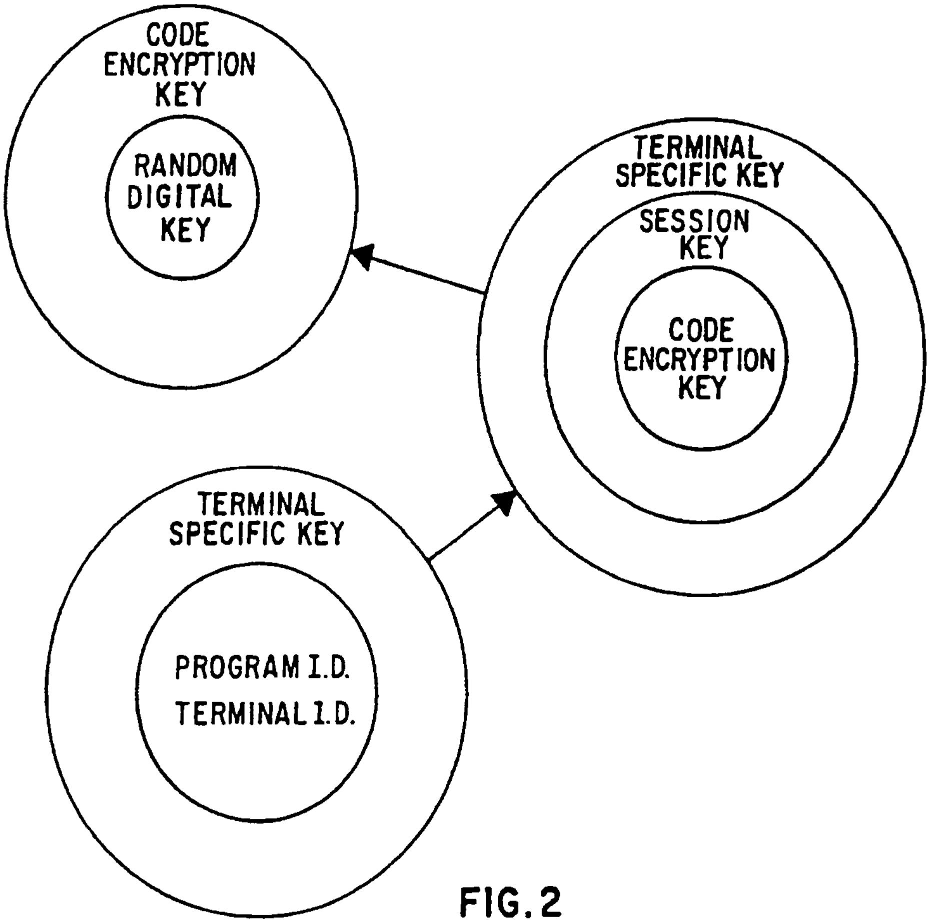

CODE ENCRYPTION KEY
RANDOM DIGITAL KEY
TERMINAL SPECIFIC KEY
SESSION KEY
CODE ENCRYPTION KEY
TERMINAL SPECIFIC KEY
PROGRAM I.D.
TERMINAL I.D.

FIG. 2

⑲ **Europäisches Patentamt**
European Patent Office
Office européen des brevets

⑪ Publication number: **0 613 073 A1**

⑫ EUROPEAN PATENT APPLICATION

㉑ Application number: **93306468.5**

(51) Int. Cl.^{5}: **G06F 1/00**

㉒ Date of filing: **17.08.93**

㉚ Priority: **23.02.93 GB 9303595**

㊸ Date of publication of application:
31.08.94 Bulletin 94/35

(84) Designated Contracting States:
DE FR GB SE

(71) Applicant: **INTERNATIONAL COMPUTERS LIMITED**
ICL House
Putney, London, SW15 1SW (GB)

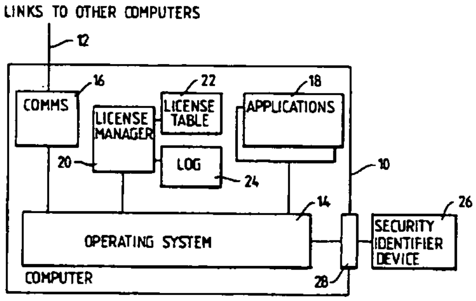
(72) Inventor: **Archer, Barrie**
Lilac Cottage,
Honey Hall
Wokingham, Berkshire RG11 3BA (GB)

(74) Representative: **Guyatt, Derek Charles**
Intellectual Property Department
International Computers Limited et al
Cavendish Road
Stevenage, Herts, SG1 2DY (GB)

(54) Licence management mechanism for a computer system.

(57) A computer system includes a license manager for regulating usage of software items. The license manager checks the host identity of the computer on which it runs and permits usage only if the host identity matches an identity value in a license key. The host identity of the computer is supplied by a security identification device removably coupled to an external port on the computer. Communication of the host identity between the security identifier device and the license manager is protected by encryption.

Fig.1.


LINKS TO OTHER COMPUTERS
12
16
COMMS
22
LICENSE TABLE
18
APPLICATIONS
LICENSE MANAGER
20
LOG
24
10
14
26
OPERATING SYSTEM
SECURITY IDENTIFIER DEVICE
COMPUTER
28

EP 0 613 073 A1

Rank Xerox (UK) Business Services
(3.10/3.09/3.3.4)

Background to the invention

This invention relates to a license management mechanism for a computer system, for controlling use of licensed software.

Software is normally licensed rather than sold in order that restrictions on unauthorised use can be legally enforced. Various schemes have been tried to make the software enforce these restrictions itself, including copy protection, hardware keys, etc., but the current trend is to the use of license keys that are packets of data which permit the software to work only on a particular machine.

One way in which this has been implemented is through the provision of a mechanism referred to as a license manager to which the handling of these license keys is delegated. By centralising the handling of the license keys it is possible to restrict the use of software not just to a single machine but to a network of machines. This provides additional flexibility for the user as well as providing the potential for more sophisticated control over the use of the software within a user organisation.

Central to the use of license managers to control the use of software in this way is the ability to identify which machine the license manager is running. If this were not done it would be possible to obtain license keys for use on one machine and use them on any number of machines. Various schemes have been used to achieve this identification, including serial numbers built into the machine processor, use of Ethernet DTE addresses, etc.

The object of the present invention is to provide a novel way of identifying the machine on which a license manager is running.

Summary of the invention

According to the invention there is provided a computer system including a license manager for regulating usage of software items in accordance with license keys issued to the license manager, the license manager being arranged to check the host identity of the computer on which it runs and to permit usage only if the host identity matches an identity value in the license keys, characterised in that the host identity of the computer is supplied by a security identification device removably coupled to an external port on the computer.

Such identification devices have been used for PC software to permit the software to run only on machines that have the device attached. These devices are usually referred to as dongles. The present invention differs from such known use of dongles in that in the present case the device is used to identify the machine to the license manager, rather than to authorise a particular item of software.

Brief description of the drawings

Figure 1 is a block diagram of a computer system embodying the invention.

Figure 2 is a flow chart showing the operation of a license manager in response to a request to use a feature.

Figure 3 is a flow chart showing a host identity checking function performed by the license manager.

Description of an embodiment of the invention

One embodiment of the invention will now be described by way of example with reference to the accompanying drawing.

Referring to Figure 1, the system comprises a number of computers 10, linked together by means of communications links 12 to form a data processing network.

Each of the computers runs an operating system 14 which controls and coordinates the operation of the computer, and communications software 16 which allows the computer to communicate with the other computers in the system over the links 12. Each computer also runs a number of applications 18 (where an application is any logical software entity).

At least one of the computers runs a program referred to herein as the license manager (LM) 20. The function of the LM is to regulate the applications within a particular domain, so that each application can be used only to the extent permitted by licenses granted to the system owner. The domain comprises those applications that can communicate with the LM. In this example, the domain extends over a multi-computer network, but in other examples it could consist of a single computer.

Each application has a number of features associated with it. A "feature" is defined herein as an aspect of an application that is subject to license control by the LM. A feature may, for example, simply be the invocation of the application by a user. However, more complex features may be defined such as number of users, number of communication links and database size.

Each application also has an application key associated to it, which is unique to the application. As will be described, application keys are used to ensure security of communication between the applications and the LM.

The LM has a private area of memory in which it maintains a license table 22 and a log 24.

The license table holds a number of license keys that have been issued for this system. Each license key contains the following package of information:-

Machine identifier: the identity of the computer on which the license manager is permitted to run.

Expiry date: the date until which the license key is valid.

Limit: the number of units of a particular feature that are licensed (eg the number of users, number of communication links, or database size).

Application key: the key value of the application to which the license key relates.

Signature: a cryptographic signature which ensures that the license key cannot be changed without detection.

Whenever one of the applications requires to use a feature, it sends a request message to the LM. The request message includes:

- the identity of the feature required
- the number of units of the feature required
- the application key
- a timestamp value.

Referring to Figure 2, when the LM receives this request message, it checks that the timestamp value is current. Assuming the timestamp value is current, the LM then checks whether there is a license key in the license table for the required feature.

If there is a license key in the table, the LM then checks whether the expiry date of the license has passed, and checks the signature of the license key to ensure that it has not been modified. The LM also checks whether the required number of units are available for the feature (ie whether the number of requested units plus the number of units already granted is less than or equal to the limit value in the license key).

If all these checks are satisfactory, the LM returns a "license granted" message to the application, sealed under the application key. The LM keeps a record of the number of units granted for each feature. If, on the other hand, any of the checks fails, the LM returns a "license denied" message to the application. The LM also writes a record in the log 24 to indicate whether a license has been granted or denied.

If the application receives a "license granted" message, it proceeds to use the requested features as required. If, on the other hand, it receives a "license denied" message, it performs one of the following actions, as determined by the designer of the application:

- the application may simply shut itself down.
- in the case where the license was denied because there were not enough units of the requested feature available, the application may display a "call again later" message to the user.
- the application may continue running in a reduced service mode eg a demonstration mode.

When an application terminates, it sends a "license relinquish" message to the LM. The LM will then withdraw any licenses issued to this application, making the units available to other applications.

Each application is required to send a revalidation message periodically to the LM, to re-validate its license. For example, a revalidation message may be required every 5 minutes. If the application does not receive any response to this message, it assumes that it has lost contact with the LM, and shuts down or continues in a reduced service mode.

The LM periodically checks whether it has received revalidation messages from all the application to which it has granted licenses. If a revalidation message has not been received from an application, the LM assumes that the application has failed, and therefore withdraws the license, making the units available to other applications.

In order to ensure that unauthorised copies of the LM cannot be run on other systems, it is necessary to provide a way of identifying the machine on which the LM runs. This is achieved by means of a security identification device (SID) 26, which stores an identifier unique to this device, referred to as the secure host identifier. The SID is attached to the computer 10 by way of an external port 28. In this example, the port is a standard parallel printer port, and the SID is designed so that a printer may be plugged into the back of the SID, so that both the printer and SID share the same port. Messages for the SID are identified by special commands.

In other embodiments of the invention, the SID may be attached to a special dedicated port, or to some other type of standard port. The port may be serial rather than parallel.

Referring to Figure 3, in order to check the host identity, the LM sends a request message to the SID at regular intervals, requesting it to supply the secure host identifier.

The SID responds to this by returning a message encrypted under a key known only to the SID and the LM.

The message contains:

- the secure host identifier
- a sequence number, which is incremented each time the SID returns a message.

When the LM receives this message, it decrypts it, and checks the sequence number to ensure that it is the next expected sequential value. This ensures that it is not possible to replace the SID by a program which intercepts the requests from the LM and returns a copy of the SID's response, or which passes the request to a SID on another system.

The LM then checks whether the returned secure host identifier matches the machine identifiers of the license keys held in the license table 22.

If the LM does not receive any response to a request to the SID, or if the response does not contain the correct sequence number, or if the secure host identifier does not match the machine identifiers in the license keys, the LM closes down. This means that the LM will not issue any more licenses to applications. Also, because the LM will not now respond to the revalidation message from the application, any outstanding licenses are effectively cancelled.

In summary, it can be seen that the LM will issue licenses, permitting applications to operate, only if a security identification device SID is connected to the computer, and if the machine identifiers in the individual license keys issued to the LM match the secure host identifier held in the SID.

It should be noted that the LM can grant licenses to applications running in any of the computers 10 in the network, not just to applications running in the same computer as the LM. The number of licenses that may be granted is restricted by the limit in the license keys. Thus, for example, if a license key sets a limit on the number of users, then the total number of users of a particular application in the network cannot exceed this limit.

The use of the device for the provision of the identifier to the license manager has several very important advantages:

- if the machine to which the device is attached fails, the device can be transferred to another machine (new keys are not required)
- the supplier of the device can retain title to the device, so in the event of the machine being sold the device has to be returned to the supplier. Hence all software on the machine that would only work with a license manager will no longer function as required by the terms of supply of the software which is licensed to a legal entity not to a machine.
- if the user of the software wishes to change the license he has to reduce its capability, the device can be replaced and new keys issued. Current schemes do not provide for the secure revocation of the keys.
- the device can be used to provide secure identification on standard hardware platforms which do not inherently provide such a facility, and hence can enable the use of license management on such hardware.

It should be noted that although the embodiment of the invention described above is a multi-computer system, the invention is equally applicable to single processor systems, or to multi-nodal systems, comprising a plurality of multi-processor nodes.

Claims

1. A computer system including a license manager for regulating usage of software items in accordance with license keys issued to the license manager, the license manager being arranged to check the host identity of the computer on which it runs and to permit usage only if the host identity matches an identity value in the license keys, characterised in that the host identity of the computer is supplied by a security identification device removably coupled to an external port on the computer.

2. A system according to Claim 1 wherein communication of the host identity between the security identifier device and the license manager is protected by encryption.

3. A system according to Claim 2 wherein each host identity returned by the security identifier device is encrypted together with a sequence number which is incremented each time the host identity is returned.

4. A system according to any preceding claim wherein the license manager regulates the usage of software items within a domain comprising software items that can communicate with the license manager.

5. A system according to Claim 4 wherein said domain is distributed over a network of computers.

Fig.1.

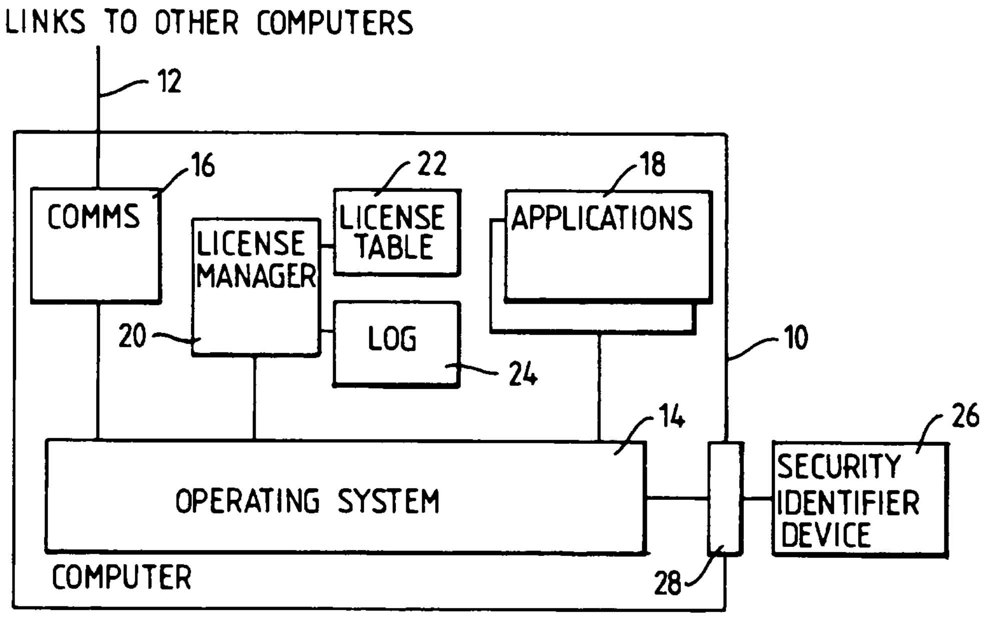

LINKS TO OTHER COMPUTERS
12
16
COMMS
22
LICENSE TABLE
18
APPLICATIONS
LICENSE MANAGER
20
LOG
24
10
14
26
OPERATING SYSTEM
SECURITY IDENTIFIER DEVICE
COMPUTER
28

Fig. 2.

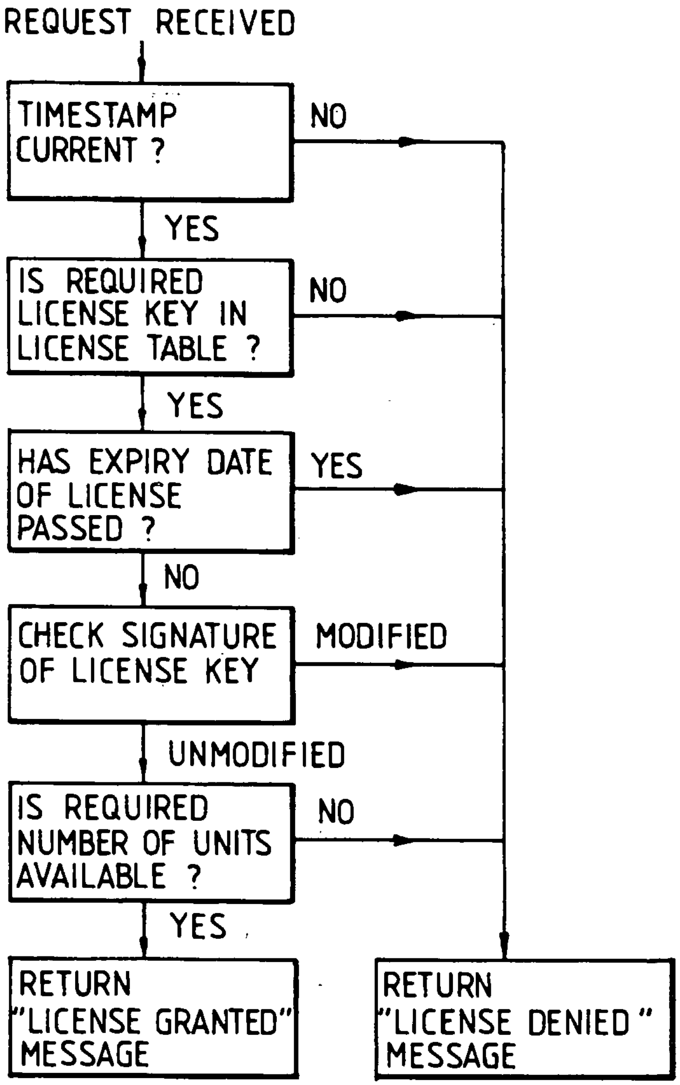
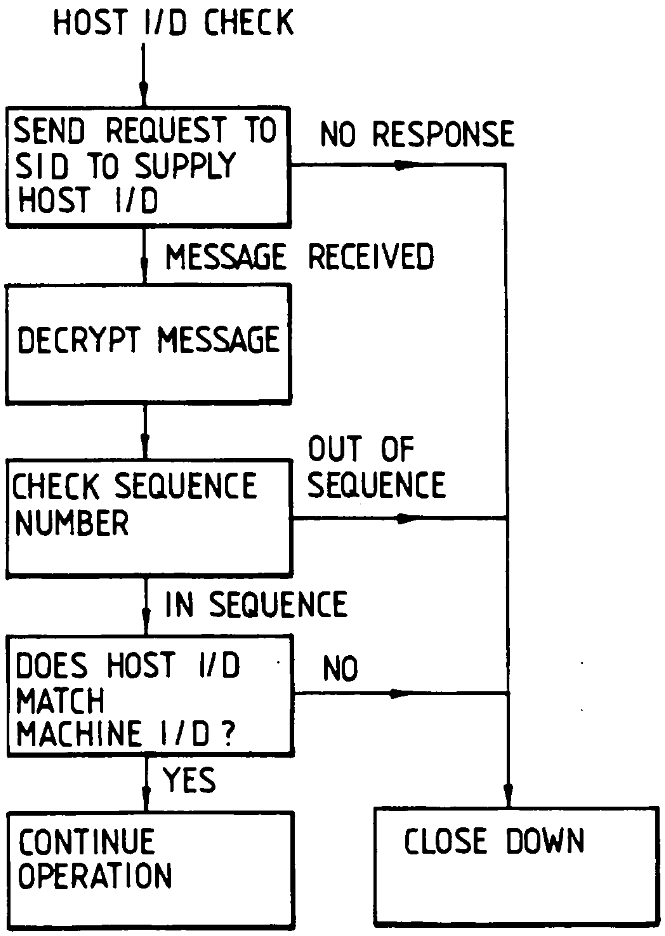

REQUEST RECEIVED
TIMESTAMP CURRENT ?
NO
YES
IS REQUIRED LICENSE KEY IN LICENSE TABLE ?
NO
YES
HAS EXPIRY DATE OF LICENSE PASSED ?
YES
NO
CHECK SIGNATURE OF LICENSE KEY
MODIFIED
UNMODIFIED
IS REQUIRED NUMBER OF UNITS AVAILABLE ?
NO
YES
RETURN "LICENSE GRANTED" MESSAGE
RETURN "LICENSE DENIED" MESSAGE

Fig.3.


HOST I/D CHECK
SEND REQUEST TO SID TO SUPPLY HOST I/D
NO RESPONSE
MESSAGE RECEIVED
DECRYPT MESSAGE
CHECK SEQUENCE NUMBER
OUT OF SEQUENCE
IN SEQUENCE
DOES HOST I/D MATCH MACHINE I/D ?
NO
YES
CONTINUE OPERATION
CLOSE DOWN

European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 93 30 6468

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.5)
Y	US-A-4 924 378 (HERSHEY ET AL) * abstract; figures 1,3,5,7 * * column 1, paragraph 2 * * column 2, line 1 - column 3, line 36 * * column 7, line 22 - column 8, line 12 * * column 10, line 27 - line 40 * * claims 1-5,11-23 * ---	1-5	G06F1/00
Y	PTR PHILIPS TELECOMMUNICATION AND DATA SYSTEMS REVIEW, vol. 47, no. 3 , September 1989 , HILVERSUM, NL; pages 1 - 19 R.C.FERREIRA 'The Smart Card: A High Security Tool in EDP' * summary; figures 4,5 * * page 5, line 6 - page 7, line 5 * * page 9, line 1 - page 11, line 40 * * page 12, line 36 - page 13, line 4 * ---	1-5	
A	EP-A-0 191 162 (IBM) * abstract; figures 4,9 * * column 6, line 8 - column 7, line 14 * * column 9, line 6 - line 39 * * column 10, line 5 - line 40 * * column 13, line 5 - line 36 * -----	1,3	**TECHNICAL FIELDS SEARCHED (Int.Cl.5)** G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	4 May 1994	Powell, D

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

⑲ Europäisches Patentamt
European Patent Office
Office européen des brevets

⑪ Publication number: **0 678 836 A1**

⑫ EUROPEAN PATENT APPLICATION

㉑ Application number: **94105573.3**

㉒ Date of filing: **11.04.94**

(51) Int. Cl.^{6}: **G07F 7/10**

(43) Date of publication of application:
25.10.95 Bulletin 95/43

(84) Designated Contracting States:
DE FR GB

(71) Applicant: **TANDEM COMPUTERS INCORPORATED**
10435 North Tantau Avenue,
Loc. 200-16
Cupertino,
California 95014-0709 (US)

(72) Inventor: **Atalla, Martin M.**
18 Monte Vista
Atherton
CA 94025 (US)
Inventor: **Hopkins, W. Dale**
2425 Rio Drive
Gilroy
CA 95020 (US)

(74) Representative: **KUHNEN, WACKER & PARTNER**
Alois-Steinecker-Strasse 22
D-85354 Freising (DE)

(54) Method and means for combining and managing personal verification and message authentication encryptions for network transmission.

(57) The method and means of transmitting a user's transaction message to a destination node in a computer-secured network operates on the message, and a sequence number that is unique to the transaction message to form a message authentication code in combination with the user's personal identification number. The message authentication code is encrypted with a generated random number and a single session encryption key which also encrypts the user's personal identification number. An intermediate node may receive the encryptions to reproduce the personal identification number that is then used to encrypt the received message and sequence number to produce the random number and a message authentication code for comparison with a decrypted message authentication code. Upon favorable comparison, the random number and the message authentication code are encrypted with a second session encryption key to produce an output code that is transmitted to the destination node along with an encrypted personal identification number. There, the received encryptions are decrypted using the second session key to provide the personal identification number for use in encrypting the message and sequence number to produce a message authentication code for comparison with a decrypted message authentication code. Upon favorable comparison, the transaction is completed and a selected portion of the decrypted random number is returned to the originating node for comparison with the corresponding portion of the random number that was generated there. Upon unfavorable comparison at the destination node or at an intermediate node, a different portion of the decrypted random number is returned to the originating node for comparison with the corresponding portion of the random number that was generated there. The comparisons at the originating node provide an unambiguous indication of the completion or non-completion of the transaction at the destination node.

Rank Xerox (UK) Business Services
(3.10/3.09/3.3.4)

Related Cases

The subject matter of this application is related to the subject matter disclosed in U.S. Patents 4,268,715; 4,281,215; 4,283,599; 4,288,659; 4,315,101; 4,357,529; 4,536,647 and pending application for U.S. Patent Serial No. 547,207, entitled POCKET TERMING, METHOD AND SYSTEM FOR SECURED BANKING TRANSACTIONS, filed October 31, 1983 by M.M. Atalla.

Background of the Invention

Conventional data encryption networks commonly encrypt a Personal Identification Number with a particular encryption key for transmission along with data messages, sequence numbers, and the like, from one location node in the data network to the next location or node in the network. There, the encrypted PIN is decrypted using the encryption key, and re-encrypted with another encryption key for transmission to the next node in the network, and so on to the final node destination in the network.

In addition, such conventional data encryption networks also develop a Message Authentication Codes in various ways, and then encrypt such MAC for transmission to the next node using a MAC-encryption key that is different from the encryption key used to encrypt the PIN. At such next node, the MAC is decrypted using the MAC encryption key and then re-encrypted using a new MAC-encryption key for transmission to the next node, and so on to the final destination node in the network.

Further, such conventional networks operate upon the PIN, MAC, data message, sequence number, and the like; received and decrypted at the final destination node to consummate a transaction, or not, and then communicate an ACKnowledgment or Non-ACKnowledgment message back to the originating node of the network. Such ACK or NACK codes may be encrypted and decrypted in the course of transmission node by node through the network back to the originating node to provide an indication there of the status of the intended transaction at the final destination node.

Conventional data encryption networks of this type are impeded from handling greater volumes of messages from end to end by the requirement for separately encrypting and decrypting the PIN and MAC codes at each node using different encryption/decryption keys for each, and by the requirement for encrypting/decrypting at least the ACK code at each node along the return path in the network.

In addition, such conventional data encryption networks are susceptible to unauthorized intrusion and compromise of the security and message authenticity from node to node because of the separated PIN and MAC encryption/decryption techniques involved. For example, the encrypted PIN is vulnerable to being "stripped" away from the associated MAC, message, sequence number, and the like, and to being appended to a different MAC, message, sequence number, and the like, for faithful transmission over the network. Further, the return acknowledgment code may be intercepted and readily converted to a non-acknowledgment code or simply be altered in transmission after the transaction was completed at the destination node. Such a return code condition could, for example, cause the user to suffer the debiting of his account and, at the same time, the denial of completion of a credit purchase at point-of-sale terminal or other originating node.

Summary of the Invention

Accordingly, the method and means for integrating the encryption keys associated with the PIN and MAC codes according to the present invention assure that these codes are sufficiently interrelated and that alteration of one such code will adversely affect the other such code and inhibit message authentication in the network. In addition, the return acknowledgment or non-acknowledgment code may be securely returned from node to node in the network without the need for encryption and decryption at each node, and will still be securely available for proper validation as received at the originating node. This is accomplished according to the present invention by using one session key to encrypt the PIN along with the MAC, a random number, the message, and the sequence number which are also encrypted with the PIN such that re-encryption thereof in the transmission from location to location, or node to node over a network is greatly facilitated and validatable at each node, if desired. In addition, portions of the random number are selected for use as the Acknowledgment or Non-Acknowledgment return codes which can be securely returned and which can then only be used once to unambiguously validate the returned code only at the originating node in the network.

Description of the Drawings

Figure 1 is graphic representation of a typical conventional encryption scheme which operates with two independent session keys;

Figure 2 is a schematic representation of a second network according to the present inventions; and

Figure 3 is a graphic representation of the signal processing involved in the operation of the network of Figure 2.

Description of the Preferred Embodiment

Referring now to Figure 1, there is shown a graphic representation of the encoding scheme commonly used to produce the PIN and MAC codes using two session keys for transmission separately to the next network node. As illustrated, one session key 5 may be used to encrypt the PIN entered 7 by a user (plus a block of filler bits such as the account number, as desired) in a conventional encryption module 9 which may operate according to the Data Encryption Standard (DES) established by the American National Standards Institute (ANSI) to produce the encrypted PIN signal 11 (commonly referred to as the PIN block" according to ANSI standard 9.3) for transmission to the next network node. In addition, the message or transaction data which is entered 13 by the user and which is to be transmitted to another node, is combined with a sequence number 15 that may comprise the date, time, station code, and the like, for encryption by a DES encryption module 17 with another session key 19 to produce a Message Authentication Code (MAC) 21 for that message and sequence number. The MAC may comprise only a selected number of significant bits of the encrypted code. The message and MAC are separately transmitted to the next node along with the encrypted PIN, and these codes are separately decrypted with the respective session keys and then re-encrypted with new separate session keys for transmission to the next network node, and so on, to the destination node. Conventional PIN validation at the destination node, and message authentication procedures may be performed on the received, encrypted PIN and MAC, (not illustrated) and the message is then acted upon to complete a transaction if the PIN is valid and the MAC is unaltered. A return ACKnowledgment (or Non-ACKnowledgment) code may be encrypted and returned to the next node in the network over the return path to the originating node. At each node in the return path, the ACK code is commonly decrypted and re-encrypted for transmission to the next node in the return path, and so on (not illustrated), to the originating node where receipt of the ACK is an indication that the transaction was completed at the destination node. Conventional systems with operating characteristics similar to those described above are more fully described, for example, in U.S. Patent 4,283,599.

One disadvantage associated with such conventional systems is the need to encrypt and decrypt at each node using two separate session keys. Another disadvantage is that such conventional systems are vulnerable to unauthorized manipulation at a network node by which the message and MAC may be "stripped away" from the encrypted PIN associated with such message and replaced with a new message and MAC for transmission with the same encrypted PIN to the next network node. Further, the acknowledgement code that is to be returned to the originating node not only must be decrypted and re-encrypted at each node along the return path, but the return of an acknowledgment code that is altered along the return path may connote non-acknowledgment or non-completion of the intended transaction at the destination node. This condition can result in the account of the user being debited (the PIN and MAC were valid and authentic as received at the destination node), but the user being denied completion of a credit transaction (e.g., transfer of goods) at the originating node.

Referring now to Figures 2 and 3, there are shown schematic and graphic representations, respectively, of network operations according to the present invention. Specifically, there is shown a system for transmitting a message over a network 29 from an originating node 31 to a destination node 33 via an intermediate node 35. At the originating node 31, an authorized user enters his PIN 37 of arbitrary bit length with the aid of a key board, or card reader, or the like, and the entered PIN is then filled or blocked 39 with additional data bits (such as the user's account number in accordance with ANSI standard 9.3) to configure a PIN of standard bit length.

In addition, the transaction data or message 41 entered through a keyboard, or the like, by the user is combined with a sequence number 43 which is generated to include date, time of day, and the like. The combined message and sequence number is encrypted 45 with the PIN (or blocked PIN) in a conventional DES module to produce a multi-bit encrypted output having selected fields of bits, one field of which 51 serves as the Message Authentication Code (MAC). Other schemes may also be used to produce a MAC, provided the PIN (or blocked PIN) is used as the encryption key, and the resulting MAC, typically of 64-bit length, may be segregated into several sectors or fields 51. A random number (R/N) is generated 52 by conventional means and is segregated into several sectors or fields 54, 56, 58. The first sector or field 54 of, say 32-bits length, is then encrypted with the selected MAC field 53 in a conventional DES encryption module 55 (or in DES module 45 in time share operation) using the session key K_1 as the encryption key 50. In addition, the PIN (or blocked PIN) 39 is encrypted in DES encryption module 60 (or in DES module 45 in time share operation) using the session key K1 as the encryption Key 50. The session key 50 may be transmitted to successive nodes 35, 33 in secured manner, for example, as disclosed in U.S. Patent 4,288,659. The resulting encrypted output codes 62, 64 are then transmitted along with sequence number 43 and the message 41 (in clear or cypher text) over the network 29 to the next node 35 in the path toward the destination node 33. Thus, only a single session key K_1 is used to encrypt the requisite data for transmission over the network, and the residual sectors or fields 56, 58 of the random number from generator 52 remain available to verify successful completion of the transaction at the destination node 33, as later described herein.

At the intermediate node 35, the encrypted PIN 64 received from the originating node 31 is decrypted in conventional DES module 70 using the session key K_1 to produce the blocked PIN 63. In addition, the encrypted MAC and R/N 68 received from the originating node is decrypted in conventional DES module 61 (or in DES module 70 operating in timeshare relationship) using session key K_1 to produce the MAC and the R/N in segregated fields. An initial validation may be performed by encrypting the received message 41 and sequence number 43 in conventional DES module 67 using the decrypted PIN 63 as the encryption key. Of course, the original PIN as entered by the user may be extracted from the decrypted, blocked PIN 63 to use as the encryption key in module 67 if the corresponding scheme was used in node 31. (It should be understood that the PIN or blocked PIN does not appear in clear text outside of such decryption or encryption modules 70, 67 (or 69, later described herein), and that these modules may be the same DES module operated in time-shared relationship.)

The encrypted output of module 67 includes several sectors, or fields, similar to those previously described in connection with the encrypted output of module 45. The selected sector 53 of significant bits that constitutes the MAC is selected for comparison with the MAC 65 that is decrypted in DES module 61. This decryption also provides the R/N having several selected sectors or fields 72. If the comparison of the decrypted and encrypted MAC's in comparator 74 is favorable, gate 76 is enabled and the decrypted MAC and R/N are encrypted in conventional DES module 69 using new session key K_2 as the encryption key, and gate 88 is enabled to encrypt the decrypted PIN in DES module 78 (or in DES module 67 or 69 in time share operating). If comparison is unfavorable, the transaction may be aborted and the gate 80 is enabled to transmit back to the originating node 31 the sector or field 58 of the R/N which constitutes the Non ACKnowledge sector of the decrypted R/N output of module 61. The encrypted PIN output 82 of module 78 and the encrypted MAC and R/N output 84 of the module 69 are thus transmitted along with the message 41 and sequence number 43 over the network 29 to the destination node 35 upon favorable comparison 74 of the encrypted and decrypted MACs.

At the destination node 33, the encrypted PIN output 86 received from the intermediate node 35 is decrypted in conventional DES module 71 using the session key K_2 to produce the PIN 73. An initial validation may be performed by encrypting the received message 41 and sequence number 43 in conventional DES module 77, using the decrypted PIN 73 as the encryption key. As was described in connection with the intermediate node 35, the original PIN as entered by the user may be extracted from the decrypted, blocked PIN 73 to use as the encryption Key in module 77 if the corresponding scheme was used in node 31. And, it should be understood that the PIN or blocked PIN does not appear in clear text outside of the decryption or encryption modules 71, 77, which modules may be the same DES module operated in time-shared relationship. In addition, the encrypted MAC and R/N received at the destination node 33 is decrypted in DES module 92 using the session key K_2 to produce the MAC 75 and the R/N 94 in segregated sectors or fields. The selected sector 53 of significant bits that constitutes the MAC in the encrypted output of module 77 is compared 79 for parity with the decrypted MAC 75. If comparison is favorable, the transaction may be completed in response to the message 41, and gate 81 may be enabled to transmit 29 back to the intermediate node 35 a second selected sector or field 56 which constitutes the ACKnowledge output sector of the R/N decrypted output from module 92. If comparison 79 is unfavorable, the transaction is not completed and gate 83 is enabled to transmit 29 back to the intermediate node 35 a third selected sector or field 58 which constitutes the Non-ACKnowledge sector of the R/N decrypted output from module 92.

In accordance with one aspect of the present invention, the returned ACK or NACK codes do not require decryption and re-encryption when transmitted from node to node along the return path in the network back to the originating node 31. Instead, these codes are already in encoded form and may be transmitted directly from node to node without encumbering a node with additional operational overhead. These codes are therefore secured in transmission over the network and are only cypherable in the originating node 31 which contains the ACK and NACK fields or sectors 56 and 58 of the random number from generator 52. At the originating node 31, the second and third sectors or fields 56 and 58 of the random number are compared 98 with the corresponding sectors of decrypted R/N outputs received from the destination node 33 (or the sector 58 of the decrypted R/N output received from intermediate node 35) to provide an indication at the originating node that the transaction was either completed 89 or aborted 91. Of course, the ACK and NACK may be encrypted as a network option when returned to the originating node 31. And, it should be understood that the encryption and decryption modules at each node may be the same conventional DES module operated in timeshare relationship.

Therefore, the system and method of combining the management of PIN and MAC codes and the session keys associated therewith from node to node along a data communication network obviates the conventional need for separate session keys for the PIN and the MAC, and also obviates the need for conventional encryption/decryption schemes for an acknowledgment code at each node along the return path back to the originating node. If desired, PIN validations may be performed at each node since the PIN is available within the DES module circuitry. In addition, the present system and method also reduces the vulnerability of a secured transmission system to unauthorized separation of a valid PIN code from its associated message and MAC code for unauthorized attachment to a different message and MAC code. Further, the method and means of the present invention reduces the ambiguity associated with the return or not of only an acknowledgment code in conventional systems by returning either one of the ACK and NACK codes without additional operational overhead at each node.

Claims

1. The method of securing transaction data between two locations in response to a user's message ad personal identification number, the method comprising:

   forming a sequence number representative of the user's transaction;

   encoding in a first logical combination at the first location the user's message and the sequence number in accordance with the personal identification number received from the user to produce a message authentication code having a plural number of digit sectors;

   generating a random number;

   establishing a first encoding key;

   encoding in a second logical combination at the first location the random number and a selected number of sectors of the message authentication code in accordance with the first encryption key to produce a first coded output;

   encoding in a third logical combination at the first location the user's personal identification number in accordance with the first encoding key to produce a second coded output;

   transmitting to another location the user's message and the sequence number and the first ad second coded outputs;

   establishing the first encoding key at such other location;

   decoding the first coded output received at such other location with the first encoding key according to said second logical combination thereof to provide the random number and message authentication code;

   decoding the second coded output received at such other location with the first encoding key according to said third logical combination to provide the user's personal identification number;

   encoding in the first logical combination at such other location the user's message and sequence number received thereat in accordance with the decoded personal identification number to produce a message authentication code having a plural number of digit sectors; and

   comparing selected corresponding digit sectors of the decoded message authentication code and the encoded message authentication code to provide an indication upon favorable comparison of the valid transmission of the user's message between the two locations.

2. The method according to claim 1 comprising the steps of:

   establishing a second encoding key at the other location;

   encoding in a fourth logical combination at such other location the decoded random number and selected sector of the message authentication code in accordance with the second encoding key to produce a third coded output;

   encoding in a fifth logical combination at the other location the decoded user's personal identification number in accordance with the second encoding key to produce a fourth coded output;

   transmitting to a remote location the user's message and the sequence number and the third and fourth coded outputs;

   establishing the second encoding key at the remote location;

   decoding the third coded output as received at the remote location according to the fourth logical combination in accordance with the second encoding key to provide the random number and the message authentication code having a plural number of digit sectors;

   decoding the fourth coded output received at the remote location according to the fifth logical combination to provide the user's personal identification number;

encoding the message and the sequence number received at the remote location according to the first logical combination in accordance with the decoded personal identification number to produce a message authentication code having a plural number of digit sectors; and

comparing corresponding digit sectors of the decoded message authentication code and the encoded message authentication code at the remote location to provide an indication upon favorable comparison of the unaltered transmission of the message, or an indication upon unfavorable comparison of an alteration in the transmission of the message.

3. The method according to claim 1 comprising the steps of:

transmitting a selected sector of the decoded random number from the other location to the one location in response to unfavorable comparison; and

comparing the selected sector of the random number received at the one location from the other location with the corresponding selected sector at the one location to provide an indication of the altered transmission of the message to the other location.

4. The method according to claim 2 comprising the steps of:

completing the transaction and returning a second selected sector of the decoded random number from the remote location to the one location in response to said favorable comparison, and inhibiting completion of the transaction and returning a third selected sector of the decoded random number from the remote location to the one location in response to said unfavorable comparison; and

comparing the selected sector of the random number received at the one location from the remote location with the corresponding selected sector of the number generated at the one location to provide an indication of the completion or non-completion of the transaction at the remote location.

5. Apparatus for securing transaction data between two locations in response to a user's message and personal identification number, the apparatus comprising:

means for generating a sequence number associated with a user's transaction;

means for generating a random number;

first encryption means at one location for encrypting according to a first logical combination of the user's message and the sequence number applied thereto with the personal identification number received from the user for producing a message authentication code therefrom having a plural number of digit sectors;

means at said one location for producing a first session key;

second encryption means coupled to receive the random number from the user and a selected sector of the message identification code for encrypting the same with the first session key according to a second logical combination thereof to produce a first encoded output;

third encryption means coupled to receive the personal identification number from the user for encrypting the same with the first session key according to a third logical combination thereof to produce a second encoded output;

means for transmitting the first and second encoded outputs and message and sequence number from the one location to the next location;

means at the next location for producing the first session key;

first decryption means at the next location coupled to receive the transmitted first encoded output and the first session key for decrypting in accordance with said second logical combination to provide the random number and the message authentication code;

second decryption means at the next location coupled to receive the transmitted second encoded output and the first session key for decrypting in accordance with the third logical combination thereof to produce the user's personal identification number;

third encrpytion means at the next location coupled to receive the transmitted message and sequence number for encoding the same according to said first logical combination with the decrypted personal identification number to produce a message authentication code having a plural number of digit sectors;

comparison means at the next location coupled to receive the corresponding selected sectors of the decrypted message authentication code and of the encrypted message authentication code for producing an output indication of the parity thereof; and

means at the next location responsive to said output indication for operating upon the received message in response to favorable comparison.

6. Apparatus as in claim 5 comprising:
   means at the next location responsive to the unfavorable comparison for transmitting to the one location a selected sector of the random number.

7. Apparatus as in claim 5 comprising:
   means at the next location for producing a second encoding key;
   first encryption means at the next location coupled to receive the decrypted message authentication code and random number for encoding the same with the second encoding key in accordance with a fourth logical combination in response to said favorable comparison for producing a third output code for transmission to a destination location;
   second encryption means at the next location coupled to receive the decrypted personal identification number for encoding the same with the second encoding key in accordance with a fifth logical combination in response to said favorable comparison for producing a fourth output code for transmission to a destination location;
   means at the destination location for producing the second encoding key;
   first decryption means at the destination location for receiving the third output code transmitted from said next location and the second encoding key for decoding the same according to said fourth logical combination to provide the random number and the message authentication code;
   second decryption means at the destination location for receiving the fourth output code transmitted from said next location and the second encoding key for decoding the same according to said fifth logical combination to provide the personal identificaton number;
   encryption means at the destination location for receiving the message and the sequence number for encoding the same with the decrypted personal identification number in accordance with the first logical combination to produce a message authentication code having a plural number of digit sectors;
   means at the destination location for comparing corresponding selected sectors of the encrypted message authentication code and the decrypted message authentication code to produce output indications of favorable and unfavorable comparisons;
   means at the destination location responsive to favorable output indication for operating upon the transmitted message and for transmitting a selected sector of the random number to said one location, and responsive to unfavorable comparison for transmitting another selected sector of the random number to said one location; and
   comparator means at the one location coupled to receive the corresponding selected sectors of the random number for providing an output indication of the status of operation upon the message at the destination location.

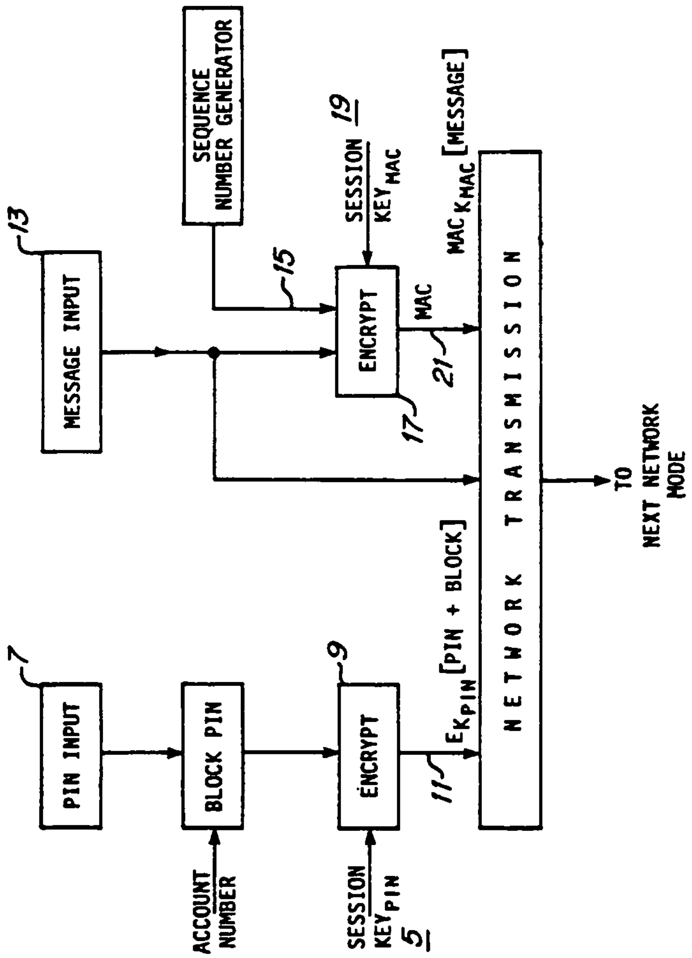

PIN INPUT
7
BLOCK PIN
ACCOUNT NUMBER
ENCRYPT
9
SESSION KEY PIN
5
11
E KPIN [PIN + BLOCK]
N E T W O R K T R A N S M I S S I O N
MESSAGE INPUT
13
SEQUENCE NUMBER GENERATOR
15
ENCRYPT
17
MAC
21
SESSION KEY MAC
19
MAC K MAC [MESSAGE]
TO NEXT NETWORK MODE

Figure 1
(PRIOR ART)

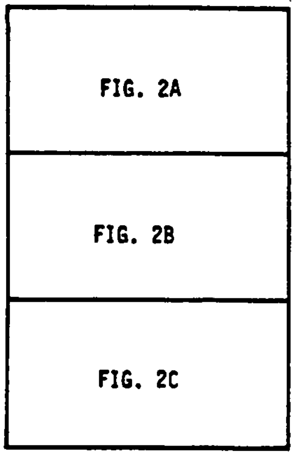


FIG. 2A
FIG. 2B
FIG. 2C

Figure 2

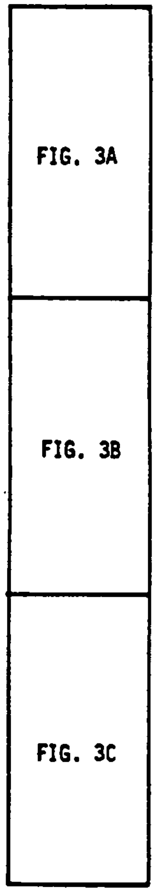


FIG. 3A
FIG. 3B
FIG. 3C

Figure 3

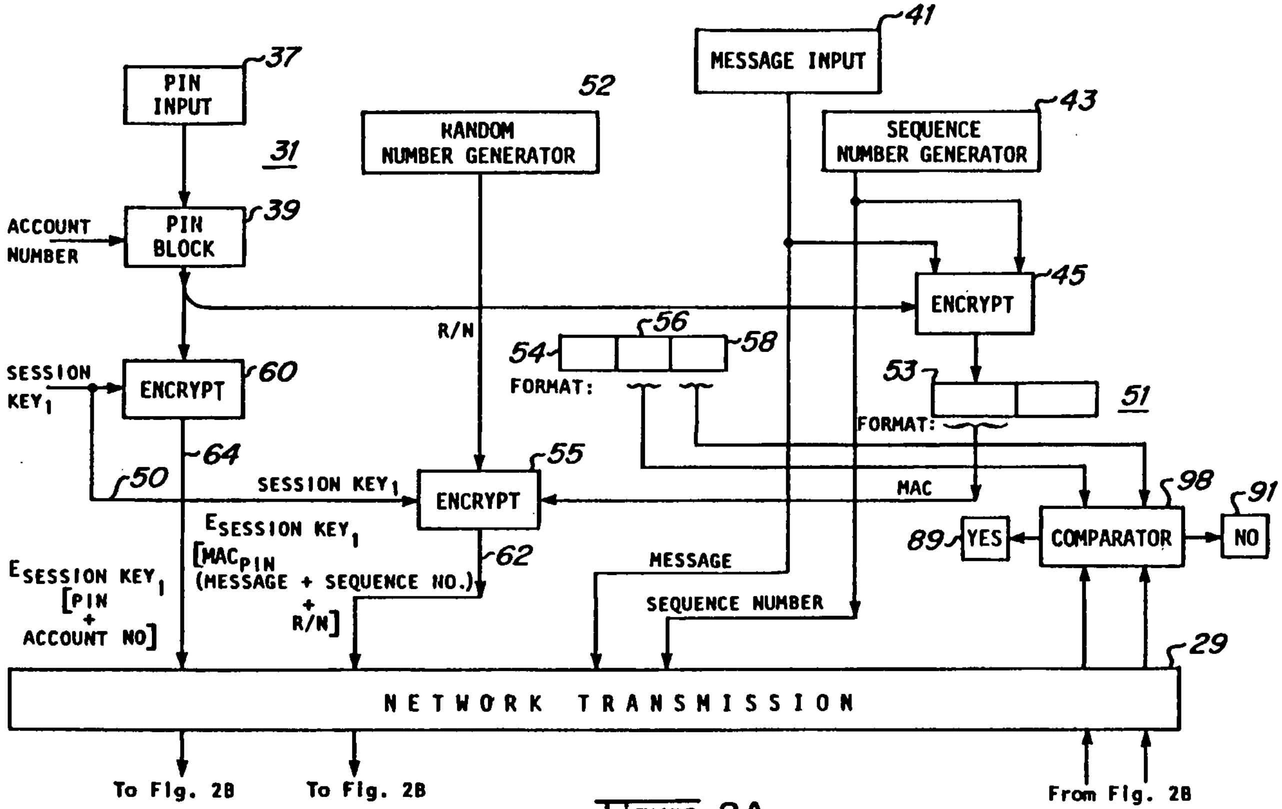

PIN INPUT
37
31
RANDOM NUMBER GENERATOR
52
MESSAGE INPUT
41
SEQUENCE NUMBER GENERATOR
43
ACCOUNT NUMBER
PIN BLOCK
39
ENCRYPT
45
R/N
54
56
58
FORMAT:
53
51
FORMAT:
SESSION KEY$_1$
ENCRYPT
60
64
50
SESSION KEY$_1$
ENCRYPT
55
MAC
98
91
89
YES
COMPARATOR
NO
$E_{SESSION\ KEY_1}$ [PIN + ACCOUNT NO]
$E_{SESSION\ KEY_1}$ [MAC_{PIN} (MESSAGE + SEQUENCE NO.) + R/N]
62
MESSAGE
SEQUENCE NUMBER
29
NETWORK TRANSMISSION
To Fig. 2B
To Fig. 2B
From Fig. 2B

Figure 2A

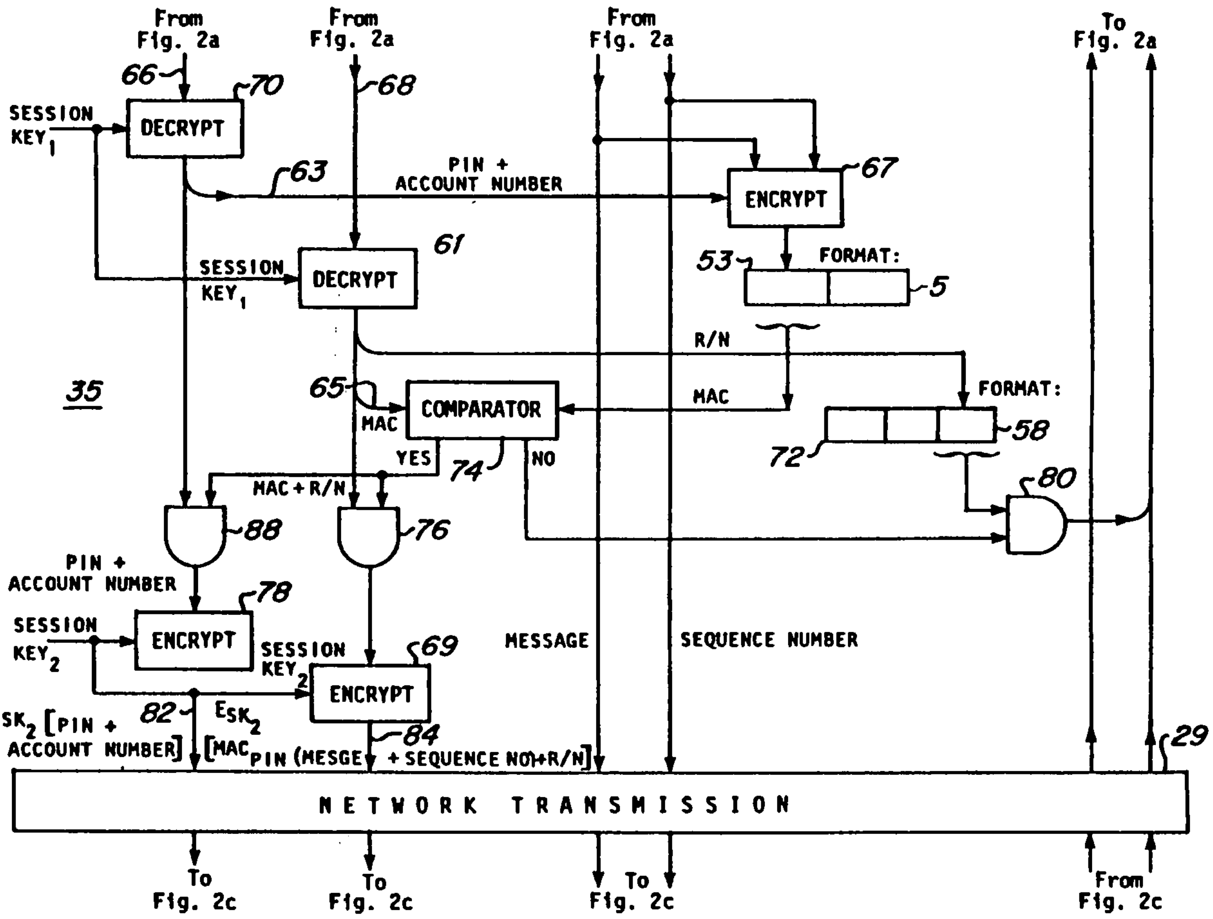

From Fig. 2a
From Fig. 2a
From Fig. 2a
To Fig. 2a
66
70
68
SESSION KEY1
DECRYPT
63
PIN + ACCOUNT NUMBER
ENCRYPT
67
61
SESSION KEY1
DECRYPT
53
FORMAT:
5
R/N
35
65
MAC
COMPARATOR
MAC
FORMAT:
58
72
YES
74
NO
80
MAC + R/N
88
76
PIN + ACCOUNT NUMBER
78
SESSION KEY2
ENCRYPT
SESSION KEY2
69
ENCRYPT
MESSAGE
SEQUENCE NUMBER
ESK2 [PIN + ACCOUNT NUMBER]
82
ESK2 [MACPIN (MESGE + SEQUENCE NO + R/N]
84
29
NETWORK TRANSMISSION
To Fig. 2c
To Fig. 2c
To Fig. 2c
From Fig. 2c

Figure 2B

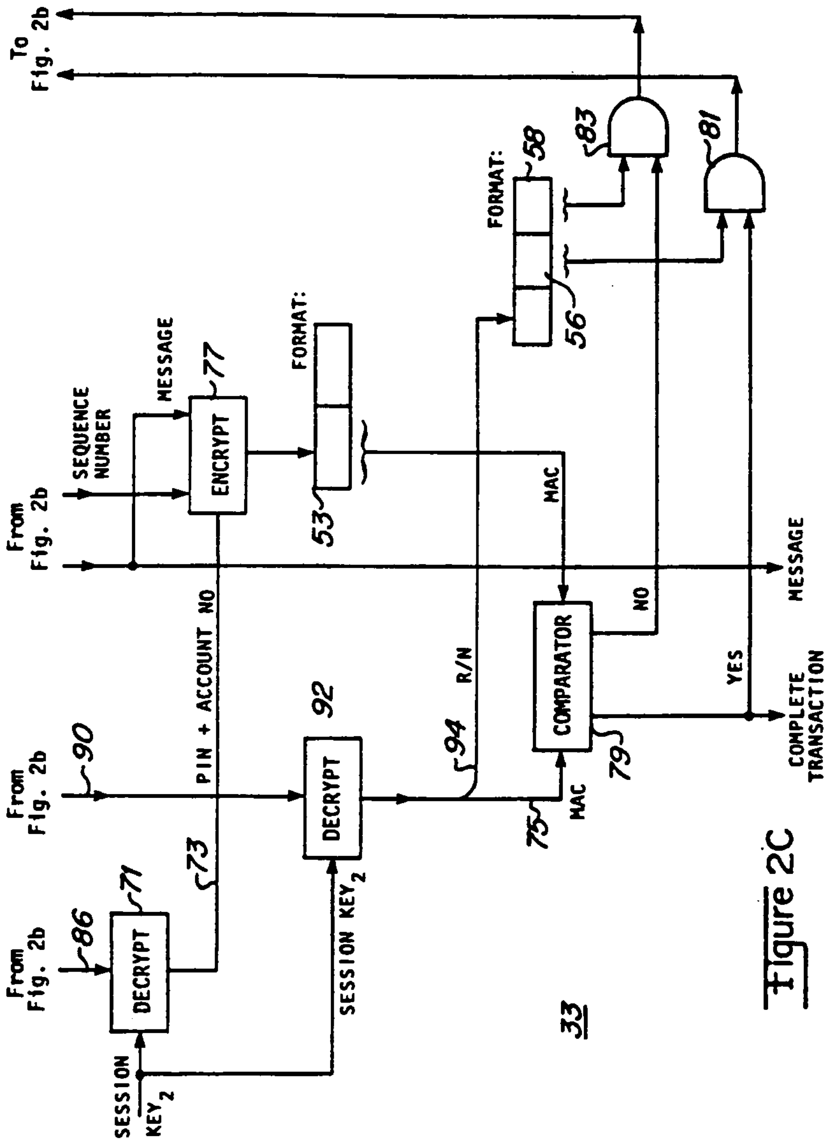

From Fig. 2b
86
DECRYPT
71
SESSION KEY$_2$
From Fig. 2b
90
PIN + ACCOUNT NO
73
DECRYPT
92
SESSION KEY$_2$
From Fig. 2b
SEQUENCE NUMBER
MESSAGE
ENCRYPT
77
FORMAT:
53
MAC
R/N
94
75
MAC
COMPARATOR
79
NO
YES
COMPLETE TRANSACTION
MESSAGE
FORMAT:
58
56
83
81
To Fig. 2b
33

Figure 2C

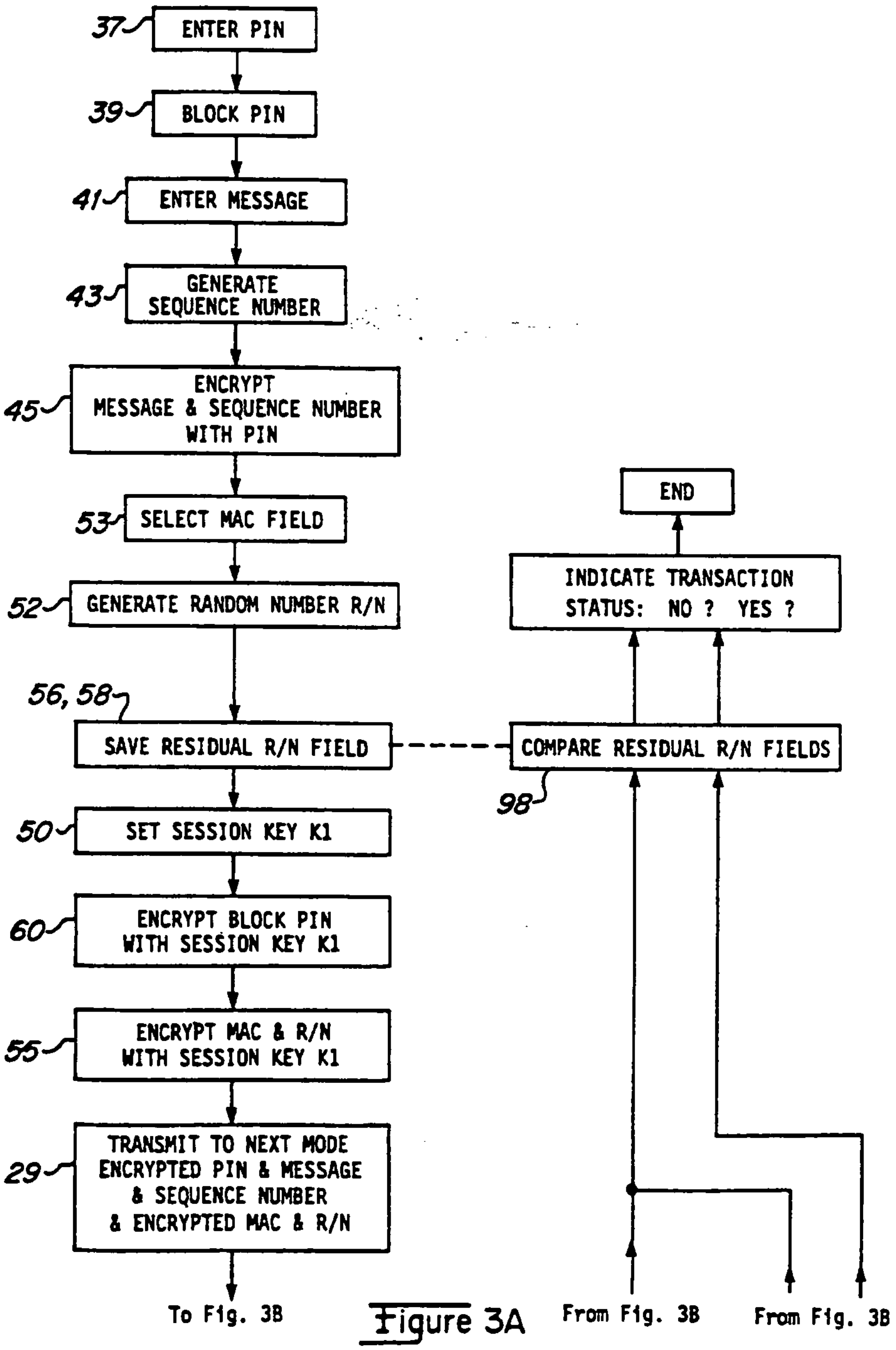

37
ENTER PIN
39
BLOCK PIN
41
ENTER MESSAGE
43
GENERATE SEQUENCE NUMBER
45
ENCRYPT MESSAGE & SEQUENCE NUMBER WITH PIN
53
SELECT MAC FIELD
52
GENERATE RANDOM NUMBER R/N
56, 58
SAVE RESIDUAL R/N FIELD
50
SET SESSION KEY K1
60
ENCRYPT BLOCK PIN WITH SESSION KEY K1
55
ENCRYPT MAC & R/N WITH SESSION KEY K1
29
TRANSMIT TO NEXT MODE ENCRYPTED PIN & MESSAGE & SEQUENCE NUMBER & ENCRYPTED MAC & R/N
To Fig. 3B
END
INDICATE TRANSACTION STATUS: NO ? YES ?
COMPARE RESIDUAL R/N FIELDS
98
From Fig. 3B
From Fig. 3B

Figure 3A

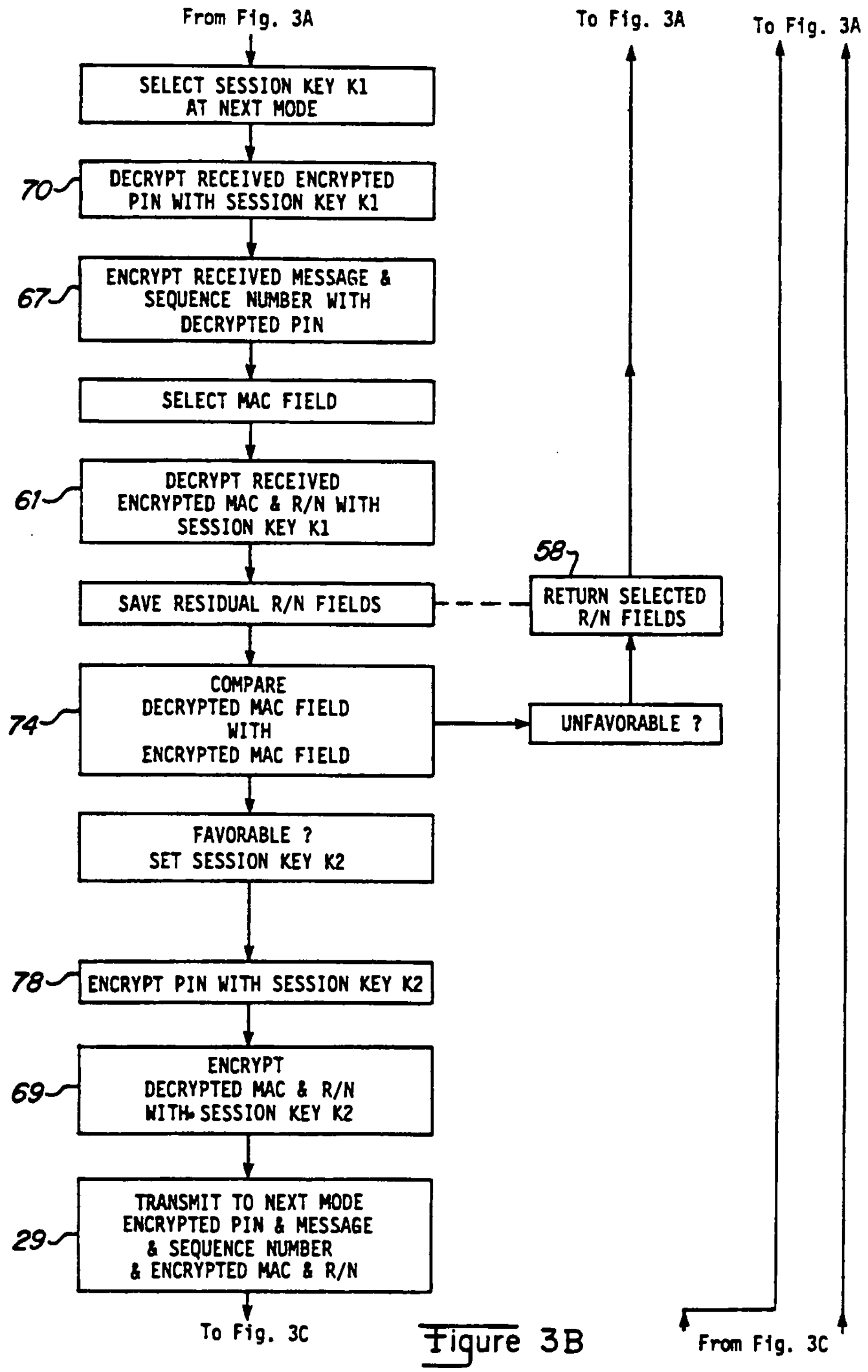

From Fig. 3A
To Fig. 3A
To Fig. 3A
SELECT SESSION KEY K1 AT NEXT MODE
70
DECRYPT RECEIVED ENCRYPTED PIN WITH SESSION KEY K1
67
ENCRYPT RECEIVED MESSAGE & SEQUENCE NUMBER WITH DECRYPTED PIN
SELECT MAC FIELD
61
DECRYPT RECEIVED ENCRYPTED MAC & R/N WITH SESSION KEY K1
58
SAVE RESIDUAL R/N FIELDS
RETURN SELECTED R/N FIELDS
74
COMPARE DECRYPTED MAC FIELD WITH ENCRYPTED MAC FIELD
UNFAVORABLE ?
FAVORABLE ? SET SESSION KEY K2
78
ENCRYPT PIN WITH SESSION KEY K2
69
ENCRYPT DECRYPTED MAC & R/N WITH SESSION KEY K2
29
TRANSMIT TO NEXT MODE ENCRYPTED PIN & MESSAGE & SEQUENCE NUMBER & ENCRYPTED MAC & R/N
To Fig. 3C
From Fig. 3C

Figure 3B

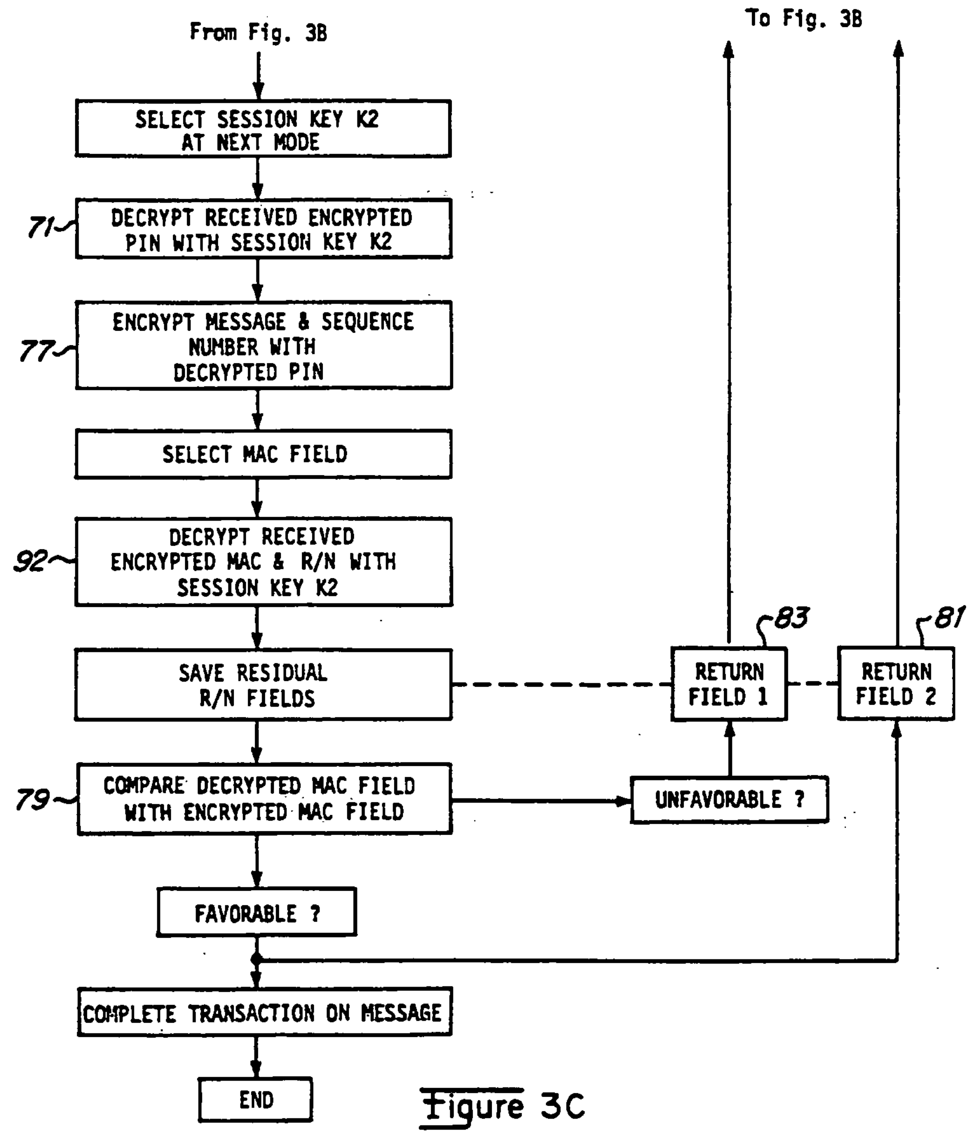

From Fig. 3B
To Fig. 3B
SELECT SESSION KEY K2 AT NEXT MODE
71
DECRYPT RECEIVED ENCRYPTED PIN WITH SESSION KEY K2
77
ENCRYPT MESSAGE & SEQUENCE NUMBER WITH DECRYPTED PIN
SELECT MAC FIELD
92
DECRYPT RECEIVED ENCRYPTED MAC & R/N WITH SESSION KEY K2
SAVE RESIDUAL R/N FIELDS
83
RETURN FIELD 1
81
RETURN FIELD 2
79
COMPARE DECRYPTED MAC FIELD WITH ENCRYPTED MAC FIELD
UNFAVORABLE ?
FAVORABLE ?
COMPLETE TRANSACTION ON MESSAGE
END

Figure 3C

European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 94 10 5573

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	EP-A-0 391 261 (NIPPON TELEGRAPH)	1	G07F7/10
A	* abstract *	2,5	
	* page 2, line 19 - line 31 *		
	* page 4, line 31 - page 5, line 12 *		
	* page 6, line 21 - line 25 *		
	* page 7, line 2 - line 11 *		
	* page 9, line 33 - line 54 *		
	* page 16, line 41 - page 17, line 32 *		
	* claim 1; figures 2A,2B *		

X	US-A-5 101 373 (KATSUAKI)	1	
A	* column 5, line 32 - line 59 *	2,3,5	
	* claims 1,4,5 *		

A	US-A-5 016 277 (HAMILTON)	1,5	
	* column 16, line 60 - column 17, line 7 *		

A	EP-A-0 547 975 (BULL CP8)	1,5	
	* abstract *		

A	EP-A-0 500 245 (TOSHIBA)	1,5	
	* abstract *		
	* claim 1 *		

A	EP-A-0 494 796 (NCR CORPORATION)	1,5	
	* abstract *		

TECHNICAL FIELDS SEARCHED (Int.Cl.6)

G07F
H04L

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	14 September 1994	Taccoen, J-F

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

⑲ Europäisches Patentamt
European Patent Office
Office européen des brevets

⑪ Publication number: **0 679 977 A1**

⑫ EUROPEAN PATENT APPLICATION

㉑ Application number: **95105400.6**

㉒ Date of filing: **10.04.95**

⑤① Int. Cl.⁶: **G06F 1/00**, G06F 12/14

③⓪ Priority: **25.04.94 US 238418**

④③ Date of publication of application:
02.11.95 Bulletin 95/44

⑧④ Designated Contracting States:
DE FR GB

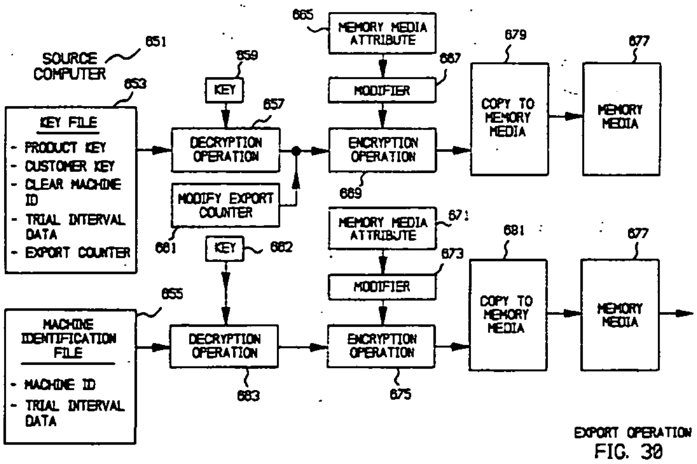
⑦① Applicant: **INTERNATIONAL BUSINESS MACHINES CORPORATION**
Old Orchard Road
Armonk, N.Y. 10504 (US)

⑦② Inventor: **Cooper, Thomas Edward**
858 West Willow Street
Louisville,
Colorado 80027 (US)
Inventor: **Nagda, Jagdish**
701 Kalmia Avenue
Boulder,
Colorado 80304 (US)
Inventor: **Pryor, Robert Franklin**
7380 Mt. Meeker Road
Lognmont,
Colorado 80503 (US)

⑦④ Representative: **Schäfer, Wolfgang, Dipl.-Ing.**
IBM Deutschland
Informationssysteme GmbH
Patentwesen und Urheberrecht
D-70548 Stuttgart (DE)

⑤④ Method and apparatus enabling software trial allowing the distribution of software objects.

⑤⑦ A method and apparatus is provided for transferring encrypted files from a source computer to one or more target computers. An export program is provided in the source computer and an import program is provided in the target computer. The export program decrypts the encrypted file and tags the export operation with an export counter value. The clear text file is then encrypted with an encryption operation utilizing a key which is unique to a transfer memory media, such as diskette serial number. The memory media is carried to a target computer which utilizes the import file to decrypt the encrypted file.


SOURCE COMPUTER
651
653
KEY FILE
- PRODUCT KEY
- CUSTOMER KEY
- CLEAR MACHINE ID
- TRIAL INTERVAL DATA
- EXPORT COUNTER
659
KEY
657
DECRYPTION OPERATION
MODIFY EXPORT COUNTER
661
KEY
662
665
MEMORY MEDIA ATTRIBUTE
667
MODIFIER
ENCRYPTION OPERATION
669
MEMORY MEDIA ATTRIBUTE
671
673
MODIFIER
ENCRYPTION OPERATION
675
679
COPY TO MEMORY MEDIA
677
MEMORY MEDIA
681
COPY TO MEMORY MEDIA
677
MEMORY MEDIA
655
MACHINE IDENTIFICATION FILE
- MACHINE ID
- TRIAL INTERVAL DATA
DECRYPTION OPERATION
663

EXPORT OPERATION
FIG. 30

EP 0 679 977 A1

Rank Xerox (UK) Business Services
(3.10/3.09/3.3.4)

CROSS-REFERENCE TO RELATED APPLICATION

The present application is related to U.S. Patent Application Serial No. 08/235,033, entitled "Method and Apparatus for Enabling Trial Period Use of Software Products: Method and Apparatus for Utilizing a Decryption Stub," further identified by Attorney Docket No. BT9-93-070; U.S. Patent Application Serial No. 08/235,035, entitled "Method and Apparatus for Enabling Trial Period Use of Software Products: Method and Apparatus for Allowing a Try-and-Buy User Interaction," further identified by Attorney Docket No. DA9-94-008; U.S. Patent Application Serial No. 08/235,032, entitled "Method and Apparatus for Enabling Trial Period Use of Software Products: Method and Apparatus for Generating a Machine-Dependent Identification," further identified by Attorney Docket No. DA9-94-009; and U.S. Patent Application Serial No. 08/235,418, entitled "Method and Apparatus for Enabling Trial Period Use of Software Products: Method and Apparatus for Utilizing an Encryption Header," further identified by Attorney Docket No. DA9-94-010, all filed of even date herewith by the inventors hereof and assigned to the assignee herein, and incorporated by reference herein.

BACKGROUND OF THE INVENTION

1.Technical Field:

The present invention relates in general to techniques for securing access to software objects, and in particular to techniques for temporarily encrypting and restricting access to software objects.

2. Description of the Related Art:

The creation and sale of software products has created tremendous wealth for companies having innovative products, and this trend will continue particularly since consumers are becoming evermore computer literate as time goes on. Computer software is difficult to market since the potential user has little opportunity to browse the various products that are available. Typically, the products are contained in boxes which are shrink-wrapped closed, and the potential customer has little or no opportunity to actually interact with or experience the software prior to purchasing. This causes considerable consumer dissatisfaction with products, since the consumer is frequently forced to serially purchase a plurality of software products until an acceptable product is discovered. This is perhaps one significant cause of the great amount of software piracy which occurs in our economy. A potential software purchaser will frequently "borrow" a set of diskettes from a friend or business associate, with the stated intention of using the software for a temporary period. Frequently, such temporary use extends for long intervals and the potential customer may never actually purchase a copy of the software product, and may instead rely upon the borrowed copy.

Since no common communication channel exists for the sampling of software products, such as those created in movie theaters by movie trailers, and in television by commercials, software manufacturers are forced to rely upon printed publications and direct mail advertisements in order to advertise new products and solicit new customers. Unfortunately, printed publications frequently fail to provide an accurate description of the product, since the user interaction with the product cannot be simulated in a static printed format. The manufacturers of computer software products and the customers would both be well served if the customers could have access to the products prior to making decisions on whether or not to purchase the product, if this could be accomplished without introducing risk of unlawful utilization of the product.

The distribution of encrypted software products is one mechanism a software vendor can utilize to distribute the product to potential users prior to purchase; however, a key must be distributed which allows the user access to the product. The vendor is then forced to rely entirely upon the honesty and integrity of a potential customer. Unscrupulous or dishonest individuals may pass keys to their friends and business associates to allow unauthorized access. It is also possible that unscrupulous individuals may post keys to publicly-accessible bulletin boards to allow great numbers of individuals to become unauthorized users. Typically, these types of breaches in security cannot be easily prevented, so vendors have been hesitant to distribute software for preview by potential customers.

SUMMARY OF THE INVENTION

It is one object of the present invention to provide a method and apparatus for distributing software objects from a producer to potential users which allows the user a temporary trial period without subjecting the software product to unnecessary risks of piracy or unauthorized utilization beyond the trial interval. Preferably this is accomplished by providing a software object on a computer-accessible memory media along with a file management program. Preferably, the software object is reversibly functionally limited, through one or more particular encryption operations. The computer-accessible memory media is shipped from the producer to the potential user utilizing conventional mail and delivery services. Upon receipt, the potential user loads the file management program into a user-controlled data processing system and associates it with the operating system for the data processing system. Then, the computer-accessible memory media is read utilizing the user-controlled data processing system. The file management program is executed by the user-controlled data processing system and serves to restrict access to the software object for a predefined and temporary trial period. During the temporary trial mode of operation, the software object is temporarily enabled by reversing the reversible functional limitation of the software object. This is preferably accomplished by decryption of the encrypted software object when the software object is called by the operating system of the user-controlled data processing system. The file management program preferably prevents copying operations, so the encrypted software project is temporarily decrypted when it is called by the operating system. If the potential user elects to purchase the software object, a permanent use mode of operation is entered, wherein the functional limitation of the software object is permanently reversed, allowing unlimited use to the software object by the potential user. This facilitates browsing operations which allow the potential user to review the software and determine whether it suits his or her needs.

The file management program continuously monitors the operating system of the user-controlled data processing system for operating system input calls and output calls. The file management program identifies when the operating system of the user-controlled data processing system calls for a software object which is subject to trial-interval browsing. Then, the file management system fetches a temporary access key associated with the software object, and then examines the temporary access key to determine if it is valid. Next, the file management program reverses the functional limitation of the software object, and passes it to the data processing system for processing.

It is another objective of the present invention to provide a method and apparatus for distributing a software object from a source to a user, wherein a software object is encrypted utilizing a long-lived encryption key, and directed from the source to the user. The encrypted software object is loaded onto a user-controlled data processing system having a particular system configuration. A numerical machine identification based at least in part upon the particular configuration of the user-controlled data processing system is then derived. Next, a temporary key is derived which is based at least in part upon the numerical machine identification and the long-lived encryption key. A long-lived key generator is provided for receiving the temporary key and producing the long-lived encryption key. The temporary key allows the user to generate for a prescribed interval the long-lived encryption key to access the software object. These operations are performed principally by a file management program which is operable in a plurality of modes. These modes include a set up mode of operation, a machine identification mode of operation, and a temporary key derivation mode of operation. During the set up mode of operation, the file management program is loaded onto a user-controlled data processing system and associated with an operating system for the user-controlled data processing system. During the machine identification mode of operation, the file management program is utilized to derive a numerical machine identification based upon at least on attribute of the user-controlled data processing system. During the temporary key derivation mode of operation, a temporary key is derived which is based at least in part upon the numerical machine identification. The file management program also allows a trial mode of operation, wherein the file management program is utilized by executing it with the user-controlled data processing system to restrict access to the software object for an interval defined by the temporary key, during which the long-lived key generator is utilized in the user-controlled data processing system to provide the long-lived key in response to receipt of at least one input including the temporary key.

It is yet another objective of the present invention to provide a method and apparatus in a data processing system for securing access to particular files which are stored in a computer-accessible memory media. A file management program is provided as an operating system component of the data processing system. A plurality of files are stored in the computer-accessible memory media, including at least one encrypted file and at least one unencrypted file. For each encrypted file, a preselected portion is recorded in computer memory, a decryption block is generated which includes information which can be utilized to decrypt the file, and the decryption block is incorporated into the file in lieu of the preselected portion which has been recorded elsewhere in computer memory. The file management program is utilized to monitor data processing operation calls for a called file stored in the computer-accessible memory media. The file management program determines whether the called file has an associated decryption block. The file management program processes the called file in a particular manner dependent upon whether or not the called file has an associated decryption block. The incorporation of the decryption block does not change the size of the encrypted file, thus preventing certain types of processing errors. During the trial interval, the encrypted file is maintained in an encrypted condition, and cannot be copied. If the potential user opts to purchase the software product, a permanent key is provided which results in replacement of the preselected portion to the file in lieu of the decryption block. Once the decryption block is removed, the encrypted file may be decrypted to allow unrestricted use by the purchaser. Preferably, the file management program is utilized to intercept files as they are called by the operating system, and to utilize the decryption block to derive a name for a key file and read the called file. The decryption block of each encrypted file includes a validation segment which is decrypted by the file management program and compared to a selected segment for the called file to determine whether the key can decrypt the particular file. If the decrypted validation segment matches a known clear text validation segment, the file is then dynamically decrypted as it is passed for further processing.

It is yet another objective of the present invention to provide a method and apparatus in a data processing system for securing access to particular files which are stored in a computer-accessible memory media. A file management program is provided as an operating system component of a data processing system. In a computer-accessible memory media available to the data processing system, at least one encrypted file and one unencrypted file are stored. The encrypted file has associated with it an unencrypted security stub which is at least partially composed of executable code. The file management program is utilized to monitor the data processing system calls for a called file stored in the computer accessible memory media, to determine whether the called file has an associated unencrypted security stub, and to process the called file in a particular manner dependent upon whether or not the called file has an associated unencrypted security stub. More particularly, if it is determined that the called file has no associated unencrypted security stub, the called file is allowed to be processed. However, if it is determined that the called file has an associated unencrypted security stub, it must be examined before a decision can be made about whether or not to allow it to be processed. First, the unencrypted security stub is examined in order to obtain information which allows decryption operations to be performed. Then, the decryption operations are performed. Finally, the called file is allowed to pass for further processing. Preferably, the called file is dynamically decrypted as it is passed to the operating system for processing. Also, the unencrypted security stub is separated from the called file prior to execution of the called file. However, if the unencrypted security stub accidentally remains attached to the called file, processing operations must be stopped, and a message must be posted in order to prevent the processor from becoming locked-up.

It is still another objective of the present invention to provide a method and apparatus for distributing a software object from a source to a user. A computer-accessible memory media is distributed from the source to a potential user. It includes a software object which is encrypted utilizing a predetermined encryption engine and a long-lived and secret key. An interface program is provided which facilitates interaction between the source and the user. The interface program includes machine identification module which generates a machine identification utilizing at least on predetermined attribute of the user-controlled data processing system. It also further includes a long-lived and secret key generator which receives as an input at least a temporary key and produces as an output a long-lived and secret key. A validation module is provided which tests temporary key determined its validity. The source of the software object maintains a temporary key generator which receives as an input at least a machine identification and produces an output of the temporary key. An interface program is loaded onto the user-controlled data processing system. The machine identification module is utilized to examine at least one predetermined attribute of the user-controlled data processing system and to generate the machine identification. During interaction between the source and the user, the machine identification is communicated over an insecure communication channel. At the source of the software object, the temporary key is generated utilizing the machine identification (and other information) as an input to the temporary key generator. During interaction between the source and the user, the temporary key is communicated, typically over an insecure communication channel. Next, the validation module is utilized to determine the validity of the temporary key. The long-lived and secret key generator is then utilized to receive the temporary key and generate the long-lived and secret key in order to decrypt and temporarily gain access to the software object. The user is also provided with an import module and an export module which allow for the utilization of portable memory media to transfer the encrypted software object, a key file, and a machine identification file from one machine in a distributed data processing system to another machine in the distributed data processing system, while allowing the temporary key to allow temporary trial access to the software object.

The above as well as additional objectives, features, and advantages of the present invention will become apparent in the following detailed written description.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objectives and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

Figure 1 is a pictorial representation of a stand-alone data processing system, a telephone, and a variety of computer-accessible memory media all of which may be utilized in the implementation of the preferred technique of enabling trial period use of software products;

Figure 2 is a pictorial representation of a distributed data processing system which may utilize the technique of the present invention of enabling trial period use of software products;

Figure 3 is a block diagram representation of data processing system attributes which may be utilized to generate a machine identification, in accordance with the present invention;

Figure 4 is a block diagram depiction of a routine for encrypting software objects;

Figure 5 is a pictorial representation of the exchange of information between a source (a software vendor) and a user (a customer), in accordance with the teachings of the present invention;

Figure 6 is a flowchart representation of the broad steps employed in building a user interface shell, in accordance with the present invention;

Figure 7 is a flowchart representation of vendor and customer interaction in accordance with the present invention;

Figures 8, 9, 10a, and 10b depict user interface screens which facilitate trial period operations in accordance with the present invention;

Figure 11 depicts a user interface which is used to initiate a temporary access key;

Figure 12 is a block diagram depiction of the preferred technique of generating a machine identification;

Figure 13 is a block diagram depiction of an encryption operation which is utilized to encrypt a machine identification, in accordance with the present invention;

Figure 14 is a block diagram representation of the preferred technique for generating a product key, in accordance with the present invention;

Figure 15 is a block diagram representation of a preferred technique utilizing a temporary product key to generate a real key which can be utilized to decrypt one or more software objects;

Figures 16 and 17 depict a preferred technique of validating the real key which is derived in accordance with the block diagram of Figure 15;

Figure 18 is a block diagram depiction of the preferred routine for encyrpting a key file which contains information including a temporary product key;

Figure 19 is a block diagram depiction of the preferred technique of handling an encryption header in an encrypted file, in accordance with the present invention;

Figure 20 depicts in block diagram form the technique of utilizing a plurality of inputs in the user-controlled data processing system to derive the real key which may be utilized to decrypt an encrypted software object;

Figure 21 depicts a decryption operation utilizing the real key derived in accordance with Figure 20;

Figure 22 is a block diagram depiction of a comparison operation which is utilized to determine the validity of the real key;

Figure 23 depicts a decryption operation utilizing a validated real key;

Figures 24, 25, 26, 27, 28 depict the utilization of an encryption header in accordance with the present invention;

Figure 29 is a flowchart representation of the preferred technique of providing a trial period of use for an encrypted software object;

Figures 30 and 31 depict export and import operations which may be utilized to perform trial period use operations in a distributed data processing system;

Figures 32 and 33 provide an alternative view of the import and export operations which are depicted in Figures 30 and 31;

Figures 34 and 35 provide a block diagram depiction of an alternative technique for performing an export/import operation.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENT

The method and apparatus of the present invention for enabling trail period use of software products can be utilized in stand-alone PCs such as that depicted in Figure 1, or in distributed data processing systems, such as that depicted in Figure 2. In either event, temporary trial period access to one or more software products depends upon utilization of the trial product on a particular data processing system with particular data processing system attributes. This is accomplished by encrypting the trial software product utilizing a temporary access key which is based upon one or more data processing system attributes. Figure 3 graphically depicts a plurality of system configuration attributes, which may be utilized in developing a temporary access key, as will be described in greater detail herebelow. To begin with, the environment of the stand-alone data processing system of Figure 1, and the distributed data processing system of Figure 2 will be described in detail, followed by a description of particular system configuration attributes which are depicted in Figure 3.

With reference now to the figures and in particular with reference to Figure 1, there is depicted a pictorial representation of data processing system 10 which may be programmed in accordance with the present invention. As may be seen, data processing system 10 includes processor 12 which preferably includes a graphics processor, memory device and central processor (not shown). Coupled to processor 12 is video display 16 which may be implemented utilizing either a color or monochromatic monitor, in a manner well known in the art. Also coupled to processor 12 is keyboard 14. Keyboard 14 preferably comprises a standard computer keyboard which is coupled to the processor by means of a cable.

Also coupled to processor 12 is a graphical pointing device, such as mouse 20. Mouse 20 is coupled to processor 12, in a manner well known in the art, via a cable. As is shown, mouse 20 may include left button 24, and right button 26, each of which may be depressed, or "clicked", to provide command and control signals to data processing system 10. While the disclosed embodiment of the present invention utilizes a mouse, those skilled in the art will appreciate that any graphical pointing device such as a light pen or touch sensitive screen may be utilized to implement the method of the present invention. Upon reference to the foregoing, those skilled in the art will appreciate that data processing system 10 may be implemented utilizing a so-called personal computer, such as the Model 80 PS/2 computer manufactured by International Business Machines Corporation of Armonk, New York.

While the present invention may be utilized in stand-alone data processing systems, it may also be utilized in a distributed data processing system, provided the import and export routines of the present invention are utilized to transfer one or more encrypted files, their encrypted key files, and associated file management programs through a portable memory media (such as diskettes or tapes) between particular data processing units within the distributed data processing system. While the import and export routines of the present invention will be described in greater detail herebelow, it is important that a basic distributed data processing system be described and understood.

Figure 3 provides a block diagram depiction of a plurality of data processing system attributes which may be utilized to uniquely identify a particular data processing system (whether a stand-alone or a node in a distributed data processing system), and which further can be utilized to generate in the machine identification value which is utilized to derive or generate a temporary access product key which may be utilized to gain access to an encrypted product for a particular predefined trial interval. A data processing system may include a particular system bus 60 architecture, a particular memory controller 74, bus controller 76, interrupt controller 78, keyboard mouse controller 80, DMA controller 66, VGA video controller 82, parallel controller 84, serial controller 86, diskette controller 88, and disk controller 82. Additionally, a plurality of empty or occupied slots 106 may be used to identify the particular data processing system. Each particular data processing system may have attributes which may be derived from RAM 70, ROM 68, or CMOS RAM 72. End devices such as printer 96, monitor 94, mouse 92, keyboard 90, diskette 100, or disk drive 104 may be utilized to derive one or more attributes of the data processing system which may be processed in a predetermined manner to derive a machine identification value. The derivation of the machine identification value will be described in greater detail below. The present invention is directed to an efficient method of distributing software programs to users which would provide to them a means to try the program before obtaining (by purchasing) a license for it. In accordance with this concept, complete programs are distributed to potential users on computer-accessible memory media such as diskettes or CD-ROMs. The concept is to generate keys that allow the user to access the programs from the distributed media. In this environment, a file management program provides a plurality of interfaces which allows the user to browse the different products. The interfaces allow ordering and unlocking of the software products contained on the distributed media. Unlocking of the software product is accomplished by the reception, validation, and recording of a temporary access (decryption) key.

The file management program is resident in the user-controlled data processing system and becomes a part of the operating system in the user's computer. An example of such a resident program (in the PC DOS environment) would be a resident program TSR, for "terminate and stay resident" operations, that intercepts and handles DOS file input and output operations. When a temporary access key is provided to a user, system files are checked to see if this file has been used in a trial mode of operation before. If the product has never been used in a trial mode of operation, the temporary key is saved. Once the trial mode of operation key exists, an encrypted application can only be run if it is initiated by the file management program. The file management program will recognize that the application is encrypted and that a valid trial mode of operation key exists for the particular operation. A valid trial mode of application key is one that has not expired. The trial mode of operation may be defined by either a timer, or a counter. A timer can be used to count down a particular predefined period (such as thirty days); alternatively, the counter can be used to decrement through a predefined number of trial "sessions" which are allowed during the trial mode of operation. If the key is valid, the file management program communicates directly with the TSR and enables the trial mode of operation for a particular encrypted application. The file management program then kicks off the encrypted application. The code which is resident in the operating system of the user-controlled data processing system maintains control over the operating system. It monitors the use of the trial mode of operation keys to allow files to be decrypted and loaded into memory, but prevents the encrypted files from being decrypted and copied to media. This is done by using the operating system to determine which applications are trying to access the data and only allowing the applications that have permission to access the data to do so.

Figure 4 is a block diagram depiction of a routine for encrypting software objects. The binary characters which make up software object 201 are supplied as an input to encryption engine 205. Real key 203 is utilized as an encryption key in encryption engine 205. The output of encryption engine 205 is an encrypted software object 207. Encryption engine 205 may be any conventional encryption operation such as the published and well known DES algorithm; alternatively, the encryption engine 205 may be an exclusive-OR operation which randomizes software object 201.

Figure 5 is a pictorial representation of the exchange of information between a source 209 (a software vendor) and a user 211 (a potential customer, in accordance with the teachings of the present invention. The arrows between source 209 and user 211 represent exchanges of objects or information between vendor 209 and 211. In the exchange of flow 203, computer-accessible memory media is directed from source 209 to user 211. This transfer may occur by US mail delivery, courier delivery, express service delivery, or by delivery through printed publications such as books and magazines. Alternatively, an electronic document may be transferred from source 209 to user 211 utilizing electronic mail or other transmission techniques. In flow 215, user-specific information, preferably including a unique machine identification number which identifies the data processing system of user 211, is transferred from user 211 to source 209 via an insecure communication channel; typically, this information is exchanged over the telephone, but may be passed utilizing electronic mail or other communication techniques. In flow 217, source 209 provides a product key to user 211. The product key allows the product contained in the memory media to be temporarily accessed for a prescribed and predefined interval. This interval is considered to be a "trial" interval during which user 211 may become familiar with the software and make a determination on whether or not he or she wishes to purchase the software product. User 211 must communicate additionally with source 209 in order to obtain permanent access to the software product. The product key allows user 211 to obtain access to the software product for a particular predefined time interval, or for a particular number of predefined "sessions." As time passes, the user's clock or counter runs down. At the termination of the trial period, further access is denied. Therefore, the user 211 must take affirmative steps to contact source 209 and purchase a permanent key which is communicated to user 211 and which permanently unlocks a product to allow unrestricted access to the software product.

The communication between source 209 and user 211 is facilitated by a user interface. The creation of the interface is depicted in flowchart form in Figure 6. The process begins at software block 219, and continues at software block 221, wherein source 209 makes language and locale selections which will determine the language and currencies utilized in the interface which facilitates implementation of the trial period use of the software products. A plurality of software products may be bundled together and delivered to user 211 on a single computer-accessible memory media. Therefore, in accordance with software block 223, source 209 must make a determination as to the programs which will be made available on a trial basis on the computer-accessible memory media, and the appropriate fields are completed, in accordance with software block 223. Next, in accordance with software block 225, the programs are functionally limited or encrypted. Then, in accordance with software block 227, the shell is loaded along with the computer program products onto a computer-accessible memory media such as a diskette or CD ROM. The process ends at software block 229.

Figure 7 is a flowchart representation of vendor and customer interaction in accordance with the present invention. The flow begins at software block 231, and continues at step 233, wherein computer-accessible memory media are distributed to users for a try-and-buy trial interval. Then, in accordance with step 235, the file management program is loaded from the computer-accessible memory media onto a user-controlled data processing system for execution. The file management program includes a plurality of interface screens which facilitate interaction between the vendor and the customer, which and which set forth the options available to the customer. Thus, in accordance with step 237, the file management program allows browsing and displays appropriate user interfaces. Next, in accordance with step 239, the customer and the vendor interact, typically over the telephone or electronic mail, to allow the vendor to gather information about the customer and to distribute a temporary key which allows access to one or more software products which are contained on the computer-accessible memory media for a predefined trial interval. Typically, the interval will be defined by an internal clock, or by a counter which keeps track of the number of sessions the potential purchaser has with a particular software product or products. Step 241 represents the allowance of the trial interval use. Then, in accordance with software block 243, the file management program monitors and oversees all input and output calls in the data processing system to prevent unauthorized use of the encrypted software products contained on the computer-accessible memory media. In the preferred embodiment of the present invention, the file management program monitors for calls to encrypted files, and then determines whether access should be allowed or denied before the file is passed for further processing. The customer can assess the software product and determine whether he or she desires to purchase it. If a decision is made to purchase the product, the customer must interact once again with the vendor, and the vendor must deliver to the customer a permanent key, as is set forth in step 245. The process ends when the customer receives the permanent key, decrypts the one or more software products that he or she has purchased, and is then allowed ordinary and unrestricted access to the software products.

Figures 8, 9, 10a, and 10b depict user interface screens which facilitate trial period operations in accordance with the present invention. Figure 8 depicts an order form user interface 249 which is displayed when the customer selects a "view order" option from another window. The order form user interface 249 includes a title bar 251 which identifies the software vendor and provides a telephone number to facilitate interaction between the potential customer and the vendor. An order form field 255 is provided which identifies one or more software products which may be examined during a trial interval period of operation. A plurality of subfields are provided including quantity subfield 259, item subfield 257, description subfield 260, and price subfield 253. Delete button 261 allows the potential customer to delete items from the order form field. Subtotal field 263 provides a subtotal of the prices for the ordered software. Payment method icons 265 identify the acceptable forms of payment. Of course, a potential user may utilize the telephone number to directly contact the vendor and purchase one or more software products; alternatively, the user may select one or more software products for a trial period mode of operation, during which a software product is examined to determine its adequacy. A plurality of function icons 267 are provided at the lowermost portion of order form interface 249. These include a close icon, fax icon, mail icon, print icon, unlock icon, and help icon. The user may utilize a graphical pointing device in a conventional point-and-click operation to select one or more of these operations. The fax icon facilitates interaction with the vendor utilizing a facsimile machine or facsimile board. The print icon allows the user to generate a paper archival copy of the interaction with the software vendor.

The customer, the computer-accessible memory media, and the computer system utilized by the customer are identified by media identification 269, customer identification 273, and machine identification 271. The media identification is assigned to the computer-accessible memory media prior to shipping to the potential customer. It is fixed, and cannot be altered. The customer identification 273 is derived from interaction between the potential customer and the vendor. Preferably, the customer provides answers to selected questions in a telephone dialogue, and the vendor supplies a customer identification 273, which is unique to the particular customer. The machine identification 271 is automatically derived utilizing the file management program which is resident on the computer-accessible memory media, and which is unique to the particular data processing system being utilized by the potential customer. The potential customer will provide the machine identification to the vendor, typically through telephone interaction, although fax interaction and regular mail interaction is also possible.

Figure 9 is a representation of an order form dialog interface 275. This interface facilitates the acquisition of information which uniquely identifies the potential customer, and includes name field 277, address field 279, phone number field 281, facsimile number field 283, payment method field 285, shipping method field 287, account number field 289, expiration date field 291, value added tax ID field 293. Order information dialog interface 275 further includes print button 295 and cancel button 297 which allow the potential user to delete information from these fields, or to print a paper copy of the interface screen.

Figures 10a and 10b depict unlock dialog interface screens 301, 303. The user utilizes a graphical pointing device to select one or more items which are identified by the content item number field 307 and description field 309 which are components of unlock list 305. The interface further includes customer ID field 313 and machine ID field 315. Preferably, the vendor provides the customer identification to the customer in an interaction via phone, fax, or mail. Preferably, the customer provides to the vendor the machine identification within machine identification field 315 during interaction via phone, fax, or mail. Once the information is exchanged, along with an identification of the products which are requested for a trial interval period of operation, a temporary access key is provided which is located within key field 311. The key will serve to temporarily unlock the products identified and selected by the customer. Close button 319, save button 317, and help button 321 are also provided in this interface screen to facilitate user interaction.

Figure 10b depicts a single-product unlock interface screen 303. This interface screen includes only machine identification field 315, customer identification field 315, and key field 311. The product which is being unlocked need not be identified in this interface, since the dialog pertains only to a single product, and it is assumed that the user knows the product for which a temporary trial period of operation is being requested. Save button 317, cancel button 319, and help button 321 are also provided in this interface to facilitate operator interaction.

Figure 11 depicts a user interface screen which is utilized in unlocking the one or more encrypted products for the commencement of a trial interval mode of operation. The starting date dialog of Figure 11 is displayed after the "SAVE" push button is selected in the unlock dialog of either Figure 10a or Figure 10b. The user will be prompted to verify the correct starting date which is provided in date field 310. The user responds to the query by pointing and clicking to either the "continue" button 312, the "cancel" button 314, or the "help" button 316. The date displayed in field 310 is derived from the system clock of the user-controlled data processing system. The user may have to modify the system clock to make the date correspond to the official or stated date of commencement of the trial period of operation.

A trial interval operation can take two forms: one form is a functionally disabled product that allows a user to try all the features, but may not allow a critical function like printing or saving of data files. Another type of trial interval is a fully functional product that may be used for a limited time. This requires access protection, and allows a customer to try all the functions of a product for free or for a nominal fee. Typically, in accordance with the present invention, access to the product is controlled through a "timed" key. The trial period for using the product is a fixed duration determined by the vendor. The trial period begins when the key is issued. In accordance with the present invention, the products being previewed during the trial interval of operation can only be run from within a customer shell. A decryption driver will not allow the encrypted products to be copied in the clear, nor will it allow the product to be run outside the customer's shell. In an alternative embodiment, the trial interval is defined by a counter which is incremented or decremented with each "session" the customer has with the product. This may allow the customer a predefined number of uses of the product before decryption is no longer allowed with the temporary key.

The limits of the temporary access key are built into a "control vector" of the key. Typically, a control vector will include a short description of the key, a machine identification number, and a formatted text string that includes the trial interval data (such as a clock value or a counter value). The control vector cannot be altered without breaking the key. When a protected software product is run, the usage data must be updated to enforce the limits of the trial interval period of operation. In order to protect the clock or counter from tampering, its value is recorded in a multiple number of locations, typically in encrypted files. In the preferred embodiment of the present invention, the trial interval information (clock value and/or counter value) is copied to a "key file" which will be described in further detail herebelow, to a machine identification file, which will also be discussed herebelow, and to a system file. When access to an encrypted program is requested, all of these locations are checked to determine if the value for the clock and/or counter is the same. It is unlikely that an average user has the sophistication to tamper successfully with all three files. In the preferred embodiment, a combination of a clock and a counter is utilized to prevent extended use of backup and restore operations to reset the system clock. Although it is possible to reset a PC's clock each time a trial use is requested, this can also be detected by tracking the date/time stamps of certain files on the system and using the most recent date between file date/time stamps and the system clock. As stated above, one of the three locations the timer and/or counter information is stored is a system file. When operating in an OS/2 operating system, the time and usage data can be stored in the system data files, such as the OS2.INI in the OS/2 operating system. The user will have to continuously backup and restore these files to reset the trial and usage data. These files contain other data that is significant to the operation of the user system. The casual user can accidentally lose important data for other applications by restoring these files to an older version. In the present invention, these protection techniques greatly hinder a dishonest user's attempts to extend the trial interval use beyond the authorized interval.

In broad overview, in the present invention, the vendor loads a plurality of encrypted software products onto a computer-accessible memory media, such as a CD ROM or magnetic media diskette. Also loaded onto the computer-accessible memory media is a file management program which performs a plurality of functions, including the function of providing a plurality of user interface screens which facilitate interaction between the software vendor and the software customer. The computer-accessible memory media is loaded onto a user-controlled data processing system, and the file management program is loaded for execution. The file management program provides a plurality of user-interface screens to the software customer which gathers information about the customer (name, address, telephone number, and billing information) and receives the customer selections of the software products for which a trial interval is desired. Information is exchanged between the software vendor card customer, including: a customer identification number, a product identification number, a media identification number, and a machine identification number. The vendor generates the customer identification number in accordance with its own internal record keeping. Preferably, the representative of the software vendor gathers information from the software customer and types this information into a established blank form in order to identify the potential software customer. Alternatively, the software vendor may receive a facsimile or mail transmission of the completed order information dialog interface screen 275 (of Figure 9). The distributed memory media (such as CDs and diskettes) also include a file management program which is used to generate a unique machine identification based at least in part upon one attribute of the user-controlled data processing system. This machine identification is preferably a random eight-bit number which is created during a one-time setup process. Preferably, eight random bits are generated from a basic random number generator using the system time as the "seed" for the random number generator. Preferably, check bits are added in the final result. Those check bits are critical to the order system because persons taking orders must key in the machine ID that the customer reads over the phone. The check bits allow for instant verification of the machine ID without requiring the customer to repeat the number. Preferably, a master file is maintained on the user-controlled data processing system which contains the clear text of the machine identification and an encrypted version of the machine identification.

When the software customer places an order for a temporary trial use of the software products, he or she verbally gives to the telephone representative of the software vendor the machine identification. In return, the telephone representative gives the software customer a product key which serves as a temporary access key to the encrypted software products on the computer-accessible memory media, as well as a customer identification number. Preferably, the product key is a function of the machine identification, the customer number, the real encryption key for the programs or programs ordered, and a block of control data. The software customer may verify the product key by combining it with the customer number, and an identical block of control data to produce the real encryption key. This key is then used to decrypt an encrypted validation segment, to allow a compare operation. If the encrypted validation segment is identical to known clear text for the validation segment, then the user's file management program has determined that the product key is a good product key and can be utilized for temporary access to the software products. Therefore, if the compare matches, the key is stored on the user-controlled data processing system in a key file. Preferably, the key file contains the product key, a customer key (which is generated from the customer number and an internal key generating key) and a clear ASCII string containing the machine identification. All three items must remain unchanged in order for the decryption tool to derive the real encryption key. To further tie the key file to this particular user-controlled data processing system, the same key file is encrypted with a key that is derived from system parameters. These system parameters may be derived from the configuration of the data processing system.

Stated broadly, in the present invention the temporary key (which is given verbally over the phone, typically) is created from an algorithm that utilizes encryption to combine the real key with a customer number, the machine identification number, and other predefined clear text. Thus, the key is only effective for a single machine: even if the key were to be given to another person, it would not unlock the program on that other person's machine. This allows the software vendor to market software programs by distributing complete programs on computer-accessible memory media such as diskettes or CD ROMs, without significant risk of the loss of licensing revenue.

Some of the preferred unique attributes of the system which may be utilized for encryption operations include the hard disk serial number, the size and format of the hard disk, the system model number, the hardware interface cards, the hardware serial number, and other configuration parameters. The result of this technique is that a machine identification file can only be decrypted on a system which is an identical clone of the user-controlled data processing system. This is very difficult to obtain, since most data processing systems have different configurations, and the configurations can only be matched through considerable effort. These features will be described in detail in the following written description.

Turning now to Figure 12, the file management program receives the distributed computer-accessible memory media with encrypted software products and a file management program contained therein. The file management program assesses the configuration of the user-controlled data processing system, as represented in step 351 of Figure 12. The user-specific attributes of the data processing system are derived in step 353, and provided as an input to machine identification generator 355, which is preferably a random number generator which receives a plurality of binary characters as an input, and generates a pseudo-random output which is representative of machine identification 357. The process employed by machine identification generator 355 is any conventional pseudo-random number generator which receives as an input of binary characters, and produces as an output a plurality of pseudo-random binary characters, in accordance with a predefined algorithm.

With reference now to Figure 13, machine identification 357 is also maintained within the file management program in an encrypted form. Machine identification 357 is supplied as an input to encryption engine 359 to produce as an output the encrypted machine identification 361. Encryption engine 359 may comprise any convention encryption routine, such as the DES algorithm. A key 363 is provided also as an input to encryption engine 359, and impacts the encryption operation in a conventional manner. Key 363 is derived from system attribute selector 365. The types of system attributes which are candidates for selection include system attribute listing 367 which includes: the hard disk serial number, the size of the hard disk, the format of the hard disk, the system model number, the hardware interface card, the hardware serial number, or other configuration parameters.

In accordance with the present invention, the clear text machine identification 357 and the encrypted machine identification 361 are maintained in memory. Also, in accordance with the present invention, the file management program automatically posts the clear text machine identification 357 to the appropriate user interface screens. The user then communicates the machine identification to the software vendor where it is utilized in accordance with the block diagram of Figure 14. As is shown, product key encryption engine 375 is maintained within the control of the software vendor. This product key encryption engine 375 receives as an input: the machine identification 357, a customer number 369 (which is assigned to the customer in accordance with the internal record keeping of this software vendor), the real encryption key 371 (which is utilized to decrypt the software products maintained on the computer-accessible memory media within the custody of the software customer), a control block text 373 (which can be any predefined textural portion), and trial interval data 374 (such as clock and/or counter value which defines the trial interval of use). Product key encryption engine produces as an output a product key 377. Product key 377 may be communicated to the software customer via an insecure communication channel, without risk of revealing real key 371. Real key 371 is masked by the encryption operation, and since the product key 377 can only be utilized on a data processing system having a configuration identical to that from which machine identification 357 has been derived, access to the encrypted software product is maintained in a secure condition.

Upon delivery of product key 377, the file management program resident in the user-controlled data processing system utilizes real key generator 379 to receive a plurality of inputs, including product key 377, customer number 369, control block text 373, machine identification 357 and trial interval data 374. Real key generator 379 produces as an output the derived real key 381.

Encryption and decryption algorithm utilized to perform the operations of the product key encryption engine 375 and the real key generator 379 (of Figures 14 and 15) is described and claimed in copending U.S. Patent Application Serial No. 07/964,324, filed October 21, 1992, entitled "Method and System for Multimedia Access Control Enablement", which is incorporated herein as if fully set forth.

Next, as is depicted in Figures 16 and 17, the derived real key 381 is tested to determine the validity and authenticity of the product key 377 which has been provided by the software vendor. As is shown, the derived real key 381 is supplied as an input to encryption engine 385. A predetermined encrypted validation data segment 383 is supplied as the other input to encryption engine 385. Encryption engine supplies as an output derived clear validation text 387. Then, in accordance with Figure 17, the derived clear validation text 387 is compared to the known clear validation text 391 in comparator 389. Comparator 389 simply performs a bit-by-bit comparison of the derived clear validation text 387 with the known clear validation text 391. If the derived clear validation text 387 matches the known clear validation text 391, a key file is created in accordance with step 393; however, if the derived clear validation text 387 does not match the known clear validation text 391, a warning is posted to the user-controlled data processing system in accordance with step 395.

Turning now to Figure 18, key file 397 is depicted as including the temporary product key, the customer key (which is an encrypted version of the customer number), the machine identification number in clear text and the trial interval data (such as a clock and/or counter value). This key file is supplied as an input to encryption engine 399. Key 401 is also provided as an input to encryption engine 399. Key 401 is derived from unique system attributes 403, such as those system attributes utilized in deriving the machine identification number. Encryption engine 399 provides as an output the encrypted key file 405.

Figures 19, 20, 21, 22, and 23 depict operations of the file management program after a temporary access key has been received, and validated, and recorded in key file 397 (of Figure 18).

Figure 19 is a block diagram representation of the steps which are performed when an encrypted software product is called for processing by the user-control data processing system. The encrypted file 405 is fetched, and a "header" portion 407 is read by the user-controlled data processing system. The header has a number of components including the location of the key file. The location of the key file is utilized to fetch the key file in accordance with step 409. The header further includes an encrypted validation text 411. The encrypted validation text 411 is also read by the user-controlled data processing system. As is stated above (and depicted in Figure 18) the key file includes the product key 419, a customer key 417, and the machine identification 415. These are applied as inputs to decryption engine 413. Decryption engine 413 provides as an output real key 421. Before real key 421 is utilized to decrypt encrypted software products on the distributed memory media, it is tested to determine its validity. Figure 21 is a block diagram of the validation testing. Encrypted validation text 423, which is contained in the "header", is provided as an input to decryption engine 425. Real key 421 (which was derived in the operation of Figure 20) is also supplied as an input to decryption engine 425. Decryption engine 425 provides as an output clear validation text 427. As is set forth in block diagram form in Figure 22, clear validation text 427 is supplied as an input to comparator 429. The known clear validation text 431 is also supplied as an input to comparator 429. Comparator 429 determines whether the derived clear validation text 427 matches the known clear validation text 431. If the texts match, the software object is decrypted in accordance with step 433; however, if the validation text portions do not match, a warning is post in accordance with step 435. Figure 23 is a block diagram depiction of the decryption operation of step 433 of Figure 22. The encrypted software object 437 is applied as an input to decryption engine 439. The validated real key 441 is also supplied as an input to decryption engine 439. Decryption engine 439 supplies as an output the decrypted software object 443.

The encryption header is provided to allow for the determination of whether or not a file is encrypted when that file is stored with clear-text files. In providing the encryption header for the encrypted file, it is important that the file size not be altered because the size may be checked as part of a validation step (unrelated in any way to the concept of the present invention) during installation. Therefore, making the file larger than it is suppose to be can create operational difficulties during installation of the software. The encryption header is further necessary since the file names associated with the encrypted software products cannot be modified to reflect the fact that the file is encrypted, because the other software applications that may be accessing the encrypted product will be accessing those files utilizing the original file names. Thus, altering the file name to indicate that the file is encrypted would prevent beneficial and desired communication between the encrypted software product and other, perhaps related, software products. For example, spreadsheet applications can usually port portions of the spreadsheet to a related word processing program to allow the integration of financial information into printed documents. Changing the hard-coded original file name for the word processing program would prevent the beneficial communication between these software products. The encryption header of the present invention resolves these problems by maintaining the encrypted file at its nominal file length, and by maintaining the file name for the software product in an unmodified form.

Figure 24 graphically depicts an encrypted file with encryption header 451. The encryption header 451 includes a plurality of code segments, including: unique identifier portion 453, the name of the key file portion 455, encrypted validation segment 457, encryption type 459, offset to side file 461, and encrypted file data 463. Of course, in this view, the encrypted file data 463 is representative of the encrypted software product, such as a word processing program or spreadsheet. The encryption header 451 is provided in place of encrypted data which ordinarily would comprise part of the encrypted software product. The encryption header is substituted in the place of the first portion of the encrypted software product. In order to place the encryption header 451 at the front of the encrypted software product of encrypted file data 463, a portion of the encrypted file data must be copied to another location. Offset to side file 461 identifies that side file location where the displaced file data is contained.

Figure 25 graphically depicts the relationship between the directory of encrypted files and the side files. As is shown, the directory of encrypted files 465 includes file aaa, file bbb, file ccc, file ddd, through file nnn. Each of these files is representative of a directory name for a particular encrypted software product. Each encrypted software product has associated with it a side file which contains the front portion of the file which has been displaced to accommodate encryption header 451 without altering the size of the file, and without altering the file name. File aaa has associated with it a side file AAA. Software product file bbb has associated with it a side file BBB. Encrypted software product ccc has associated with it a side file CCC. Encrypted software product ddd has associated with it a side file DDD. Encrypted software product nnn has associated with it a side file NNN. In Figure 25, directory names 467, 469, 471, 473, 475 are depicted as being associated with side files 477, 479, 481, 483, and 485. The purpose of the side files is to allow each of the encrypted software products to be tagged with an encryption header without changing the file size.

Encryption type segment 459 of the encryption header 451 identifies the type of encryption utilized to encrypt the encrypted software product. Any one of a number of conventional encryption techniques can be utilized to encrypt the product, and different encryption types can be utilized to encrypt different software products contained on the same memory media. Encryption type segment 459 ensures that the appropriate encryption/decryption routine is called so that the encrypted software product may be decrypted, provided the temporary access keys are valid and not expired. The name of key file segment 455 of encryption header 451 provides an address (typically a disk drive location) of the key file. As is stated above (in connection with Figure 18) the key file includes the product key, a customer key, and the clear machine ID. All three of these pieces of information are required in order to generate the real key (in accordance with Figure 20). Encrypted validation segment 457 includes the encrypted validation text which is utilized in the routine depicted in Figure 21 which generates a derived clear validation text which may be compared utilizing the routine of Figure 22 to the known clear validation text. Only if the derived clear validation text exactly matches the known clear validation text can the process continue by utilizing the derived and validated real key to decrypt the encrypted software product in accordance with the routine of Figure 23. However, prior to performing the decryption operations of Figure 23, the contents of the corresponding side file must be substituted back into the encrypted software product in lieu of encryption header 451. This ensures that the encrypted software product is complete prior to the commencement of decryption operations.

Each time a file is called for processing by the operating system of the user-controlled data processing system, the file management program which is resident in the operating system intercepts the input/output requests and examines the front portion of the file to determine if a decryption block identifier, such as unique identifier 453, exists at a particular known location. For best performance, as is depicted in Figure 24, this location will generally be at the beginning of the file. If the file management program determines that the file has the decryption block, the TSR will read the block into memory. The block is then parsed in order to build a fully qualified key file name by copying an environment variable that specifies the drive and directory containing the key files and concatenating the key file name from the encryption block. The TSR then attempts to open the key file. If the key file does not exist, the TSR returns an "access denied" response to the application which is attempting to open the encrypted file. If the key file is determined to exist, the TSR opens the key file and reads in the keys (the product key, the customer key, and the machine identification) and generates the real key. This real key is in use to decrypt the decryption block validation data. As is stated above, a comparison operation determines whether this decryption operation was successful. If the compare fails, the key file is determined to be "invalid", and the TSR returns an "access denied message" to the application which is attempting to open the encrypted software product. However, if the compare is successful, the file management program prepares to decrypt the file according to the encryption type found in the encryption header. The TSR then returns a valid file handle to the calling application to indicate that the file has been opened. When the application reads data from the encrypted file, the TSR reads and decrypts this data before passing it back to the application. If the data requested is part of the displaced data that is stored in the side file, the TSR will read the side file and return the appropriate decrypted block to the calling application without the calling application being aware that the data came from a separate file.

While the broad concepts of the encryption header are depicted in Figures 24 and 25, the more particular aspects of creating the encrypted files are depicted in Figures 26, 27, and 28. Figures 27 and 28 depict two types of data files. Figure 27 depicts a non-executing data file, while Figure 28 depicts an executing data file. Figure 26 depicts a header 499 which includes signature segment 501, header LEN 503, side file index 505, side file LEN 507, decryption type identifier 509, verification data 511, and key file name 518. As is shown in Figure 27, a software product begins as a clear file 521, and is encrypted in accordance with a particular encryption routine into encrypted file 523. Encryption type segment 509 of header 499 identifies the type of encryption utilized to change clear file 521 to encrypted file 523. Next, the front portion of encrypted file 523 is copied to side file 527 which is identified by side file index 505 and side file LEN 507 of header 499. Additionally, a copy of the clear text of the verification data is also included in side file 527. Then, header 499 is copied to the front portion of encrypted file 523 to form modified encrypted files 525. A similar process is employed for executing files, as depicted in Figure 28. The clear text copy of the software product (represented as clear file 531) is encrypted in accordance with a conventional routine, to form encrypted file 533. The front portion of encrypted file 533 is copied to side file 539 so that the overlaid data of encrypted file 533 is preserved. Furthermore, side file 539 includes a copy of the clear text of the verification data. Then, the encrypted file 533 is modified by overlaying and executable stub 537 and header 599 onto the first portion of encrypted file 553.

The purpose of executable stub 537 of Figure 28 will now be described. The DOS operating system for a personal computer will try to execute an encrypted application. This can result in a system "hang" or unfavorable action. The executable stub 357 of the executing file of Figure 28 is utilized to protect the user from attempting to execute applications that are encrypted: there would be considerable risk that a user would hang his system or format a drive if he or she try to run an encrypted file. The executable stub is attached to the front portion of the encrypted software product so that this stub is executed whenever the application is run without the installed TSR or run from a drive the TSR is not "watching". This stub will post a message to the user that explains why the application cannot run. In addition to providing a message, this executable stub can be used to perform sophisticated actions, such as:

(1) it can duplicate the functionality of the TSR and install dynamic encryption before kicking off the application a second time;
(2) it can turn on a temporary access key and kick off the application a second time;
(3) it can communicate with the TSR and inform it to look at the drive the application is being run from.

The executable stub is saved or copied into the encrypted program as follows:

(1) the application is encrypted;
(2) a decryption block is created for this program;
(3) a pre-built executable stub is attached to the front end of the decryption block;
(4) the length of the combined decryption header and executable stub is determined;
(5) the bytes at the front of the executable file equal to this length are then read into memory, preferably into a predefined side file location; and
(6) the encryption header and executable stub are then written over the leading bytes in the executable code.

The TSR can determine if an executable is encrypted by searching beyond the "known size" of the executable stub for the decryption block portion. When the TSR decrypts the executable stub it accesses the side file to read in the bytes that were displaced by the stub and header block.

Figure 29 provides a flowchart representation of operation during a trial period interval, which begins at software block 601. In accordance with software block 603, the file management program located in the operating system of the user-controlled data processing system continually monitors for input/output calls to the memory media. Then, in accordance with software block 605, for each input/output call, the called file is intercepted, and in accordance with software block 607 the operating system is denied access to the called file, until the file management program can determine whether access should be allowed or not. A portion of the called file is read where the decryption block should be located. This portion of the called file is then read, in accordance with software block 609, to derive a key file address in accordance with software block 611. The address which is derived is utilized to fetch the key file, in accordance with software block 613. In accordance with decision block 615, if the key file cannot be located, the process ends at software block 617; however, if it is determined in decision block 615 that the key file can be located, the key is derived in accordance with software block 619. The derived key is then utilized to decrypt the validation segment which is located within the encryption header, in accordance with software block 621. In decision block 623, the decryption validation segment is compared to the clear text for the decryption validation segment; if it is determined that the decrypted segment does not match the known clear text segment, the process continues at software block 625 by ending; however, if it is determined in decision block 623 that the decrypted validation segment does match the known clear text validation segment, the process continues as software block 627, wherein access to the called file is allowed. Then, the decryption type is read from the decryption header in accordance with software block 629, and the called file is dynamically decrypted in accordance with software block 631 as it is passed for processing by the operating system of the user-controlled data processing system, in accordance with software block 633. The process terminates at software block 635.

If unauthorized execution of an encrypted file is attempted, the executable stub will at least temporarily deny access and post a message to the system, but may handle the problem in a number of sophisticated ways which were enumerated above.

In accordance with the preferred embodiment of the present invention, during the trial interval, or at the conclusion of the trial interval, the prospective purchaser may contact the vendor to make arrangements for the purchase of a copy of the one or more software products on the computer-accessible memory media. Preferably, CD ROMs or floppy disks have been utilized to ship the product to the potential user. Preferably, the computer-accessible memory media includes the two encrypted copies of each of the products which are offered for a trial interval of use. One encrypted copy may be decrypted utilizing the file management program and the temporary key which is communicated from the vendor to the purchaser. The other encrypted copy is not provided for use in the trial interval mode of operation, but instead is provided as the permanent copy which may be decrypted and utilized once the software product has been purchased. In broad overview, the user selects a software product for a trial interval mode of operation, and obtains from the vendor temporary access keys, which allow the user access to the product (through the file management program) for a predefined trial interval. Before or after the conclusion of the trial interval, the user may purchase a permanent copy of the software product from the vendor by contacting the vendor by facsimile, electronic mail, or telephone. Once payment is received, the vendor communicates to the user a permanent access key which is utilized to decrypt the second encrypted copy of the software product. This encrypted product may be encrypted utilizing any conventional encryption routine, such as the DES algorithm. The permanent key allows the software product to be decrypted for unrestricted use. Since multiple copies of the product may be purchased in one transaction, the present invention is equipped with a technique for providing movable access keys, which will be discussed below in connection with Figures 30 through 35. In the preferred embodiment of the present invention, the encryption algorithm employed to encrypt and decrypt the second copy of the software product is similar to that employed in the trial interval mode of operation.

The present invention includes an export/import function which allows for the distribution of permanent access keys, after the conclusion of a trial interval period. Typically, an office administrator or data processing system manager will purchase a selected number of "copies" of the encrypted product after termination of a trial interval period. Certain individuals within the organization will then be issued permanent keys which allow for the unrestricted and permanent access to the encrypted product. In an office or work environment where the computing devices are not connected in a distributed data processing network, the permanent access keys must be communicated from the office administrator or data processing manager to the selected individuals within an organization who are going to receive copies of the encrypted software product. The permanent keys allow for permanent access to the product. Since not all employees within an organization may be issued copies of the particular encrypted product, the vendor would like to have the distribution occur in a manner which minimizes or prevents the distribution beyond the sales agreement or license agreement. Since the products are encrypted, they may be liberally distributed in their encrypted form. It is the keys which allow unrestricted access to the product which are to be protected in the current invention. To prevent the distribution of keys on electronic mail or printed communications, the present invention includes an export program which is resident in a source computer and an import program which is resident in a target computer which allow for the distribution of the access keys via a removable memory media, such as a floppy diskette. This ensures that the access keys are not subject to inadvertent or accidental distribution or disclosure. There are two principal embodiments which accomplish this goal.

In the first embodiment, one or more encrypted files which are maintained in the source computer are first decrypted, and then encrypted utilizing an encryption algorithm and an encryption key which is unique to the transportable memory media (such as a diskette serial number). The key file may then be physically carried via the diskette to a target computer, where it is decrypted utilizing a key which is derived by the target computer from interaction with the transferable memory media. Immediately, the key file or files are then encrypted utilizing an encryption operation which is keyed with a key which is derived from a unique system attribute of the target computer.

In the alternative embodiment, the transferrable memory media is loaded onto the target computer to obtain from the target computer import file a transfer key which is uniquely associated with the target computer, and which may be derived from one or more unique system attributes of the target computer. The memory media is then transferred to the source computer, where the one or more key files are decrypted, and then encrypted utilizing the transfer key. The memory media is then carried to the target computer where the transfer key is generated and utilized in a decryption operation to decrypt the one or more key files. Preferably, immediately the key files are encrypted utilizing an encryption operation which is keyed with a key which is uniquely associated with the target computer, and which may be derived from one or more unique computer configuration attributes. The first embodiment is discussed herein in connection with Figures 30, 31, 32, and 33. The second embodiment is discussed in connection with Figures 34 and 35.

Figures 30 and 31 depict in block diagram form export and import operations which allow an authorized user to move his permanent key to another data processing system using an "export" facility that produces a unique diskette image of the access key that has been enabled for import into another system. In accordance with the present invention, the access keys which are delivered over the telephone by the software vendor to the customer are less than 40 bytes in length. The key file that is produced is over 2,000 bytes in length. An export facility is provided for copying the key file and the machine identification file to a diskette. Both files are then encrypted with a modified diskette serial number to inhibit these files from being copied to a public forum where anyone could use them. An import facility provided in another system decrypts these files and adds the product key and machine identification from the diskette to a list of import product keys and machine identifications in the import systems master file, and copies the key file to the import system hard disk. The key file is encrypted on the import system as is disclosed above.

Figure 30 is a block diagram depiction of an export operation in accordance with the preferred embodiment of the present invention. As is shown, source computer 651 includes a key file 653 and a machine identification file 655. Key file 653 includes the product key, the customer key, the clear text of the machine identification for source computer 653, trial interval data (such as a clock and/or counter which define the trial interval period), and an export counter which performs the dual functions of defining the maximum number of export operations allowed for the particular protected software products and keeping track of the total number of export operations which have been accomplished. The machine identification file includes the machine identification number and trial interval data (such as a clock and/or counter which defines the trial interval period). Both key file 653 and machine identification file 655 are encrypted with any conventional encryption operation (such as the DES algorithm), which is keyed with a key which is derived from a unique system attribute of source computer 651. At the commencement of an export operation, key file 653 and machine identification file 655 are decrypted. Key file 653 is supplied as an input to decryption operation 657 which is keyed with key 659. Likewise, machine identification file 655 is supplied as an input to decryption operation 663 which is keyed with key 661. Decryption operations 657, 663 generate a clear text version of key file 653 and machine identification file 655. Once the clear text is obtained, the export counter which is contained within key file 653 is modified in accordance with block 661. For example, if this is the seventh permitted export operation out of ten permissible operations, the counter might read "7:10". The clear text version of key file 653 is supplied as an input to encryption operation 669. Encryption operation 669 may be any conventional encryption operation (such as the DES algorithm), which is keyed with a memory media attribute 665 which is unique to a memory media which is coupled to source computer 651, which has been subjected to modification of modifier 667. For example, a unique diskette serial number may be supplied as the "memory media attribute" which is unique to memory media 677. The diskette serial number is modified in accordance with modifier 667 to alter it slightly, and supply it as an input to encryption operations 669. The same operation is performed for the clear text of machine identification file 655. A unique memory media attribute 671 is modified by modifier 673 and utilized as a key for encryption operation 675, which may comprise any conventional encryption operation, such as the DES operation. Finally, the output of encryption operations 669 and 675 are supplied as inputs to copy operations 679, 681 which copy the encrypted key file 653 and machine identification file 655 to memory media 677.

Figure 31 is a block diagram depiction of an import operation. Memory media 677 (of Figure 30) is physically removed from source computer 651 (of Figure 30) and physically carried over to computer 707 (of Figure 31); alternatively, in a distributed data processing system, this transfer may occur without the physical removal of memory media 677. With reference now to Figure 31, in accordance with block 683, the machine identification of the target machine is copied to memory media 677 to maintain a record of which particular target computer received the key file and machine identification file. Then, in accordance with blocks 685, 693 the encrypted key file 653 and machine identification file 655 are copied from the memory media to target computer 707. The encrypted key file 653 is supplied as an input to decryption operation 689 which is keyed with key 687. Decryption operation 689 reverses the encryption operation of block 669, and provides as an output a clear text version of key file 653. Likewise, machine identification file 655 is supplied as an input to decryption operation 697, which is keyed with key 695. Decryption operation 697 reverses the encryption of encryption operation 675 and provides as an output the clear text of machine identification file 655. In accordance with block 691, the machine identification of the source computer 651 is retrieved and recorded in memory in the clear text of key file 653. Next, the clear text of key file 653 is supplied as an input to encryption operation 699. Encryption operation 699 is a conventional encryption operation, such as the DES operation, which is keyed with a target computer unique attribute, such as the machine identification or modified machine identification for the target computer 707. The clear text of machine identification file 655 is supplied as an input to encryption operation 703. Encryption operation 703 is any conventional encryption operation, such as the DES encryption operation, which is keyed with a unique target computer attribute 705, such as machine identification or modified machine identification of target computer 707. The output of encryption operation 699 produces an encrypted key file 709 which includes a product key (which is the same temporary product key of key file 653 of source computer 651), a customer number (which is the same customer number of key file 653 of source computer 651), and clear machine identification (which is the machine identification for target computer 707, and not that of source computer 651), trial interval data (which is identical to the trail interval data of key file 653 of source 651), and an identification of the machine identification of the source computer 651. The output of encryption operation 703 defines machine identification file 711, which includes the machine identification of the target computer 707 (and not that of the source computer 651), and the trial interval data (which is identical to that of machine identification file 655 of source computer 651).

Figures 32 and 33 provide alternative views of the import and export operations which are depicted in Figures 30 and 31, and emphasize several of the important features of the present invention. As is shown, source computer 801 includes machine identification file 803 which is encrypted with a system attribute key which is unique to the source computer 801. The machine identification file includes machine identification file number as well as count of the number of exports allowed for each protected software product, and a count of the total number of exports which have been utilized. For example, the first export operation carries a count of "1:10", which signifies that one export operation of ten permitted export operations has occurred. In the next export operation, the counter is incremented to "2:20" which signifies that two of the total number of ten permitted export operations has occurred. Each target computer which receives the results of the export operation is tagged with this particular counter value, to identify that it is the recipient of a particular export operation. For example, one source computer system may carry a counter value of "1:10", which signifies that it is the recipient of the first export operation of ten permitted export operations. Yet another target computer may carry the counter value of "7:10", which signifies that this particular target computer received the seventh export operation of a total of ten permitted export operations. In this fashion, the target computer maintains a count of a total number of used export operations, while the source computers each carry a different counter value which identifies it a the recipient of the machine identification file and key file from the source computer from particular ones of the plurality of permitted export operations.

Note that in source computer 801 machine identification file 803 and key file 805 are encrypted with an encryption algorithm which utilizes as a key a system attribute which is unique to source computer 801; however, once machine identification file 803 and key file 805 are transferred to a memory media, such as export key diskette 807, machine identification file 809 and key file 811 are encrypted in any conventional encryption operation which utilizes as an encryption key a unique diskette attribute, such as the diskette's serial number. This minimizes the possibility that the content of the machine ID file 809 and/or key file 811 can be copied to another diskette or other memory media and then utilized to obtain unauthorized access to the software products. This is so because for an effective transfer of the content of machine ID file 809 and key file 811 to a target computer to occur, the target computer must be able to read and utilize the unique diskette attribute from the export key diskette 807. Only when the machine ID file 809 and key file 811 are presented to a target computer on the diskette onto which these items were copied can an effective transfer occur. The presentation of the machine ID file 809 and key file 811 on a diskette other than export key diskette 807 to a potential target computer will result in the transfer of meaningless information, since the unique attribute of export key diskette 807 (such as the diskette serial number) is required by the target computer in order to successfully accomplish the decryption operation.

As is shown in Figure 33, export key diskette 807 is presented to target computer 813. Of course, the machine identification file 809 and key file 811 are in encrypted form. In the transfer from export key diskette 807 to target computer 813, the content of machine ID file 809 is updated with the machine identification of the target computer 813, and the count of imports utilized. In accomplishing the transfer to target computer 813, a machine identification file 815 is constructed which includes a number of items such as machine identification for the target computer 813, customer information, as well as a list of the machine identification number of the source computer 801. Both machine identification file 815 and the key file 817 are encrypted utilizing a conventional encryption operation which uses as a key a unique attribute of target computer 813. This ties machine identification file 815 and key file 817 to the particular target computer 813.

By using an export/import counter to keep track of the total number of authorized export/import operations, and the total number of used export/import operations, the present invention creates an audit trail which can be utilized to keep track of the distribution of software products during the trial interval. Each source computer will carry a record of the total number of export operations which have been performed. Each source computer will carry a record of which particular export/import operation was utilized to transfer one or more protected software products to the target computer. The memory media utilized to accomplish the transfer (such as a diskette, or group of diskettes) will carry a permanent record of the machine identification numbers of both the source computer and the target computer's utilized in all export/import operations.

The procedure for implementing export and import operations ensures that the protected software products are never exposed to unnecessary risks. When the machine identification file and key file are passed from the source computer to the export diskette, they are encrypted with the unique attribute of the export diskette which prevents or inhibits copying of the export diskette or posting of its contents to a bulletin board as a means for illegally distributing the keys. During the import operations, the machine identification and key files are encrypted with system attributes which are unique to the target computer to ensure that the software products are maintained in a manner which is consistent with the security of the source computer, except that those software products are encrypted with attributes which are unique to the target computer, thus preventing illegal copying and posting of the keys.

The second embodiment of the export/import function is depicted in block diagram form in Figures 34 and 35. In broad overview, memory media 1677 is first utilized to interact with target computer 1707 to obtain from target computer 1707 a transfer key which is unique to target computer 1707, and which is preferably derived from one or more unique system attributes of target computer 1707. The transfer key may he a modification of the machine identification for target computer 1707. Next, the memory media 1677 is utilized to interact with source computer 1651 in an export mode of operation, wherein key file 1653 and machine identification file 1655 are first decrypted, and then encrypted utilizing the transfer key.

Figure 34 is a block diagram depiction of an export operation in accordance with the preferred embodiment of the present invention. As is shown, source computer 1651 includes a key file 1653 and a machine identification file 1655. Key file 1653 includes the product key, the customer key, the clear text of the machine identification for source computer 1653, trial interval data (such as a clock and/or counter which define the trial interval period), and an export counter which performs the dual functions of defining the maximum number of export operations allowed for the particular protected software products and keeping track of the total number of export operations which have been accomplished. The machine identification file includes the machine identification number and trial interval data (such as a clock and/or counter which defines the trial interval period). Both key file 1653 and machine identification file 1655 are encrypted with any conventional encryption operation (such as the DES algorithm), which is keyed with a key which is derived from a unique system attribute of source computer 1651. At the commencement of an export operation, key file 1653 and machine identification file 1655 are decrypted. Key file 1653 is supplied as an input to decryption operation 1657 which is keyed with key 1659. Likewise, machine identification file 1655 is supplied as an input to decryption operation 1663 which is keyed with key 1661. Decryption operations 1657, 1663 generate a clear text version of key file 1653 and machine identification file 1655. Once the clear text is obtained, the export counter which is contained within key file 1653 is modified in accordance with block 1661. For example, if this is the seventh permitted export operation out of ten permissible operations, the counter might read "7:10". The clear text version of key file 1653 is supplied as an input to encryption operation 1669. Encryption operation 1669 may be any conventional encryption operation (such as the DES algorithm), which is keyed with the transfer key 1665 which was previously obtained. The same operation is performed for the clear text of machine identification file 1655. Transfer key 1671 is utilized as a key for encryption operation 1675, which may comprise any conventional encryption operation, such as the DES operation. Finally, the output of encryption operations 1669 and 1675 are supplied as inputs to copy operations 1679, 1681 which copy the encrypted key file 1653 and machine identification file 1655 to memory media 1677.

Figure 35 is a block diagram depiction of an import operation. Memory media 1677 (of Figure 34) is physically removed from source computer 1651 (of Figure 34) and physically carried over to computer 1707 (of Figure 35); alternatively, in a distributed data processing system, this transfer may occur without the physical removal of memory media 1677. With reference now to Figure 35, in accordance with block 1683, the machine identification of the target machine is copied to memory media 1677 to maintain a record of which particular target computer received the key file and machine identification file. Then, in accordance with blocks 1685, 1693 the encrypted key file 1653 and machine identification file 1655 are copied from the memory media to target computer 1707. The encrypted key file 1653 is supplied as an input to decryption operation 1689 which is keyed with key 1687. Decryption operation 1689 reverses the encryption operation of block 1669, and provides as an output a clear text version of key file 1653. Likewise, machine identification file 1655 is supplied as an input to decryption operation 1697, which is keyed with key 1695. Decryption operation 1697 reverses the encryption of encryption operation 1675 and provides as an output the clear text of machine identification file 1655. In accordance with block 1691, the machine identification of the source computer 1651 is retrieved and recorded in memory in the clear text of key file 1653. Next, the clear text of key file 1653 is supplied as an input to encryption operation 1699. Encryption operation 1699 is a conventional encryption operation, such as the DES operation, which is keyed with a target computer unique attribute, such as the machine identification or modified machine identification for the target computer 1707. The clear text of machine identification file 1655 is supplied as an input to encryption operation 1703. Encryption operation 1703 is any conventional encryption operation, such as the DES encryption operation, which is keyed with a unique target computer attribute 1705, such as machine identification or modified machine identification of target computer 1707. The output of encryption operation 1699 produces an encrypted key file 1709 which includes a product key (which is the same temporary product key of key file 1653 of source computer 1651), a customer number (which is the same customer number of key file 1653 of source computer 1651), and clear machine identification (which is the machine identification for target computer 1707, and not that of source computer 1651), trial interval data (which is identical to the trail interval data of key file 1653 of source 1651), and an identification of the machine identification of the source computer 1651. The output of encryption operation 1703 defines machine identification file 1711, which includes the machine identification of the target computer 1707 (and not that of the source computer 1651), and the trial interval data (which is identical to that of machine identification file 1655 of source computer 1651).

While the invention has been particularly shown and described with reference to a preferred embodiment, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention.

Claims

1. A method of passing encrypted files between data processing systems, comprising:
   at a source computer providing at least one file which is encrypted with a key which is at least partially derived from at least one unique source computer system attribute;
   providing a transfer memory medium;
   at said source computer, decrypting said at least one file;
   at said source computer, encrypting said at least one file with a key which is derived from at least one unique transfer memory media attribute;
   at said source computer, copying said encrypted file to said transfer memory media;
   at a target computer, decrypting said at least one file;
   at said target computer, encrypting said at least one file with a key which is at least partially derived from at least one target computer system attribute.

2. A method of passing encrypted files between data processing systems, comprising:

at a source computer providing at least one file which is encrypted with a key which is at least partially derived from at least one unique source computer system attribute;
providing a transfer memory medium;
at a target computer copying a transfer encryption key which is unique to said target computer to said transfer memory media;
at said source computer, decrypting said at least one file;
at said source computer, encrypting said at least one file with said transfer encryption key;
at said source computer, copying said encrypted file to said transfer memory media;
at a target computer, decrypting said at least one file;
at said target computer, encrypting said at least one file with a key which is at least partially derived from at least one target computer system attribute.

3. A method of passing encrypted files according to Claims 1 or 2, further comprising:
providing a export counter in said source computer which defines a maximum number of permissible transfer operations; and
actuating said export counter for each transfer operation.

4. A method of passing encrypted files according to one of Claims 1 to 3, further comprising:
identifying each one of said permissible transfer operations to a particular target computer.

5. A method of passing encrypted files according to one of Claims 1 to 4, further comprising:
recording the occurrence of all transfer operations involving said transfer memory medium by obtaining identifying information from each target computer.

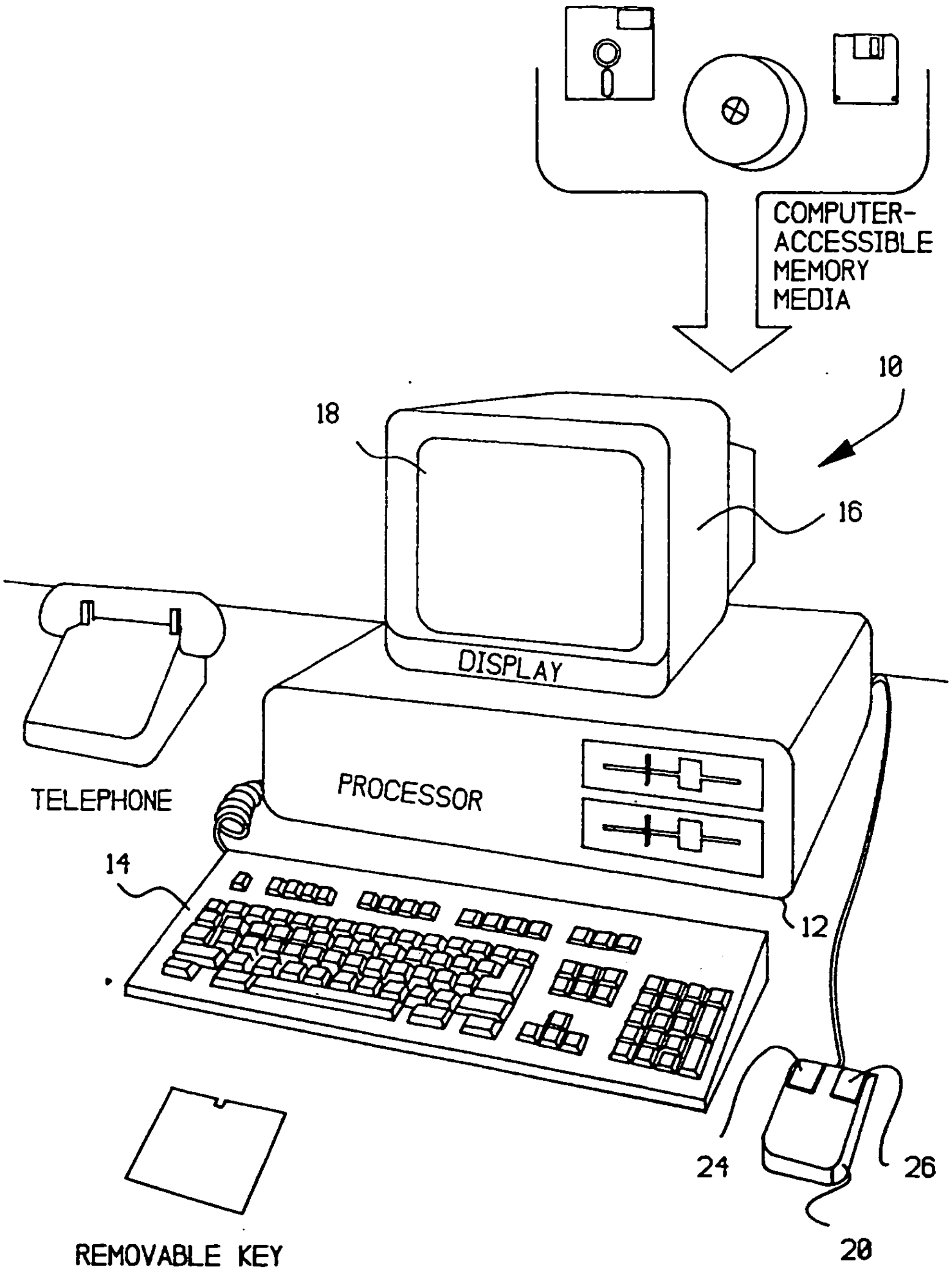
6. A method of passing encrypted files between data processing systems, comprising:
at a source computer providing at least one file which is encrypted with a key which is at least partially derived from at least one unique source computer system attribute;
providing a transfer memory medium;
initiating a particular transfer operation;
at said source computer, decrypting said at least one file;
including in said at least one file a transfer identifier which uniquely identifies said particular transfer operation;
at said source computer, encrypting said at least one file with a key which is derived from at least one unique transfer memory media attribute;
at said source computer, copying said encrypted file to said transfer memory media;
at a target computer, decrypting said at least one file;
at said target computer, encrypting said at least one file with a key which is at least partially derived from at least one target computer system attribute.

7. A method of passing encrypted files according to Claim 6, further comprising:
at said target computer, passing a unique target computer identification to said transfer memory media.

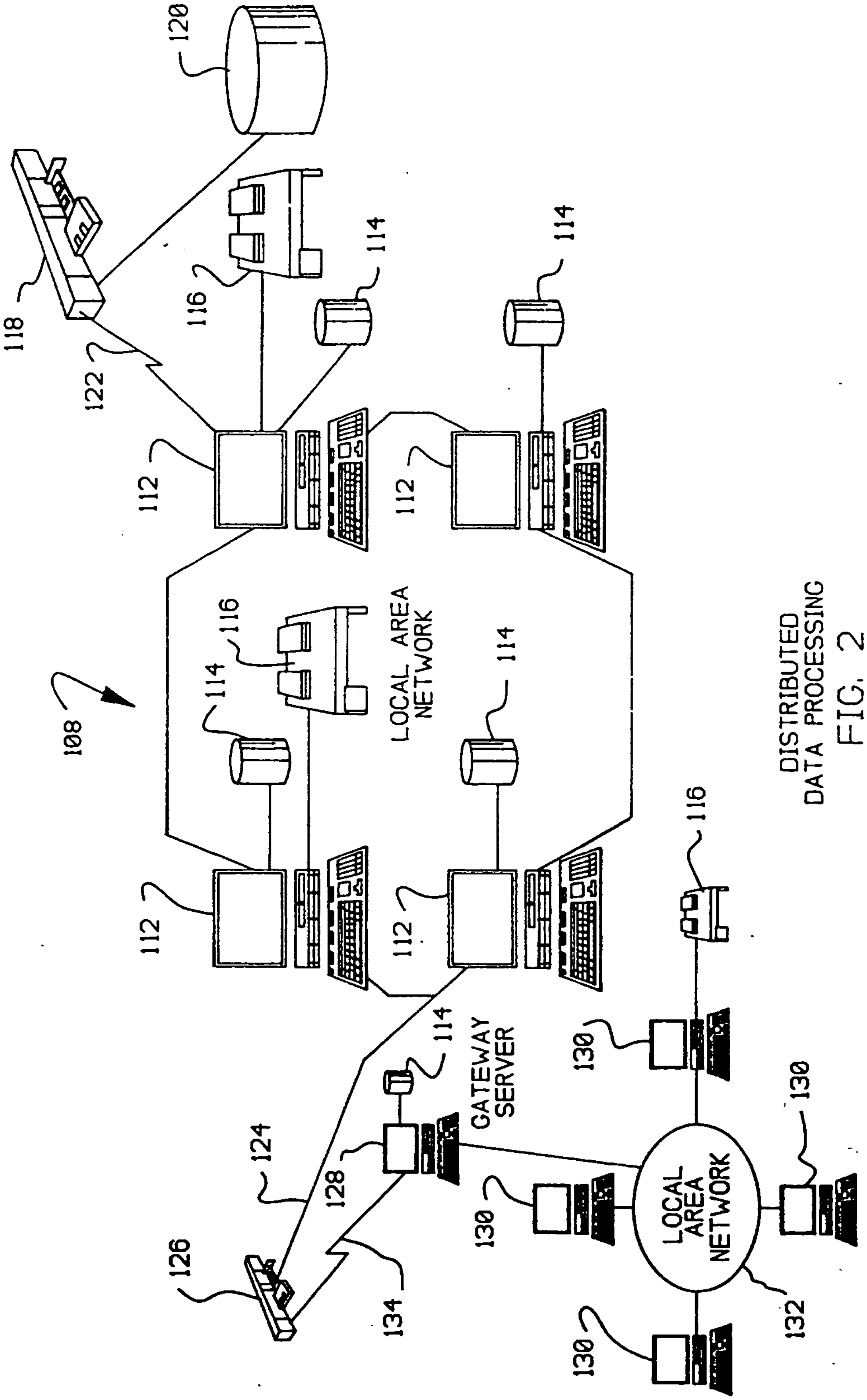
8. A method of passing encrypted files according to Claim 6 or 7, further comprising:
at said target computer, updating said at least one file to provide an identification of said source computer.

9. An apparatus passing encrypted files between data processing systems, comprising:
at least one file in a source computer which is encrypted with a key which is at least partially derived from at least one unique source computer system attribute;
a removable transfer memory medium having a unique attribute;
an export program for decrypting said at least one file and encrypting said at least one file with a key which is derived from said unique attribute and copying said encrypted file to said transfer memory media;
an import program at a target computer for decrypting said at least one file, and encrypting said at least one file with a key which is at least partially derived from at least one target computer system attribute.

10. An apparatus for passing encrypted files according to Claim 9, further comprising:
a export counter in said export program in said source computer which defines a maximum number of permissible transfer operations, and for counting each transfer operation.


COMPUTER-
ACCESSIBLE
MEMORY
MEDIA
10
18
16
DISPLAY
PROCESSOR
TELEPHONE
14
12
24
26
20
REMOVABLE KEY

STAND ALONE PC
FIG. 1


120
118
122
116
114
112
112
114
108
114
116
LOCAL AREA NETWORK
114
112
112
116
114
GATEWAY SERVER
124
128
126
134
130
130
130
130
LOCAL AREA NETWORK
132

DISTRIBUTED DATA PROCESSING
FIG. 2

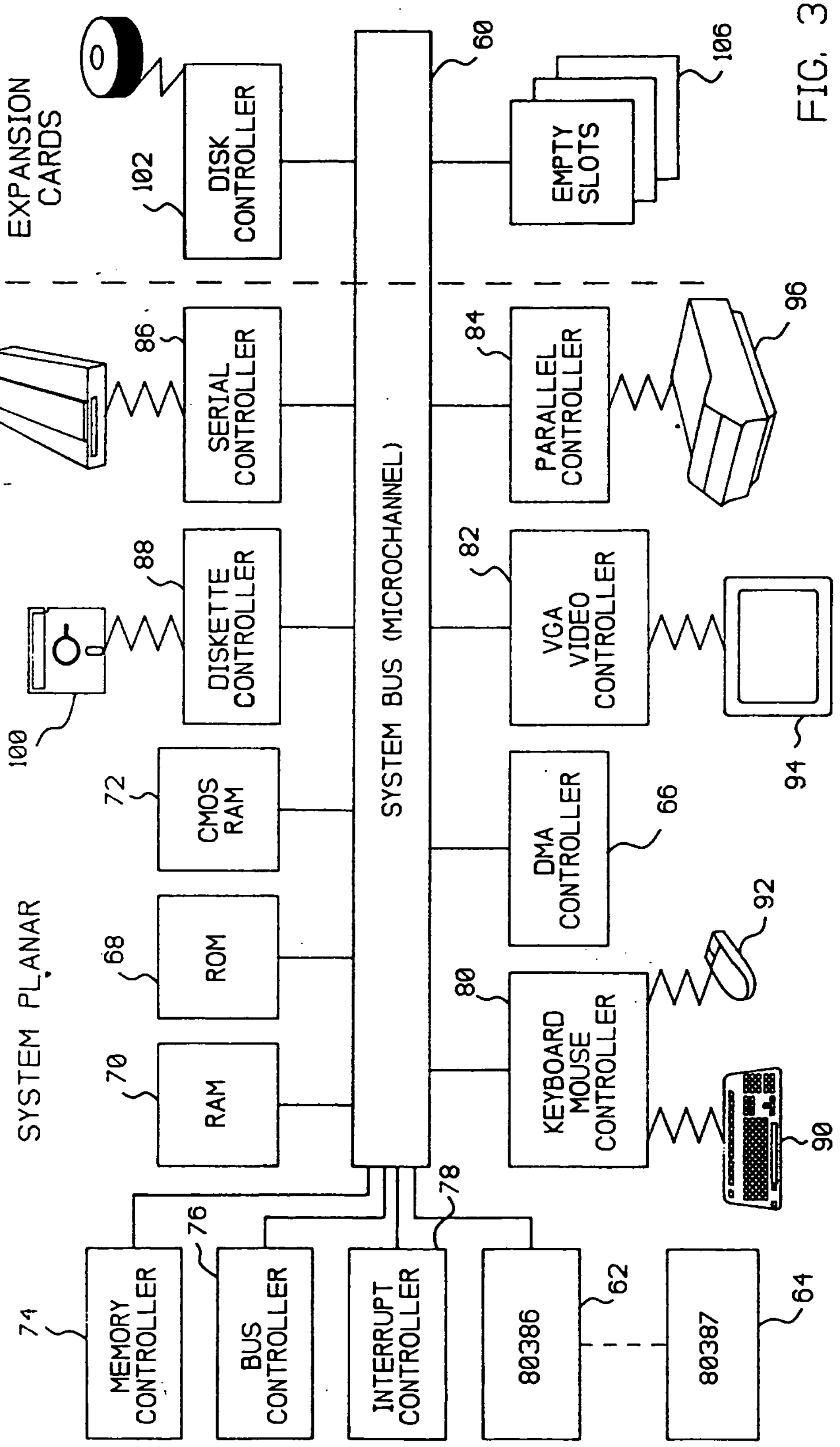

SYSTEM PLANAR
EXPANSION CARDS
SYSTEM BUS (MICROCHANNEL)
60
MEMORY CONTROLLER
74
BUS CONTROLLER
76
INTERRUPT CONTROLLER
78
80386
62
80387
64
RAM
70
ROM
68
CMOS RAM
72
DISKETTE CONTROLLER
88
100
SERIAL CONTROLLER
86
98
DISK CONTROLLER
102
KEYBOARD MOUSE CONTROLLER
80
90
92
DMA CONTROLLER
66
VGA VIDEO CONTROLLER
82
94
PARALLEL CONTROLLER
84
96
EMPTY SLOTS
106

FIG. 3

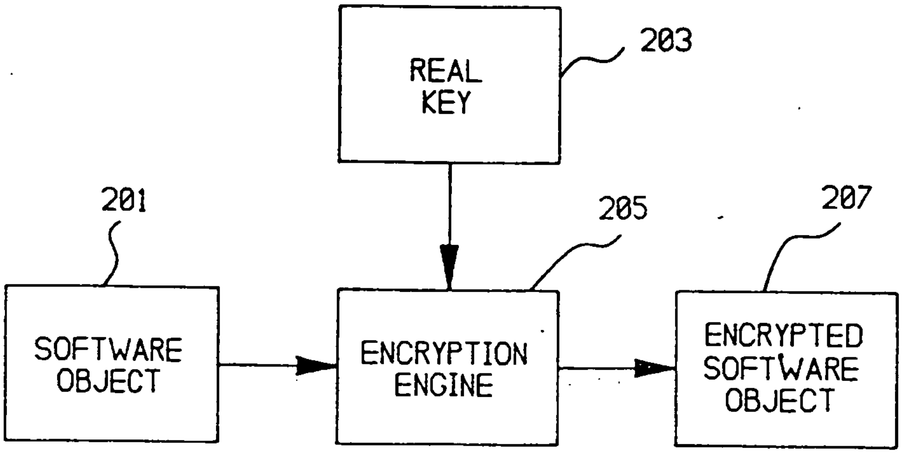

REAL
KEY
203
201
205
207
SOFTWARE
OBJECT
ENCRYPTION
ENGINE
ENCRYPTED
SOFTWARE
OBJECT

FIG. 4

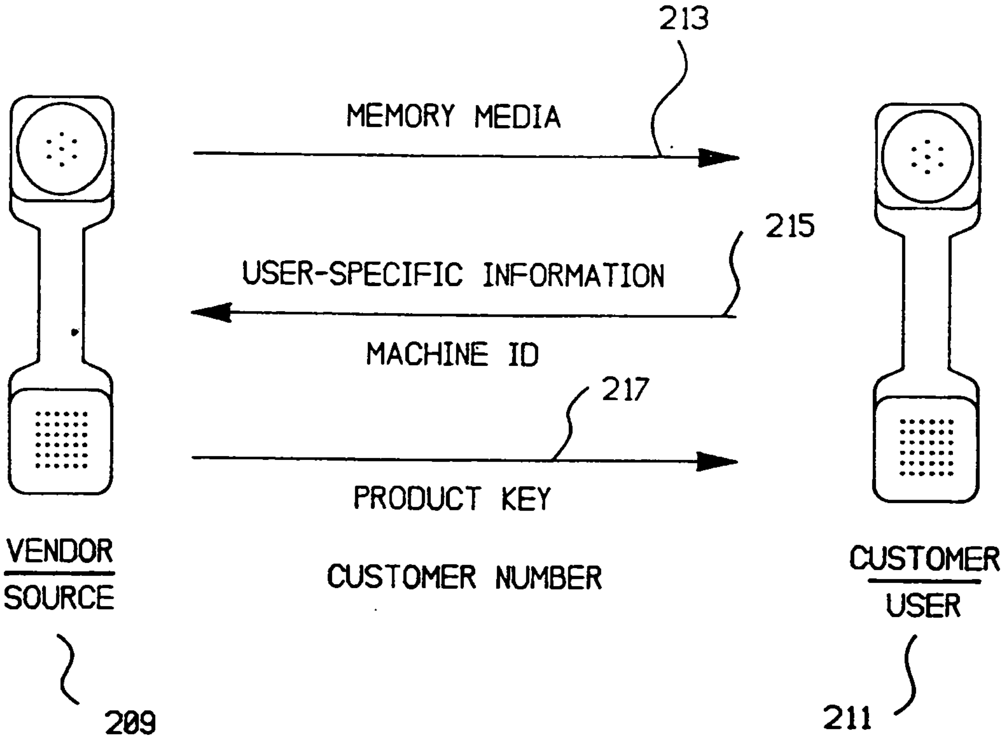
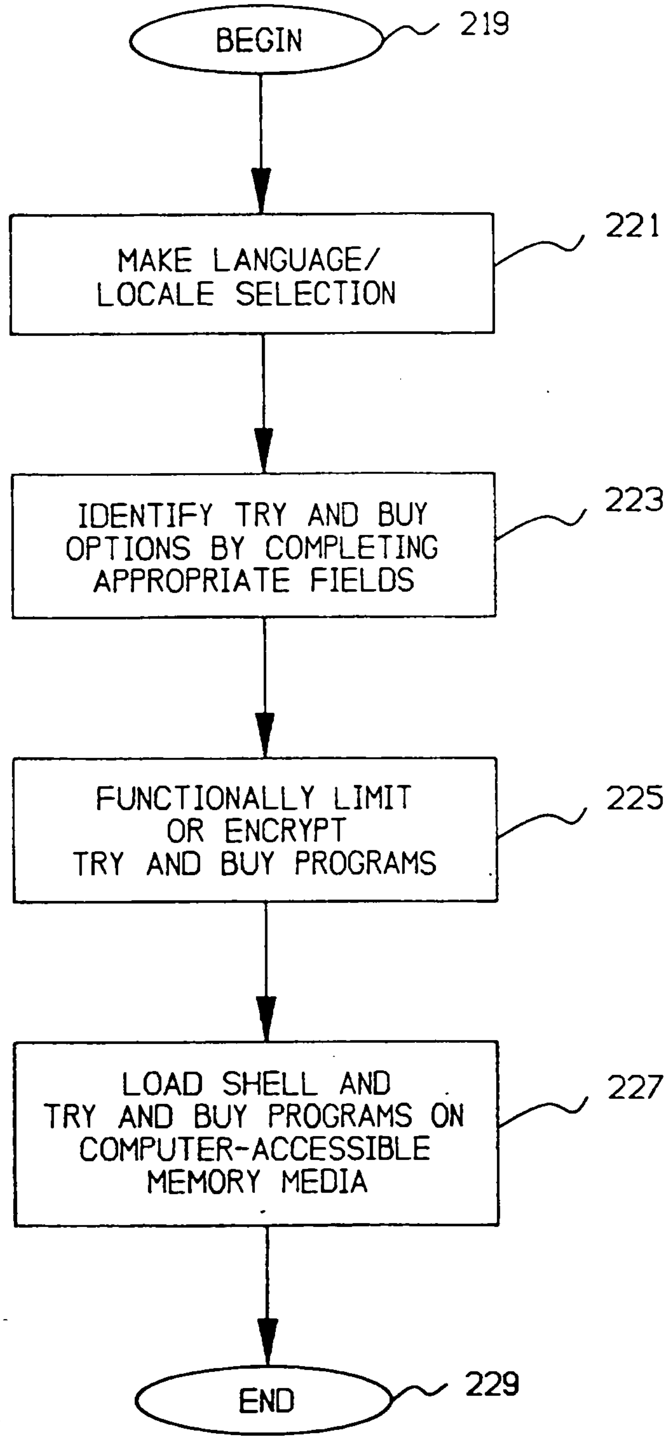
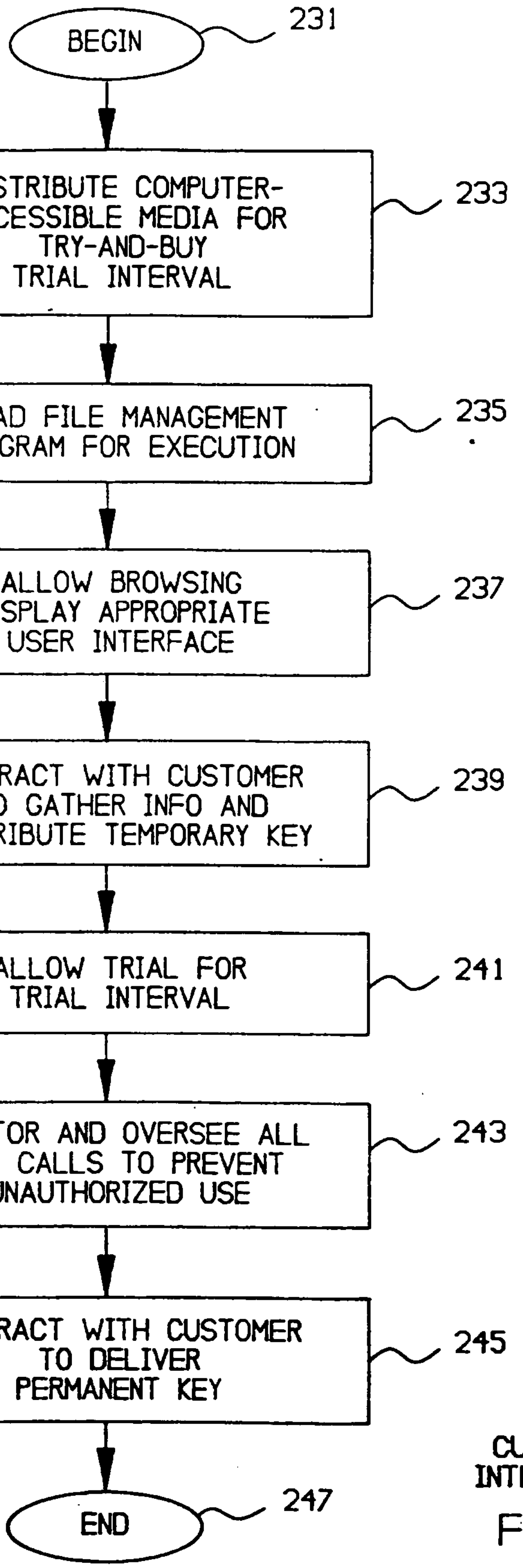

213
MEMORY MEDIA
215
USER-SPECIFIC INFORMATION
MACHINE ID
217
PRODUCT KEY
CUSTOMER NUMBER
VENDOR
SOURCE
209
CUSTOMER
USER
211

FIG. 5

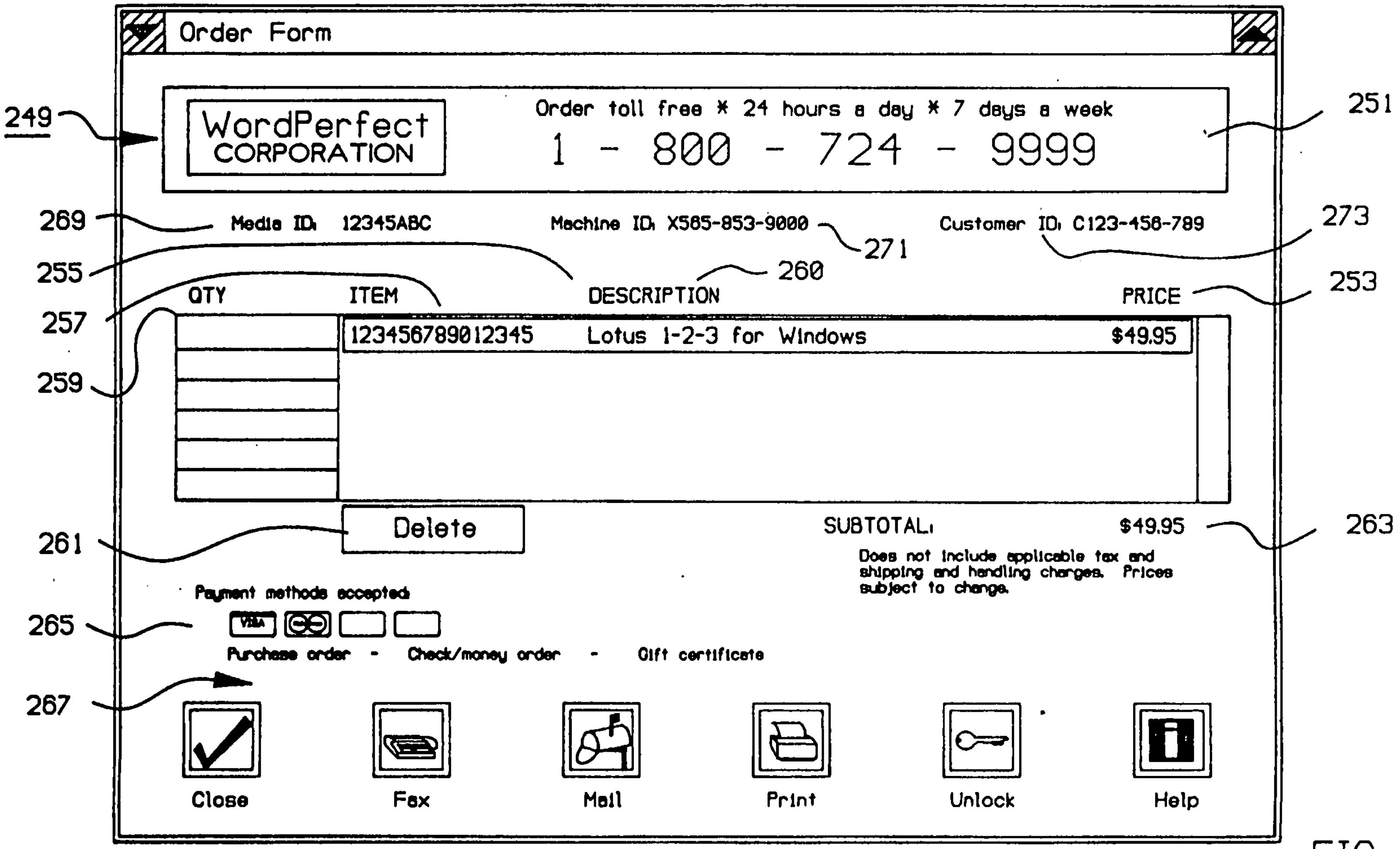

BEGIN
219
MAKE LANGUAGE/ LOCALE SELECTION
221
IDENTIFY TRY AND BUY OPTIONS BY COMPLETING APPROPRIATE FIELDS
223
FUNCTIONALLY LIMIT OR ENCRYPT TRY AND BUY PROGRAMS
225
LOAD SHELL AND TRY AND BUY PROGRAMS ON COMPUTER-ACCESSIBLE MEMORY MEDIA
227
END
229

BUILDING THE SHELL
FIG. 6


BEGIN
231
DISTRIBUTE COMPUTER-ACCESSIBLE MEDIA FOR TRY-AND-BUY TRIAL INTERVAL
233
LOAD FILE MANAGEMENT PROGRAM FOR EXECUTION
235
ALLOW BROWSING DISPLAY APPROPRIATE USER INTERFACE
237
INTERACT WITH CUSTOMER TO GATHER INFO AND DISTRIBUTE TEMPORARY KEY
239
ALLOW TRIAL FOR TRIAL INTERVAL
241
MONITOR AND OVERSEE ALL I/O CALLS TO PREVENT UNAUTHORIZED USE
243
INTERACT WITH CUSTOMER TO DELIVER PERMANENT KEY
245
END
247

CUSTOMER INTERACTION

FIG. 7



Order Form

WordPerfect CORPORATION

Order toll free * 24 hours a day * 7 days a week

1 - 800 - 724 - 9999

Media ID: 12345ABC

Machine ID: X585-853-9000

Customer ID: C123-456-789

QTY	ITEM	DESCRIPTION	PRICE
	123456789012345	Lotus 1-2-3 for Windows	$49.95

SUBTOTAL: $49.95

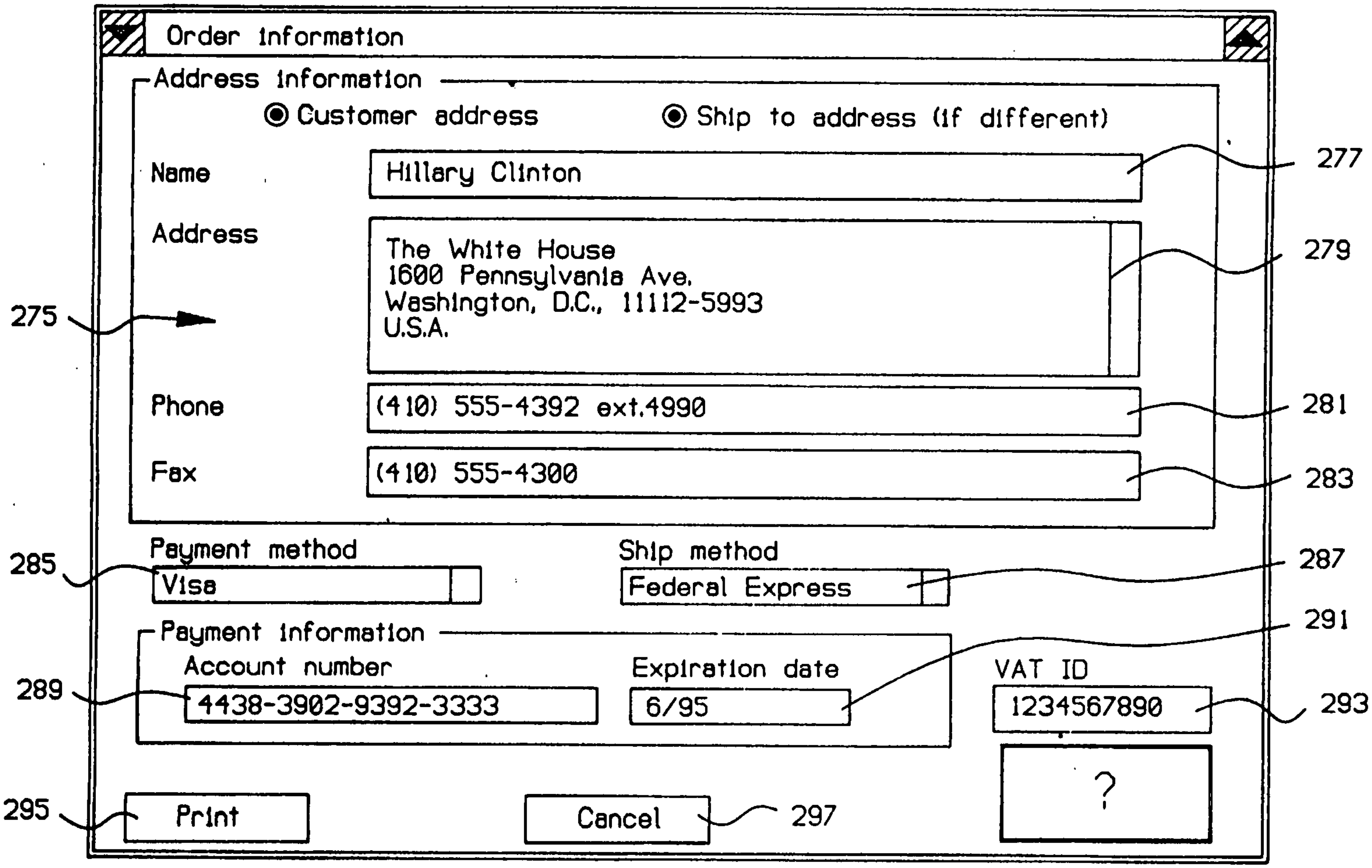
Does not include applicable tax and shipping and handling charges. Prices subject to change.

Payment methods accepted:

Purchase order - Check/money order - Gift certificate

249
251
269
271
273
255
260
257
253
259
261
263
265
267
Delete
VISA
Close
Fax
Mail
Print
Unlock
Help

FIG. 8


Order information
Address information
Print
Cancel
?
275
277
279
281
283
285
287
289
291
293
295
297

● Customer address
● Ship to address (if different)

Name: Hillary Clinton

Address:
The White House
1600 Pennsylvania Ave.
Washington, D.C., 11112-5993
U.S.A.

Phone: (410) 555-4392 ext.4990

Fax: (410) 555-4300

Payment method: Visa

Ship method: Federal Express

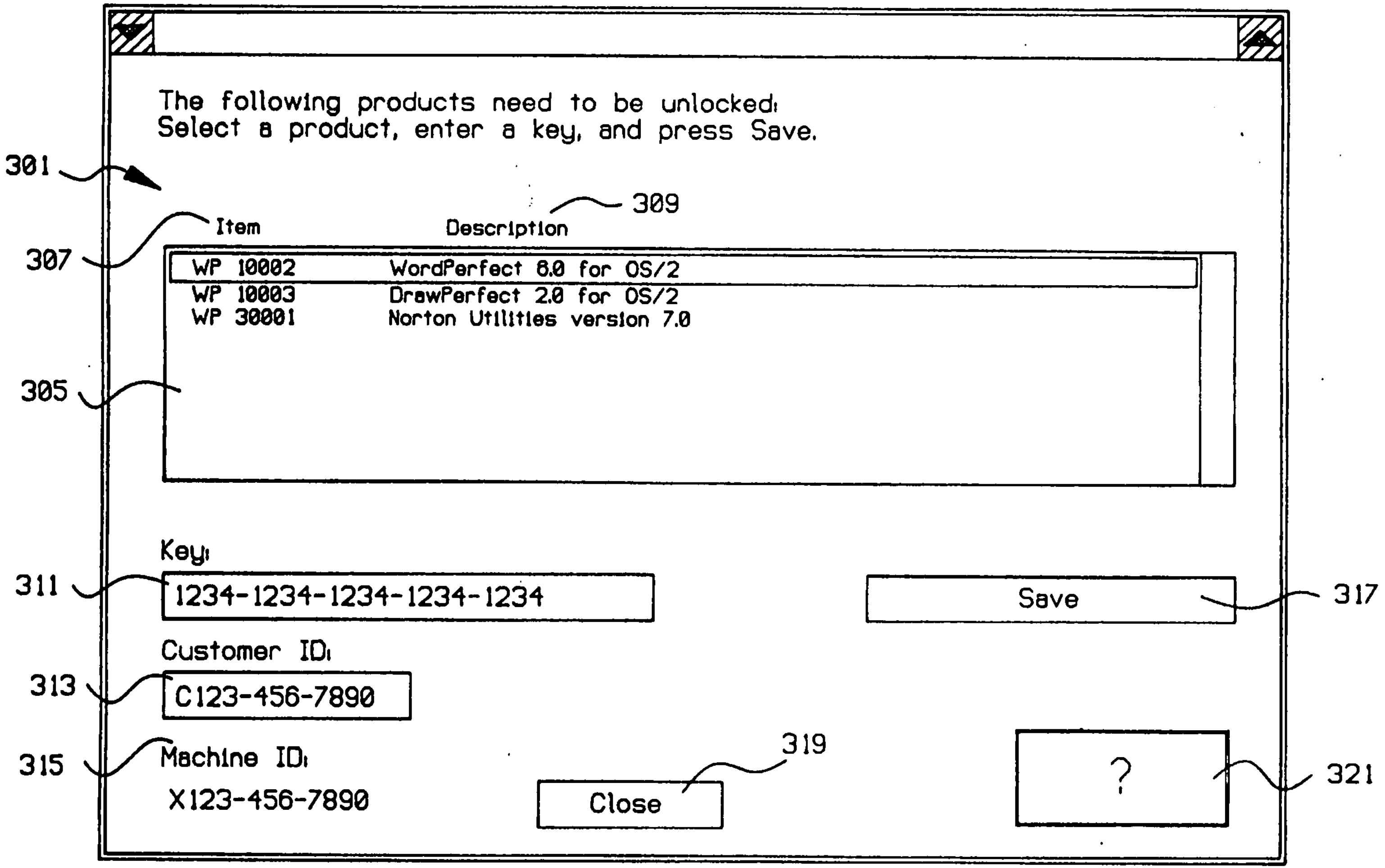
Payment information

Account number: 4438-3902-9392-3333

Expiration date: 6/95

VAT ID: 1234567890

FIG. 9



The following products need to be unlocked:
Select a product, enter a key, and press Save.

Item	Description
WP 10002	WordPerfect 6.0 for OS/2
WP 10003	DrawPerfect 2.0 for OS/2
WP 30001	Norton Utilities version 7.0

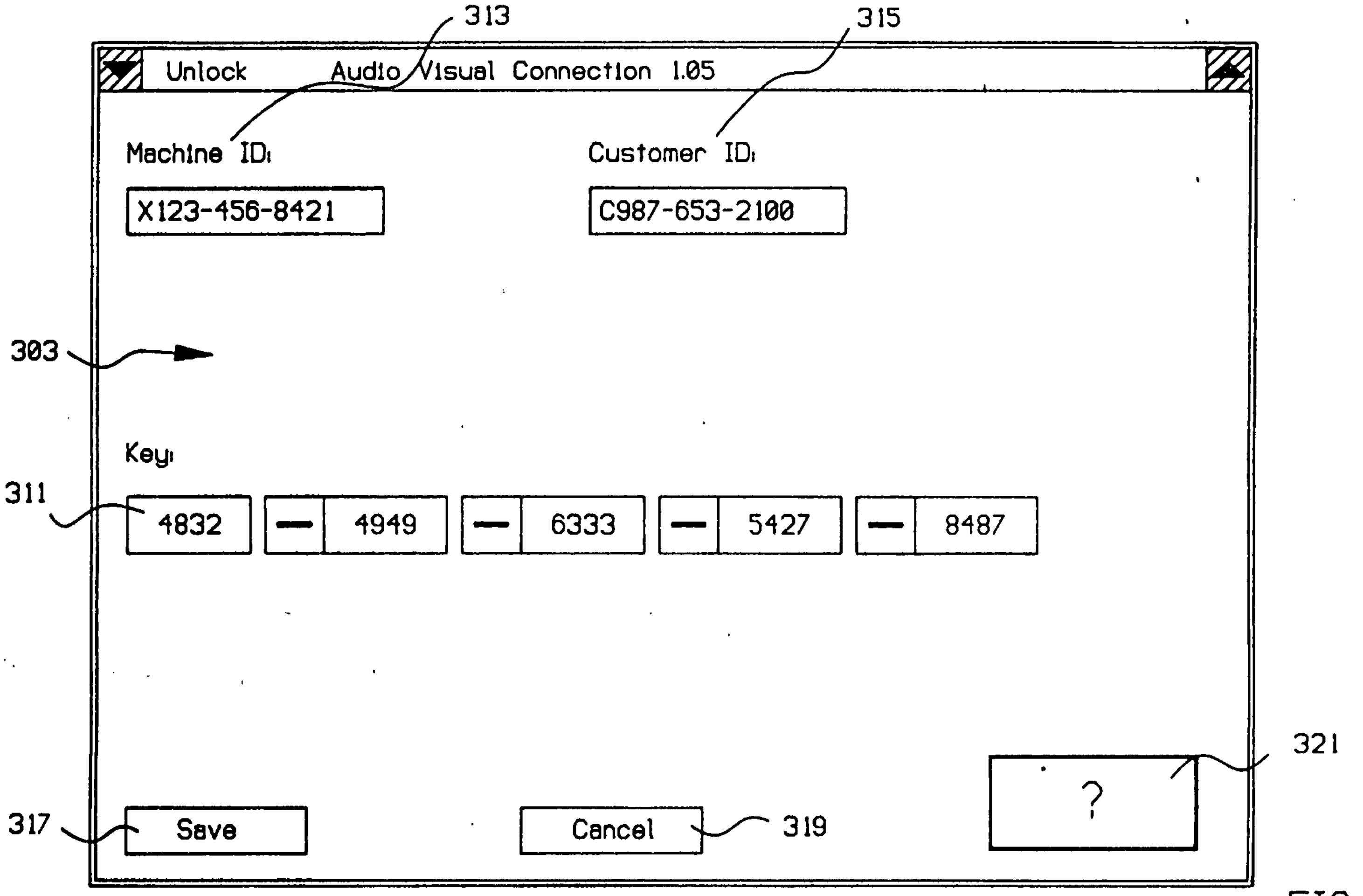
Key: 1234-1234-1234-1234-1234

Customer ID: C123-456-7890

Machine ID: X123-456-7890

301
307
309
305
311
313
315
Save
317
Close
319
?
321

FIG. 10A

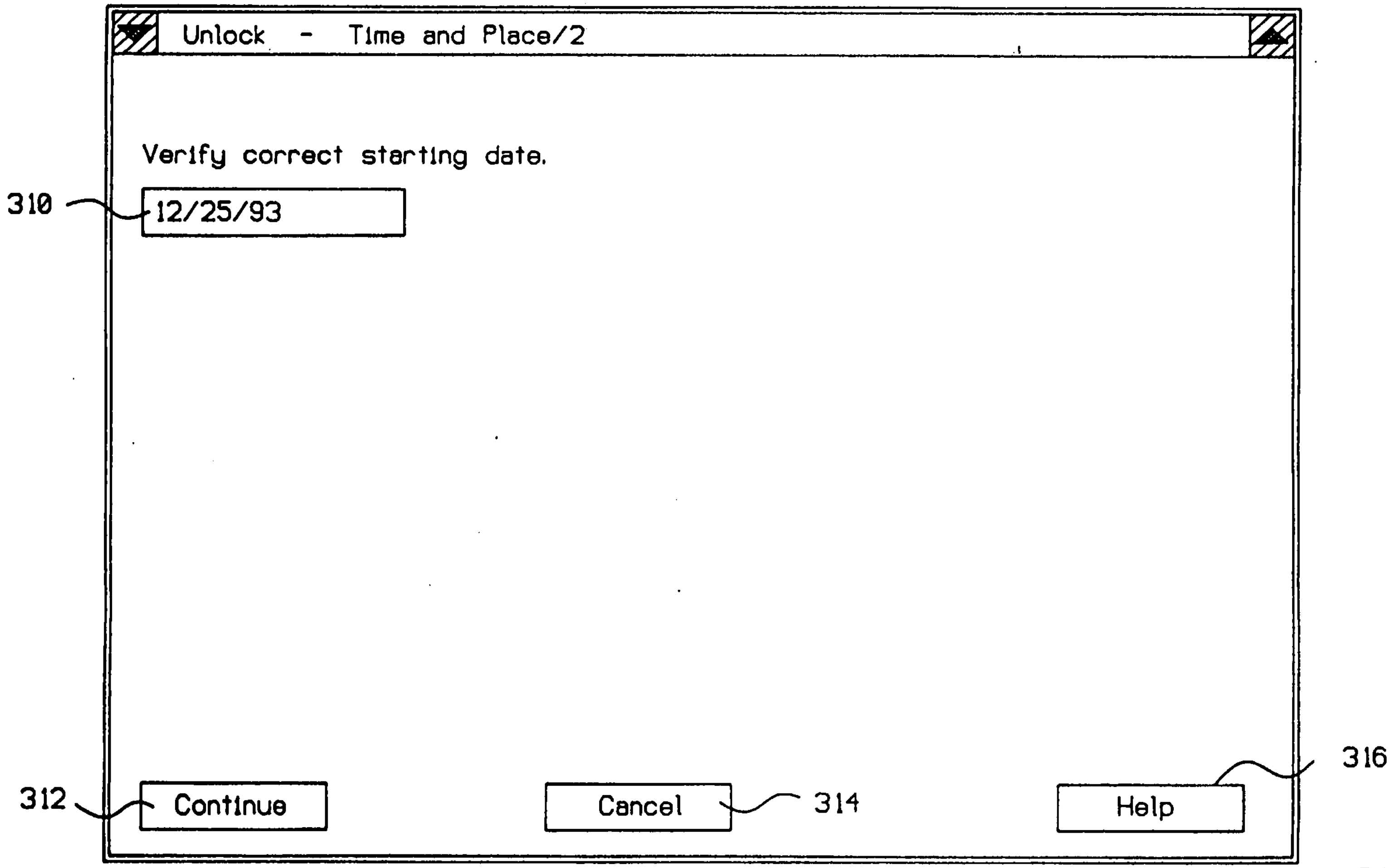

313
315
Unlock
Audio Visual Connection 1.05
303
311
317
Save
Cancel
319
321
?

Machine ID: X123-456-8421

Customer ID: C987-653-2100

Key: 4832 — 4949 — 6333 — 5427 — 8487

FIG. 10B


Unlock - Time and Place/2
Continue
Cancel
Help
310
312
314
316

Verify correct starting date.

12/25/93

FIG. 11

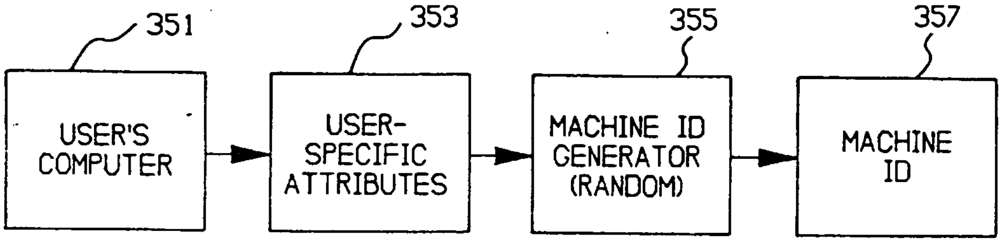

351
353
355
357
USER'S COMPUTER
USER-SPECIFIC ATTRIBUTES
MACHINE ID GENERATOR (RANDOM)
MACHINE ID

FIG. 12

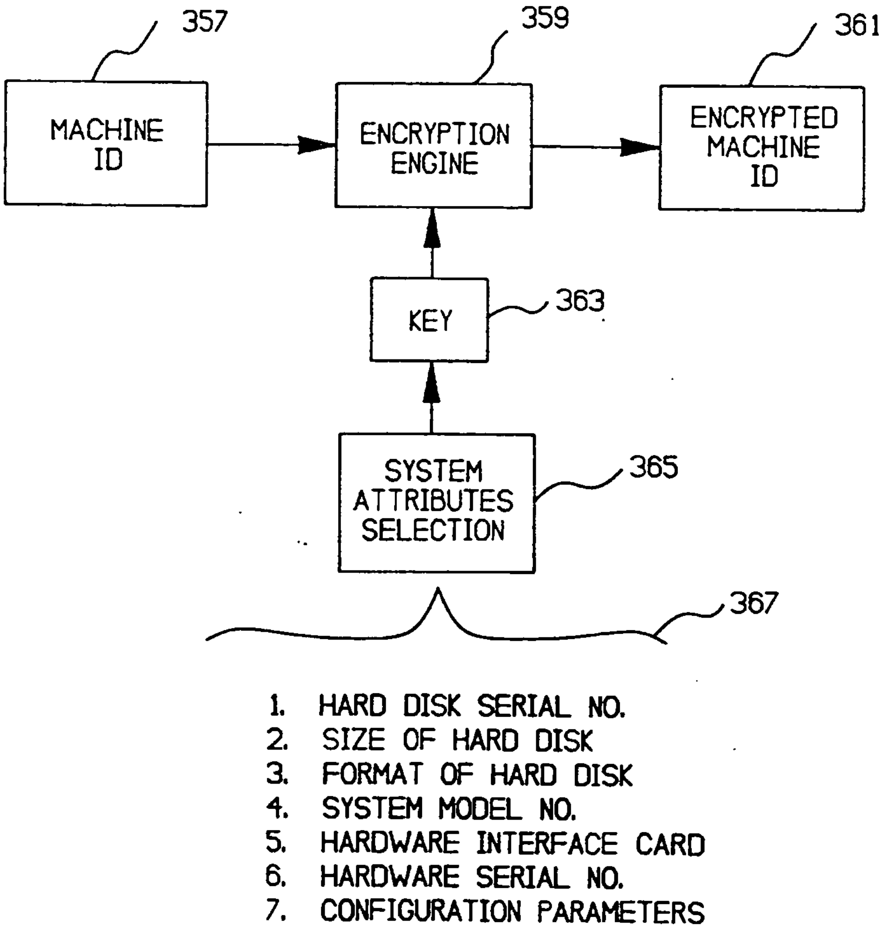
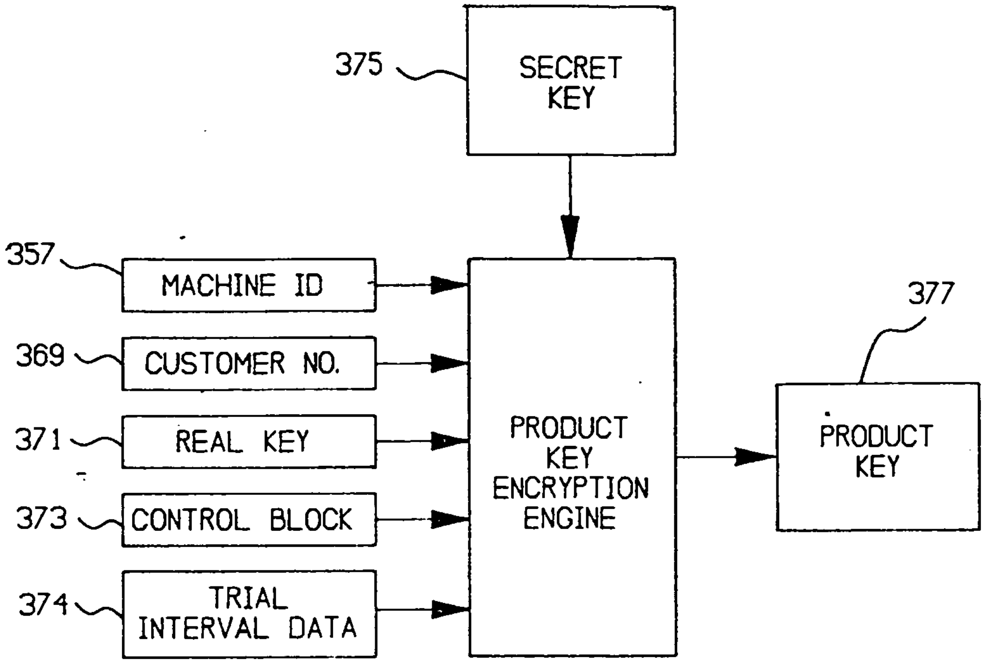
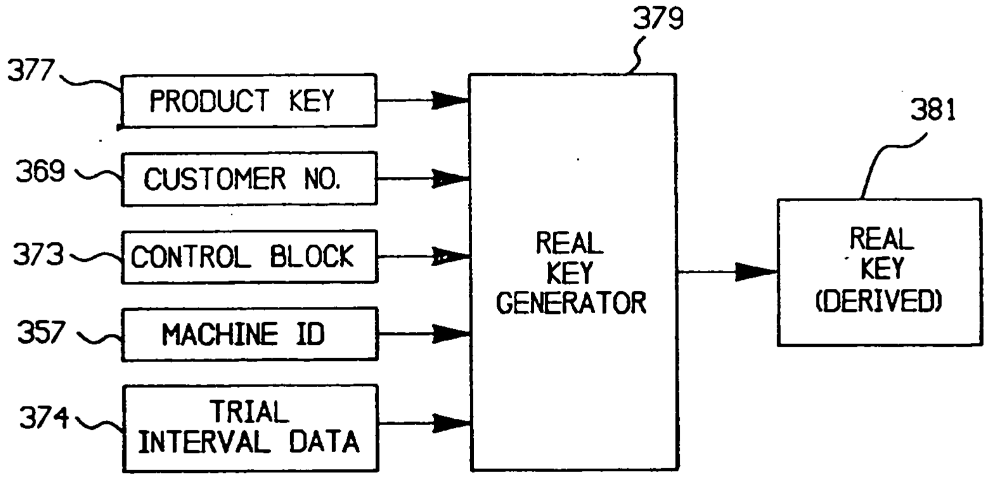

357
359
361
MACHINE ID
ENCRYPTION ENGINE
ENCRYPTED MACHINE ID
KEY
363
SYSTEM ATTRIBUTES SELECTION
365
367
1. HARD DISK SERIAL NO.
2. SIZE OF HARD DISK
3. FORMAT OF HARD DISK
4. SYSTEM MODEL NO.
5. HARDWARE INTERFACE CARD
6. HARDWARE SERIAL NO.
7. CONFIGURATION PARAMETERS

FIG. 13


375
SECRET KEY
357
MACHINE ID
369
CUSTOMER NO.
371
REAL KEY
373
CONTROL BLOCK
374
TRIAL INTERVAL DATA
PRODUCT KEY ENCRYPTION ENGINE
377
PRODUCT KEY

GENERATION OF PRODUCT KEY
FIG. 14


379
377
PRODUCT KEY
369
CUSTOMER NO.
373
CONTROL BLOCK
357
MACHINE ID
374
TRIAL INTERVAL DATA
REAL KEY GENERATOR
381
REAL KEY (DERIVED)

VALIDATION OF PRODUCT KEY
FIG. 15

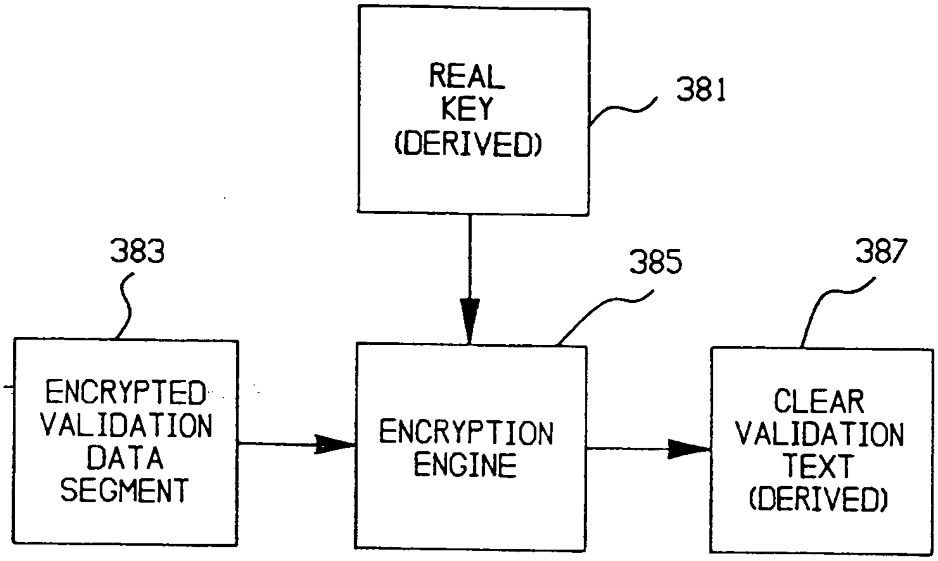

REAL KEY (DERIVED)
381
383
385
387
ENCRYPTED VALIDATION DATA SEGMENT
ENCRYPTION ENGINE
CLEAR VALIDATION TEXT (DERIVED)

FIG. 16

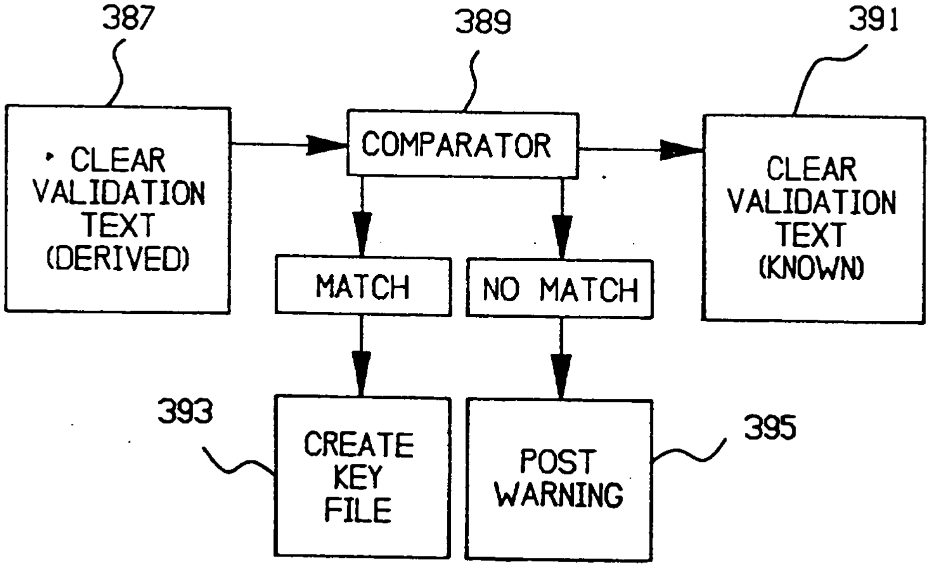

387
389
391
CLEAR VALIDATION TEXT (DERIVED)
COMPARATOR
CLEAR VALIDATION TEXT (KNOWN)
MATCH
NO MATCH
393
CREATE KEY FILE
POST WARNING
395

FIG. 17

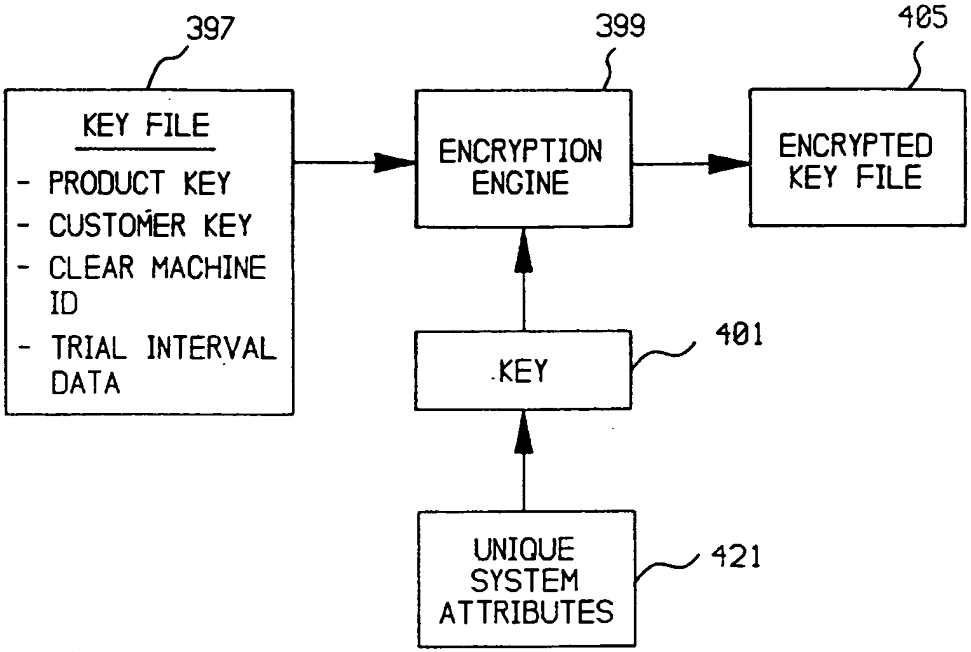

397
399
405
KEY FILE
- PRODUCT KEY
- CUSTOMER KEY
- CLEAR MACHINE ID
- TRIAL INTERVAL DATA
ENCRYPTION ENGINE
ENCRYPTED KEY FILE
KEY
401
UNIQUE SYSTEM ATTRIBUTES
421

FIG. 18

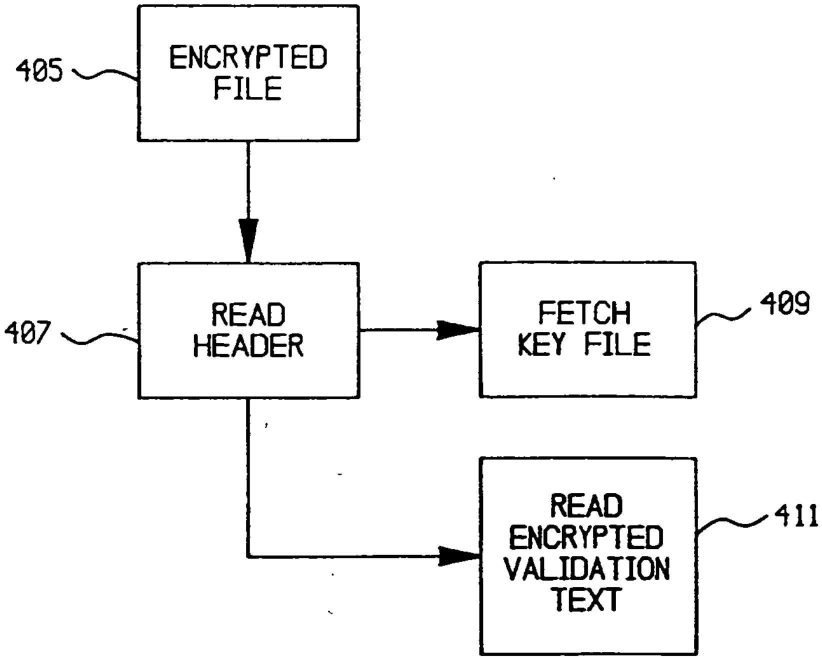

405
ENCRYPTED FILE
407
READ HEADER
FETCH KEY FILE
409
READ ENCRYPTED VALIDATION TEXT
411

FIG. 19

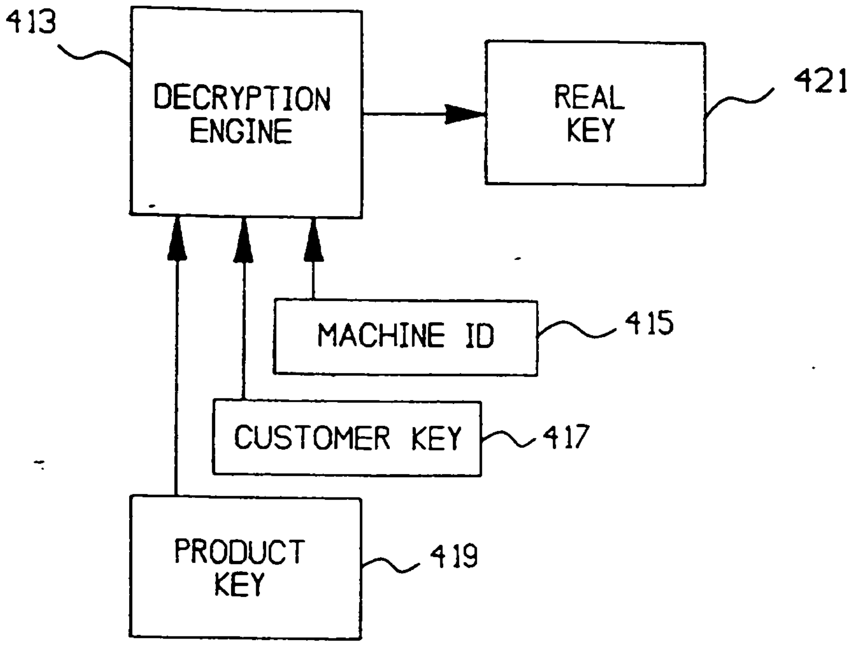

413
DECRYPTION ENGINE
REAL KEY
421
MACHINE ID
415
CUSTOMER KEY
417
PRODUCT KEY
419

FIG. 20

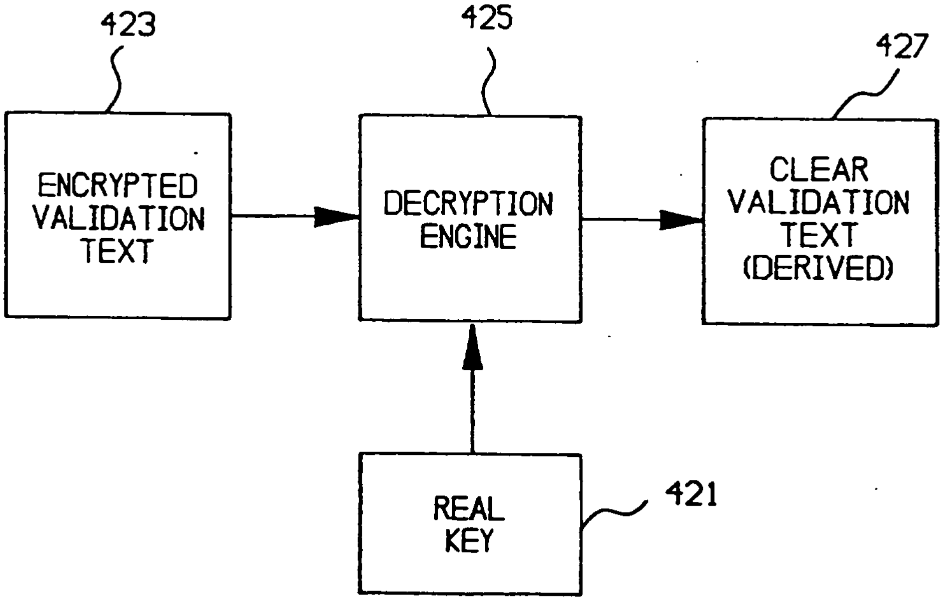

423
425
427
ENCRYPTED VALIDATION TEXT
DECRYPTION ENGINE
CLEAR VALIDATION TEXT (DERIVED)
REAL KEY
421

FIG. 21

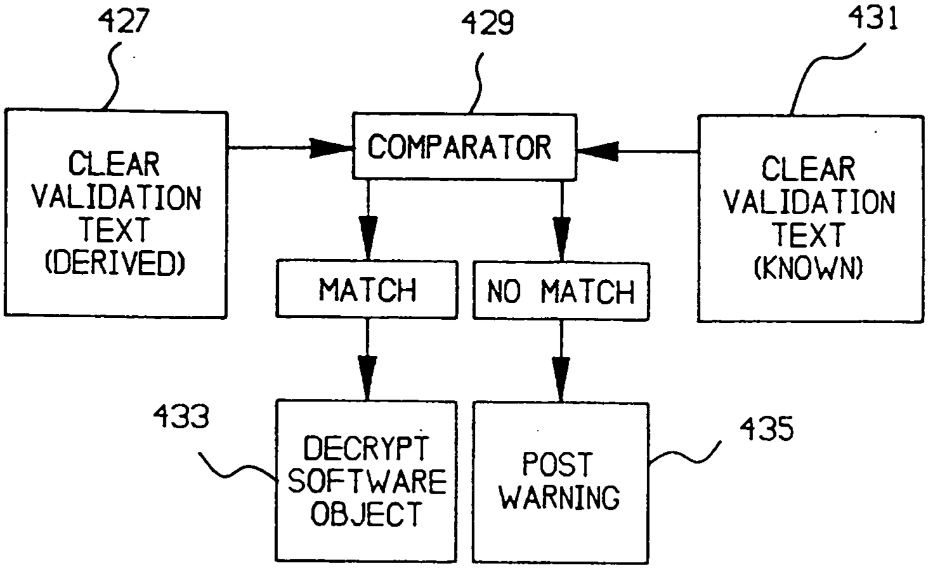

427
429
431
CLEAR VALIDATION TEXT (DERIVED)
COMPARATOR
CLEAR VALIDATION TEXT (KNOWN)
MATCH
NO MATCH
433
DECRYPT SOFTWARE OBJECT
POST WARNING
435

FIG. 22

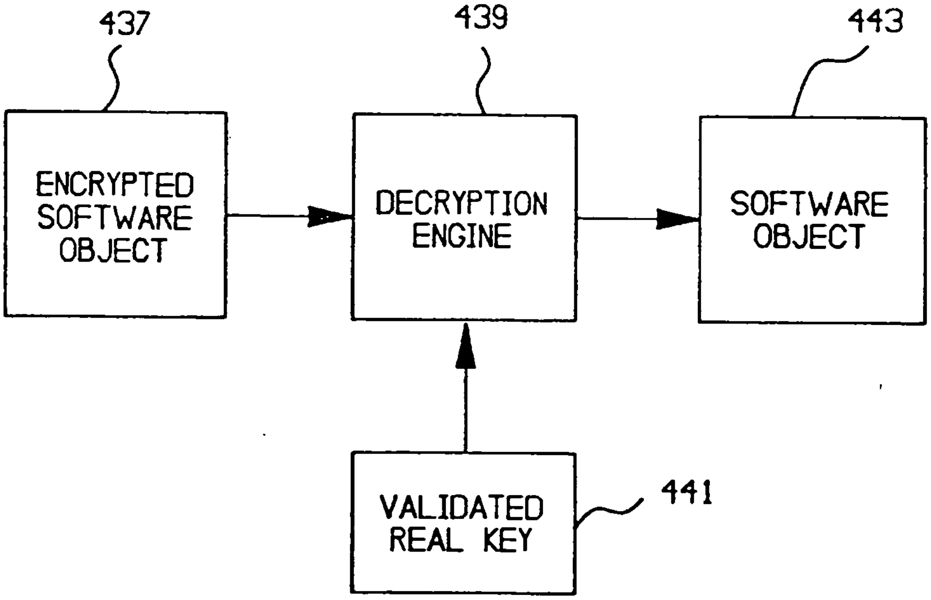
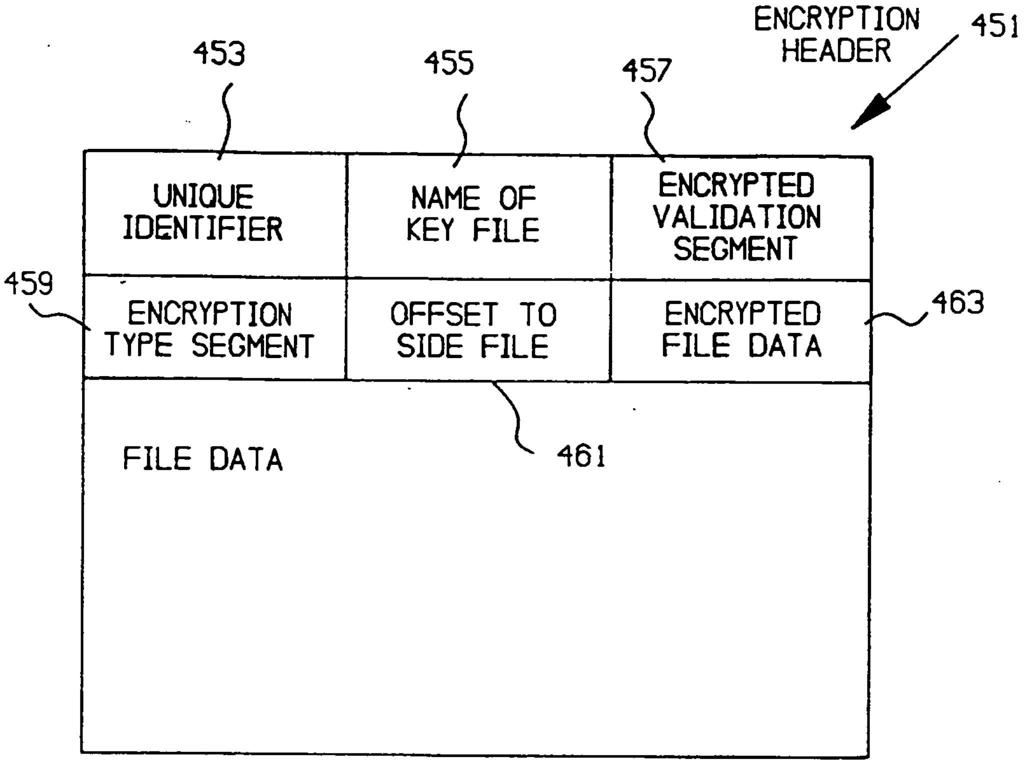

437
439
443
ENCRYPTED SOFTWARE OBJECT
DECRYPTION ENGINE
SOFTWARE OBJECT
VALIDATED REAL KEY
441

FIG. 23


ENCRYPTION HEADER
451
453
455
457
UNIQUE IDENTIFIER
NAME OF KEY FILE
ENCRYPTED VALIDATION SEGMENT
459
ENCRYPTION TYPE SEGMENT
OFFSET TO SIDE FILE
ENCRYPTED FILE DATA
463
FILE DATA
461

ENCRYPTED FILE

FIG. 24

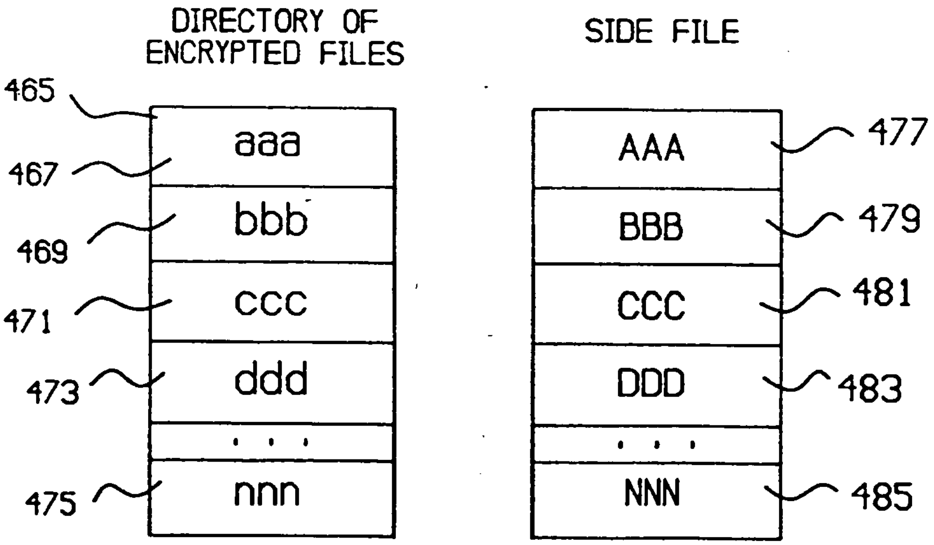

DIRECTORY OF ENCRYPTED FILES
SIDE FILE
465
467
aaa
AAA
477
469
bbb
BBB
479
471
ccc
CCC
481
473
ddd
DDD
483
. . .
. . .
475
nnn
NNN
485

FIG. 25

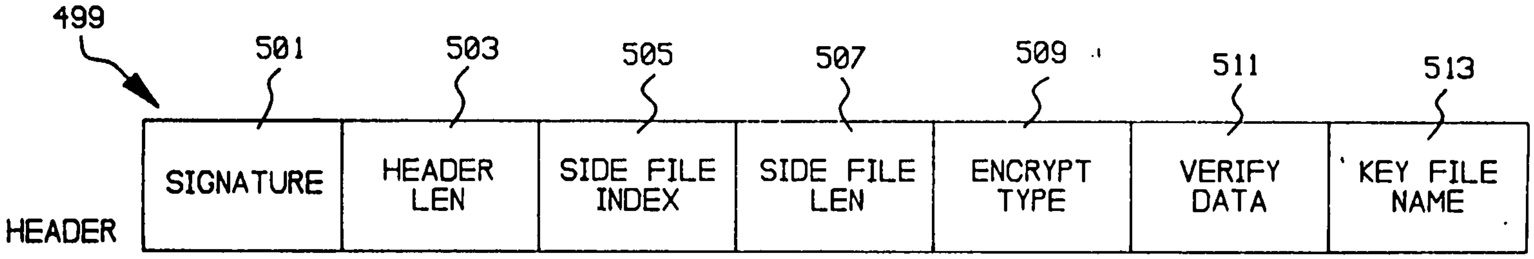

499
HEADER
501
SIGNATURE
503
HEADER LEN
505
SIDE FILE INDEX
507
SIDE FILE LEN
509
ENCRYPT TYPE
511
VERIFY DATA
513
KEY FILE NAME

FIG. 26

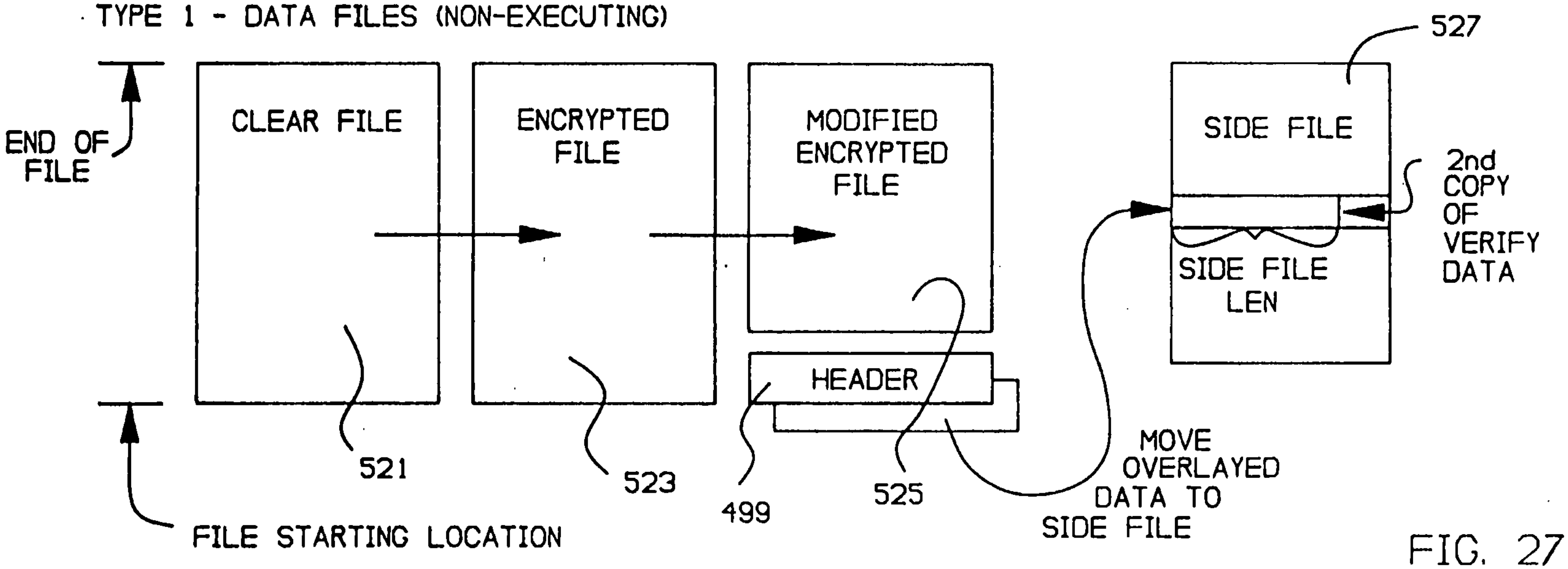

TYPE 1 - DATA FILES (NON-EXECUTING)
END OF FILE
FILE STARTING LOCATION
CLEAR FILE
521
ENCRYPTED FILE
523
MODIFIED ENCRYPTED FILE
HEADER
499
525
MOVE OVERLAYED DATA TO SIDE FILE
527
SIDE FILE
SIDE FILE LEN
2nd COPY OF VERIFY DATA

FIG. 27

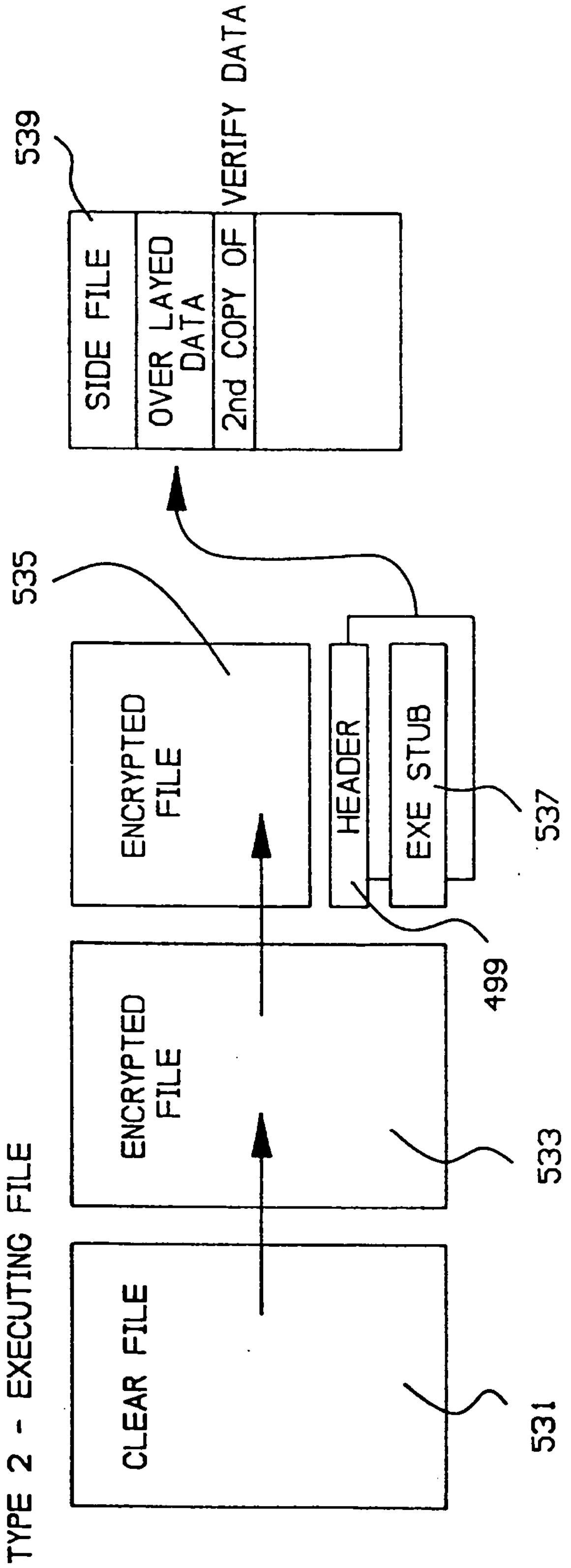
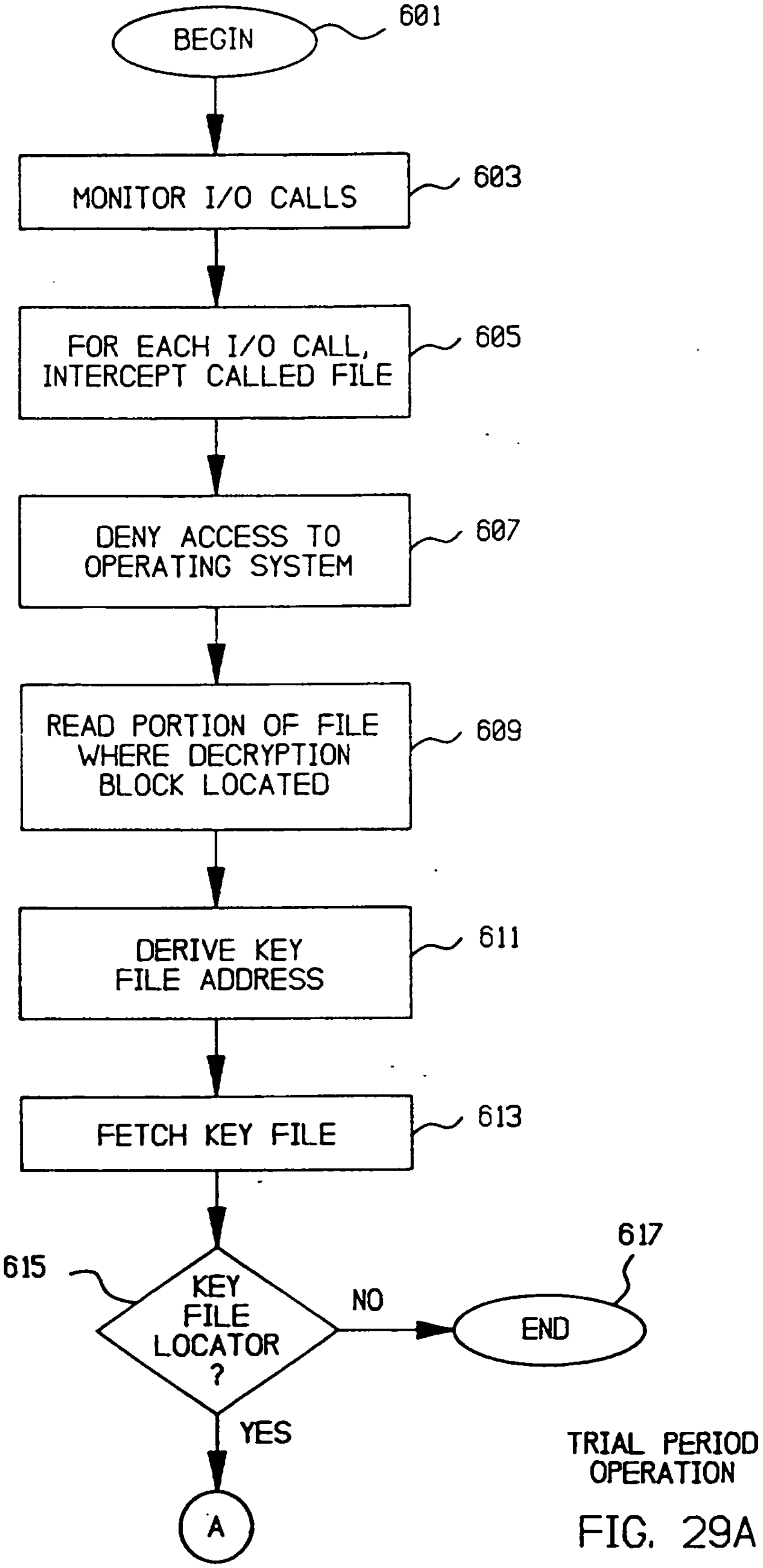
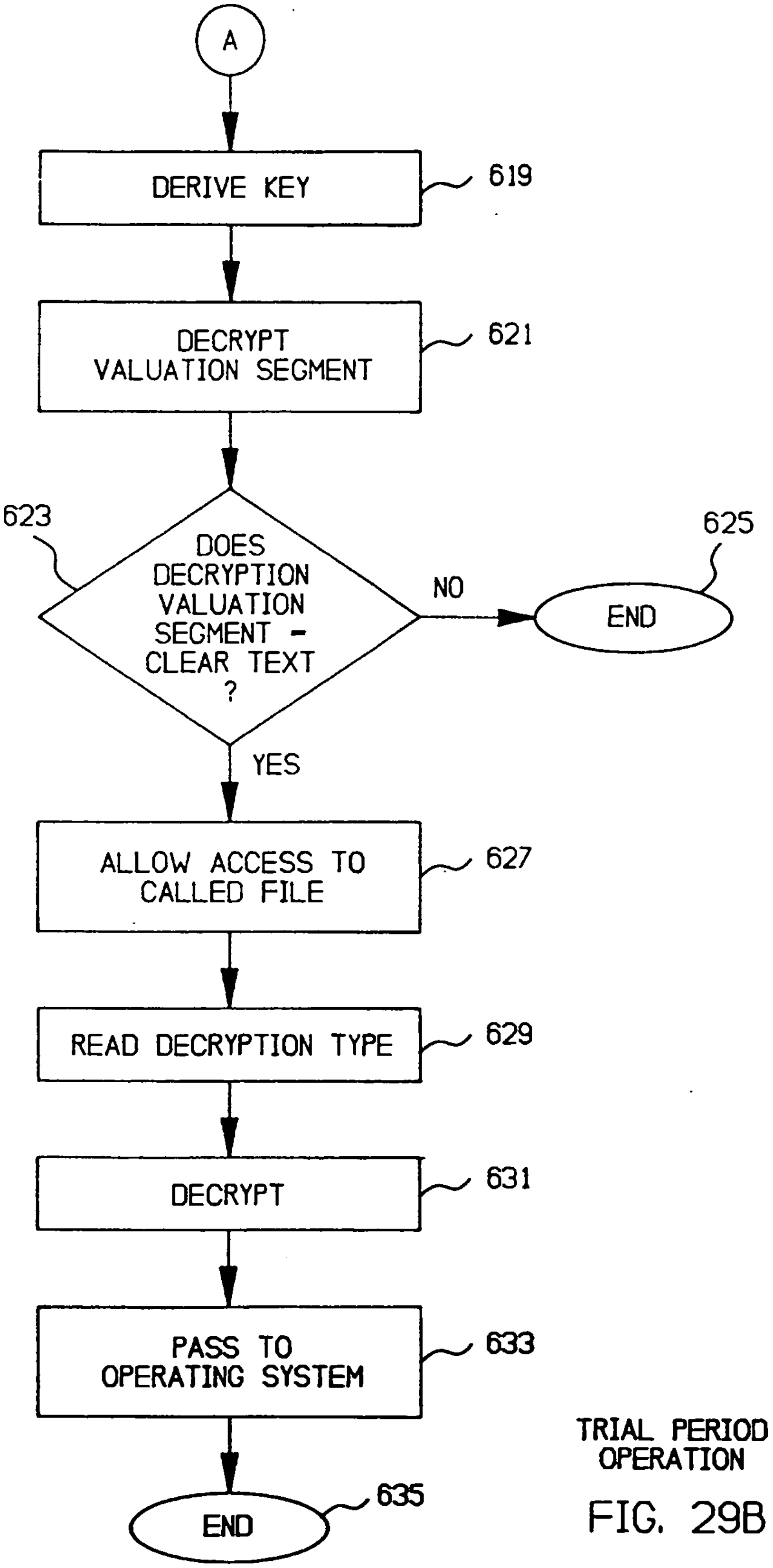

TYPE 2 - EXECUTING FILE
CLEAR FILE
531
ENCRYPTED FILE
533
ENCRYPTED FILE
535
HEADER
499
EXE STUB
537
SIDE FILE
OVER LAYED DATA
2nd COPY OF VERIFY DATA
539

FIG. 28


BEGIN
601
MONITOR I/O CALLS
603
FOR EACH I/O CALL, INTERCEPT CALLED FILE
605
DENY ACCESS TO OPERATING SYSTEM
607
READ PORTION OF FILE WHERE DECRYPTION BLOCK LOCATED
609
DERIVE KEY FILE ADDRESS
611
FETCH KEY FILE
613
615
KEY FILE LOCATOR ?
NO
END
617
YES
A

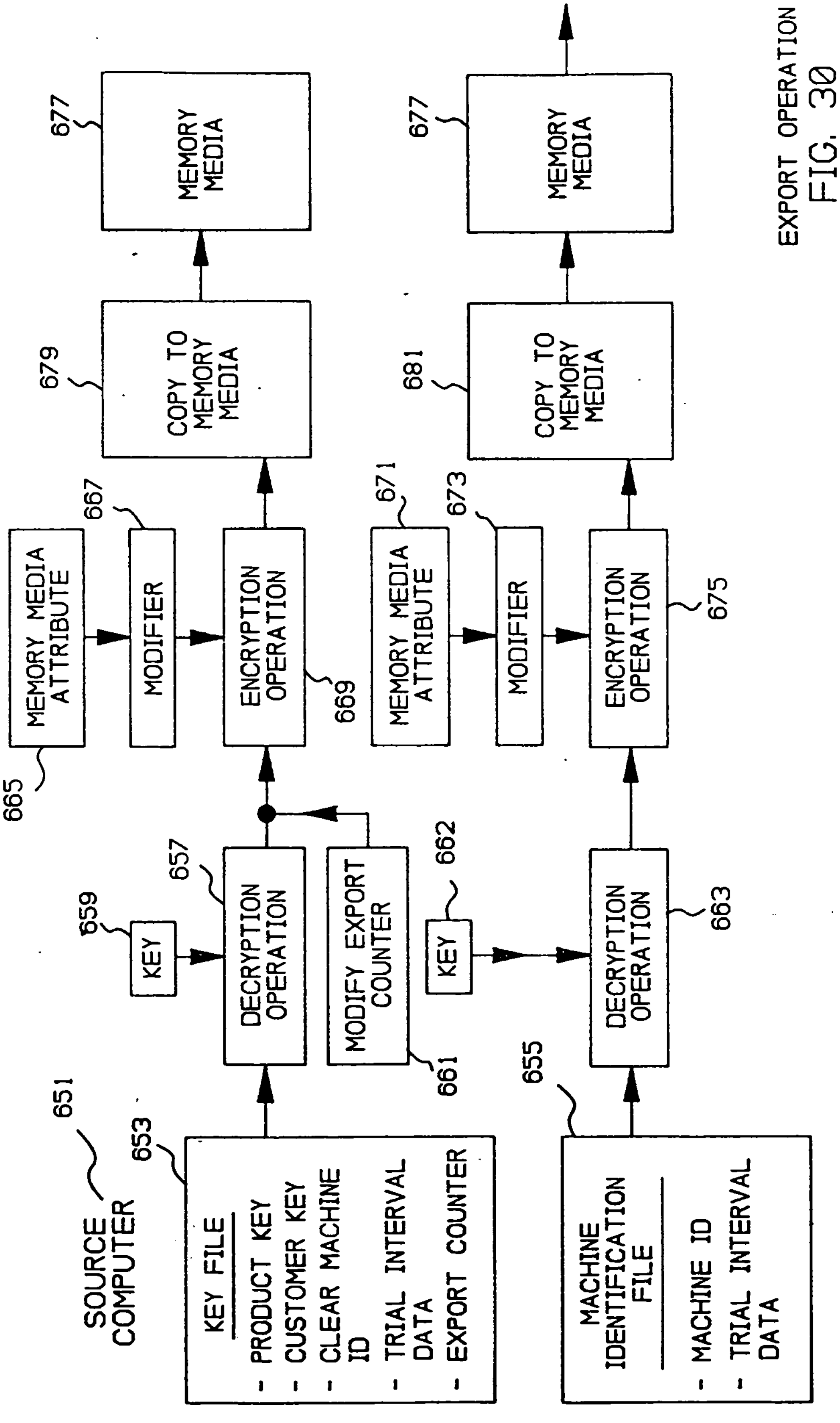
TRIAL PERIOD OPERATION

FIG. 29A

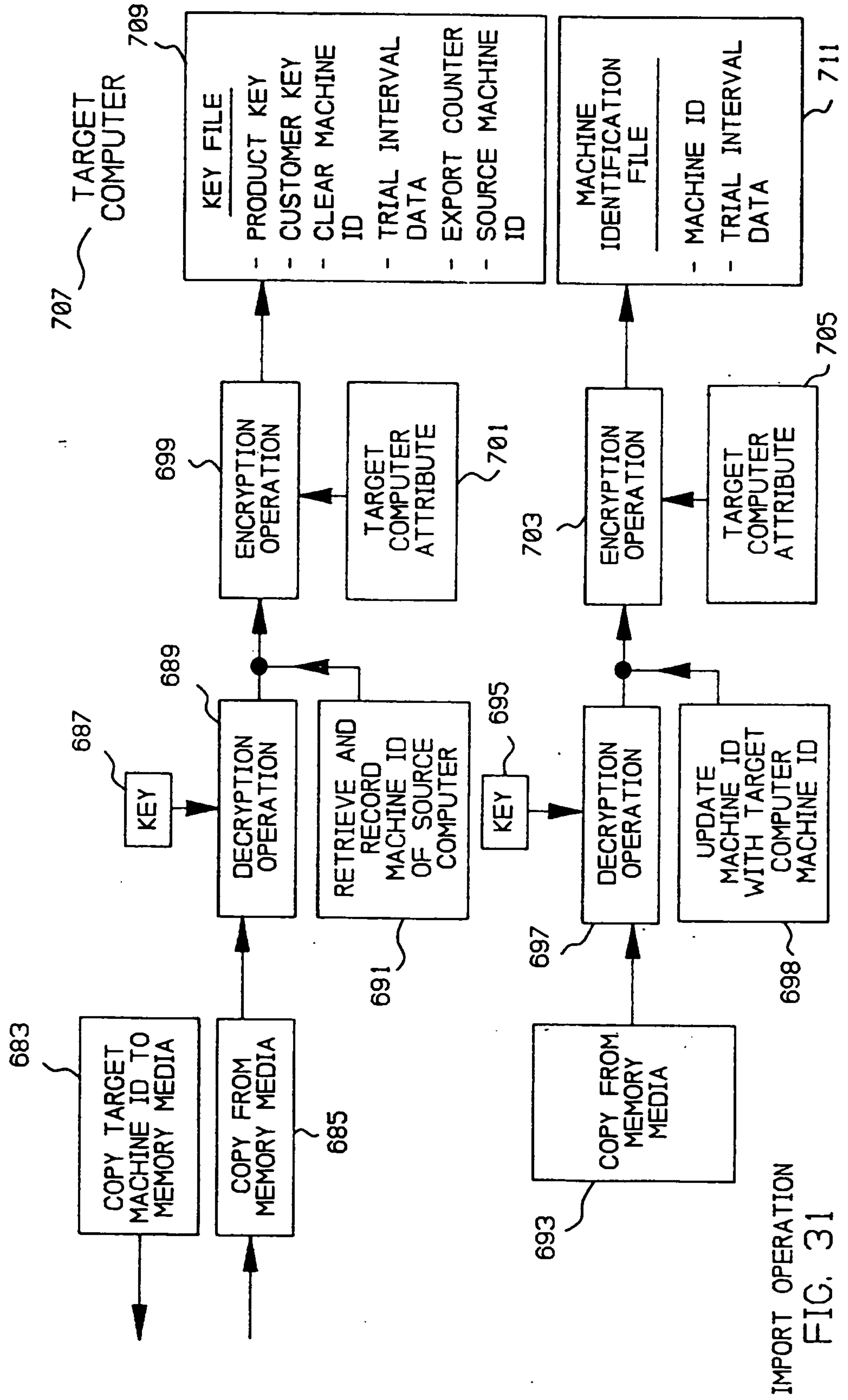

A
DERIVE KEY
619
DECRYPT VALUATION SEGMENT
621
623
DOES DECRYPTION VALUATION SEGMENT - CLEAR TEXT ?
NO
625
END
YES
ALLOW ACCESS TO CALLED FILE
627
READ DECRYPTION TYPE
629
DECRYPT
631
PASS TO OPERATING SYSTEM
633
END
635

TRIAL PERIOD OPERATION

FIG. 29B


SOURCE COMPUTER
651
653
KEY FILE
- PRODUCT KEY
- CUSTOMER KEY
- CLEAR MACHINE ID
- TRIAL INTERVAL DATA
- EXPORT COUNTER
659
KEY
657
DECRYPTION OPERATION
MODIFY EXPORT COUNTER
661
665
MEMORY MEDIA ATTRIBUTE
667
MODIFIER
ENCRYPTION OPERATION
669
679
COPY TO MEMORY MEDIA
677
MEMORY MEDIA
655
MACHINE IDENTIFICATION FILE
- MACHINE ID
- TRIAL INTERVAL DATA
662
KEY
DECRYPTION OPERATION
663
671
MEMORY MEDIA ATTRIBUTE
673
MODIFIER
ENCRYPTION OPERATION
675
681
COPY TO MEMORY MEDIA
677
MEMORY MEDIA

EXPORT OPERATION
FIG. 30


683
COPY TARGET MACHINE ID TO MEMORY MEDIA
COPY FROM MEMORY MEDIA
685
687
KEY
689
DECRYPTION OPERATION
RETRIEVE AND RECORD MACHINE ID OF SOURCE COMPUTER
691
699
ENCRYPTION OPERATION
TARGET COMPUTER ATTRIBUTE
701
707
TARGET COMPUTER
709
KEY FILE
- PRODUCT KEY
- CUSTOMER KEY
- CLEAR MACHINE ID
- TRIAL INTERVAL DATA
- EXPORT COUNTER
- SOURCE MACHINE ID
693
COPY FROM MEMORY MEDIA
695
KEY
697
DECRYPTION OPERATION
UPDATE MACHINE ID WITH TARGET COMPUTER MACHINE ID
698
703
ENCRYPTION OPERATION
TARGET COMPUTER ATTRIBUTE
705
MACHINE IDENTIFICATION FILE
- MACHINE ID
- TRIAL INTERVAL DATA
711

IMPORT OPERATION
FIG. 31

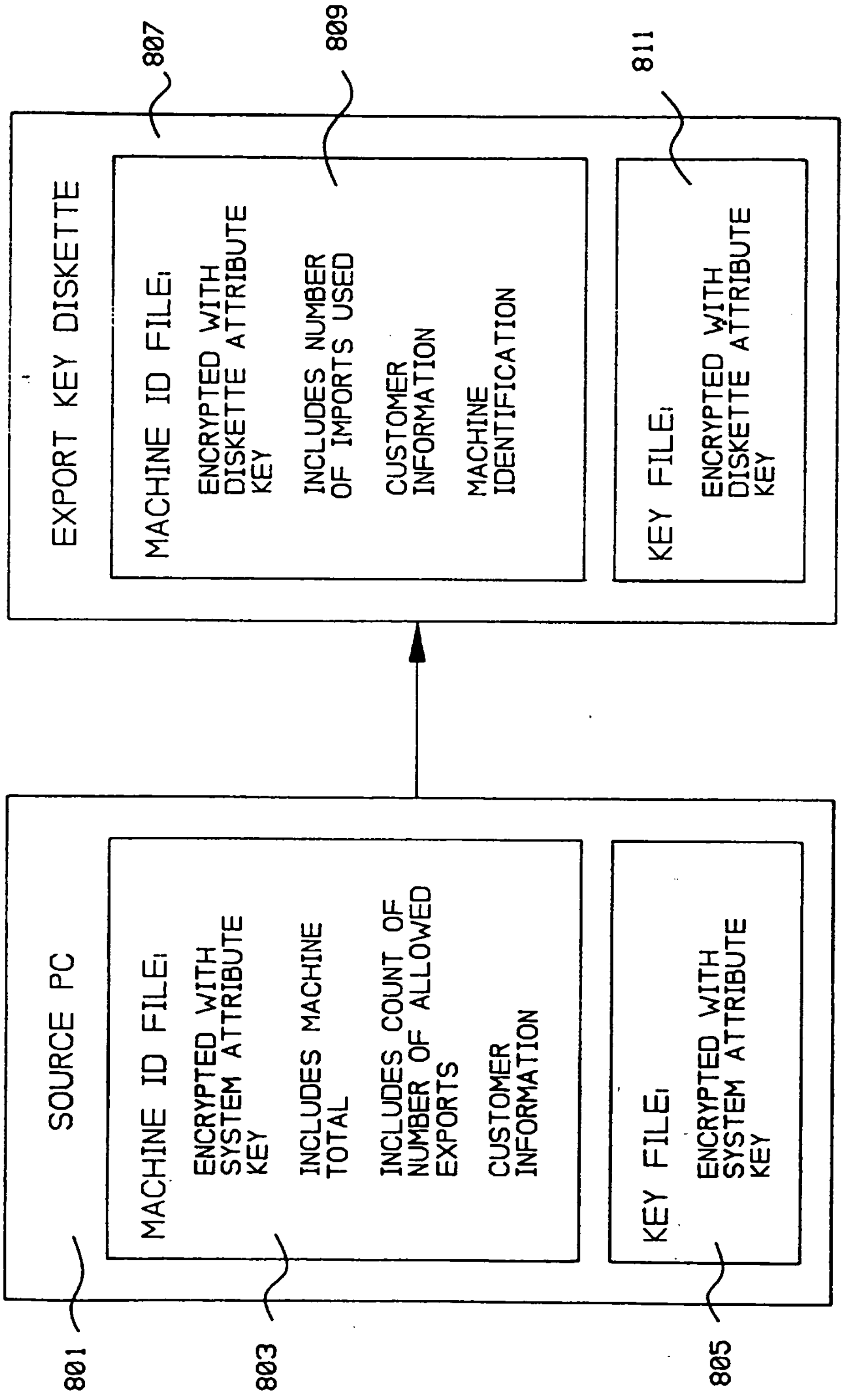

SOURCE PC
MACHINE ID FILE:
ENCRYPTED WITH SYSTEM ATTRIBUTE KEY
INCLUDES MACHINE TOTAL
INCLUDES COUNT OF NUMBER OF ALLOWED EXPORTS
CUSTOMER INFORMATION
KEY FILE:
ENCRYPTED WITH SYSTEM ATTRIBUTE KEY
801
803
805
EXPORT KEY DISKETTE
MACHINE ID FILE:
ENCRYPTED WITH DISKETTE ATTRIBUTE KEY
INCLUDES NUMBER OF IMPORTS USED
CUSTOMER INFORMATION
MACHINE IDENTIFICATION
KEY FILE:
ENCRYPTED WITH DISKETTE ATTRIBUTE KEY
807
809
811

FIG. 32

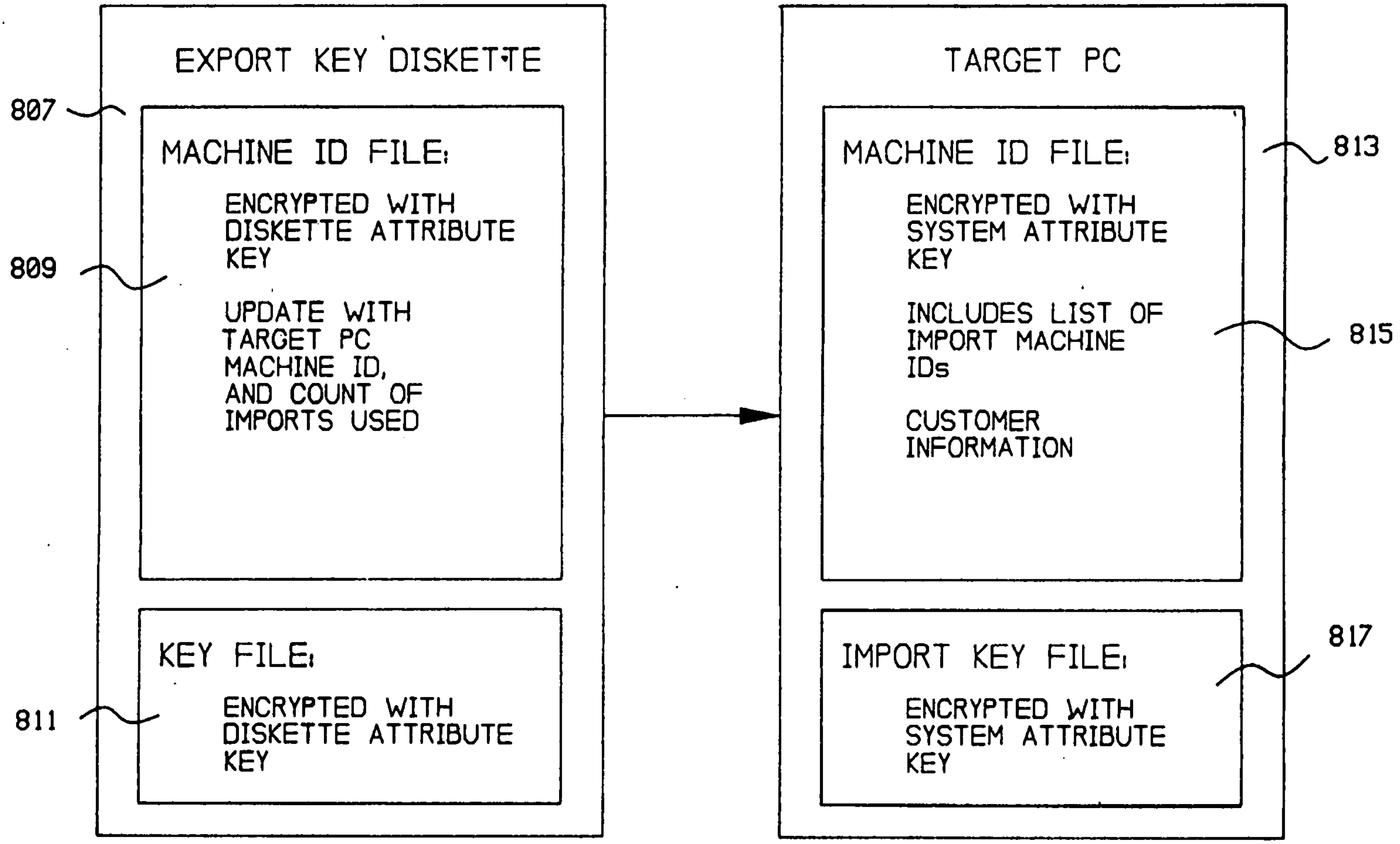
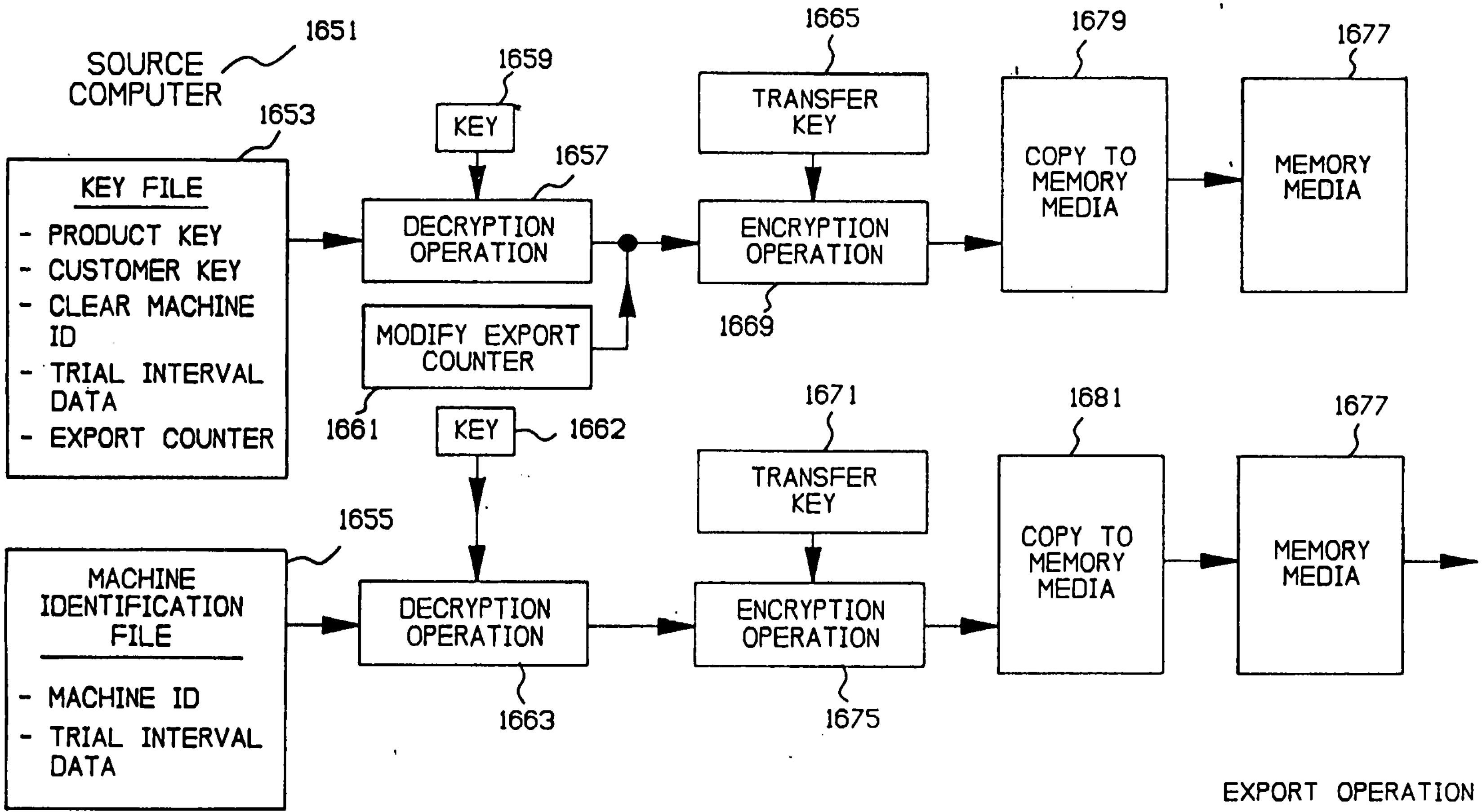
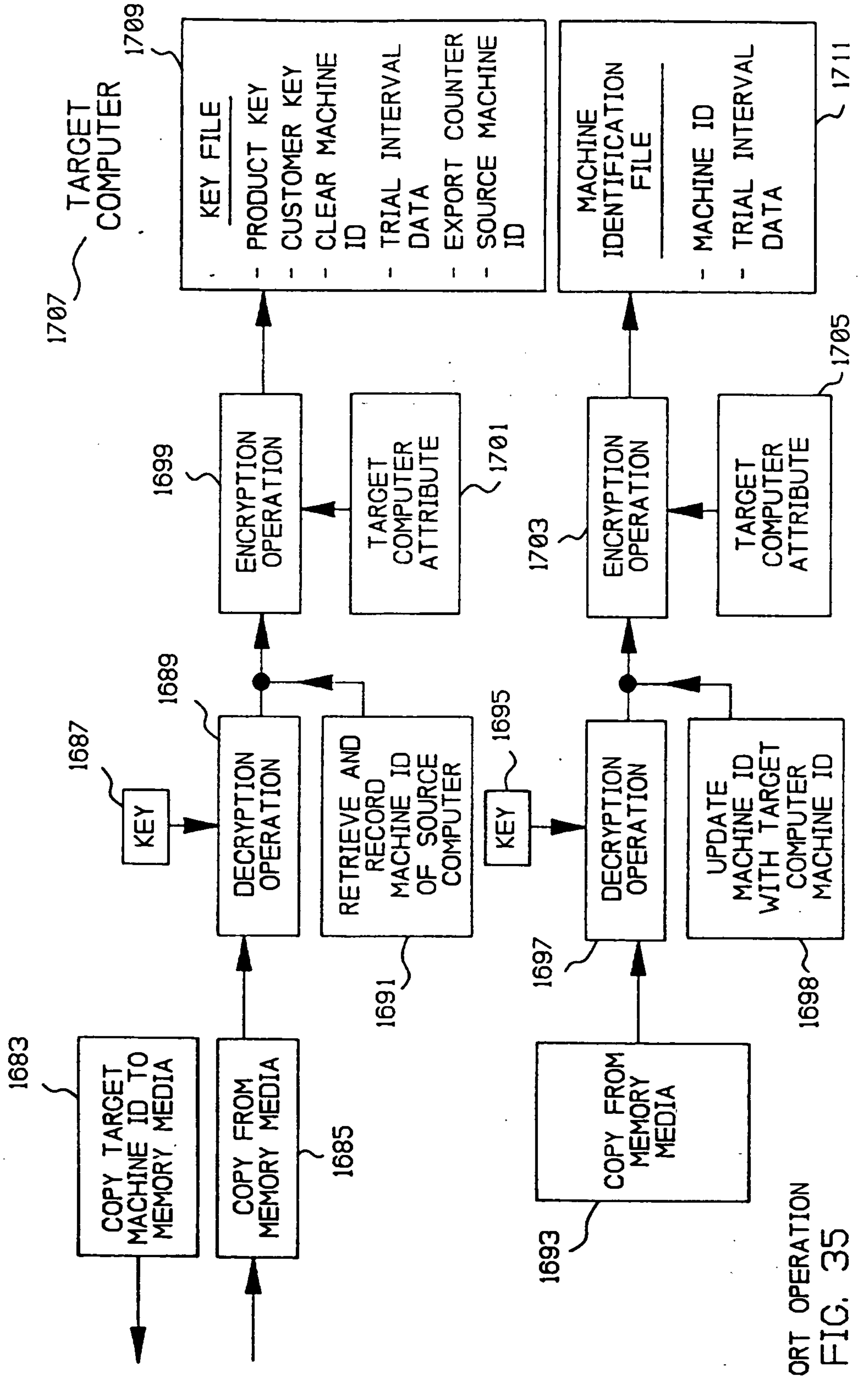

EXPORT KEY DISKETTE
807
MACHINE ID FILE:
ENCRYPTED WITH DISKETTE ATTRIBUTE KEY
809
UPDATE WITH TARGET PC MACHINE ID, AND COUNT OF IMPORTS USED
KEY FILE:
811
ENCRYPTED WITH DISKETTE ATTRIBUTE KEY
TARGET PC
MACHINE ID FILE:
813
ENCRYPTED WITH SYSTEM ATTRIBUTE KEY
INCLUDES LIST OF IMPORT MACHINE IDs
815
CUSTOMER INFORMATION
IMPORT KEY FILE:
817
ENCRYPTED WITH SYSTEM ATTRIBUTE KEY

FIG. 33


SOURCE COMPUTER
1651
1653
KEY FILE
- PRODUCT KEY
- CUSTOMER KEY
- CLEAR MACHINE ID
- TRIAL INTERVAL DATA
- EXPORT COUNTER
1659
KEY
1657
DECRYPTION OPERATION
MODIFY EXPORT COUNTER
1661
KEY
1662
1665
TRANSFER KEY
ENCRYPTION OPERATION
1669
1679
COPY TO MEMORY MEDIA
1677
MEMORY MEDIA
1655
MACHINE IDENTIFICATION FILE
- MACHINE ID
- TRIAL INTERVAL DATA
DECRYPTION OPERATION
1663
1671
TRANSFER KEY
ENCRYPTION OPERATION
1675
1681
COPY TO MEMORY MEDIA
1677
MEMORY MEDIA

EXPORT OPERATION
FIG. 34


1683
COPY TARGET MACHINE ID TO MEMORY MEDIA
COPY FROM MEMORY MEDIA
1685
1687
KEY
1689
DECRYPTION OPERATION
RETRIEVE AND RECORD MACHINE ID OF SOURCE COMPUTER
1691
1699
ENCRYPTION OPERATION
TARGET COMPUTER ATTRIBUTE
1701
1707
TARGET COMPUTER
1709
KEY FILE
- PRODUCT KEY
- CUSTOMER KEY
- CLEAR MACHINE ID
- TRIAL INTERVAL DATA
- EXPORT COUNTER
- SOURCE MACHINE ID
1693
COPY FROM MEMORY MEDIA
1695
KEY
1697
DECRYPTION OPERATION
UPDATE MACHINE ID WITH TARGET COMPUTER MACHINE ID
1698
1703
ENCRYPTION OPERATION
TARGET COMPUTER ATTRIBUTE
1705
MACHINE IDENTIFICATION FILE
- MACHINE ID
- TRIAL INTERVAL DATA
1711

IMPORT OPERATION
FIG. 35

European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 95 10 5400

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	WO-A-94 07204 (UNILOC) * abstract; figures 41,2,8 * * page 6, line 11 - page 9, line 5 * * page 10, line 3 - line 10 * * page 12, line 7 - page 17, line 13 *	1,9	G06F1/00 G06F12/14
Y	---	2,4-8,10	
Y	GB-A-2 136 175 (ATALLA) * the whole document * ---	2	
Y	EP-A-0 268 139 (IBM) * column 1, line 1 - column 3, line 1 * * column 6, line 7 - column 7, line 50 * * column 9, line 20 - line 29 * * column 19, line 9 - line 50 * * column 21, line 6 - line 18 * * claims 2,9 *	4-8,10	
A	---	3	
A	EP-A-0 561 685 (FUJITSU) * the whole document * -----	9	

TECHNICAL FIELDS SEARCHED (Int.Cl.6)
G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	25 July 1995	Powell, D

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 715 243 A1

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
05.06.1996 Bulletin 1996/23

(51) Int Cl.^{6}: **G06F 1/00**, G06F 17/60

(21) Application number: **95308414.2**

(22) Date of filing: **23.11.1995**

(84) Designated Contracting States:
DE FR GB

(30) Priority: **23.11.1994 US 344773**

(71) Applicant: **XEROX CORPORATION**
Rochester New York 14644 (US)

(72) Inventors:
- **Stefik, Mark J.**
  Woodside, California 94062 (US)
- **Pirolli, Peter L.T.**
  El Cerrrito, California 94530 (US)
- **Merkle, Ralph C.**
  Sunnyvale, California 94087 (US)

(74) Representative: **Goode, Ian Roy et al**
Rank Xerox Ltd
Patent Department
Parkway
Marlow Buckinghamshire SL7 1YL (GB)

(54) System for controlling the distribution and use of digital works having a fee reporting mechanism

(57) A fee accounting mechanism for reporting fees associated with the distribution and use of digital works. Usage rights and fees are attached to digital works. The usage rights define how the digital work may be used or further distributed. Usage fees are specified as part of a usage right. The digital works and their usage rights and fees are stored in repositories (201). The repositories control access to the digital works. Upon determination that the exercise of a usage right requires a fee, the repository generates a fee reporting transaction (302). Fee reporting is done to a credit server (301). The credit server collects the fee information and periodically transmits it to a billing clearinghouse (303).

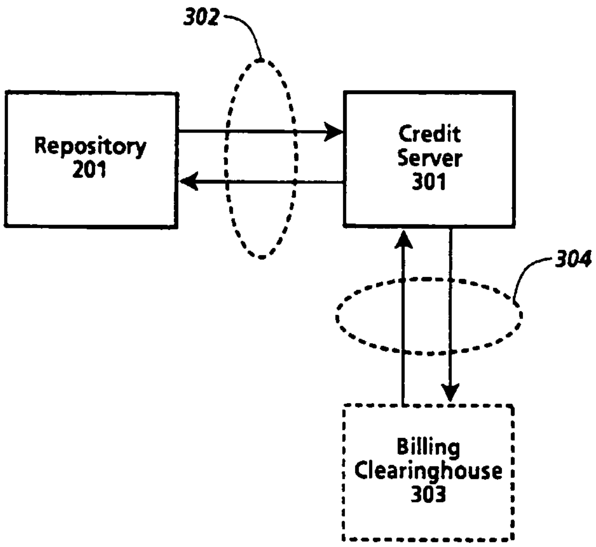


Fig. 3

EP 0 715 243 A1

Printed by Jouve, 75001 PARIS (FR)

Description

The present invention relates to the field of distribution and usage rights enforcement for digitally encoded works.

A fundamental issue facing the publishing and information industries as they consider electronic publishing is how to prevent the unauthorized and unaccounted distribution or usage of electronically published materials. Electronically published materials are typically distributed in a digital form and recreated on a computer based system having the capability to recreate the materials. Audio and video recordings, software, books and multimedia works are all being electronically published. Companies in these industries receive royalties for each accounted for delivery of the materials, e.g. the sale of an audio CD at a retail outlet. Any unaccounted distribution of a work results in an unpaid royalty (e.g. copying the audio recording CD to another digital medium.)

The ease in which electronically published works can be "perfectly" reproduced and distributed is a major concern. The transmission of digital works over networks is commonplace. One such widely used network is the Internet. The Internet is a widespread network facility by which computer users in many universities, corporations and government entities communicate and trade ideas and information. Computer bulletin boards found on the Internet and commercial networks such as CompuServ and Prodigy allow for the posting and retrieving of digital information. Information services such as Dialog and LEXIS/NEXIS provide databases of current information on a wide variety of topics. Another factor which will exacerbate the situation is the development and expansion of the National Information Infrastructure (the NII). It is anticipated that, as the NII grows, the transmission of digital works over networks will increase many times over. It would be desirable to utilize the NII for distribution of digital works without the fear of widespread unauthorized copying.

The most straightforward way to curb unaccounted distribution is to prevent unauthorized copying and transmission. For existing materials that are distributed in digital form, various safeguards are used. In the case of software, copy protection schemes which limit the number of copies that can be made or which corrupt the output when copying is detected have been employed. Another scheme causes software to become disabled after a predetermined period of time has lapsed. A technique used for workstation based software is to require that a special hardware device must be present on the workstation in order for the software to run, e.g., see US-A-4,932,054 entitled "Method and Apparatus for Protecting Computer Software Utilizing Coded Filter Network in Conjunction with an Active Coded Hardware Device." Such devices are provided with the software and are commonly referred to as dongles.

Yet another scheme is to distribute software, but which requires a "key" to enable its use. This is employed in distribution schemes where "demos" of the software are provided on a medium along with the entire product. The demos can be freely used, but in order to use the actual product, the key must be purchased. These schemes do not hinder copying of the software once the key is initially purchased.

It is an object of the present invention to provide an improved system and method for controlling the use and distribution of digital works.

The invention accordingly provides a system and method as claimed in the accompanying claims.

In a system for the control of distribution and use of digital works, a fee reporting mechanism for reporting fees associated with such distribution and use is disclosed. The system includes a means for attaching usage rights to a digital work. The usage rights define how the digital work may be used or further distributed by a possessor of the digital work. Usage fees are specified as part of a usage right. The ability to report usage fees may be a condition to the exercise of a usage right. Further, different fees may be assigned to different usage rights.

The present invention enables various usage fee scenarios to be used. Fees may be assessed on a per use basis, on a metered basis or based on a predetermined schedule. Fees may also be discounted on a predetermined schedule, or they can be marked-up a predetermined percentage (e.g. as a distributor fee). Fee reporting may also be deferred to a later time, to accommodate special deals, rebates or some other external information not yet available.

The present invention supports usage fees in an additive fashion. Usage fees may be reported for a composite digital work, i.e. a digital work comprised of a plurality of discrete digital works each having their own usage rights, and for distributors of digital works. Accordingly, fees to multiple revenue owners can be reported.

Usage fee reporting is done to a credit server. The credit server collects the fee information and periodically transmits it to a billing clearinghouse. Alternatively, the credit server may have a pre-allocated credit which is decremented as fees are incurred. In this alternative embodiment, the credit server would have to be periodically reallocated with credits to enable further use.

A system and method in accordance with the invention will now be described, by way of example, with reference to the accompanying drawings, in which:-

Figure 1 is a flowchart illustrating a simple instantiation of the operation of the currently preferred embodiment of the present invention.

Figure 2 is a block diagram illustrating the various repository types and the repository transaction flow between them in the currently preferred embodiment of the present invention.

Figure 3 is a block diagram of a repository coupled with a credit server in the currently preferred embodiment of the present invention.

Figures 4a and 4b are examples of rendering systems as may be utilized in the currently preferred embodiment of the present invention.

Figure 5 illustrates a contents file layout for a digital work as may be utilized in the currently preferred embodiment of the present invention.

Figure 6 illustrates a contents file layout for an individual digital work of the digital work of Figure 5 as may be utilized in the currently preferred embodiment of the present invention.

Figure 7 illustrates the components of a description block of the currently preferred embodiment of the present invention.

Figure 8 illustrates a description tree for the contents file layout of the digital work illustrated in Figure 5.

Figure 9 illustrates a portion of a description tree corresponding to the individual digital work illustrated in Figure 6.

Figure 10 illustrates a layout for the rights portion of a description block as may be utilized in the currently preferred embodiment of the present invention.

Figure 11 is a description tree wherein certain d-blocks have PRINT usage rights and is used to illustrate "strict" and "lenient" rules for resolving usage rights conflicts.

Figure 12 is a block diagram of the hardware components of a repository as are utilized in the currently preferred embodiment of the present invention.

Figure 13 is a block diagram of the functional (logical) components of a repository as are utilized in the currently preferred embodiment of the present invention.

Figure 14 is diagram illustrating the basic components of a usage right in the currently preferred embodiment of the present invention.

Figure 15 lists the usage rights grammar of the currently preferred embodiment of the present invention.

Figure 16 is a flowchart illustrating the steps of certificate delivery, hotlist checking and performance testing as performed in a registration transaction as may be performed in the currently preferred embodiment of the present invention.

Figure 17 is a flowchart illustrating the steps of session information exchange and clock synchronization as may be performed in the currently preferred embodiment of the present invention, after each repository in the registration transaction has successfully completed the steps described in Figure 16.

Figure 18 is a flowchart illustrating the basic flow for a usage transaction, including the common opening and closing step, as may be performed in the currently preferred embodiment of the present invention.

Figure 19 is a state diagram of server and client repositories in accordance with a transport protocol followed when moving a digital work from the server to the client repositories, as may be performed in the currently preferred embodiment of the present invention.

OVERVIEW

A system for controlling use and distribution of digital works is disclosed. The present invention is directed to supporting commercial transactions involving digital works.

Herein the terms "digital work", "work" and "content" refer to any work that has been reduced to a digital representation. This would include any audio, video, text, or multimedia work and any accompanying interpreter (e.g. software) that may be required for recreating the work. The term composite work refers to a digital work comprised of a collection of other digital works. The term "usage rights" or "rights" is a term which refers to rights granted to a recipient of a digital work. Generally, these rights define how a digital work can be used and if it can be further distributed. Each usage right may have one or more specified conditions which must be satisfied before the right may be exercised.

Figure 1 is a high level flowchart omitting various details but which demonstrates the basic operation of the present invention. Referring to Figure 1, a creator creates a digital work, step 101. The creator will then determine appropriate usage rights and fees, attach them to the digital work, and store them in Repository 1, step 102. The determination of appropriate usage rights and fees will depend on various economic factors. The digital work remains securely in Repository 1 until a request for access is received. The request for access begins with a session initiation by another repository. Here a Repository 2 initiates a session with Repository 1, step 103. As will be described in greater detail below, this session initiation includes steps which helps to insure that the respective repositories are trustworthy. Assuming that a session can be established, Repository 2 may then request access to the Digital Work for a stated purpose, step 104. The purpose may be, for example, to print the digital work or to obtain a copy of the digital work. The purpose will correspond to a specific usage right. In any event, Repository 1 checks the usage rights associated with the digital work to determine if the access to the digital work may be granted, step 105. The check of the usage rights essentially involves a determination of whether a right associated with the access request has been attached to the digital work and if all conditions associated with the right are satisfied. If the access is denied, repository 1 terminates the session with an error message, step 106. If access is granted, repository 1 transmits the digital work to repository

2, step 107. Once the digital work has been transmitted to repository 2, repository 1 and 2 each generate billing information for the access which is transmitted to a credit server, step 108. Such double billing reporting is done to insure against attempts to circumvent the billing process.

Figure 2 illustrates the basic interactions between repository types in the present invention. As will become apparent from Figure 2, the various repository types will serve different functions. It is fundamental that repositories will share a core set of functionality which will enable secure and trusted communications. Referring to Figure 2, a repository 201 represents the general instance of a repository. The repository 201 has two modes of operation; a server mode and a requester mode. When in the server mode, the repository will be receiving and processing access requests to digital works. When in the requester mode, the repository will be initiating requests to access digital works. Repository 201 is general in the sense that its primary purpose is as an exchange medium for digital works. During the course of operation, the repository 201 may communicate with a plurality of other repositories, namely authorization repository 202, rendering repository 203 and master repository 204. Communication between repositories occurs utilizing a repository transaction protocol 205.

Communication with an authorization repository 202 may occur when a digital work being accessed has a condition requiring an authorization. Conceptually, an authorization is a digital certificate such that possession of the certificate is required to gain access to the digital work. An authorization is itself a digital work that can be moved between repositories and subjected to fees and usage rights conditions. An authorization may be required by both repositories involved in an access to a digital work.

Communication with a rendering repository 203 occurs in connection with the rendering of a digital work. As will be described in greater detail below, a rendering repository is coupled with a rendering device (e.g. a printer device) to comprise a rendering system.

Communication with a master repository 205 occurs in connection with obtaining an identification certificate. Identification certificates are the means by which a repository is identified as "trustworthy". The use of identification certificates is described below with respect to the registration transaction.

Figure 3 illustrates the repository 201 coupled to a credit server 301. The credit server 301 is a device which accumulates billing information for the repository 201. The credit server 301 communicates with repository 201 via billing transactions 302 to record billing transactions. Billing transactions are reported to a billing clearinghouse 303 by the credit server 301 on a periodic basis. The credit server 301 communicates to the billing clearinghouse 303 via clearinghouse transactions 304. The clearinghouse transactions 304 enable a secure and encrypted transmission of information to the billing clearinghouse 303.

RENDERING SYSTEMS

A rendering system is generally defined as a system comprising a repository and a rendering device which can render a digital work into its desired form. Examples of a rendering system may be a computer system, a digital audio system, or a printer. A rendering system has the same security features as a repository. The coupling of a rendering repository with the rendering device may occur in a manner suitable for the type of rendering device.

Figure 4a illustrates a printer as an example of a rendering system. Referring to Figure 4, printer system 401 has contained therein a printer repository 402 and a print device 403. It should be noted that the the dashed line defining printer system 401 defines a secure system boundary. Communications within the boundary are assumed to be secure. Depending on the security level, the boundary also represents a barrier intended to provide physical integrity. The printer repository 402 is an instantiation of the rendering repository 205 of Figure 2. The printer repository 402 will in some instances contain an ephemeral copy of a digital work which remains until it is printed out by the print engine 403. In other instances, the printer repository 402 may contain digital works such as fonts, which will remain and can be billed based on use. This design assures that all communication lines between printers and printing devices are encrypted, unless they are within a physically secure boundary. This design feature eliminates a potential "fault" point through which the digital work could be improperly obtained. The printer device 403 represents the printer components used to create the printed output.

Also illustrated in Figure 4a is the repository 404. The repository 404 is coupled to the printer repository 402. The repository 404 represents an external repository which contains digital works.

Figure 4b is an example of a computer system as a rendering system. A computer system may constitute a "multi-function" device since it may execute digital works (e.g. software programs) and display digital works (e.g. a digitized photograph). Logically, each rendering device can be viewed as having its own repository, although only one physical repository is needed. Referring to Figure 4b, a computer system 410 has contained therein a display/execution repository 411. The display/execution repository 411 is coupled to display device, 412 and execution device 413. The dashed box surrounding the computer system 410 represents a security boundary within which communications are assumed to be secure. The display/execution repository 411 is further coupled to a credit server 414 to report any fees to be billed for access to a digital work and a repository 415 for accessing digital works stored therein.

STRUCTURE OF DIGITAL WORKS

Usage rights are attached directly to digital works. Thus, it is important to understand the structure of a digital work. The structure of a digital work, in particular composite digital works, may be naturally organized into an acyclic structure such as a hierarchy. For example, a magazine has various articles and photographs which may have been created and are owned by different persons. Each of the articles and photographs may represent a node in a hierarchical structure. Consequently, controls, i.e. usage rights, may be placed on each node by the creator. By enabling control and fee billing to be associated with each node, a creator of a work can be assured that the rights and fees are not circumvented.

In the currently preferred embodiment, the file information for a digital work is divided into two files: a "contents" file and a "description tree" file. From the perspective of a repository, the "contents" file is a stream of addressable bytes whose format depends completely on the interpreter used to play, display or print the digital work. The description tree file makes it possible to examine the rights and fees for a work without reference to the content of the digital work. It should be noted that the term description tree as used herein refers to any type of acyclic structure used to represent the relationship between the various components of a digital work.

Figure 5 illustrates the layout of a contents file. Referring to Figure 5, a digital work is comprised of story A 510, advertisement 511, story B 512 and story C 513. It is assumed that the digital work is stored starting at a relative address of 0. Each of the parts of the digital work are stored linearly so that story A 510 is stored at approximately addresses 0-30,000, advertisement 511 at addresses 30,001-40,000, story B 512 at addresses 40,001-60,000 and story C 513 at addresses 60,001-85K. The detail of story A 510 is illustrated in Figure 6. Referring to Figure 6, the story A 510 is further broken down to show text 614 stored at address 0-1500, soldier photo 615 at addresses 1501-10,000, graphics 616 stored at addresses 10,001-25,000 and sidebar 617 stored address 25,001-30,000. Note that the data in the contents file may be compressed (for saving storage) or encrypted (for security).

From Figures 5 and 6 it is readily observed that a digital work can be represented by its component parts as a hierarchy. The description tree for a digital work is comprised of a set of related descriptor blocks (d-blocks). The contents of each d-block is described with respect to Figure 7. Referring to Figure 7, a d-block 700 includes an identifier 701 which is a unique identifier for the work in the repository, a starting address 702 providing the start address of the first byte of the work, a length 703 giving the number of bytes in the work, a rights portion 704 wherein the granted usage rights and their status data are maintained, a parent pointer 705 for pointing to a parent d-block and child pointers 706 for pointing to the child d-blocks. In the currently preferred embodiment, the identifier 701 has two parts. The first part is a unique number assigned to the repository upon manufacture. The second part is a unique number assigned to the work upon creation. The rights portion 704 will contain a data structure, such as a look-up table, wherein the various information associated with a right is maintained. The information required by the respective usage rights is described in more detail below. D-blocks form a strict hierarchy. The top d-block of a work has no parent; all other d-blocks have one parent. The relationship of usage rights between parent and child d-blocks and how conflicts are resolved is described below.

A special type of d-block is a "shell" d-block. A shell d-block adds no new content beyond the content of its parts. A shell d-block is used to add rights and fee information, typically by distributors of digital works.

Figure 8 illustrates a description tree for the digital work of Figure 5. Referring to Figure 8, a top d-block 820 for the digital work points to the various stories and advertisements contained therein. Here, the top d-block 820 points to d-block 821 (representing story A 510), d-block 822 (representing the advertisement 511), d-block 823 (representing story B 512) and and d-block 824 (representing story C 513).

The portion of the description tree for Story A 510 is illustrated in Figure 9. D-block 925 represents text 614, d-block 926 represents photo 615, d-block 927 represents graphics 616 by and d-block 928 represents sidebar 617.

The rights portion 704 of a descriptor block is further illustrated in Figure 10. Figure 10 illustrates a structure which is repeated in the rights portion 704 for each right. Referring to Figure 10, each right will have a right code field 1050 and status information field 1052. The right code field 1050 will contain a unique code assigned to a right. The status information field 1052 will contain information relating to the state of a right and the digital work. Such information is indicated below in Table 1. The rights as stored in the rights portion 704 may typically be in numerical order based on the right code.

TABLE 1

DIGITAL WORK STATE INFORMATION		
Property	**Value**	**Use**
Copies-in-Use	Number	A counter of the number of copies of a work that are in use. Incremented when another copy is used; decremented when use is completed.
Loan-Period	Time-Units	Indicator of the maximum number of time-units that a document can be loaned out
Loaner-Copy	Boolean	Indicator that the current work is a loaned out copy of an authorized digital work.
Remaining-Time	Time-Units	Indicator of the remaining time of use on a metered document right.
Document-Descr	String	A string containing various identifying information about a document. The exact format of this is not specified, but it can include information such as a publisher name, author name, ISBN number, and so on.
Revenue-Owner	RO-Descr	A handle identifying a revenue owner for a digital work. This is used reporting usage fees.
Publication-Date	Date-Descr	The date that the digital work was published.
History-list	History-Rec	A list of events recording the repostories and dates for operations that copy, transfer, backup, or restore a digital work.

The approach for representing digtal works by separating description data from content assumes that parts of a file are contiguous but takes no position on the actual representation of content. In particular, it is neutral to the question of whether content representation may take an object oriented approach. It would be natural to represent content as objects. In principle, it may be convenient to have content objects that include the billing structure and rights information that is represented in the d-blocks. Such variations in the design of the representation are possible and are viable alternatives but may introduce processing overhead, e.g. the interpretation of the objects.

Digital works are stored in a repository as part of a hierarchical file system. Folders (also termed directories and sub-directories) contain the digital works as well as other folders. Digital works and folders in a folder are ordered in alphabetical order. The digital works are typed to reflect how the files are used. Usage rights can be attached to folders so that the folder itself is treated as a digital work. Access to the folder would then be handled in the same fashion as any other digital work As will be described in more detail below, the contents of the folder are subject to their own rights. Moreover, file management rights may be attached to the folder which define how folder contents can be managed.

ATTACHING USAGE RIGHTS TO A DIGITAL WORK

It is fundamental to the present invention that the usage rights are treated as part of the digital work. As the digital work is distributed, the scope of the granted usage rights will remain the same or may be narrowed. For example, when a digital work is transferred from a document server to a repository, the usage rights may include the right to loan a copy for a predetermined period of time (called the original rights). When the repository loans out a copy of the digital work, the usage rights in the loaner copy (called the next set of rights) could be set to prohibit any further rights to loan out the copy. The basic idea is that one cannot grant more rights than they have.

The attachment of usage rights into a digital work may occur in a variety of ways. If the usage rights will be the same for an entire digital work, they could be attached when the digital work is processed for deposit in the digital work server. In the case of a digital work having different usage rights for the various components, this can be done as the digital work is being created. An authoring tool or digital work assembling tool could be utilized which provides for an automated process of attaching the usage rights.

As will be described below, when a digital work is copied, transferred or loaned, a "next set of rights" can be specified. The"next set of rights" will be attached to the digital work as it is transported.

Resolving Conflicting Rights

Because each part of a digital work may have its own usage rights, there will be instances where the rights of a "contained part" are different from its parent or container part. As a result, conflict rules must be established to dictate when and how a right may be exercised. The hierarchical structure of a digital work facilitates the enforcement of such rules. A "strict" rule would be as follows: a right for a part in a digital work is sanctioned if and only if it is sanctioned for the part, for ancestor d-blocks containing the part and for all descendent d-blocks. By sanctioned, it is meant that (1) each of the respective parts must have the right, and (2) any conditions for exercising the right are satisfied.

It also possible to implement the present invention using a more lenient rule. In the more lenient rule, access to the part may be enabled to the descendent parts which have the right, but access is denied to the descendents which do not.

An example of applying both the strict rule and lenient is illustrated with reference to Figure 11. Referring to Figure 11, a root d-block 1101 has child d-blocks 1102-1105. In this case, root d-block represents a magazine, and each of the child d-blocks 1102-1105 represent articles in the magazine. Suppose that a request is made to PRINT the digital work represented by root d-block 1101 wherein the strict rule is followed. The rights for the root d-block 1101 and child d-blocks 1102-1105 are then examined. Root d-block 1101 and child d-blocks 1102 and 1105 have been granted PRINT rights. Child d-block 1103 has not been granted PRINT rights and child d-block 1104 has PRINT rights conditioned on payment of a usage fee.

Under the strict rule the PRINT right cannot be exercised because the child d-block does not have the PRINT right. Under the lenient rule, the result would be different. The digital works represented by child d-blocks 1102 and 1105 could be printed and the digital work represented by d-block 1104 could be printed so long as the usage fee is paid. Only the digital work represented by d-block 1103 could not be printed. This same result would be accomplished under the strict rule if the requests were directed to each of the individual digital works.

The present invention supports various combinations of allowing and disallowing access. Moreover, as will be described below, the usage rights grammar permits the owner of a digital work to specify if constraints may be imposed on the work by a container part. The manner in which digital works may be sanctioned because of usage rights conflicts would be implementation specific and would depend on the nature of the digital works.

REPOSITORIES

In the description of Figure 2, it was indicated that repositories come in various forms. All repositories provide a core set of services for the transmission of digital works. The manner in which digital works are exchanged is the basis for all transaction between repositories. The various repository types differ in the ultimate functions that they perform. Repositories may be devices themselves, or they may be incorporated into other systems. An example is the rendering repository 203 of Figure 2.

A repository will have associated with it a repository identifier. Typically, the repository identifier would be a unique number assigned to the repository at the time of manufacture. Each repository will also be classified as being in a particular security class. Certain communications and transactions may be conditioned on a repository being in a particular security class. The various security classes are described in greater detail below.

As a prerequisite to operation, a repository will require possession of an identification certificate. Identification certificates are encrypted to prevent forgery and are issued by a Master repository. A master repository plays the role of an authorization agent to enable repositories to receive digital works. Identification certificates must be updated on a periodic basis. Identification certificates are described in greater detail below with respect to the registration transaction.

A repository has both a hardware and functional embodiment. The functional embodiment is typically software executing on the hardware embodiment. Alternatively, the functional embodiment may be embedded in the hardware embodiment such as an Application Specific Integrated Circuit (ASIC) chip.

The hardware embodiment of a repository will be enclosed in a secure housing which if compromised, may cause the repository to be disabled. The basic components of the hardware embodiment of a repository are described with reference to Figure 12. Referring to Figure 12, a repository is comprised of a processing means 1200, storage system 1207, clock 1205 and external interface 1206. The processing means 1200 is comprised of a processor element 1201 and processor memory 1202. The processing means 1201 provides controller, repository transaction and usage rights transaction functions for the repository. Various functions in the operation of the repository such as decryption and/or decompression of digital works and transaction messages are also performed by the processing means 1200. The processor element 1201 may be a microprocessor or other suitable computing component. The processor memory 1202 would typically be further comprised of Read Only Memories (ROM) and Random Access Memories (RAM). Such memories would contain the software instructions utilized by the processor element 1201 in performing the functions of the repository.

The storage system 1207 is further comprised of descriptor storage 1203 and content storage 1204. The description tree storage 1203 will store the description tree for the digital work and the content storage will store the associated content. The description tree storage 1203 and content storage 1204 need not be of the same type of storage medium, nor are they necessarily on the same physical device. So for example, the descriptor storage 1203 may be stored on a solid state storage (for rapid retrieval of the description tree information), while the content storage 1204 may be on a high capacity storage such as an optical disk.

The clock 1205 is used to time-stamp various time based conditions for usage rights or for metering usage fees which may be associated with the digital works. The clock 1205 will have an uninterruptable power supply, e.g. a battery, in order to maintain the integrity of the time-stamps. The external interface means 1206 provides for the signal connection to other repositories and to a credit server. The external interface means 1206 provides for the exchange of signals via such standard interfaces such as RS-232 or Personal Computer Manufacturers Card Industry Association (PCMCIA) standards, or FDDI. The external interface means 1206 may also provide network connectivity.

The functional embodiment of a repository is described with reference to Figure 13. Referring to Figure 13, the functional embodiment is comprised of an operating system 1301, core repository services 1302, usage transaction handlers 1303, repository specific functions, 1304 and a user interface 1305. The operating system 1301 is specific to the repository and would typically depend on the type of processor being used. The operating system 1301 would also provide the basic services for controlling and interfacing between the basic components of the repository.

The core repository services 1302 comprise a set of functions required by each and every repository. The core repository services 1302 include the session initiation transactions which are defined in greater detail below. This set of services also includes a generic ticket agent which is used to "punch" a digital ticket and a generic authorization server for processing authorization specifications. Digital tickets and authorizations are specific mechanisms for controlling the distribution and use of digital works and are described in more detail below. Note that coupled to the core repository services are a plurality of identification certificates 1306. The identification certificates 1306 are required to enable the use of the repository.

The usage transactions handlers 1303 comprise functionality for processing access requests to digital works and for billing fees based on access. The usage transactions supported will be different for each repository type. For example, it may not be necessary for some repositories to handle access requests for digital works.

The repository specific functionality 1304 comprises functionality that is unique to a repository. For example, the master repository has special functionality for issuing digital certificates and maintaining encryption keys. The repository specific functionality 1304 would include the user interface implementation for the repository.

Repository Security Classes

For some digital works the losses caused by any individual instance of unauthorized copying is insignificant and the chief economic concern lies in assuring the convenience of access and low-overhead billing. In such cases, simple and inexpensive handheld repositories and network-based workstations may be suitable repositories, even though the measures and guarantees of security are modest.

At the other extreme, some digital works such as a digital copy of a first run movie or a bearer bond or stock certificate would be of very high value so that it is prudent to employ caution and fairly elaborate security measures to ensure that they are not copied or forged. A repository suitable for holding such a digital work could have elaborate measures for ensuring physical integrity and for verifying authorization before use.

By arranging a universal protocol, all kinds of repositories can communicate with each other in principle. However, creators of some works will want to specify that their works will only be transferred to repositories whose level of security is high enough. For this reason, document repositories have a ranking system for classes and levels of security. The security classes in the currently preferred embodiment are described in Table 2.

TABLE 2

REPOSITORY SECURITY LEVELS	
Level	Description of Security
0	Open system.Document transmission is unencrypted. No digital certificate is required for identification. The security of the system depends mostly on user honesty, since only modest knowledge may be needed to circumvent the security measures. The repositery has no provisions for preventing unauthorized programs from running and accessing or copying files. The system does not prevent the use of removable storage and does not encrypt stored files.
1	Minimal security. Like the previous class except that stored files are minimally encrypted, including ones on removable storage.
2	Basic security. Like the previous class except that special tools and knowledge are required to compromise the programming, the contents of the repository, or the state of the clock. All digital communications are encrypted. A digital certificate is provided as identification. Medium level encryption is used. Repository identification number is unforgeable.

Continuation of the Table on the next page

TABLE 2 (continued)

REPOSITORY SECURITY LEVELS	
Level	Description of Security
3	General security. Like the previous class plus the requirement of special tools are needed to compromise the physical integrity of the repository and that modest encryption is used on all transmissions. Password protection is required to use the local user interface. The digital clock system cannot be reset without authorization. No works would be stored on removable storage. When executing works as programs, it runs them in their own address space and does not give them direct access to any file storage or other memory containing system code or works. They can access works only through the transmission transaction protocol.
4	Like the previous class except that high level encryption is used on all communications. Sensors are used to record attempts at physical and electronic tampering. After such tampering, the repository will not perform other transactions until it has reported such tampering to a designated server.
5	Like the previous class except that if the physical or digital attempts at tampering exceed some preset thresholds that threaten the physical integrity of the repository or the integrity of digital and cryptographic barriers, then the repository will save only document description records of history but will erase or destroy any digital identifiers that could be misused if released to an unscrupulous. It also modifies any certificates of authenticity to indicate that the physical system has been compromised. It also erases the contents of designated documents.
6	Like the previous class except that the repository will attempt wireless communication to report tampering and will employ noisy alarms.
10	This would correspond to a very high level of security. This server would maintain constant communications to remote security systems reporting transactions, sensor readings, and attempts to circumvent security.

The characterization of security levels described in Table 2 is not intended to be fixed. More important is the idea of having different security levels for different repositories. It is anticipated that new security classes and requirements will evolve according to social situations and changes in technology.

Repository User Interface

A user interface is broadly defined as the mechanism by which a user interacts with a repository in order to invoke transactions to gain access to a digital work, or exercise usage rights. As described above, a repository may be embodied in various forms. The user interface for a repository will differ depending on the particular embodiment. The user interface may be a graphical user interface having icons representing the digital works and the various transactions that may be performed. The user interface may be a generated dialog in which a user is prompted for information.

The user interface itself need not be part of the repository. As a repository may be embedded in some other device, the user interface may merely be a part of the device in which the repository is embedded. For example, the repository could be embedded in a "card" that is inserted into an available slot in a computer system. The user interface may be a combination of a display, keyboard, cursor control device and software executing on the computer system.

At a minimum, the user interface must permit a user to input information such as access requests and alpha numeric data and provide feedback as to transaction status. The user interface will then cause the repository to initiate the suitable transactions to service the request. Other facets of a particular user interface will depend on the functionality that a repository will provide.

CREDIT SERVERS

In the present invention, fees may be associated with the exercise of a right. The requirement for payment of fees is described with each version of a usage right in the usage rights language. The recording and reporting of such fees is performed by the credit server. One of the capabilities enabled by associating fees with rights is the possibility of supporting a wide range of charging models. The simplest model, used by conventional software, is that there is a single fee at the time of purchase, after which the purchaser obtains unlimited rights to use the work as often and for as long as he or she wants. Alternative models, include metered use and variable fees. A single work can have different fees for different uses. For example, viewing a photograph on a display could have different fees than making a hardcopy or including it in a newly created work. A key to these alternative charging models is to have a low overhead means of establishing fees and accounting for credit on these transactions.

A credit server is a computational system that reliably authorizes and records these transactions so that fees are billed and paid. The credit server reports fees to a billing clearinghouse. The billing clearinghouse manages the financial transactions as they occur. As a result, bills may be generated and accounts reconciled. Preferably, the credit server would store the fee transactions and periodically communicate via a network with the billing clearinghouse for reconciliation. In such an embodiment, communications with the billing clearinghouse would be encrypted for integrity and security reasons. In another embodiment, the credit server acts as a "debit card" where transactions occur in "real-time" against a user account.

A credit server is comprised of memory, a processing means, a clock, and interface means for coupling to a repository and a financial institution (e.g. a modem). The credit server will also need to have security and authentication functionality. These elements are essentially the same elements as those of a repository. Thus, a single device can be both a repository and a credit server, provided that it has the appropriate processing elements for carrying out the corresponding functions and protocols. Typically, however, a credit server would be a cardsized system in the possession of the owner of the credit. The credit server is coupled to a repository and would interact via financial transactions as described below. Interactions with a financial institution may occur via protocols established by the financial institutions themselves.

In the currently preferred embodiment credit servers associated with both the server and the repository report the financial transaction to the billing clearinghouse. For example, when a digital work is copied by one repository to another for a fee, credit servers coupled to each of the repositories will report the transaction to the billing clearinghouse. This is desirable in that it insures that a transaction will be accounted for in the event of some break in the communication between a credit server and the billing clearinghouse. However, some implementations may embody only a single credit server reporting the transaction to minimize transaction processing at the risk of losing some transactions.

USAGE RIGHTS LANGUAGE

The present invention uses statements in a high level "usage rights language" to define rights associated with digital works and their parts. Usage rights statements are interpreted by repositories and are used to determine what transactions can be successfully carried out for a digital work and also to determine parameters for those transactions. For example, sentences in the language determine whether a given digital work can be copied, when and how it can be used, and what fees (if any) are to be charged for that use. Once the usage rights statements are generated, they are encoded in a suitable form for accessing during the processing of transactions.

Defining usage rights in terms of a language in combination with the hierarchical representation of a digital work enables the support of a wide variety of distribution and fee schemes. An example is the ability to attach multiple versions of a right to a work. So a creator may attach a PRINT right to make 5 copies for $10.00 and a PRINT right to make unlimited copies for $100.00. A purchaser may then choose which option best fits his needs. Another example is that rights and fees are additive. So in the case of a composite work, the rights and fees of each of the components works is used in determining the rights and fees for the work as a whole.

The basic contents of a right are illustrated in Figure 14. Referring to Figure 14, a right 1450 has a transactional component 1451 and a specifications component 1452. A right 1450 has a label (e.g. COPY or PRINT) which indicates the use or distribution privileges that are embodied by the right. The transactional component 1451 corresponds to a particular way in which a digital work may be used or distributed. The transactional component 1451 is typically embodied in software instructions in a repository which implement the use or distribution privileges for the right. The specifications components 1452 are used to specify conditions which must be satisfied prior to the right being exercised or to designate various transaction related parameters. In the currently preferred embodiment, these specifications include copy count 1453, Fees and Incentives 1454, Time 1455, Access and Security 1456 and Control 1457. Each of these specifications will be described in greater detail below with respect to the language grammar elements.

The usage rights language is based on the grammar described below. A grammar is a convenient means for defining valid sequence of symbols for a language. In describing the grammar the notation "[a|b|c]" is used to indicate distinct choices among alternatives. In this example, a sentence can have either an "a", "b" or "c". It must include exactly one of them. The braces {} are used to indicate optional items. Note that brackets, bars and braces are used to describe the language of usage rights sentences but do not appear in actual sentences in the language.

In contrast, parentheses are part of the usage rights language. Parentheses are used to group items together in lists. The notation (x^*) is used to indicate a variable length list, that is, a list containing one or more items of type x. The notation $(x)^*$ is used to indicate a variable number of lists containing x.

Keywords in the grammar are words followed by colons. Keywords are a common and very special case in the language. They are often used to indicate a single value, typically an identifier. In many cases, the keyword and the parameter are entirely optional. When a keyword is given, it often takes a single identifier as its value. In some cases, the keyword takes a list of identifiers.

In the usage rights language, time is specified in an hours:minutes:seconds (or hh:mm:ss) representation. Time zone indicators, e.g. PDT for Pacific Daylight Time, may also be specified. Dates are represented as year/ month/day (or YYYY/MMM/DD). Note that these time and date representations may specify moments in time or units of time Money units are specified in terms of dollars.

Finally, in the usage rights language, various "things" will need to interact with each other. For example, an instance of a usage right may specify a bank account, a digital ticket, etc.. Such things need to be identified and are specified herein using the suffix "-ID.".

The Usage Rights Grammar is listed in its entirety in Figure 15 and is described below.

Grammar element 1501 **"Digital Work Rights: = (Rights*)"** define the digital work rights as a set of rights. The set of rights attached to a digital work define how that digital work may be transferred, used, performed or played. A set of rights will attach to the entire digital work and in the case of compound digital works, each of the components of the digital work. The usage rights of components of a digital may be different.

Grammar element 1502 **"Right : = (Right-Code {Copy-Count} {Control-Spec} {Time-Spec} {Access-Spec} {Fee-Spec})"** enumerates the content of a right. Each usage right must specify a right code. Each right may also optionally specify conditions which must be satisfied before the right can be exercised. These conditions are copy count, control, time, access and fee conditions. In the currently preferred embodiment, for the optional elements, the following defaults apply: copy count equals 1, no time limit on the use of the right, no access tests or a security level required to use the right and no fee is required. These conditions will each be described in greater detail below.

It is important to note that a digital work may have multiple versions of a right, each having the same right code. The multiple version would provide alternative conditions and fees for accessing the digital work.

Grammar element 1503 **"Right-Code : = Render-Code I Transport-Code I File-Management-CodeI Derivative-Works- Code Configuration-Code"** distinguishes each of the specific rights into a particular right type (although each right is identified by distinct right codes). In this way, the grammar provides a catalog of possible rights that can be associated with parts of digital works. In the following, rights are divided into categories for convenience in describing them.

Grammar element 1504 **"Render-Code : = [Play : {Player: Player-ID} I Print: {Printer: Printer-ID}]"** lists a category of rights all involving the making of ephemeral, transitory, or non-digital copies of the digital work. After use the copies are erased.

• Play	A process of rendering or performing a digital work on some processor. This includes such things as playing digital movies, playing digital music, playing a video game, running a computer program, or displaying a document on a display.
• Print	To render the work in a medium that is not further protected by usage rights, such as printing on paper.

Grammar element 1505 **"Transport-Code : = [Copy I Transfer I Loan (Remaining-Rights: Next-Set-of-Rights}] {(Next-Copy-Rights: Next-Set of Rights)}"** lists a category of rights involving the making of persistent, usable copies of the digital work on other repositories. The optional Next-Copy-Rights determine the rights on the work after it is transported. If this is not specified, then the rights on the transported copy are the same as on the original. The optional Remaining-Rights specify the rights that remain with a digital work when it is loaned out. If this is not specified, then the default is that no rights can be exercised when it is loaned out.

• Copy	Make a new copy of a work
• Transfer	Moving a work from one repository to another.
• Loan	Temporarily loaning a copy to another repository for a specified period of time.

Grammar element 1506 **"File-Management-Code : = Backup {Back-Up-Copy-Rights: Next-Set -of Rights}I Restore I Delete I Folder I Directory {Name:Hide-Local I Hide - Remote}{Parts:Hide-Local I Hide-Remote}"** lists a category of rights involving operations for file management, such as the making of backup copies to protect the copy owner against catastrophic equipment failure.

Many software licenses and also copyright law give a copy owner the right to make backup copies to protect against catastrophic failure of equipment. However, the making of uncontrolled backup copies is inherently at odds with the ability to control usage, since an uncontrolled backup copy can be kept and then restored even after the authorized copy was sold.

The File management rights enable the making and restoring of backup copies in a way that respects usage rights, honoring the requirements of both the copy owner and the rights grantor and revenue owner. Backup copies of work descriptions (including usage rights and fee data) can be sent under appropriate protocol and usage rights control to other document repositories of sufficiently high security. Further rights permit organization of digital works into folders which themselves are treated as digital works and whose contents may be "hidden" from a party seeking to determine the contents of a repository.

• Backup	To make a backup copy of a digital work as protection against media failure.
• Restore	To restore a backup copy of a digital work.
• Delete	To delete or erase a copy of a digital work.
• Folder	To create and name folders, and to move files and folders between folders.
• Directory	To hide a folder or its contents.

Grammar element 1507 "**Derivative-Works-Code : [Extract | Embed | Edit {Process: Process-ID}] {Next-Copy-Rights : Next-Set-of Rights}**" lists a category of rights involving the use of a digital work to create new works.

• Extract	To remove a portion of a work, for the purposes of creating a new work.
• Embed	To include a work in an existing work.
• Edit	To alter a digital work by copying, selecting and modifying portions of an existing digital work.

Grammar element 1508 "**Configuration-Code: = Install | Uninstall**" lists a category of rights for installing and uninstalling software on a repository (typically a rendering repository.) This would typically occur for the installation of a new type of player within the rendering repository.

• Install:	To install new software on a repository.
• Uninstall:	To remove existing software from a repository.

Grammar element 1509 "**Next-Set-of-Rights : = {(Add: Set-Of-Rights)} {(Delete: Set-Of-Rights)} {(Replace: Set-Of-Rights)} {(Keep: Set-Of-Rights)}**" defines how rights are carried forward for a copy of a digital work. If the Next-Copy-Rights is not specified, the rights for the next copy are the same as those of the current copy. Otherwise, the set of rights for the next copy can be specified. Versions of rights after Add: are added to the current set of rights. Rights after Delete: are deleted from the current set of rights. If only right codes are listed after Delete:, then all versions of rights with those codes are deleted. Versions of rights after Replace: subsume all versions of rights of the same type in the current set of rights.

If Remaining-Rights is not specified, then there are no rights for the original after all Loan copies are loaned out. If Remaining-Rights is specified, then the Keep: token can be used to simplify the expression of what rights to keep behind. A list of right codes following keep means that all of the versions of those listed rights are kept in the remaining copy. This specification can be overridden by subsequent Delete: or Replace: specifications.

Copy Count Specification

For various transactions, it may be desirable to provide some limit as to the number of "copies" of the work which may be exercised simultaneously for the right. For example, it may be desirable to limit the number of copies of a digital work that may be loaned out at a time or viewed at a time.

Grammar element 1510 "**Copy-Count : = (Copies: positive-integer | 0 | unlimited)**" provides a condition which defines the number of "copies" of a work subject to the right . A copy count can be 0, a fixed number, or unlimited. The copy-count is associated with each right, as opposed to there being just a single copy-count for the digital work. The Copy-Count for a right is decremented each time that a right is exercised. When the Copy-Count equals zero, the right can no longer be exercised. If the Copy-Count is not specified, the default is one.

Control Specification

Rights and fees depend in general on rights granted by the creator as well as further restrictions imposed by later distributors. Control specifications deal with interactions between the creators and their distributors governing the imposition of further restrictions and fees. For example, a distributor of a digital work may not want an end consumer of a digital work to add fees or otherwise profit by commercially exploiting the purchased digital work.

Grammar element 1511 "**Control-Spec : = (Control: {Restrictable | Unrestrictable} {Unchargeable | Chargeable})**" provides a condition to specify the effect of usage rights and fees of parents on the exercise of the right. A digital work is restrictable if higher level d-blocks can impose further restrictions (time specifications and access specifications) on the right. It is unrestrictable if no further restrictions can be imposed. The default setting is restrictable. A right is unchargeable if no more fees can be imposed on the use of the right. It is chargeable if more fees can be imposed. The default is chargeable.

Time Specification

It is often desirable to assign a start date or specify some duration as to when a right may be exercised. Grammar element 1512 **"Time-Spec : = ({Fixed-Interval | Sliding-Interval | Meter-Time} Until: Expiration-Date)"** provides for specification of time conditions on the exercise of a right. Rights may be granted for a specified time. Different kinds of time specifications are appropriate for different kinds of rights. Some rights may be exercised during a fixed and predetermined duration. Some rights may be exercised for an interval that starts the first time that the right is invoked by some transaction. Some rights may be exercised or are charged according to some kind of metered time, which may be split into separate intervals. For example, a right to view a picture for an hour might be split into six ten minute viewings or four fifteen minute viewings or twenty three minute viewings.

The terms "time" and "date" are used synonymously to refer to a moment in time. There are several kinds of time specifications. Each specification represents some limitation on the times over which the usage right applies. The Expiration-Date specifies the moment at which the usage right ends. For example, if the Expiration-Date is "Jan 1, 1995," then the right ends at the first moment of 1995. If the Expiration-Date is specified as "forever", then the rights are interpreted as continuing without end. If only an expiration date is given, then the right can be exercised as often as desired until the expiration date.

Grammar element 1513 **"Fixed-Interval : = From: Start-Time"** is used to define a predetermined interval that runs from the start time to the expiration date.

Grammar element 1514 **"Sliding-Interval : = Interval: Use-Duration"** is used to define an indeterminate (or "open") start time. It sets limits on a continuous period of time over which the contents are accessible. The period starts on the first access and ends after the duration has passed or the expiration date is reached, whichever comes first. For example, if the right gives 10 hours of continuous access, the use-duration would begin when the first access was made and end 10 hours later.

Grammar element 1515 **"Meter-Time: = Time-Remaining: Remaining-Use"** is used to define a "meter time," that is, a measure of the time that the right is actually exercised. It differs from the Sliding-Interval specification in that the time that the digital work is in use need not be continuous. For example, if the rights guarantee three days of access, those days could be spread out over a month. With this specification, the rights can be exercised until the meter time is exhausted or the expiration date is reached, whichever comes first.

Remaining-Use: = Time-Unit
Start-Time: = Time-Unit
Use-Duration: = Time-Unit

All of the time specifications include time-unit specifications in their ultimate instantiation.

Security Class and Authorization Specification

The present invention provides for various security mechanisms to be introduced into a distribution or use scheme. Grammar element 1516 **"Access-Spec : = ({SC: Security-Class} {Authorization: Authorization-ID*} {Other-Authorization: Authorization-ID*} {Ticket: Ticket-ID})"** provides a means for restricting access and transmission. Access specifications can specify a required security class for a repository to exercise a right or a required authorization test that must be satisfied.

The keyword **"SC:"** is used to specify a minimum security level for the repositories involved in the access. If "SC: " is not specified, the lowest security level is acceptable.

The optional **"Authorization:"** keyword is used to specify required authorizations on the same repository as the work. The optional **"Other-Authorization:"** keyword is used to specify required authorizations on the other repository in the transaction.

The optional **"Ticket:"** keyword specifies the identity of a ticket required for the transaction. A transaction involving digital tickets must locate an appropriate digital ticket agent who can "punch" or otherwise validate the ticket before the transaction can proceed. Tickets are described in greater detail below.

In a transaction involving a repository and a document server, some usage rights may require that the repository have a particular authorization, that the server have some authorization, or that both repositories have (possibly different) authorizations. Authorizations themselves are digital works (hereinafter referred to as an authorization object) that can be moved between repositories in the same manner as other digital works. Their copying and transferring is subject to the same rights and fees as other digital works. A repository is said to have an authorization if that authorization object is contained within the repository.

In some cases, an authorization may be required from a source other than the document server and repository. An authorization object referenced by an Authorization-ID can contain digital address information to be used to set up a communications link between a repository and the authorization source. These are analogous to phone numbers. For such access tests, the communication would need to be established and authorization obtained before the right could be exercised.

For one-time usage rights, a variant on this scheme is to have a digital ticket. A ticket is presented to a digital ticket agent, whose type is specified on the ticket. In the simplest case, a certified generic ticket agent, available on all repositories, is available to "punch" the ticket. In other cases, the ticket may contain addressing information for locating a "special" ticket agent. Once a ticket has been punched, it cannot be used again for the same kind of transaction (unless it is unpunched or refreshed in the manner described below.) Punching includes marking the ticket with a timestamp of the date and time it was used. Tickets are digital works and can be copied or transferred between repositories according to their usage rights.

In the currently preferred embodiment, a "punched" ticket becomes "unpunched" or "refreshed" when it is copied or extracted. The Copy and Extract operations save the date and time as a property of the digital ticket. When a ticket agent is given a ticket, it can simply check whether the digital copy was made after the last time that it was punched. Of course, the digital ticket must have the copy or extract usage rights attached thereto.

The capability to unpunch a ticket is inportant in the following cases:

- A digital work is circulated at low cost with a limitation that it can be used only once.
- A digital work is circulated with a ticket that can be used once to give discounts on purchases of other works.
- A digital work is circulated with a ticket (included in the purchase price and possibly embedded in the work) that can be used for a future upgrade.

In each of these cases, if a paid copy is made of the digital work (including the ticket) the new owner would expect to get a fresh (unpunched) ticket, whether the copy seller has used the work or not. In contrast, loaning a work or simply transferring it to another repository should not revitalize the ticket.

Usage Fees and Incentives Specification

The billing for use of a digital work is fundamental to a commercial distribution system. Grammar Element 1517 **"Fee-Spec: = {Scheduled-Discount} Regular-Fee-Spec I Scheduled-Fee-Spec I Markup-Spec"** provides a range of options for billing for the use of digital works.

A key feature of this approach is the development of low-overhead billing for transactions in potentially small amounts. Thus, it becomes feasible to collect fees of only a few cents each for thousands of transactions.

The grammar differentiates between uses where the charge is per use from those where it is metered by the time unit. Transactions can support fees that the user pays for using a digital work as well as incentives paid by the right grantor to users to induce them to use or distribute the digital work.

The optional scheduled discount refers to the rest of the fee specification--discounting it by a percentage over time. If it is not specified, then there is no scheduled discount. Regular fee specifications are constant over time. Scheduled fee specifications give a schedule of dates over which the fee specifications change. Markup specifications are used in d-blocks for adding a percentage to the fees already being charged.

Grammar Element 1518 **"Scheduled-Discount: = (Scheduled-Discount: (Time-Spec Percentage)*)"** A Scheduled-Discount is a essentially a scheduled modifier of any other fee specification for this version of the right of the digital work. (It does not refer to children or parent digital works or to other versions of rights.). It is a list of pairs of times and percentages. The most recent time in the list that has not yet passed at the time of the transaction is the one in effect. The percentage gives the discount percentage. For example, the number 10 refers to a 10% discount.

Grammar Element 1519 **"Regular-Fee-Spec : =({Fee: I Incentive: } [Per-Use-Spec I Metered-Rate-Spec I Best-Price-Spec I Call-For-Price-Spec] {Min: Money-Unit Per: Time-Spec}{Max: Money-Unit Per: Time-Spec} To: Account-ID)"** provides for several kinds of fee specifications.

Fees are paid by the copy-owner/user to the revenue-owner if **Fee**: is specified. Incentives are paid by the revenue-owner to the user if Incentive: is specified. If the **Min**: specification is given, then there is a minimum fee to be charged per time-spec unit for its use. If the **Max**: specification is given, then there is a maximum fee to be charged per time-spec for its use. When **Fee**: is specified, **Account-ID** identifies the account to which the fee is to be paid. When Incentive: is specified, **Account-ID** identifies the account from which the fee is to be paid.

Grammar element 1520 **"Per-Use-Spec: = Per-Use: Money-unit"** defines a simple fee to be paid every time the right is exercised, regardless of how much time the transaction takes.

Grammar element 1521 **"Metered-Rate-Spec := Metered: Money-Unit Per: Time-Spec"** defines a metered-rate fee paid according to how long the right is exercised. Thus, the time it takes to complete the transaction determines the fee.

Grammar element 1522 **"Best-Price-Spec := Best-Price: Money-unit Max: Money-unit"** is used to specify a best-price that is determined when the account is settled. This specification is to accommodate special deals, rebates, and pricing that depends on information that is not available to the repository. All fee specifications can be combined with tickets or authorizations that could indicate that the consumer is a wholesaler or that he is a preferred customer, or that the seller be authorized in some way. The amount of money in the **Max**: field is the maximum amount that the use will cost. This is the amount that is tentatively debited from the credit server. However, when the transaction is ultimately reconciled, any excess amount will be returned to the consumer in a separate transaction.

Grammar element 1523 "**Call-For-Price-Spec : = Call-For-Price** " is similar to a "**Best-Price-Spec**" in that it is intended to accommodate cases where prices are dynamic. **A Call-For-Price Spec** requires a communication with a dealer to determine the price. This option cannot be exercised if the repository cannot communicate with a dealer at the time that the right is exercised. It is based on a secure transaction whereby the dealer names a price to exercise the right and passes along a deal certificate which is referenced or included in the billing process.

Grammar element 1524 "**Scheduled-Fee-Spec: = (Schedule: (Time-Spec Regular-Fee-Spec)*)**" is used to provide a schedule of dates over which the fee specifications change. The fee specification with the most recent date not in the future is the one that is in effect. This is similar to but more general than the scheduled discount. It is more general, because it provides a means to vary the fee agreement for each time period.

Grammar element 1525 "**Markup-Spec: = Markup: percentage To: Account-ID**" is provided for adding a percentage to the fees already being charged. For example, a 5% markup means that a fee of 5% of cumulative fee so far will be allocated to the distributor. A markup specification can be applied to all of the other kinds of fee specifications. It is typically used in a shell provided by a distributor. It refers to fees associated with d-blocks that are parts of the current d-block. This might be a convenient specification for use in taxes, or in distributor overhead.

REPOSITORY TRANSACTIONS

When a user requests access to a digital work, the repository will initiate various transactions. The combination of transactions invoked will depend on the specifications assigned for a usage right. There are three basic types of transactions, Session Initiation Transactions, Financial Transactions and Usage Transactions. Generally, session initiation transactions are initiated first to establish a valid session. When a valid session is established, transactions corresponding to the various usage rights are invoked. Finally, request specific transactions are performed.

Transactions occur between two repositories (one acting as a server), between a repository and a document playback platform (e.g. for executing or viewing), between a repository and a credit server or between a repository and an authorization server. When transactions occur between more than one repository, it is assumed that there is a reliable communication channel between the repositories. For example, this could be a TCP/IP channel or any other commercially available channel that has built-in capabilities for detecting and correcting transmission errors. However, it is not assumed that the communication channel is secure. Provisions for security and privacy are part of the requirements for specifying and implementing repositories and thus form the need for various transactions.

Message Transmission

Transactions require that there be some communication between repositories. Communication between repositories occurs in units termed as messages. Because the communication line is assumed to be unsecure, all communications with repositories that are above the lowest security class are encrypted utilizing a public key encryption technique. Public key encryption is a well known technique in the encryption arts. The term key refers to a numeric code that is used with encryption and decryption algorithms. Keys come in pairs, where "writing keys" are used to encrypt data and "checking keys" are used to decrypt data. Both writing and checking keys may be public or private. Public keys are those that are distributed to others Private keys are maintained in confidence.

Key management and security is instrumental in the success of a public key encryption system. In the currently preferred embodiment, one or more master repositories maintain the keys and create the identification certificates used by the repositories.

When a sending repository transmits a message to a receiving repository, the sending repository encrypts all of its data using the public writing key of the receiving repository. The sending repository includes its name, the name of the receiving repository, a session identifier such as a nonce (described below), and a message counter in each message.

In this way, the communication can only be read (to a high probability) by the receiving repository, which holds the private checking key for decryption. The auxiliary data is used to guard against various replay attacks to security. If messages ever arrive with the wrong counter or an old nonce, the repositories can assume that someone is interfering with communication and the transaction terminated.

The respective public keys for the repositories to be used for encryption are obtained in the registration transaction described below.

Session Initiation Transactions

A usage transaction is carried out in a session between repositories. For usage transactions involving more than one repository, or for financial transactions between a repository and a credit server, a registration transaction is performed. A second transaction termed a login transaction, may also be needed to initiate the session. The goal of the registration transaction is to establish a secure channel between two repositories who know each others identities. As it is assumed that the communication channel between the repositories is reliable but not secure, there is a risk that a non-repository may mimic the protocol in order to gain illegitimate access to a repository.

The registration transaction between two repositories is described with respect to Figures 16 and 17. The steps described are from the perspective of a "repository-1" registering its identity with a "repository-2". The registration must be symmetrical so the same set of steps will be repeated for repository-2 registering its identity with repository-1. Referring to Figure 16, repository-1 first generates an encrypted registration identifier, step 1601 and then generates a registration message, step 1602. A registration message is comprised of an identifier of a master repository, the identification certificate for the repository-1 and an encrypted random registration identifier. The identification certificate is encrypted by the master repository in its private key and attests to the fact that the repository (here repository-1) is a bona fide repository. The identification certificate also contains a public key for the repository, the repository security level and a timestamp (indicating a time after which the certificate is no longer valid.) The registration identifier is a number generated by the repository for this registration. The registration identifier is unique to the session and is encrypted in repository-1's private key. The registration identifier is used to improve security of authentication by detecting certain kinds of communications based attacks. Repository-1 then transmits the registration message to repository-2, step 1603.

Upon receiving the registration message, repository-2 determines if it has the needed public key for the master repository, step 1604. If repository-2 does not have the needed public key to decrypt the identification certificate, the registration transaction terminates in an error, step 1618.

Assuming that repository-2 has the proper public key the identification certificate is decrypted, step 1605. Repository-2 saves the encrypted registration identifier, step 1606, and extracts the repository identifier, step 1607. The extracted repository identifier is checked against a "hotlist" of compromised document repositories, step 1608. In the currently preferred embodiment, each repository will contain "hotlists" of compromised repositories. If the repository is on the "hotlist', the registration transaction terminates in an error per step 1618. Repositories can be removed from the hotlist when their certificates expire, so that the list does not need to grow without bound. Also, by keeping a short list of hotlist certificates that it has previously received, a repository can avoid the work of actually going through the list. These lists would be encrypted by a master repository. A minor variation on the approach to improve efficiency would have the repositories first exchange lists of names of hotlist certificates, ultimately exchanging only those lists that they had not previously received. The "hotlists" are maintained and distributed by Master repositories.

Note that rather than terminating in error, the transaction could request that another registration message be sent based on an identification certificate created by another master repository. This may be repeated until a satisfactory identification certificate is found, or it is determined that trust cannot be established.

Assuming that the repository is not on the hotlist, the repository identification needs to be verified. In other words, repository-2 needs to validate that the repository on the other end is really repository-1. This is termed performance testing and is performed in order to avoid invalid access to the repository via a counterfeit repository replaying a recording of a prior session initiation between repository-1 and repository-2. Performance testing is initiated by repository-2 generating a performance message, step 1609. The performance message consists of a nonce, the names of the respective repositories, the time and the registration identifier received from repository-1. A nonce is a generated message based on some random and variable information (e.g. the time or the temperature.) The nonce is used to check whether repository-1 can actually exhibit correct encrypting of a message using the private keys it claims to have, on a message that it has never seen before. The performance message is encrypted using the public key specified in the registration message of repository-1. The performance message is transmitted to repository-1, step 1610, where it is decrypted by repository-1 using its private key, step 1611. Repository-1 then checks to make sure that the names of the two repositories are correct, step 1612, that the time is accurate, step 1613 and that the registration identifier corresponds to the one it sent, step 1614. If any of these tests fails, the transaction is terminated per step 1616. Assuming that the tests are passed, repository-1 transmits the nonce to repository-2 in the clear, step 1615. Repository-2 then compares the received nonce to the original nonce, step 1617. If they are not identical, the registration transaction terminates in an error per step 1618. If they are the same, the registration transaction has successfully completed.

At this point, assuming that the transaction has not terminated, the repositories exchange messages containing session keys to be used in all communications during the session and synchronize their clocks. Figure 17 illustrates the session information exchange and clock synchronization steps (again from the perspective of repository-1.) Referring to Figure 17, repository-1 creates a session key pair, step 1701. A first key is kept private and is used by repository-1 to encrypt messages. The second key is a public key used by repository-2 to decrypt messages. The second key is encrypted using the public key of repository-2, step 1702 and is sent to repository-2, step 1703. Upon receipt, repository-2 decrypts the second key, step 1704. The second key is used to decrypt messages in subsequent communications. When each repository has completed this step, they are both convinced that the other repository is bona fide and that they are communicating with the original. Each repository has given the other a key to be used in decrypting further communications during the session. Since that key is itself transmitted in the public key of the receiving repository only it will be able to decrypt the key which is used to decrypt subsequent messages.

After the session information is exchanged, the repositories must synchronize their clocks Clock synchronization is used by the repositories to establish an agreed upon time base for the financial records of their mutual transactions. Referring back to Figure 17, repository-2 initiates clock synchronization by generating a time stamp exchange message, step 1705, and transmits it to repository- 1, step 1706. Upon receipt, repository-1 generates its own time stamp message, step 1707 and transmits it back to repository-2, step 1708. Repository-2 notes the current time, step 1709 and stores the time received from repository-1, step 1710. The current time is compared to the time received from repository-1, step 1711. The difference is then checked to see if it exceeds a predetermined tolerance (e.g. one minute), step 1712. If it does, repository-2 terminates the transaction as this may indicate tampering with the repository, step 1713. If not repository-2 computes an adjusted time delta, step 1714. The adjusted time delta is the difference between the clock time of repository-2 and the average of the times from repository-1 and repository-2.

To achieve greater accuracy, repository-2 can request the time again up to a fixed number of times (e.g. five times), repeat the clock synchronization steps, and average the results.

A second session initiation transaction is a Login transaction. The Login transaction is used to check the authenticity of a user requesting a transaction. A Login transaction is particularly prudent for the authorization of financial transactions that will be charged to a credit server. The Login transaction involves an interaction between the user at a user interface and the credit server associated with a repository. The information exchanged here is a login string supplied by the repository/credit server to identify itself to the user, and a Personal Identification Number (PIN) provided by the user to identify himself to the credit server. In the event that the user is accessing a credit server on a repository different from the one on which the user interface resides, exchange of the information would be encrypted using the public and private keys of the respective repositories.

Billing Transactions

Billing Transactions are concerned with monetary transactions with a credit server. Billing Transactions are carried out when all other conditions are satisfied and a usage fee is required for granting the request. For the most part, billing transactions are well understood in the state of the art. These transactions are between a repository and a credit server, or between a credit server and a billing clearinghouse. Briefly, the required transactions include the following:

- Registration and LOG IN transactions by which the repository and user establish their bona fides to a credit server. These transactions would be entirely internal in cases where the repository and credit server are implemented as a single system.
- Registration and LOG IN transactions, by which a credit server establishes its bona fides to a billing clearinghouse.
- An Assign-fee transaction to assign a charge. The information in this transaction would include a transaction identifier, the identities of the repositories in the transaction, and a list of charges from the parts of the digital work. If there has been any unusual event in the transaction such as an interruption of communications, that information is included as well.
- A Begin-charges transaction to assign a charge. This transaction is much the same as an assign-fee transaction except that it is used for metered use. It includes the same information as the assign-fee transaction as well as the usage fee information. The credit-server is then responsible for running a clock.
- An End-charges transaction to end a charge for metered use. (In a variation on this approach, the repositories would exchange periodic charge information for each block of time.)
- A report-charges transaction between a personal credit server and a billing clearinghouse. This transaction is invoked at least once per billing period. It is used to pass along information about charges. On debit and credit cards, this transaction would also be used to update balance information and credit limits as needed.

All billing transactions are given a transaction ID and are reported to the credit severs by both the server and the client. This reduces possible loss of billing information if one of the parties to a transaction loses a banking card and provides a check against tampering with the system.

Usage Transactions

After the session initiation transactions have been completed, the usage request may then be processed. To simplify the description of the steps carried out in processing a usage request, the term requester is used to refer to a repository in the requester mode which is initiating a request, and the term server is used to refer to a repository in the server mode and which contains the desired digital work. In many cases such as requests to print or view a work, the requester and server may be the same device and the transactions described in the following would be entirely internal. In such instances, certain transaction steps, such as the registration transaction, need not be performed.

There are some common steps that are part of the semantics of all of the usage rights transactions. These steps are referred to as the common transaction steps. There are two sets --the "opening" steps and the "closing" steps. For simplicity, these are listed here rather than repeating them in the descriptions of all of the usage rights transactions.

Transactions can refer to a part of a digital work, a complete digital work, or a Digital work containing other digital works. Although not described in detail herein, a transaction may even refer to a folder comprised of a plurality of digital works. The term "work" is used to refer to what ever portion or set of digital works is being accessed.

Many of the steps here involve determining if certain conditions are satisfied. Recall that each usage right may have one or more conditions which must be satisfied before the right can be exercised. Digital works have parts and parts have parts. Different parts can have different rights and fees. Thus, it is necessary to verify that the requirements are met for ALL of the parts that are involved in a transaction For brevity, when reference is made to checking whether the rights exist and conditions for exercising are satisfied, it is meant that all such checking takes place for each of the relevant parts of the work.

Figure 18 illustrates the initial common opening and closing steps for a transaction. At this point it is assumed that registration has occurred and that a "trusted" session is in place. General tests are tests on usage rights associated with the folder containing the work or some containing folder higher in the file system hierarchy. These tests correspond to requirements imposed on the work as a consequence of its being on the particular repository, as opposed to being attached to the work itself. Referring to Figure 18, prior to initiating a usage transaction, the requester performs any general tests that are required before the right associated with the transaction can be exercised, step, 1801. For example, install, uninstall and delete rights may be implemented to require that a requester have an authorization certificate before the right can be exercised. Another example is the requirement that a digital ticket be present and punched before a digital work may be copied to a requester. If any of the general tests fail, the transaction is not initiated, step, 1802. Assuming that such required tests are passed, upon receiving the usage request, the server generates a transaction identifier that is used in records or reports of the transaction, step 1803. The server then checks whether the digital work has been granted the right corresponding to the requested transaction, step 1804. If the digital work has not been granted the right corresponding to the request, the transaction terminates, step 1805. If the digital work has been granted the requested right, the server then determines if the various conditions for exercising the right are satisfied. Time based conditions are examined, step 1806. These conditions are checked by examining the time specification for the the version of the right. If any of the conditions are not satisfied, the transaction terminates per step 1805.

Assuming that the time based conditions are satisfied, the server checks security and access conditions, step 1807. Such security and access conditions are satisfied if: 1) the requester is at the specified security class, or a higher security class, 2) the server satisfies any specified authorization test and 3) the requester satisfies any specified authorization tests and has any required digital tickets. If any of the conditions are not satisfied, the transaction terminates per step 1805.

Assuming that the security and access conditions are all satisfied, the server checks the copy count condition, step 1808. If the copy count equals zero, then the transaction cannot be completed and the transaction terminates per step 1805.

Assuming that the copy count does not equal zero, the server checks if the copies in use for the requested right is greater than or equal to any copy count for the requested right (or relevant parts), step 1809. If the copies in use is greater than or equal to the copy count, this indicates that usage rights for the version of the transaction have been exhausted. Accordingly, the server terminates the transaction, step 1805. If the copy count is less than the copies in use for the transaction the transaction can continue, and the copies in use would be incremented by the number of digital works requested in the transaction, step 1810.

The server then checks if the digital work has a "Loan" access right, step 1811. The "Loan" access right is a special case since remaining rights may be present even though all copies are loaned out. If the digital work has the "Loan" access right, a check is made to see if all copies have been loaned out, step 1812. The number of copies that could be loaned is the sum of the Copy-Counts for all of the versions of the loan right of the digital work. For a composite work, the relevant figure is the minimal such sum of each of the components of the composite work. If all copies have been loaned out, the remaining rights are determined, step 1813. The remaining-rights is determined from the remaining rights specifications from the versions of the Loan right. If there is only one version of the Loan right, then the determination is simple. The remaining rights are the ones specified in that version of the Loan right, or none if Remaining-Rights: is not specified. If there are multiple versions of the Loan right and all copies of all of the versions are loaned out, then the remaining rights is taken as the minimum set (intersection) of remaining rights across all of the versions of the loan right. The server then determines if the requested right is in the set of remaining rights, step 1814. If the requested right is not in the set of remaining rights, the server terminates the transaction, step 1805.

If Loan is not a usage right for the digital work or if all copies have not been loaned out or the requested right is in the set of remaining rights, fee conditions for the right are then checked, step 1815. This will initiate various financial transactions between the repository and associated credit server. Further, any metering of usage of a digital work will commence. If any financial transaction fails, the transaction terminates per step 1805.

It should be noted that the order in which the conditions are checked need not follow the order of steps 1806-1815.

At this point, right specific steps are now performed and are represented here as step 1816. The right specific steps are described in greater detail below.

The common closing transaction steps are now performed. Each of the closing transaction steps are performed by the server after a successful completion of a transaction. Referring back to Figure 18, the copies in use value for the requested right is decremented by the number of copies involved in the transaction, step 1817. Next, if the right had a metered usage fee specification, the server subtracts the elapsed time from the Remaining-Use-Time associated with the right for every part involved in the transaction, step 1818. Finally, if there are fee specifications associated with the right, the server initiates End-Charge financial transaction to confirm billing, step 1819.

Transmission Protocol

An important area to consider is the transmission of the digital work from the server to the requester. The transmission protocol described herein refers to events occurring after a valid session has been created. The transmission protocol must handle the case of disruption in the communications between the repositories. It is assumed that interference such as injecting noise on the communication channel can be detected by the integrity checks (e.g., parity, checksum, etc.) that are built into the transport protocol and are not discussed in detail herein.

The underlying goal in the transmission protocol is to preclude certain failure modes, such as malicious or accidental interference on the communications channel. Suppose, for example, that a user pulls a card with the credit server at a specific time near the end of a transaction. There should not be a vulnerable time at which "pulling the card" causes the repositories to fail to correctly account for the number of copies of the work that have been created. Restated, there should be no time at which a party can break a connection as a means to avoid payment after using a digital work.

If a transaction is interrupted (and fails), both repositories restore the digital works and accounts to their state prior to the failure, modulo records of the failure itself.

Figure 19 is a state diagram showing steps in the process of transmitting information during a transaction. Each box represents a state of a repository in either the server mode (above the central dotted line 1901) or in the requester mode (below the dotted line 1901). Solid arrows stand for transitions between states. Dashed arrows stand for message communications between the repositories. A dashed message arrow pointing to a solid transition arrow is interpreted as meaning that the transition takes place when the message is received. Unlabeled transition arrows take place unconditionally. Other labels on state transition arrows describe conditions that trigger the transition.

Referring now to Figure 19, the server is initially in a state 1902 where a new transaction is initiated via start message 1903. This message includes transaction information including a transaction identifier and a count of the blocks of data to be transferred. The requester, initially in a wait state 1904 then enters a data wait state 1905.

The server enters a data transmit state 1906 and transmits a block of data 1907 and then enters a wait for acknowledgement state 1908. As the data is received, the requester enters a data receive state 1909 and when the data blocks are completely received it enters an acknowledgement state 1910 and transmits an Acknowledgement message 1911 to the server.

If there are more blocks to send, the server waits until receiving an Acknowledgement message from the requester. When an Acknowledgement message is received it sends the next block to the requester and again waits for acknowledgement. The requester also repeats the same cycle of states.

If the server detects a communications failure before sending the last block, it enters a cancellation state 1912 wherein the transaction is cancelled. Similarly, if the requester detects a communications failure before receiving the last block it enters a cancellation state 1913.

If there are no more blocks to send, the server commits to the transaction and waits for the final Acknowledgement in state 1914. If there is a communications failure before the server receives the final Acknowledgement message, it still commits to the transaction but includes a report about the event to its credit server in state 1915. This report serves two purposes. It will help legitimize any claims by a user of having been billed for receiving digital works that were not completely received. Also it helps to identify repositories and communications lines that have suspicious patterns of use and interruption. The server then enters its completion state 1916.

On the requester side, when there are no more blocks to receive, the requester commits to the transaction in state 1917. If the requester detects a communications failure at this state, it reports the failure to its credit server in state 1918, but still commits to the transaction. When it has committed, it sends an acknowledgement message to the server. The server then enters its completion state 1919.

The key property is that both the server and the requester cancel a transaction if it is interrupted before all of the data blocks are delivered, and commits to it if all of the data blocks have been delivered.

There is a possibility that the server will have sent all of the data blocks (and committed) but the requester will not have received all of them and will cancel the transaction. In this case, both repositories will presumably detect a communications failure and report it to their credit server. This case will probably be rare since it depends on very precise timing of the communications failure. The only consequence will be that the user at the requester repository may want to request a refund from the credit services -- and the case for that refund will be documented by reports by both repositories.

To prevent loss of data, the server should not delete any transferred digital work until receiving the final acknowledgement from the requester. But it also should not use the file. A well known way to deal with this situation is called "two-phase commit" or 2PC.

Two-phase commit works as follows. The first phase works the same as the method described above. The server sends all of the data to the requester. Both repositories mark the transaction (and appropriate files) as uncommitted. The server sends a ready-to-commit message to the requester. The requester sends back an acknowledgement. The server then commits and sends the requester a commit message. When the requester receives the commit message, it commits the file.

If there is a communication failure or other crash, the requester must check back with the server to determine the status of the transaction. The server has the last word on this. The requester may have received all of the data, but if it did not get the final message, it has not committed. The server can go ahead and delete files (except for transaction records) once it commits, since the files are known to have been fully transmitted before starting the 2PC cycle.

There are variations known in the art which can be used to achieve the same effect. For example, the server could use an additional level of encryption when transmitting a work to a client. Only after the client sends a message acknowledging receipt does it send the key. The client then agrees to pay for the digital work. The point of this variation is that it provides a clear audit trail that the client received the work. For trusted systems, however, this variation adds a level of encryption for no real gain in accountability.

The transaction for specific usage rights are now discussed.

The Copy Transaction

A Copy transaction is a request to make one or more independent copies of the work with the same or lesser usage rights. Copy differs from the extraction right discussed later in that it refers to entire digital works or entire folders containing digital works. A copy operation cannot be used to remove a portion of a digital work.

- The requester sends the server a message to initiate the Copy Transaction. This message indicates the work to be copied, the version of the copy right to be used for the transaction, the destination address information (location in a folder) for placing the work, the file data for the work (including its size), and the number of copies requested.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the client according to the transmission protocol. If a Next-Set-Of-Rights has been provided in the version of the right, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted. In any event, the Copy-Count field for the copy of the digital work being sent right is set to the number-of-copies requested.
- The requester records the work contents, data, and usage rights and stores the work. It records the date and time that the copy was made in the properties of the digital work.
- The repositories perform the common closing transaction steps.

The Transfer Transaction

A Transfer transaction is a request to move copies of the work with the same or lesser usage rights to another repository. In contrast with a copy transaction, this results in removing the work copies from the server.

- The requester sends the server a message to initiate the Transfer Transaction. This message indicates the work to be transferred, the version of the transfer right to be used in the transaction, the destination address information for placing the work, the file data for the work, and the number of copies involved.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted. In either case, the Copy-Count field for the transmitted rights are set to the number-of-copies requested.

- The requester records the work contents, data, and usage rights and stores the work.
- The server decrements its copy count by the number of copies involved in the transaction.
- The repositories perform the common closing transaction steps.
- If the number of copies remaining in the server is now zero, it erases the digital work from its memory.

The Loan Transaction

A loan transaction is a mechanism for loaning copies of a digital work. The maximum duration of the loan is determined by an internal parameter of the digital work. Works are automatically returned after a predetermined time period.

- The requester sends the server a message to initiate the Transfer Transaction. This message indicates the work to be loaned, the version of the loan right to be used in the transaction, the destination address information for placing the work, the number of copies involved, the file data for the work, and the period of the loan.
- The server checks the validity of the requested loan period, and ends with an error if the period is not valid. Loans for a loaned copy cannot extend beyond the period of the original loan to the server.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted, as modified to reflect the loan period.
- The requester records the digital work contents, data, usage rights, and loan period and stores the work.
- The server updates the usage rights information in the digital work to reflect the number of copies loaned out.
- The repositories perform the common closing transaction steps.
- The server updates the usage rights data for the digital work. This may preclude use of the work until it is returned from the loan. The user on the requester platform can now use the transferred copies of the digital work. A user accessing the original repository cannot use the digital work, unless there are copies remaining. What happens next depends on the order of events in time.

Case 1. If the time of the loan period is not yet exhausted and the requester sends the repository a Return message.

- The return message includes the requester identification, and the transaction ID.
- The server decrements the copies-in-use field by the number of copies that were returned. (If the number of digital works returned is greater than the number actually borrowed, this is treated as an error.) This step may now make the work available at the server for other users.
- The requester deactivates its copies and removes the contents from its memory.

Case 2. If the time of the loan period is exhausted and the requester has not yet sent a Return message.

- The server decrements the copies-in-use field by the number digital works that were borrowed.
- The requester automatically deactivates its copies of the digital work. It terminates all current uses and erases the digital work copies from memory. One question is why a requester would ever return a work earlier than the period of the loan, since it would be returned automatically anyway. One reason for early return is that there may be a metered fee which determines the cost of the loan. Returning early may reduce that fee.

The Play Transaction

A play transaction is a request to use the contents of a work. Typically, to "play" a work is to send the digital work through some kind of transducer, such as a speaker or a display device. The request implies the intention that the contents will not be communicated digitally to any other system. For example, they will not be sent to a printer, recorded on any digital medium, retained after the transaction or sent to another repository.

This term "play" is natural for examples like playing music, playing a movie, or playing a video game. The general form of play means that a "player" is used to use the digital work. However, the term play covers all media and kinds of recordings. Thus one would "play" a digital work, meaning, to render it for reading, or play a computer program, meaning to execute it. For a digital ticket the player would be a digital ticket agent.

- The requester sends the server a message to initiate the play transaction. This message indicates the work to be played, the version of the play right to be used in the transaction, the identity of the player being used, and the file data for the work.

- The server checks the validity of the player identification and the compatibility of the player identification with the player specification in the right. It ends with an error if these are not satisfactory.
- The repositories perform the common opening transaction steps.
- The server and requester read and write the blocks of data as requested by the player according to the transmission protocol. The requester plays the work contents, using the player.
- When the player is finished, the player and the requester remove the contents from their memory.
- The repositories perform the common closing transaction steps.

The Print Transaction

A Print transaction is a request to obtain the contents of a work for the purpose of rendering them on a "printer." We use the term "printer" to include the common case of writing with ink on paper. However, the key aspect of "printing" in our use of the term is that it makes a copy of the digital work in a place outside of the protection of usage rights. As with all rights, this may require particular authorization certificates.

Once a digital work is printed, the publisher and user are bound by whatever copyright laws are in effect. However, printing moves the contents outside the control of repositories. For example, absent any other enforcement mechanisms, once a digital work is printed on paper, it can be copied on ordinary photocopying machines without intervention by a repository to collect usage fees. If the printer to a digital disk is permitted, then that digital copy is outside of the control of usage rights. Both the creator and the user know this, although the creator does not necessarily give tacit consent to such copying, which may violate copyright laws.

- The requester sends the server a message to initiate a Print transaction. This message indicates the work to be played, the identity of the printer being used, the file data for the work, and the number of copies in the request.
- The server checks the validity of the printer identification and the compatibility of the printer identification with the printer specification in the right. It ends with an error if these are not satisfactory.
- The repositories perform the common opening transaction steps.
- The server transmits blocks of data according to the transmission protocol.
- The requester prints the work contents, using the printer.
- When the printer is finished, the printer and the requester remove the contents from their memory.
- The repositories perform the common closing transaction steps.

The Backup Transaction

A Backup transaction is a request to make a backup copy of a digital work, as a protection against media failure. In the context of repositories, secure backup copies differ from other copies in three ways: (1) they are made under the control of a Backup transaction rather than a Copy transaction, (2) they do not count as regular copies, and (3) they are not usable as regular copies. Generally, backup copies are encrypted.

Although backup copies may be transferred or copied, depending on their assigned rights, the only way to make them useful for playing, printing or embedding is to restore them.

The output of a Backup operation is both an encrypted data file that contains the contents and description of a work, and a restoration file with an encryption key for restoring the encrypted contents. In many cases, the encrypted data file would have rights for "printing" it to a disk outside of the protection system, relying just on its encryption for security. Such files could be stored anywhere that was physically safe and convenient. The restoration file would be held in the repository. This file is necessary for the restoration of a backup copy. It may have rights for transfer between repositories.

- The requester sends the server a message to initiate a backup transaction. This message indicates the work to be backed up, the version of the backup right to be used in the transaction, the destination address information for placing the backup copy, the file data for the work.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, a set of default rights for backup files of the original are transmitted by the server.
- The requester records the work contents, data, and usage rights. It then creates a one-time key and encrypts the contents file. It saves the key information in a restoration file.
- The repositories perform the common closing transaction steps.

In some cases, it is convenient to be able to archive the large, encrypted contents file to secure offline storage, such as a magneto-optical storage system or magnetic tape. This creation of a non-repository archive file is as secure as the encryption process. Such non-repository archive storage is considered a form of "printing" and is controlled by a print right with a specified "archive-printer." An archive-printer device is programmed to save the encrypted contents file (but not the description file) offline in such a way that it can be retrieved.

The Restore Transaction

A Restore transaction is a request to convert an encrypted backup copy of a digital work into a usable copy. A restore operation is intended to be used to compensate for catastrophic media failure. Like all usage rights, restoration rights can include fees and access tests including authorization checks.

- The requester sends the server a message to initiate a Restore transaction. This message indicates the work to be restored, the version of the restore right for the transaction, the destination address information for placing the work, and the file data for the work.
- The server verifies that the contents file is available (i.e. a digital work corresponding to the request has been backed-up.) If it is not, it ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The server retrieves the key from the restoration file. It decrypts the work contents, data, and usage rights.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, a set of default rights for backup files of the original are transmitted by the server.
- The requester stores the digital work.
- The repositories perform the common closing transaction steps.

The Delete Transaction

A Delete transaction deletes a digital work or a number of copies of a digital work from a repository. Practically all digital works would have delete rights.

- The requester sends the server a message to initiate a delete transaction. This message indicates the work to be deleted, the version of the delete right for the transaction.
- The repositories perform the common opening transaction steps.
- The server deletes the file, erasing it from the file system.
- The repositories perform the common closing transaction steps.

The Directory Transaction

A Directory transaction is a request for information about folders, digital works, and their parts. This amounts to roughly the same idea as protection codes in a conventional file system like TENEX, except that it is generalized to the full power of the access specifications of the usage rights language.

The Directory transaction has the important role of passing along descriptions of the rights and fees associated with a digital work. When a user wants to exercise a right, the user interface of his repository implicitly makes a directory request to determine the versions of the right that are available. Typically these are presented to the user -- such as with different choices of billing for exercising a right. Thus, many directory transactions are invisible to the user and are exercised as part of the normal process of exercising all rights.

- The requester sends the server a message to initiate a Directory transaction. This message indicates the file or folder that is the root of the directory request and the version of the directory right used for the transaction.
- The server verifies that the information is accessible to the requester. In particular, it does not return the names of any files that have a HIDE-NAME status in their directory specifications, and it does not return the parts of any folders or files that have HIDE-PARTS in their specification. If the information is not accessible, the server ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The server sends the requested data to the requester according to the transmission protocol.
- The requester records the data.
- The repositories perform the common closing transaction steps.

The Folder Transaction

A Folder transaction is a request to create or rename a folder, or to move a work between folders. Together with Directory rights, Folder rights control the degree to which organization of a repository can be accessed or modified from another repository.

- The requester sends the server a message to initiate a Folder transaction. This message indicates the folder that is the root of the folder request, the version of the folder right for the transaction, an operation, and data. The operation can be one of create, rename, and move file. The data are the specifications required for the operation, such as a specification of a folder or digital work and a name.
- The repositories perform the common opening transaction steps.
- The server performs the requested operation -- creating a folder, renaming a folder, or moving a work between folders.
- The repositories perform the common closing transaction steps.

The Extract Transaction

A extract transaction is a request to copy a part of a digital work and to create a new work containing it. The extraction operation differs from copying in that it can be used to separate a part of a digital work from d-blocks or shells that place additional restrictions or fees on it. The extraction operation differs from the edit operation in that it does not change the contents of a work, only its embedding in d-blocks. Extraction creates a new digital work.

- The requester sends the server a message to initiate an Extract transaction. This message indicates the part of the work to be extracted, the version of the extract right to be used in the transaction, the destination address information for placing the part as a new work, the file data for the work, and the number of copies involved.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the new work. Otherwise, the rights of the original are transmitted. The Copy-Count field for this right is set to the number-of-copies requested.
- The requester records the contents, data, and usage rights and stores the work. It records the date and time that new work was made in the properties of the work.
- The repositories perform the common closing transaction steps.

The Embed Transaction

An embed transaction is a request to make a digital work become a part of another digital work or to add a shell d-block to enable the adding of fees by a distributor of the work.

- The requester sends the server a message to initiate an Embed transaction. This message indicates the work to be embedded, the version of the embed right to be used in the transaction, the destination address information for placing the part as a a work, the file data for the work, and the number of copies involved.
- The server checks the control specifications for all of the rights in the part and the destination. If they are incompatible, the server ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the new work. Otherwise, the rights of the original are transmitted. The Copy-Count field for this right is set to the number-of-copies requested.
- The requester records the contents, data, and usage rights and embeds the work in the destination file.
- The repositories perform the common closing transaction steps.

The Edit Transaction

An Edit transaction is a request to make a new digital work by copying, selecting and modifying portions of an existing digital work. This operation can actually change the contents of a digital work. The kinds of changes that are permitted depend on the process being used. Like the extraction operation, edit operates on portions of a digital work. In contrast with the extract operation, edit does not affect the rights or location of the work. It only changes the contents. The kinds of changes permitted are determined by the type specification of the processor specified in the rights. In the currently preferred embodiment, an edit transaction changes the work itself and does not make a new work. However, it would be a reasonable variation to cause a new copy of the work to be made.

- The requester sends the server a message to initiate an Edit transaction. This message indicates the work to be edited, the version of the edit right to be used in the transaction, the file data for the work (including its size), the process-ID for the process, and the number of copies involved.
- The server checks the compatibility of the process-ID to be used by the requester against any process-ID specification in the right. If they are incompatible, it ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The requester uses the process to change the contents of the digital work as desired. (For example, it can select and duplicate parts of it; combine it with other information; or compute functions based on the information. This can amount to editing text, music, or pictures or taking whatever other steps are useful in creating a derivative work.)
- The repositories perform the common closing transaction steps.

The edit transaction is used to cover a wide range of kinds of works. The category describes a process that takes as its input any portion of a digital work and then modifies the input in some way. For example, for text, a process for editing the text would require edit rights. A process for "summarizing" or counting words in the text would also be considered editing. For a music file, processing could involve changing the pitch or tempo, or adding reverberations, or any other audio effect. For digital video works, anything which alters the image would require edit rights. Examples would be colorizing, scaling, extracting still photos, selecting and combining frames into story boards, sharpening with signal processing, and so on.

Some creators may want to protect the authenticity of their works by limiting the kinds of processes that can be performed on them. If there are no edit rights, then no processing is allowed at all. A processor identifier can be included to specify what kind of process is allowed. If no process identifier is specified, then arbitrary processors can be used. For an example of a specific process, a photographer may want to allow use of his photograph but may not want it to be colorized. A musician may want to allow extraction of portions of his work but not changing of the tonality.

Authorization Transactions

There are many ways that authorization transactions can be defined. In the following, our preferred way is to simply define them in terms of other transactions that we already need for repositories. Thus, it is convenient sometimes to speak of "authorization transactions," but they are actually made up of other transactions that repositories already have.

A usage right can specify an authorization-ID, which identifies an authorization object (a digital work in a file of a standard format) that the repository must have and which it must process. The authorization is given to the generic authorization (or ticket) server of the repository which begins to interpret the authorization.

As described earlier, the authorization contains a server identifier, which may just be the generic authorization server or it may be another server. When a remote authorization server is required, it must contain a digital address. It may also contain a digital certificate.

If a remote authorization server is required, then the authorization process first performs the following steps:

- The generic authorization server attempts to set up the communications channel. (If the channel cannot be set up, then authorization fails with an error.)
- When the channel is set up, it performs a registration process with the remote repository. (If registration fails, then the authorization fails with an error.)
- When registration is complete, the generic authorization server invokes a "Play" transaction with the remote repository, supplying the authorization document as the digital work to be played, and the remote authorization server (a program) as the "player." (If the player cannot be found or has some other error, then the authorization fails with an error.)
- The authorization server then "plays" the authorization. This involves decrypting it using either the public key of the master repository that issued the certificate or the session key from the repository that transmitted it. The authorization server then performs various tests. These tests vary according to the authorization server. They include such steps as checking issue and validity dates of the authorization and checking any hot-lists of known invalid authorizations. The authorization server may require carrying out any other transactions on the repository as well, such as checking directories, getting some person to supply a password, or playing some other digital work. It may also invoke some special process for checking information about locations or recent events. The "script" for such steps is contained within the authorization server.
- If all of the required steps are completed satisfactorily, the authorization server completes the transaction normally, signaling that authorization is granted.

The Install Transaction

An Install transaction is a request to install a digital work as runnable software on a repository. In a typical case, the requester repository is a rendering repository and the software would be a new kind or new version of a player. Also in a typical case, the software would be copied to file system of the requester repository before it is installed.

- The requester sends the server an Install message. This message indicates the work to be installed, the version of the Install right being invoked, and the file data for the work (including its size).
- The repositories perform the common opening transaction steps.
- The requester extracts a copy of the digital certificate for the software. If the certificate cannot be found or the master repository for the certificate is not known to the requester, the transaction ends with an error.
- The requester decrypts the digital certificate using the public key of the master repository, recording the identity of the supplier and creator, a key for decrypting the software, the compatibility information, and a tamper-checking code. (This step certifies the software.)
- The requester decrypts the software using the key from the certificate and computes a check code on it using a 1-way hash function. If the check-code does not match the tamper-checking code from the certificate, the installation transaction ends with an error. (This step assures that the contents of the software, including the various scripts, have not been tampered with.)
- The requester retrieves the instructions in the compatibility-checking script and follows them. If the software is not compatible with the repository, the installation transaction ends with an error. (This step checks platform compatibility.)
- The requester retrieves the instructions in the installation script and follows them. If there is an error in this process (such as insufficient resources), then the transaction ends with an error. Note that the installation process puts the runnable software in a place in the repository where it is no longer accessible as a work for exercising any usage rights other than the execution of the software as part of repository operations in carrying out other transactions.
- The repositories perform the common closing transaction steps.

The Uninstall Transaction

An Uninstall transaction is a request to remove software from a repository. Since uncontrolled or incorrect removal of software from a repository could compromise its behavioral integrity, this step is controlled.

- The requester sends the server an Uninstall message. This message indicates the work to be uninstalled, the version of the Uninstall right being invoked, and the file data for the work (including its size).
- The repositories perform the common opening transaction steps.
- The requester extracts a copy of the digital certificate for the software. If the certificate cannot be found or the master repository for the certificate is not known to the requester, the transaction ends with an error.
- The requester checks whether the software is installed. If the software is not installed, the transaction ends with an error.
- The requester decrypts the digital certificate using the public key of the master repository, recording the identity of the supplier and creator, a key for decrypting the software, the compatibility information, and a tamper-checking code. (This step authenticates the certification of the software, including the script for uninstalling it.)
- The requester decrypts the software using the key from the certificate and computes a check code on it using a 1-way hash function. If the check-code does not match the tamper-checking code from the certificate, the installation transaction ends with an error. (This step assures that the contents of the software, including the various scripts, have not been tampered with.)
- The requester retrieves the instructions in the uninstallation script and follows them. If there is an error in this process (such as insufficient resources), then the transaction ends with an error.
- The repositories perform the common closing transaction steps.

Claims

1. A system for controlling the distribution and use of digital works having a mechanism for reporting fees based on the distribution and use of digital works, said system comprising:

   means for attaching usage rights to a digital work, each of said usage rights specifying how a digital work may be used or distributed, each of said usage rights specifying usage fee information, said usage fee information

comprising a fee type and fee parameters which define a fee to be paid in connection with the exercise of said usage right;
a communication medium for coupling repositories to enable communication between repositories; and
a plurality of repositories, each of said repositories comprising:
an external interface for removably coupling to said communications medium;
storage means for storing digital works having attached usage rights and fees;
requesting means for generating a request to access a digital work stored in another of said plurality of repositories, said request indicating a particular usage right; and
processing means for processing requests to access digital works stored in said storage means and for generating fee transactions when a request indicates a usage right that is attached to a digital work and said usage right specifies usage fee information;
each of said plurality of repositories being removably coupled to a credit server, said credit server being arranged for recording fee transactions from said repository and subsequently reporting said fee transactions to a billing clearinghouse.

2. The fee reporting system as recited in Claim 1 wherein said fee type of said fee information is a metered use fee, a per use fee, a best price fee, a scheduled fee,or a mark-up fee.

3. A method for reporting fees associated with the distribution and use of digital works in a system for controlling the distribution and use of digital works, said method comprising the steps of:

a) attaching one or more usage rights to a digital work, each of said one or more usage rights comprising an indicator of how said digital work may be distributed or used and a usage fee to be paid upon exercise of said right;
b) storing said digital work and attached one or more usage rights in a server repository, said server repository controlling access to said digital work;
c) said server repository receiving a request to access said digital work from a requesting repository;
d) said server repository identifying a usage right associated with said access request;
e) said server repository determining if said identified usage right is the same as one of said one or more usage rights attached to said digital work;
f) if said identified usage right is not the same as any one of said one or more usage rights attached to said digital work, said server repository denying access to said digital work;
g) if said usage right is included with said digital work, said server repository determining if a usage fee is associated with the exercise of said usage right;
h) if a usage fee is associated with usage right, said server repository calculating said usage fee;
i) said server repository transmitting a first assign fee transaction identifying said requesting repository as a payer for said usage fee to a first credit server;
j) said requesting repository transmitting a second assign fee transaction identifying said requesting repository as a payer for said usage fee to a second credit server;
k) said server repository transmitting said digital work to said requesting repository;
l) said server repository transmitting a first confirm fee transaction to said first credit server; and
m) said requesting repository transmitting a second confirm fee transaction to said second credit server.

4. The method as recited in Claim 3 wherein said digital work is comprised of a plurality of independent digital works and said step of said server calculating said usage fee is further comprised of the step of reporting the usage fees for each of the plurality of independent digital works.

5. A method for reporting fees associated with the distribution and use of digital works in a system for controlling the distribution and use of digital works, said method comprising the steps of:

a) attaching one or more usage rights to a digital work, each of said one or more usage rights comprising an indicator of how said digital work may be distributed or used and a usage fee to be paid for exercise of said right;
b) storing said digital work and said attached one or more usage rights in a server repository, said server repository controlling access to said digital work;
c) said server repository receiving a request to access said digital work from a requesting repository;
d) said server repository identifying a usage right associated with said access request;
e) said server repository determining if said digital work has attached thereto said identified usage right;
f) if said identified usage right is not attached to said digital work, said server repository denying access to

said digital work;
g) if said usage right is attached to said digital work, said server repository determining if a usage fee is associated with the exercise of said usage right;
h) if a usage fee is associated with said usage right, said server repository determining a fee type;
i) said server repository transmitting a first fee transaction identifying said requesting repository as a payee for said usage fee to a credit server, said first fee transaction being dependent on said determined fee type; and
k) said server repository transmitting said digital work to said requesting repository.

6. A system for controlling the distribution and utilization of digital works having a mechanism for reporting usage fees, said system comprising:

digital works comprising a first part for storing the digitally encoded data corresponding to a digital work and a second part for storing usage rights and fees for said digital work, said usage rights specifying how a digital work may be used or distributed and said usage fees specifying a fee to be paid in connection with the exercise of a corresponding usage right;
a plurality of repositories, each of said repositories comprising:
communication means for communicating with another of said plurality of repositories;
storage means for storing digital works;
requesting means for generating a request to access a digital work stored in another of said plurality of repositories, said request indicating a particular usage right;
processing means for processing requests to access digital works stored in said storage means and granting access when said particular usage right corresponds to a stored usage right stored in said digital work, said processing means generating fee transactions when said access is granted and said stored usage right specifies a fee;
each of said plurality of repositories being removably coupled to a credit server, said credit server being arranged for recording fee transactions from said repository and subsequently reporting said fee transactions to a billing clearinghouse.

7. The system as recited in Claim 6 wherein said storage means is further comprised of a first storage device for storing said first part of said digital work and a second storage device for storing said second part of said digital work.

8. A method for reporting fees associated with use of rendering digital works by a rendering device in a system for controlling the rendering of digital works by a rendering system, said rendering system comprised of a rendering repository and a rendering device, said rendering device utilizing a rendering digital work for rendering a digital work, said method comprising the steps of:

a) storing a first digital work in a server repository, said digital work specifying a first usage fee to be reported for a use of said first digital work;
b) storing a rendering digital work in said rendering repository, said first rendering digital work specifying a second usage fee to be reported for a use of said rendering digital work;
c) said server repository receiving a request to use said first digital work from said rendering repository;
d) said server repository determining if said request may be granted;
e) if said server repository determines that said request may not be granted, said server repository denying access to said first digital work;
f) if said server repository determines that said request may be granted, said server repository transmitting said digital work to said rendering repository;
g) said server repository transmitting a first fee transaction identifying said rendering repository as a payee for said first usage fee for use of said first digital work to a first credit server;
h) said rendering device rendering said first digital work using said rendering digital work; and
i) said rendering repository transmitting a second fee transaction identifying said rendering repository as a payee for said second usage fee for use of said rendering digital work to a second credit server.

9. The method as recited in Claim 8 further comprising the step of said rendering repository transmitting a third fee transaction identifying said rendering repository as a payee for said first usage fee for use of said first digital work to said second credit server.

10. The method as recited in Claim 9 wherein said rendering digital work is a set of coded rendering instructions for controlling said rendering device.

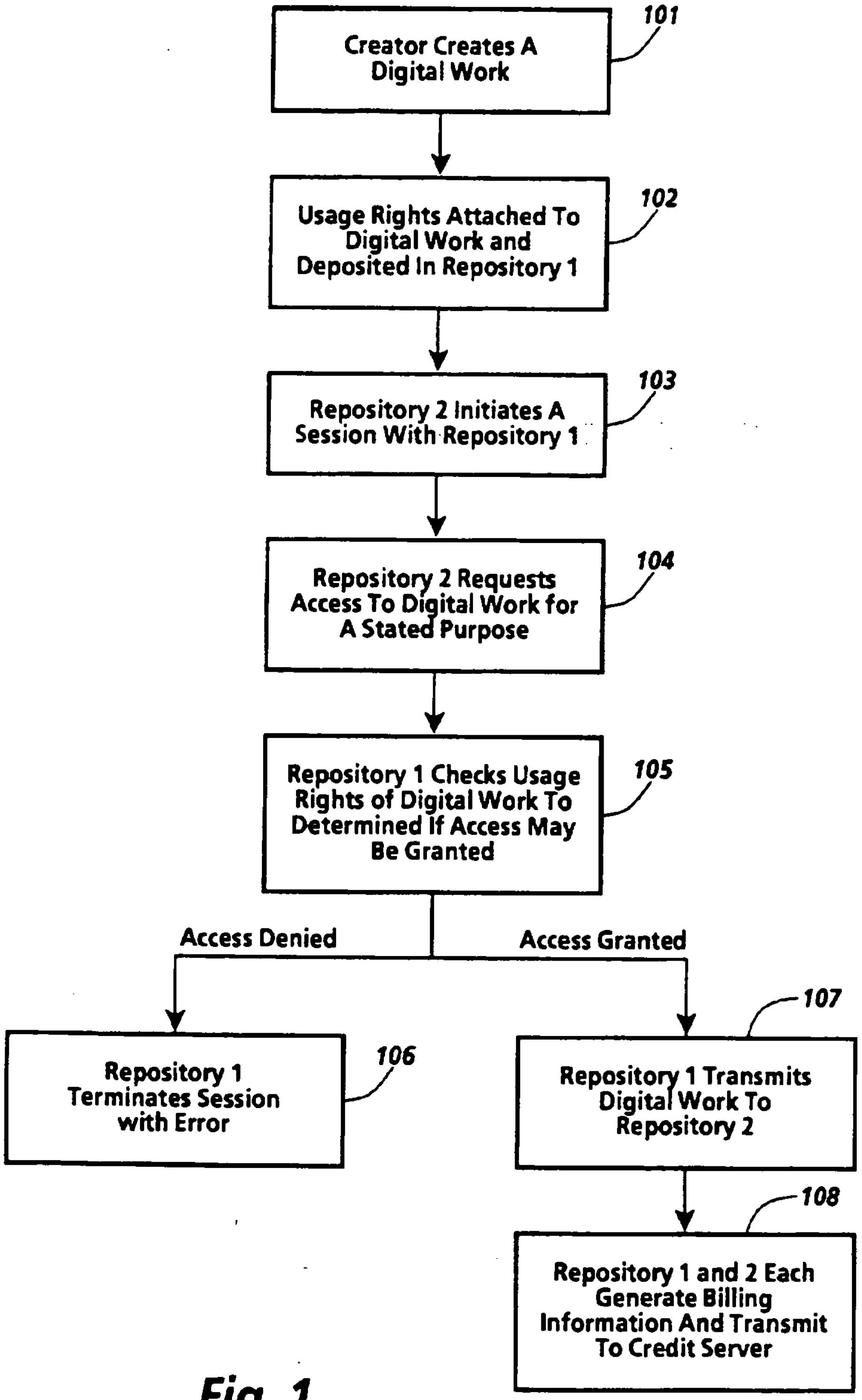

Creator Creates A Digital Work
101
Usage Rights Attached To Digital Work and Deposited In Repository 1
102
Repository 2 Initiates A Session With Repository 1
103
Repository 2 Requests Access To Digital Work for A Stated Purpose
104
Repository 1 Checks Usage Rights of Digital Work To Determined If Access May Be Granted
105
Access Denied
Access Granted
Repository 1 Terminates Session with Error
106
Repository 1 Transmits Digital Work To Repository 2
107
Repository 1 and 2 Each Generate Billing Information And Transmit To Credit Server
108

Fig. 1

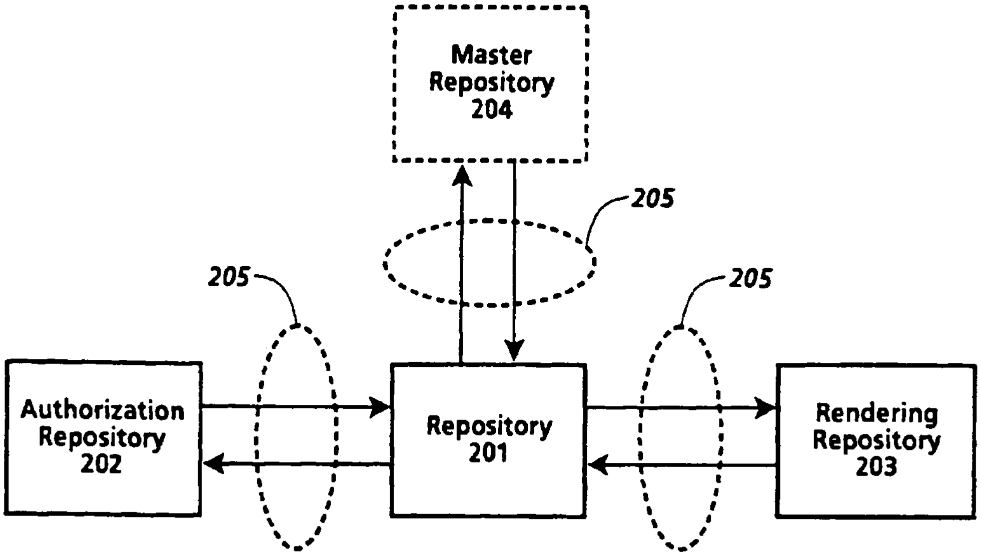

Master Repository 204
205
205
205
Authorization Repository 202
Repository 201
Rendering Repository 203

Fig. 2

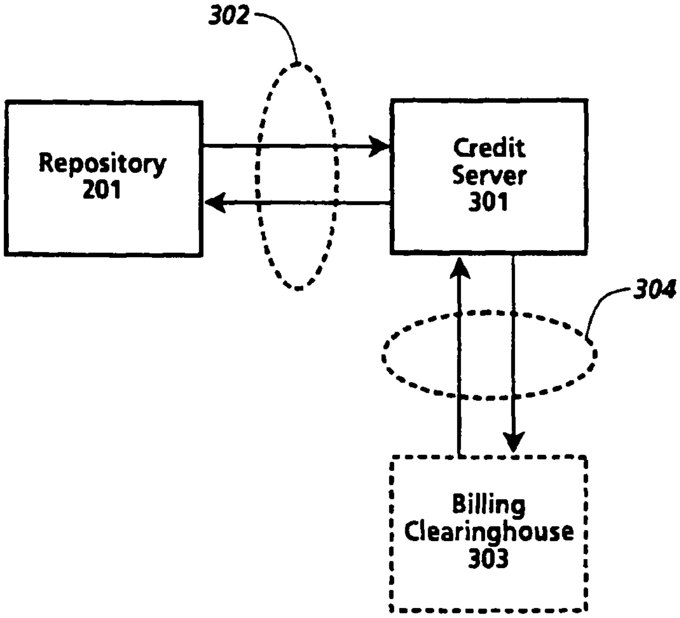
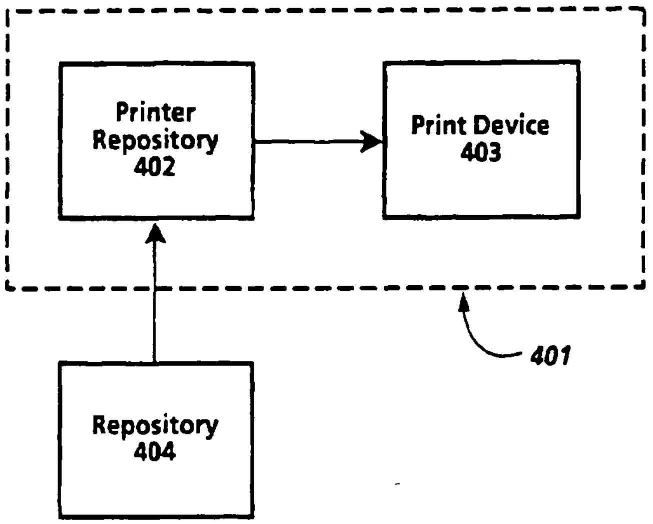
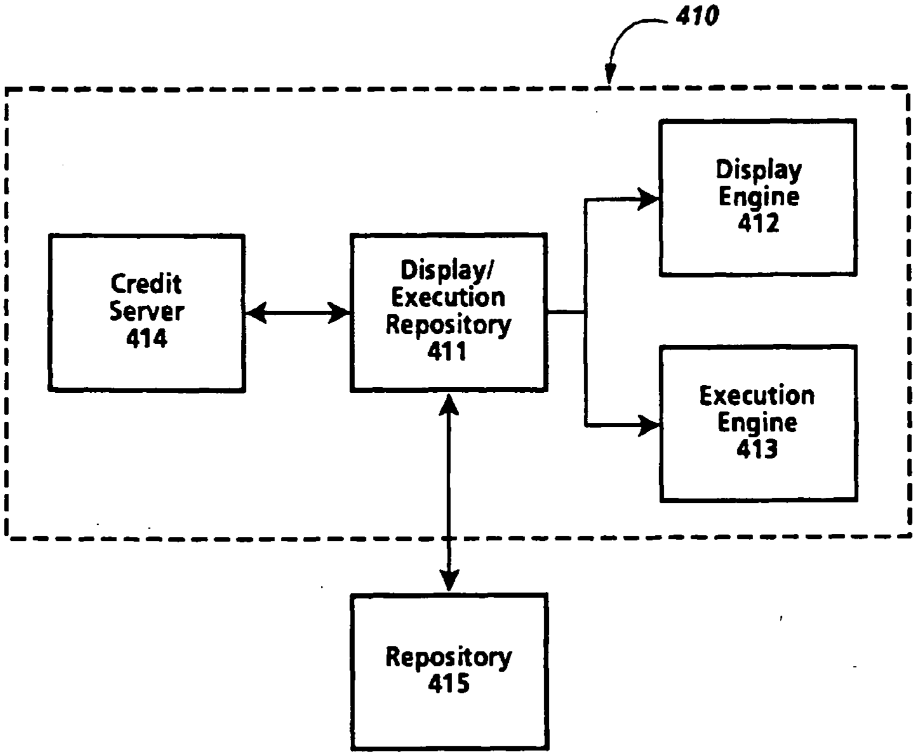

302
Repository 201
Credit Server 301
304
Billing Clearinghouse 303

Fig. 3



Printer Repository 402
Print Device 403
401
Repository 404

Fig. 4a



410
Credit Server 414
Display/ Execution Repository 411
Display Engine 412
Execution Engine 413
Repository 415

Fig. 4b

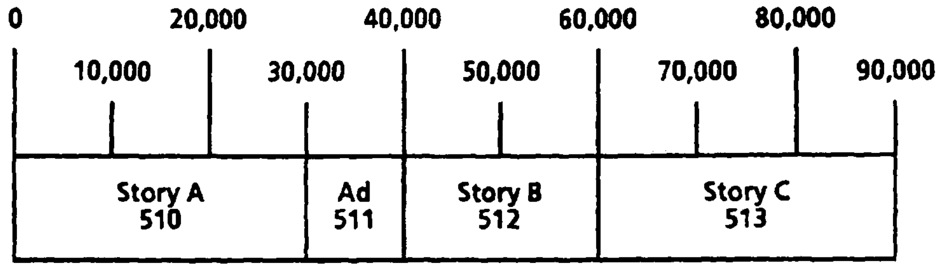

0
20,000
40,000
60,000
80,000
10,000
30,000
50,000
70,000
90,000
Story A
510
Ad
511
Story B
512
Story C
513

Fig. 5

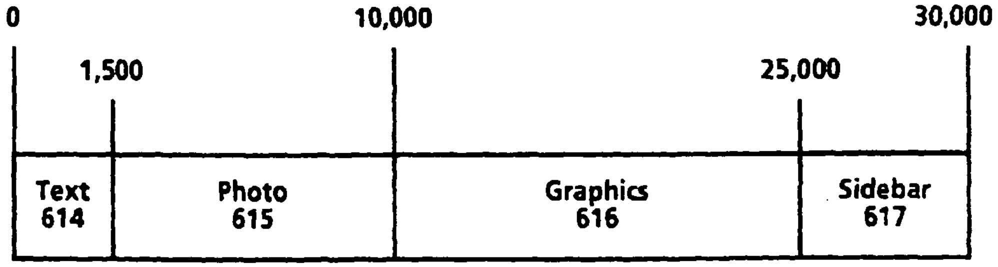

0
10,000
30,000
1,500
25,000
Text
614
Photo
615
Graphics
616
Sidebar
617

Fig. 6

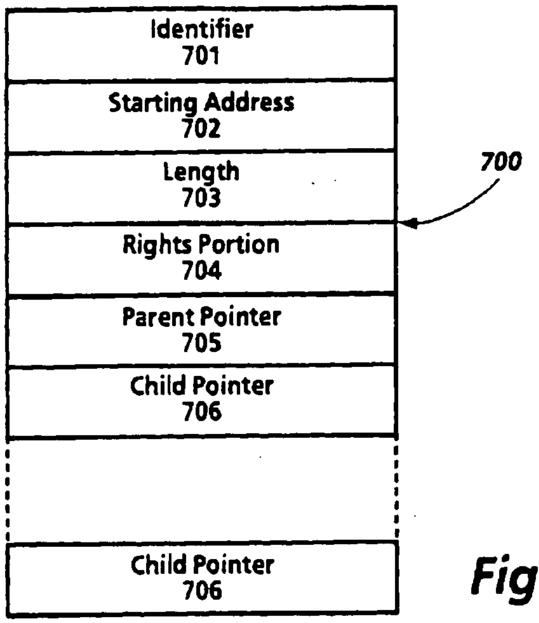

Identifier
701
Starting Address
702
Length
703
Rights Portion
704
Parent Pointer
705
Child Pointer
706
Child Pointer
706
700

Fig. 7

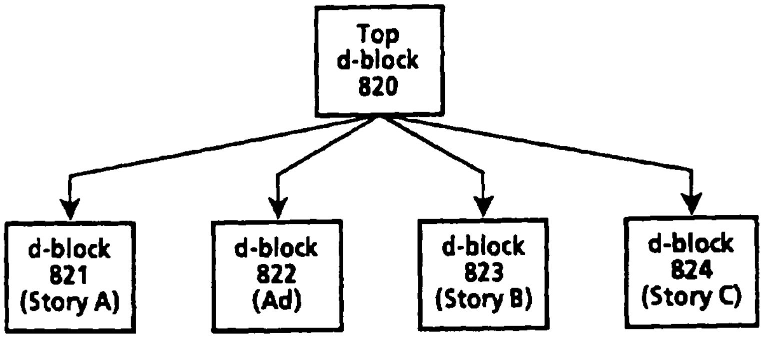

Top
d-block
820
d-block
821
(Story A)
d-block
822
(Ad)
d-block
823
(Story B)
d-block
824
(Story C)

Fig. 8

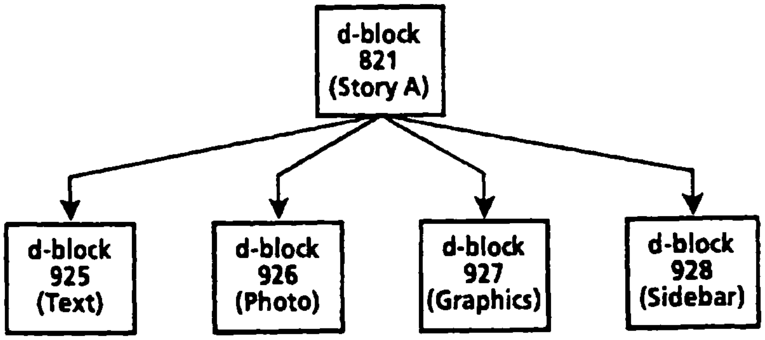

d-block
821
(Story A)
d-block
925
(Text)
d-block
926
(Photo)
d-block
927
(Graphics)
d-block
928
(Sidebar)

Fig. 9

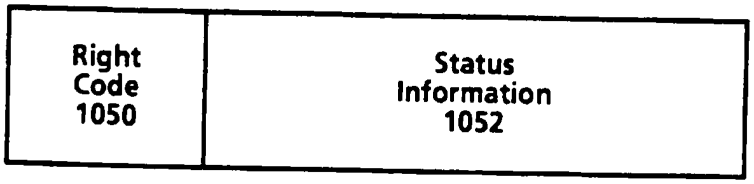

Right Code 1050
Status Information 1052

Fig.10

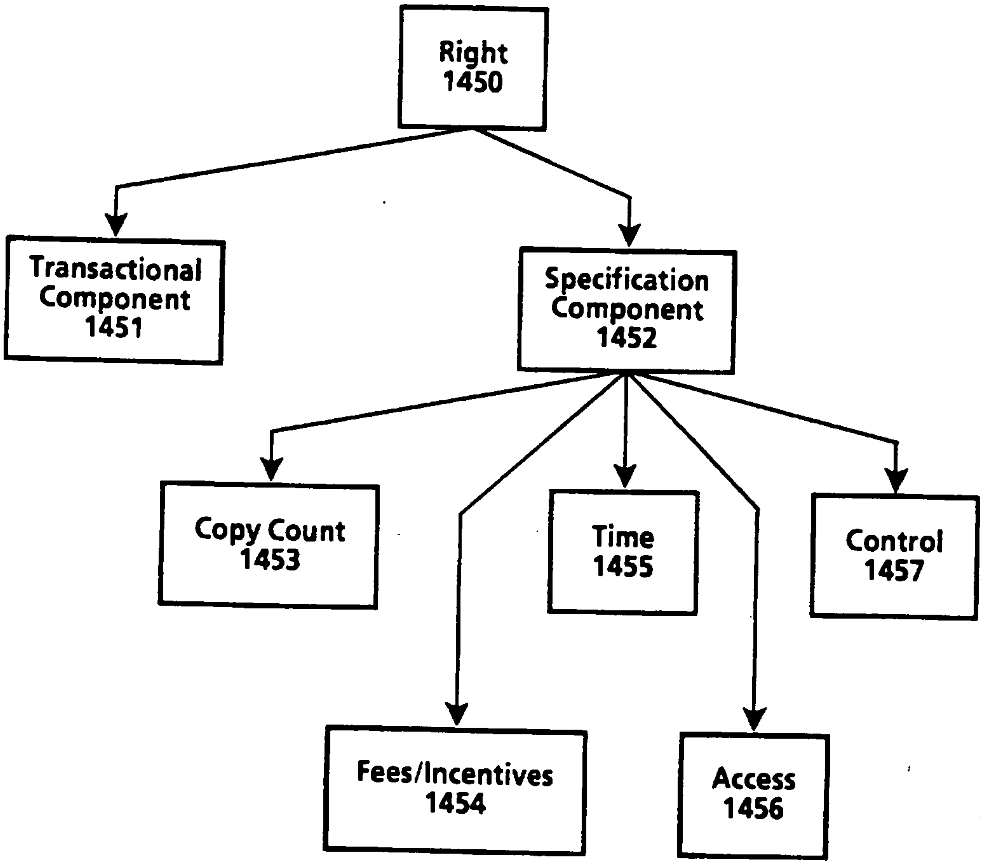

Right 1450
Transactional Component 1451
Specification Component 1452
Copy Count 1453
Time 1455
Control 1457
Fees/Incentives 1454
Access 1456

Fig.14

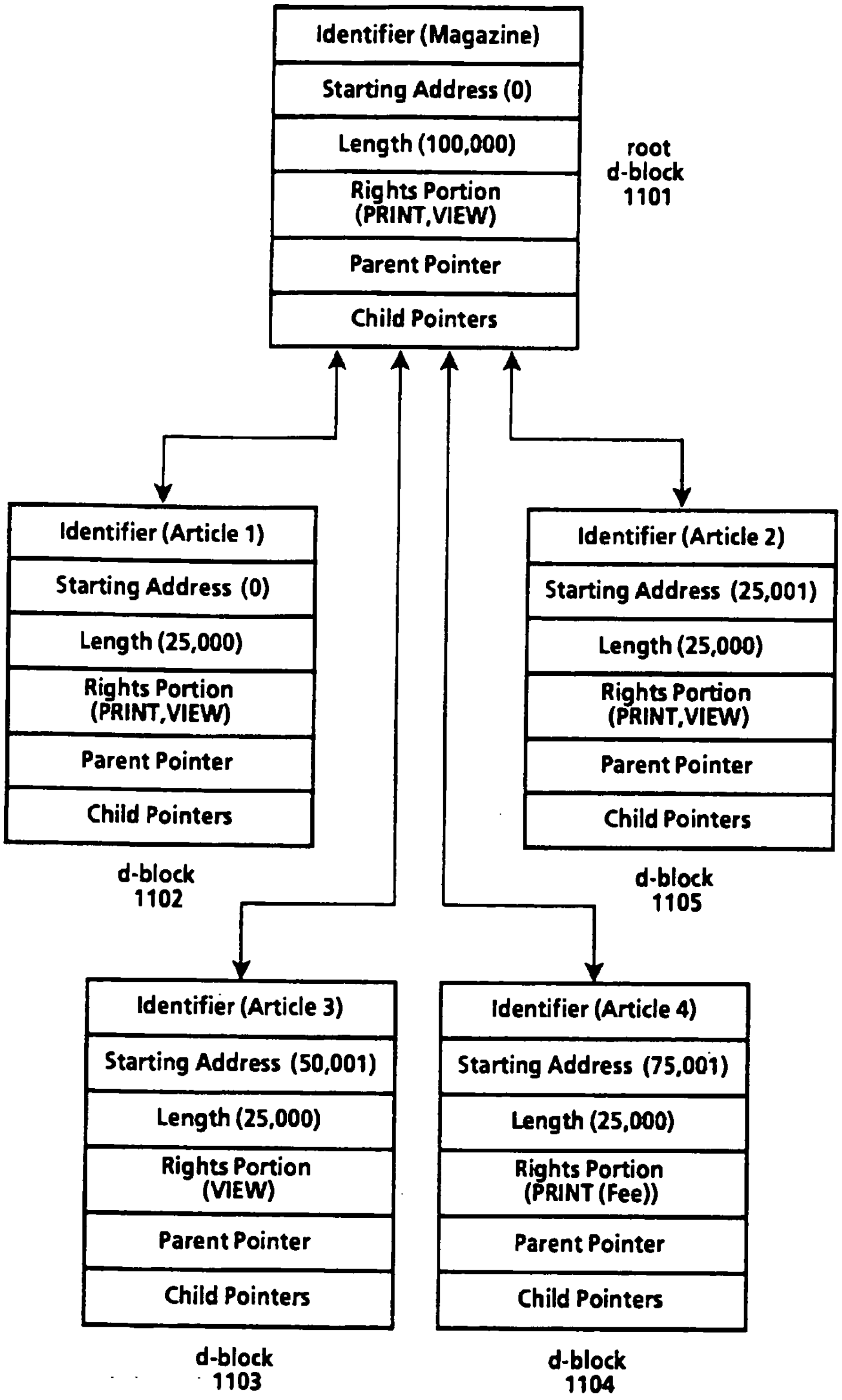

Identifier (Magazine)
Starting Address (0)
Length (100,000)
Rights Portion (PRINT,VIEW)
Parent Pointer
Child Pointers
root d-block 1101
Identifier (Article 1)
Starting Address (0)
Length (25,000)
Rights Portion (PRINT,VIEW)
Parent Pointer
Child Pointers
d-block 1102
Identifier (Article 2)
Starting Address (25,001)
Length (25,000)
Rights Portion (PRINT,VIEW)
Parent Pointer
Child Pointers
d-block 1105
Identifier (Article 3)
Starting Address (50,001)
Length (25,000)
Rights Portion (VIEW)
Parent Pointer
Child Pointers
d-block 1103
Identifier (Article 4)
Starting Address (75,001)
Length (25,000)
Rights Portion (PRINT (Fee))
Parent Pointer
Child Pointers
d-block 1104

Fig.11

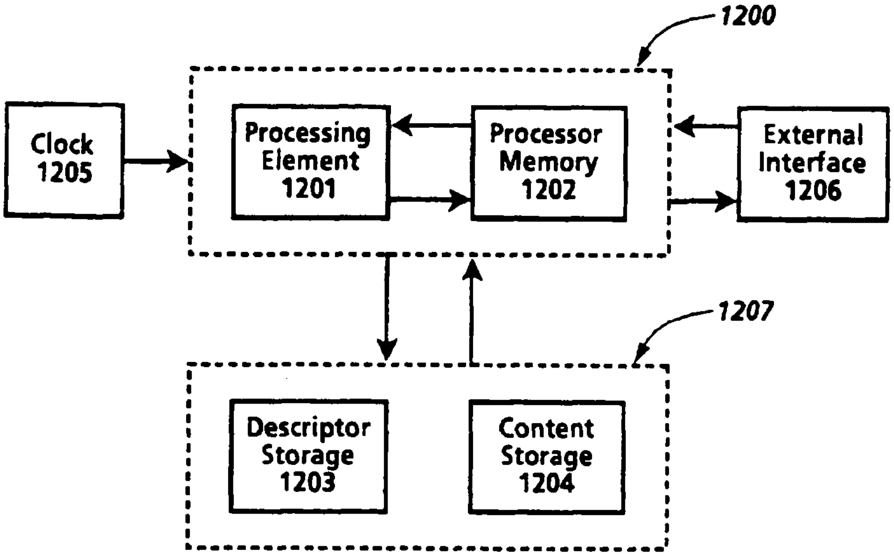

1200
Clock
1205
Processing
Element
1201
Processor
Memory
1202
External
Interface
1206
1207
Descriptor
Storage
1203
Content
Storage
1204

Fig. 12

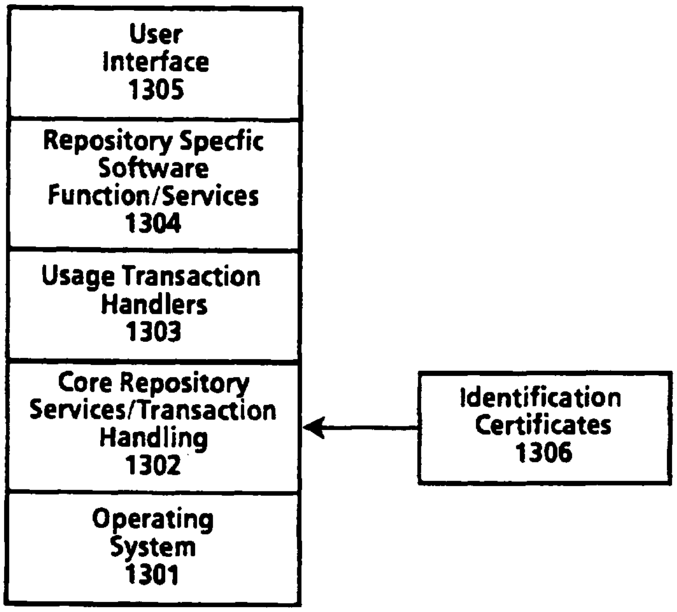

User
Interface
1305
Repository Specfic
Software
Function/Services
1304
Usage Transaction
Handlers
1303
Core Repository
Services/Transaction
Handling
1302
Operating
System
1301
Identification
Certificates
1306

Fig. 13

```
1501~Digital Work Rights: = (Rights*)
 1502~Right := (Right-Code {Copy-Count} {Control-Spec} {Time-Spec }
          {Access-Spec} {Fee-Spec})
    1503~Right-Code := Render-Code | Transport-Code | File-Management-
             Code| Derivative-Works- Code | Configuration-Code
       1504~Render-Code := [ Play : {Player: Player-ID} | Print: {Printer: Printer-ID}]
       1505~Transport-Code := [Copy | Transfer | Loan {Remaining-Rights:
                Next-Set-of-Rights}]{(Next-Copy-Rights: Next-Set-of-Rights)}
       1506~File-Management-Code    := Backup {Back-Up-Copy-Rights:
                                       Next-Set-of-Rights} | Restore | Delete | Folder
                                       | Directory {Name: Hide-Local | Hide-Remote}
                                       {Parts: Hide-Local | Hide-Remote}
       1507~Derivative-Works-Code :=    [Extract | Embed | Edit{Process:
                                          Process-ID}] {Next-Copy-Rights :
                                          Next-Set-of Rights}
       1508~Configuration-Code := Install | Uninstall
       1509~Next-Set-of-Rights := {(Add: Set-Of-Rights)} {(Delete:
                Set-Of-Rights)} {(Replace: Set-Of-Rights )}{(Keep: Set-Of-Rights )}
 1510~Copy-Count := (Copies:positive-integer | 0 | Unlimited)
 1511~Control-Spec := (Control: {Restrictable | Unrestrictable}
                        {Unchargeable | Chargeable})
 1512~Time-Spec := ({Fixed-Interval | Sliding-Interval | Meter-Time}
                     Until: Expiration-Date)
    1513~Fixed-Interval := From: Start-Time
    1514~Sliding-Interval := Interval: Use-Duration
    1515~Meter-Time:= Time-Remaining: Remaining-Use
 1516~Access-Spec := ({SC: Security-Class} {Authorization: Authorization-ID*}
          {Other-Authorization: Authorization-ID*} {Ticket: Ticket-ID})
 1517~Fee-Spec: = {Scheduled-Discount} Regular-Fee-Spec | Scheduled-Fee-Spec |
                   Markup-Spec
    1518~Scheduled-Discount: = Scheduled-Discount: (Scheduled-Discount:
                                 (Time-Spec Percentage)*)
    1519~Regular-Fee-Spec := ({Fee: | Incentive: } [Per-Use-Spec | Metered-Rate-
                               Spec | Best-Price-Spec | Call-For-Price-Spec]
                               {Min: Money-Unit Per: Time-Spec}{Max:
                               Money-Unit Per: Time-Spec} To: Account-ID)
       1520~Per-Use-Spec: = Per-Use: Money-unit
       1521~Metered-Rate-Spec := Metered: Money-Unit Per: Time-Spec
       1522~Best-Price-Spec := Best-Price: Money-unit Max: Money-unit
       1523~Call-For-Price-Spec := Call-For -Price
    1524~Scheduled-Fee-Spec: = (Schedule: (Time-Spec Regular-Fee-Spec)* )
    1525~Markup-Spec: = Markup: percentage To: Account-ID
```

Fig.15

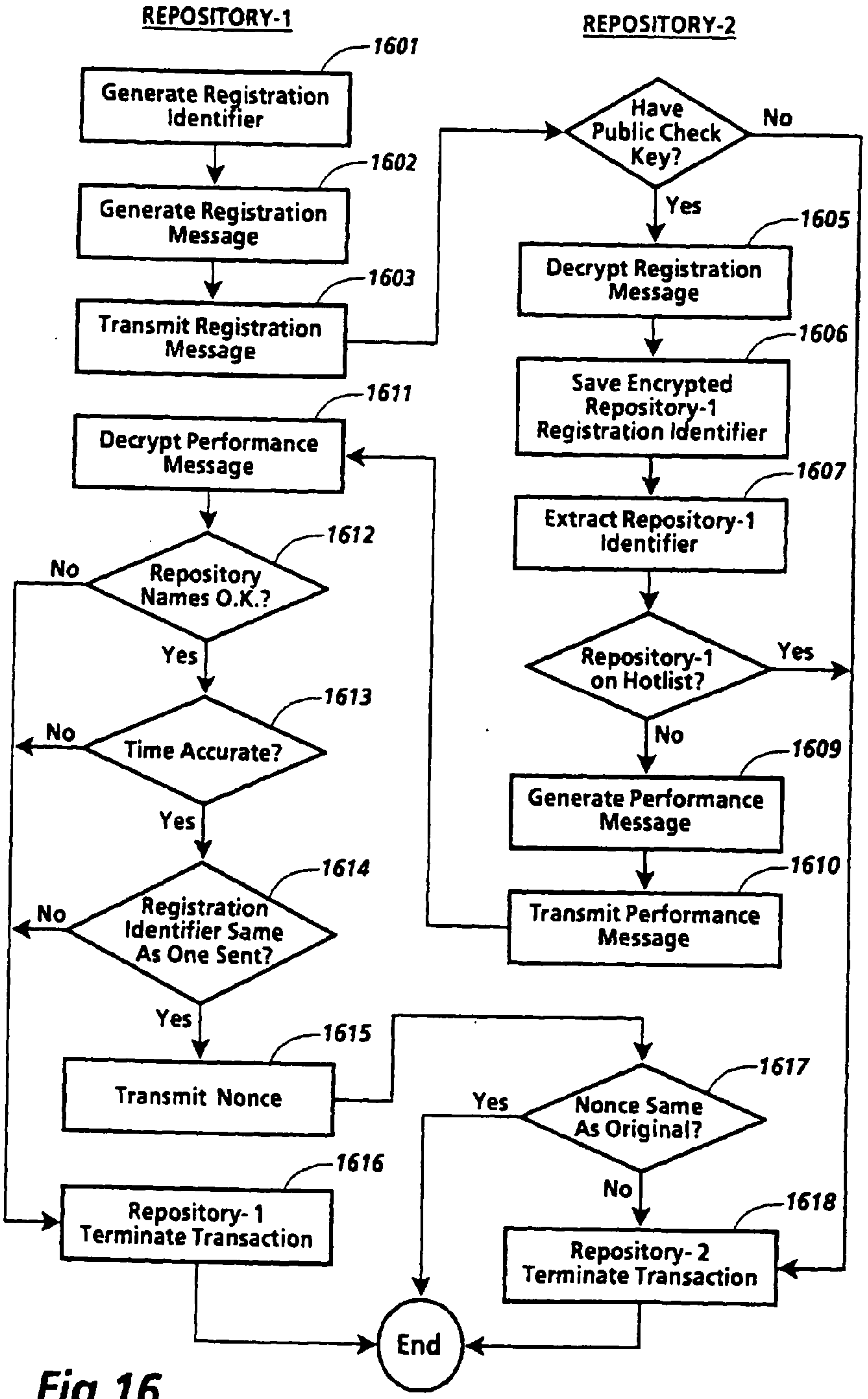

REPOSITORY-1
REPOSITORY-2
1601
Generate Registration Identifier
1602
Generate Registration Message
1603
Transmit Registration Message
Have Public Check Key?
No
Yes
1605
Decrypt Registration Message
1606
Save Encrypted Repository-1 Registration Identifier
1607
Extract Repository-1 Identifier
Repository-1 on Hotlist?
Yes
No
1609
Generate Performance Message
1610
Transmit Performance Message
1611
Decrypt Performance Message
1612
Repository Names O.K.?
No
Yes
1613
Time Accurate?
No
Yes
1614
Registration Identifier Same As One Sent?
No
Yes
1615
Transmit Nonce
1617
Nonce Same As Original?
Yes
No
1616
Repository- 1 Terminate Transaction
1618
Repository- 2 Terminate Transaction
End

Fig.16

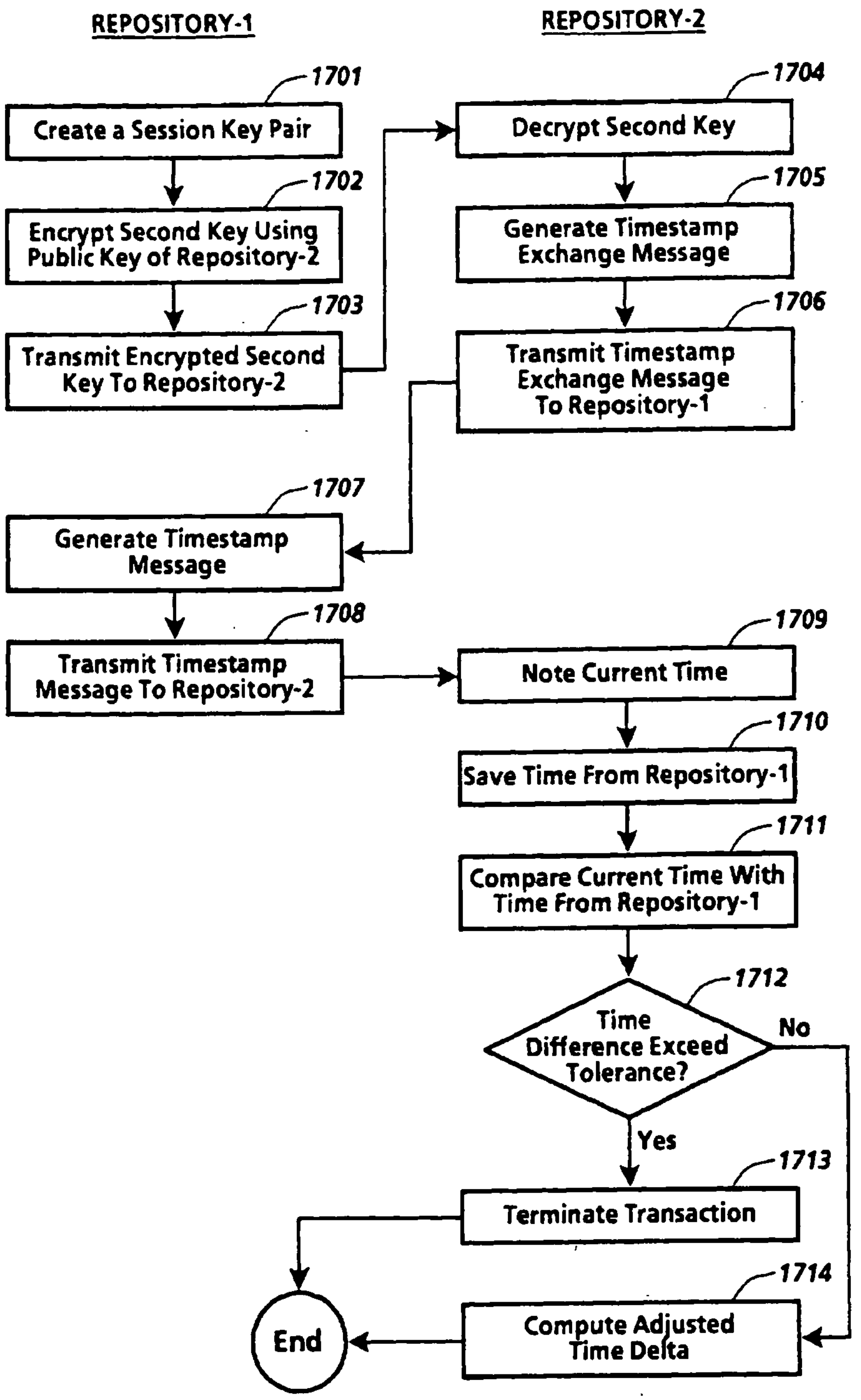

REPOSITORY-1
REPOSITORY-2
1701
Create a Session Key Pair
1702
Encrypt Second Key Using Public Key of Repository-2
1703
Transmit Encrypted Second Key To Repository-2
1704
Decrypt Second Key
1705
Generate Timestamp Exchange Message
1706
Transmit Timestamp Exchange Message To Repository-1
1707
Generate Timestamp Message
1708
Transmit Timestamp Message To Repository-2
1709
Note Current Time
1710
Save Time From Repository-1
1711
Compare Current Time With Time From Repository-1
1712
Time Difference Exceed Tolerance?
No
Yes
1713
Terminate Transaction
1714
Compute Adjusted Time Delta
End

Fig.17

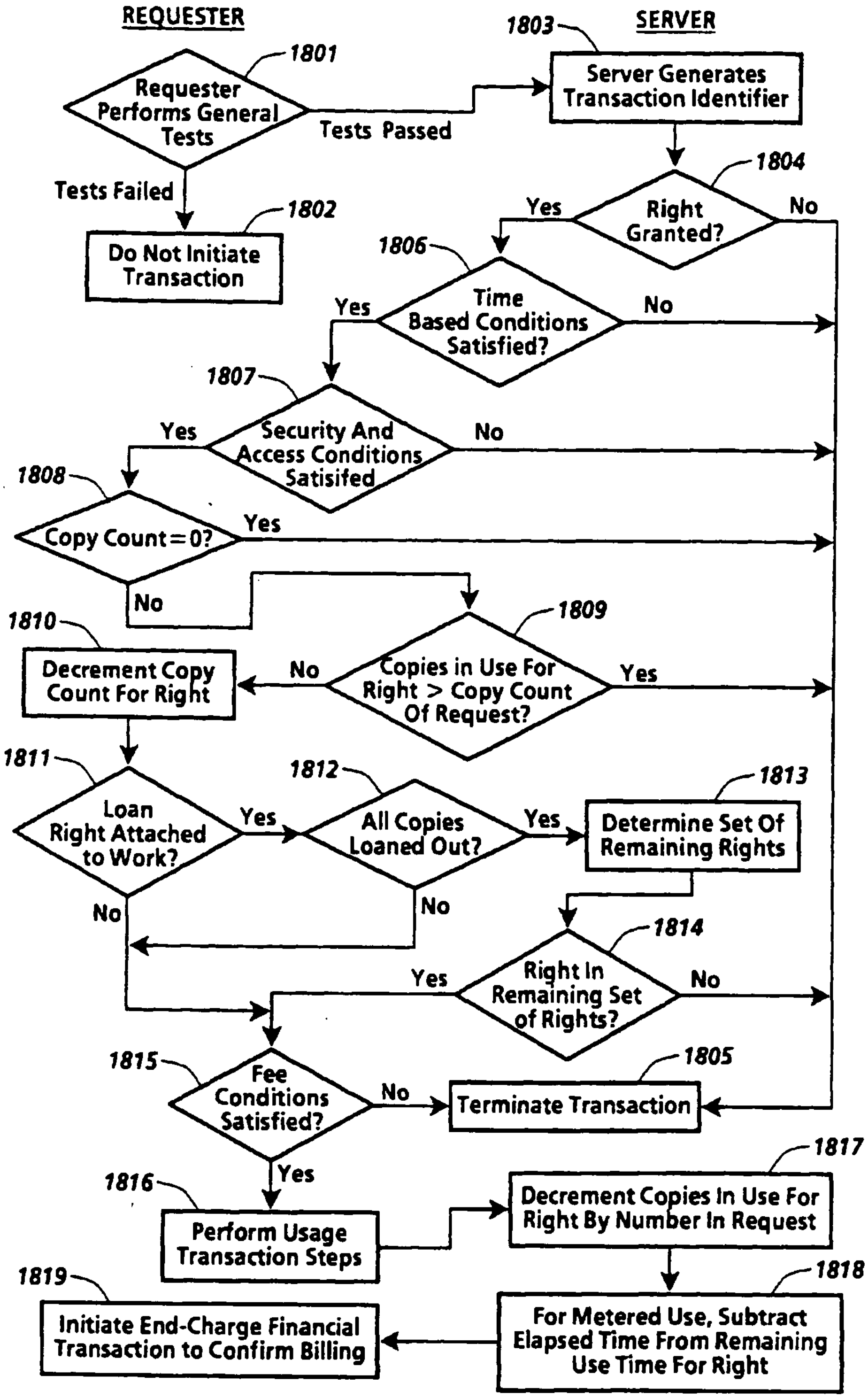

REQUESTER
SERVER
1801
Requester Performs General Tests
Tests Passed
1803
Server Generates Transaction Identifier
Tests Failed
1802
Do Not Initiate Transaction
1804
Right Granted?
Yes
No
1806
Time Based Conditions Satisfied?
Yes
No
1807
Security And Access Conditions Satisifed
Yes
No
1808
Copy Count=0?
Yes
No
1809
Copies in Use For Right > Copy Count Of Request?
No
Yes
1810
Decrement Copy Count For Right
1811
Loan Right Attached to Work?
Yes
No
1812
All Copies Loaned Out?
Yes
No
1813
Determine Set Of Remaining Rights
1814
Right In Remaining Set of Rights?
Yes
No
1815
Fee Conditions Satisfied?
No
1805
Terminate Transaction
Yes
1816
Perform Usage Transaction Steps
1817
Decrement Copies In Use For Right By Number In Request
1818
For Metered Use, Subtract Elapsed Time From Remaining Use Time For Right
1819
Initiate End-Charge Financial Transaction to Confirm Billing

Fig. 18

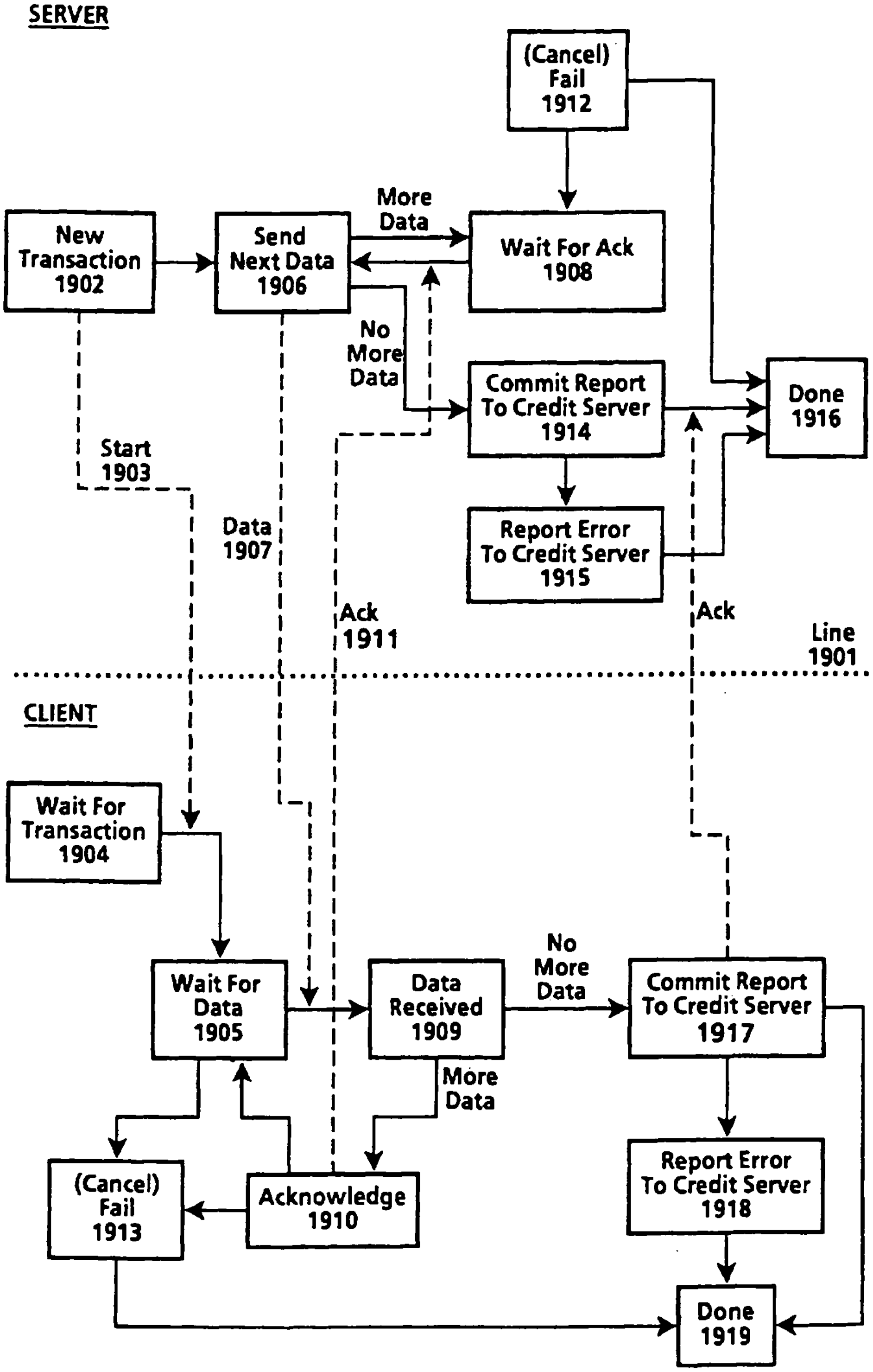

SERVER
(Cancel) Fail 1912
New Transaction 1902
Send Next Data 1906
More Data
Wait For Ack 1908
No More Data
Commit Report To Credit Server 1914
Done 1916
Start 1903
Data 1907
Report Error To Credit Server 1915
Ack 1911
Ack
Line 1901
CLIENT
Wait For Transaction 1904
Wait For Data 1905
Data Received 1909
No More Data
Commit Report To Credit Server 1917
More Data
Report Error To Credit Server 1918
(Cancel) Fail 1913
Acknowledge 1910
Done 1919

Fig.19

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 95 30 8414

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	WO-A-92 20022 (DIGITAL EQUIPMENT CORP.) * page 45, line 10 - page 64, line 17 * ---	1,3,5,6, 8	G06F1/00 G06F17/60
A	TRANSACTIONS OF THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGINEERS OF JAPAN, vol. E73, no. 7, July 1990 TOKYO JP, pages 1133-1146, XP 000159229 MORI ET AL. 'SUPERDISTRIBUTION: THE CONCEPT AND THE ARCHITECTURE' * page 1135, left column, line 17 - page 1136, left column, line 40 * ---	1,3,5,6, 8	
A	US-A-5 291 596 (MITA) * the whole document * ---	1,3,5,6, 8	
A	GB-A-2 236 604 (SUN MICROSYSTEMS INC) * page 9, line 11 - page 20, line 15 * -----	1,3,5,6, 8	TECHNICAL FIELDS SEARCHED (Int.Cl.6) G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	1 April 1996	Moens, R

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 715 244 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
05.06.1996 Bulletin 1996/23

(51) Int Cl.^{6}: **G06F 1/00**

(21) Application number: **95308417.5**

(22) Date of filing: **23.11.1995**

(84) Designated Contracting States:
DE FR GB

(30) Priority: **23.11.1994 US 334041**

(71) Applicant: **XEROX CORPORATION**
Rochester New York 14644 (US)

(72) Inventor: **Stefik, Mark J.**
Woodside, California 94062 (US)

(74) Representative: **Goode, Ian Roy**
Rank Xerox Ltd
Patent Department
Parkway
Marlow Buckinghamshire SL7 1YL (GB)

(54) System for controlling the distribution and use of digital works utilizing a usage rights grammar

(57) A system for controlling use and distribution of digital works. The present invention allows the owner of a digital work to attach usage rights (1450) to their work. The usage rights define how the individual digital work may be used and distributed (1451). Instances of usage rights are defined using a flexible and extensible usage rights grammar. Conceptually, a right in the usage rights grammar is a label associated with a predetermined behavior and conditions to exercising the right. The behavior of a usage right is embodied in a predetermined set (1452) of usage transactions steps. The usage transaction steps further check all conditions (1453-1457) which must be satisfied before the right may be exercised. These usage transaction steps define a protocol for requesting the exercise of a right and the carrying out of a right.

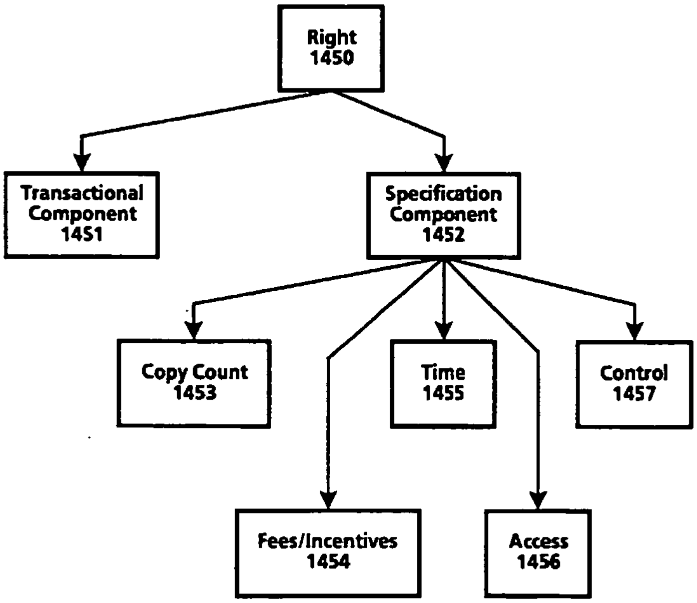


Fig.14

Printed by Jouve, 75001 PARIS (FR)

Description

The present invention relates to the field of distribution and usage rights enforcement for digitally encoded works.

A fundamental issue facing the publishing and information industries as they consider electronic publishing is how to prevent the unauthorized and unaccounted distribution or usage of electronically published materials. Electronically published materials are typically distributed in a digital form and recreated on a computer based system having the capability to recreate the materials. Audio and video recordings, software, books and multimedia works are all being electronically published. Companies in these industries receive royalties for each accounted for delivery of the materials, e.g. the sale of an audio CD at a retail outlet. Any unaccounted distribution of a work results in an unpaid royalty (e.g. copying the audio recording CD to another digital medium.)

The ease in which electronically published works can be "perfectly" reproduced and distributed is a major concern. The transmission of digital works over networks is commonplace. One such widely used network is the Internet. The Internet is a widespread network facility by which computer users in many universities, corporations and government entities communicate and trade ideas and information. Computer bulletin boards found on the Internet and commercial networks such as CompuServ and Prodigy allow for the posting and retrieving of digital information. Information services such as Dialog and LEXIS/NEXIS provide databases of current information on a wide variety of topics. Another factor which will exacerbate the situation is the development and expansion of the National Information Infrastructure (the NII). It is anticipated that, as the NII grows, the transmission of digital works over networks will increase many times over. It would be desirable to utilize the NII for distribution of digital works without the fear of widespread unauthorized copying.

The most straightforward way to curb unaccounted distribution is to prevent unauthorized copying and transmission. For existing materials that are distributed in digital form, various safeguards are used. In the case of software, copy protection schemes which limit the number of copies that can be made or which corrupt the output when copying is detected have been employed. Another scheme causes software to become disabled after a predetermined period of time has lapsed. A technique used for workstation based software is to require that a special hardware device must be present on the workstation in order for the software to run, e.g., see US-A-4,932,054 entitled "Method and Apparatus for Protecting Computer Software Utilizing Coded Filter Network in Conjunction with an Active Coded Hardware Device." Such devices are provided with the software and are commonly referred to as dongles.

Yet another scheme is to distribute software, but which requires a "key" to enable its use. This is employed in distribution schemes where "demos" of the software are provided on a medium along with the entire product. The demos can be freely used, but in order to use the actual product, the key must be purchased. These schemes do not hinder copying of the software once the key is initially purchased.

It is an object of the present invention to provide an improved system and method for controlling the use and distribution of digital works.

The invention accordingly provides a system and method as claimed in the accompanying claims.

A system for controlling use and distribution of digital works is disclosed. A digital work is any written, aural, graphical or video based work that has been translated to or created in a digital form, and which can be recreated using suitable rendering means such as software programs. The present invention allows the owner of a digital work to attach usage rights to their work. The usage rights define how the digital work may be used and distributed. These usage rights become part of the digital work and are always honored.

Instances of usage rights are defined using a flexible and extensible usage rights grammar. Conceptually, a right in the usage rights grammar is a label associated with a predetermined behavior and conditions to exercising the right. For example, a COPY right denotes that a copy of the digital work may be made. A condition to exercising the right is that the requester must pass certain security criteria. Conditions may also be attached to limit the right itself. For example, a LOAN right may be defined so as to limit the duration of which a work may be LOANed.

In the present invention a usage right is comprised of a right code along with the various conditions for exercising the right. Such conditions include a copy-count condition for limiting the number of times a right can be concurrently exercised (e.g. limit the number of copies on loan to some predetermined number), a security class condition for insuring that a repository has an appropriate level of security, access conditions for specifying access tests that must be passed, a time specification for indicating time based constraints for exercising a right and a fee specification for indicating usage fees for the exercise of a right. A digital work may have different versions of a right attached thereto. A version of a right will have the same right code as other versions, but the conditions (and typically the fees) would be different.

Digital works and their attached usage rights are stored in repositories. Digital works are transmitted between repositories. Repositories interact to exchange digital works according to a predetermined set of usage transactions steps. The behavior of a usage right is embodied in a predetermined set of usage transactions steps. The usage transaction steps further check all conditions which must be satisfied before the right may be exercised. These usage transaction steps define a protocol used by the repositories for requesting the exercise of a right and the carrying out of a right.

A system and method in accordance with the invention will now be described, by way of example, with reference to the accompanying drawings, in which:-

Figure 1 is a flowchart illustrating a simple instantiation of the operation of the currently preferred embodiment of the present invention.

Figure 2 is a block diagram illustrating the various repository types and the repository transaction flow between them in the currently preferred embodiment of the present invention.

Figure 3 is a block diagram of a repository coupled with a credit server in the currently preferred embodiment of the present invention.

Figures 4a and 4b are examples of rendering systems as may be utilized in the currently preferred embodiment of the present invention.

Figure 5 illustrates a contents file layout for a digital work as may be utilized in the currently preferred embodiment of the present invention.

Figure 6 illustrates a contents file layout for an individual digital work of the digital work of Figure 5 as may be utilized in the currently preferred embodiment of the present invention.

Figure 7 illustrates the components of a description block of the currently preferred embodiment of the present invention.

Figure 8 illustrates a description tree for the contents file layout of the digital work illustrated in Figure 5.

Figure 9 illustrates a portion of a description tree corresponding to the individual digital work illustrated in Figure 6.

Figure 10 illustrates a layout for the rights portion of a description block as may be utilized in the currently preferred embodiment of the present invention.

Figure 11 is a description tree wherein certain d-blocks have PRINT usage rights and is used to illustrate "strict" and "lenient" rules for resolving usage rights conflicts.

Figure 12 is a block diagram of the hardware components of a repository as are utilized in the currently preferred embodiment of the present invention.

Figure 13 is a block diagram of the functional (logical) components of a repository as are utilized in the currently preferred embodiment of the present invention.

Figure 14 is diagram illustrating the basic components of a usage right in the currently preferred embodiment of the present invention.

Figure 15 lists the usage rights grammar of the currently preferred embodiment of the present invention.

Figure 16 is a flowchart illustrating the steps of certificate delivery, hotlist checking and performance testing as performed in a registration transaction as may be performed in the currently preferred embodiment of the present invention.

Figure 17 is a flowchart illustrating the steps of session information exchange and clock synchronization as may be performed in the currently preferred embodiment of the present invention, after each repository in the registration transaction has successfully completed the steps described in Figure 16.

Figure 18 is a flowchart illustrating the basic flow for a usage transaction, including the common opening and closing step, as may be performed in the currently preferred embodiment of the present invention.

Figure 19 is a state diagram of server and client repositories in accordance with a transport protocol followed when moving a digital work from the server to the client repositories, as may be performed in the currently preferred embodiment of the present invention.

OVERVIEW

A system for controlling use and distribution of digital works is disclosed. The present invention is directed to supporting commercial transactions involving digital works.

Herein the terms "digital work", "work" and "content" refer to any work that has been reduced to a digital representation. This would include any audio, video, text, or multimedia work and any accompanying interpreter (e.g. software) that may be required for recreating the work. The term composite work refers to a digital work comprised of a collection of other digital works. The term "usage rights" or "rights" is a term which refers to rights granted to a recipient of a digital work. Generally, these rights define how a digital work can be used and if it can be further distributed. Each usage right may have one or more specified conditions which must be satisfied before the right may be exercised.

Figure 1 is a high level flowchart omitting various details but which demonstrates the basic operation of the present invention. Referring to Figure 1, a creator creates a digital work, step 101. The creator will then determine appropriate usage rights and fees, attach them to the digital work, and store them in Repository 1, step 102. The determination of appropriate usage rights and fees will depend on various economic factors. The digital work remains securely in Repository 1 until a request for access is received. The request for access begins with a session initiation by another repository. Here a Repository 2 initiates a session with Repository 1, step 103. As will be described in greater detail below, this session initiation includes steps which helps to insure that the respective repositories are trustworthy. Assuming that a session can be established, Repository 2 may then request access to the Digital Work for a stated purpose, step 104. The purpose may be, for example, to print the digital work or to obtain a copy of the digital work. The purpose will correspond to a specific usage right. In any event, Repository 1 checks the usage rights associated with the digital work to determine if the access to the digital work may be granted, step 105. The check of the usage rights essentially involves a determination of whether a right associated with the access request has been attached to the digital work and if all conditions associated with the right are satisfied. If the access is denied, repository 1 terminates the session with an error message, step 106. If access is granted, repository 1 transmits the digital work to repository 2, step 107. Once the digital work has been transmitted to repository 2, repository 1 and 2 each generate billing information for the access which is transmitted to a credit server, step 108. Such double billing reporting is done to insure against attempts to circumvent the billing process.

Figure 2 illustrates the basic interactions between repository types in the present invention. As will become apparent from Figure 2, the various repository types will serve different functions. It is fundamental that repositories will share a core set of functionality which will enable secure and trusted communications. Referring to Figure 2, a repository 201 represents the general instance of a repository. The repository 201 has two modes of operation; a server mode and a requester mode. When in the server mode, the repository will be receiving and processing access requests to digital works. When in the requester mode, the repository will be initiating requests to access digital works. Repository 201 is general in the sense that its primary purpose is as an exchange medium for digital works. During the course of operation, the repository 201 may communicate with a plurality of other repositories, namely authorization repository 202, rendering repository 203 and master repository 204. Communication between repositories occurs utilizing a repository transaction protocol 205.

Communication with an authorization repository 202 may occur when a digital work being accessed has a condition requiring an authorization. Conceptually, an authorization is a digital certificate such that possession of the certificate is required to gain access to the digital work. An authorization is itself a digital work that can be moved between repositories and subjected to fees and usage rights conditions. An authorization may be required by both repositories involved in an access to a digital work.

Communication with a rendering repository 203 occurs in connection with the rendering of a digital work. As will be described in greater detail below, a rendering repository is coupled with a rendering device (e.g. a printer device) to comprise a rendering system.

Communication with a master repository 205 occurs in connection with obtaining an identification certificate. Identification certificates are the means by which a repository is identified as "trustworthy". The use of identification certificates is described below with respect to the registration transaction.

Figure 3 illustrates the repository 201 coupled to a credit server 301. The credit server 301 is a device which accumulates billing information for the repository 201. The credit server 301 communicates with repository 201 via billing transactions 302 to record billing transactions. Billing transactions are reported to a billing clearinghouse 303 by the credit server 301 on a periodic basis. The credit server 301 communicates to the billing clearinghouse 303 via clearinghouse transactions 304. The clearinghouse transactions 304 enable a secure and encrypted transmission of information to the billing clearinghouse 303.

RENDERING SYSTEMS

A rendering system is generally defined as a system comprising a repository and a rendering device which can render a digital work into its desired form. Examples of a rendering system may be a computer system, a digital audio system, or a printer. A rendering system has the same security features as a repository. The coupling of a rendering repository with the rendering device may occur in a manner suitable for the type of rendering device.

Figure 4a illustrates a printer as an example of a rendering system. Referring to Figure 4, printer system 401 has contained therein a printer repository 402 and a print device 403. It should be noted that the the dashed line defining printer system 401 defines a secure system boundary. Communications within the boundary are assumed to be secure. Depending on the security level, the boundary also represents a barrier intended to provide physical integrity. The printer repository 402 is an instantiation of the rendering repository 205 of Figure 2. The printer repository 402 will in some instances contain an ephemeral copy of a digital work which remains until it is printed out by the print engine 403. In other instances, the printer repository 402 may contain digital works such as fonts, which will remain and can be billed based on use. This design assures that all communication lines between printers and printing devices are encrypted, unless they are within a physically secure boundary. This design feature eliminates a potential "fault" point through which the digital work could be improperly obtained. The printer device 403 represents the printer components used to create the printed output.

Also illustrated in Figure 4a is the repository 404. The repository 404 is coupled to the printer repository 402. The repository 404 represents an external repository which contains digital works.

Figure 4b is an example of a computer system as a rendering system. A computer system may constitute a "multi-function" device since it may execute digital works (e.g. software programs) and display digital works (e.g. a digitized photograph). Logically, each rendering device can be viewed as having its own repository, although only one physical repository is needed. Referring to Figure 4b, a computer system 410 has contained therein a display/execution repository 411. The display/execution repository 411 is coupled to display device, 412 and execution device 413. The dashed box surrounding the computer system 410 represents a security boundary within which communications are assumed to be secure. The display/execution repository 411 is further coupled to a credit server 414 to report any fees to be billed for access to a digital work and a repository 415 for accessing digital works stored therein.

STRUCTURE OF DIGITAL WORKS

Usage rights are attached directly to digital works. Thus, it is important to understand the structure of a digital work. The structure of a digital work, in particular composite digital works, may be naturally organized into an acyclic structure such as a hierarchy. For example, a magazine has various articles and photographs which may have been created and are owned by different persons. Each of the articles and photographs may represent a node in a hierarchical structure. Consequently, controls, i.e. usage rights, may be placed on each node by the creator. By enabling control and fee billing to be associated with each node, a creator of a work can be assured that the rights and fees are not circumvented.

In the currently preferred embodiment, the file information for a digital work is divided into two files: a "contents" file and a "description tree" file. From the perspective of a repository, the "contents" file is a stream of addressable bytes whose format depends completely on the interpreter used to play, display or print the digital work. The description tree file makes it possible to examine the rights and fees for a work without reference to the content of the digital work. It should be noted that the term description tree as used herein refers to any type of acyclic structure used to represent the relationship between the various components of a digital work.

Figure 5 illustrates the layout of a contents file. Referring to Figure 5, a digital work is comprised of story A 510, advertisement 511, story B 512 and story C 513. It is assumed that the digital work is stored starting at a relative address of 0. Each of the parts of the digital work are stored linearly so that story A 510 is stored at approximately addresses 0-30,000, advertisement 511 at addresses 30,001-40,000, story B 512 at addresses 40,001-60,000 and story C 513 at addresses 60,001-85K. The detail of story A 510 is illustrated in Figure 6. Referring to Figure 6, the story A 510 is further broken down to show text 614 stored at address 0-1500, soldier photo 615 at addresses 1501-10,000, graphics 616 stored at addresses 10,001-25,000 and sidebar 617 stored address 25,001-30,000. Note that the data in the contents file may be compressed (for saving storage) or encrypted (for security).

From Figures 5 and 6 it is readily observed that a digital work can be represented by its component parts as a hierarchy. The description tree for a digital work is comprised of a set of related descriptor blocks (d-blocks). The contents of each d-block is described with respect to Figure 7. Referring to Figure 7, a d-block 700 includes an identifier 701 which is a unique identifier for the work in the repository, a starting address 702 providing the start address of the first byte of the work, a length 703 giving the number of bytes in the work, a rights portion 704 wherein the granted usage rights and their status data are maintained, a parent pointer 705 for pointing to a parent d-block and child pointers 706 for pointing to the child d-blocks. In the currently preferred embodiment, the identifier 701 has two parts. The first part is a unique number assigned to the repository upon manufacture. The second part is a unique number assigned to the work upon creation. The rights portion 704 will contain a data structure, such as a look-up table, wherein the various information associated with a right is maintained. The information required by the respective usage rights is described in more detail below. D-blocks form a strict hierarchy. The top d-block of a work has no parent; all other d-blocks have one parent. The relationship of usage rights between parent and child d-blocks and how conflicts are resolved is described below.

A special type of d-block is a "shell" d-block. A shell d-block adds no new content beyond the content of its parts. A shell d-block is used to add rights and fee information, typically by distributors of digital works.

Figure 8 illustrates a description tree for the digital work of Figure 5. Referring to Figure 8, a top d-block 820 for the digital work points to the various stories and advertisements contained therein. Here, the top d-block 820 points to d-block 821 (representing story A 510), d-block 822 (representing the advertisement 511), d-block 823 (representing story B 512) and and d-block 824 (representing story C 513).

The portion of the description tree for Story A 510 is illustrated in Figure 9. D-block 925 represents text 614, d-block 926 represents photo 615, d-block 927 represents graphics 616 by and d-block 928 represents sidebar 617.

The rights portion 704 of a descriptor block is further illustrated in Figure 10. Figure 10 illustrates a structure which is repeated in the rights portion 704 for each right. Referring to Figure 10, each right will have a right code field 1050 and status information field 1052. The right code field 1050 will contain a unique code assigned to a right. The status information field 1052 will contain information relating to the state of a right and the digital work. Such information is indicated below in Table 1. The rights as stored in the rights portion 704 may typically be in numerical order based on the right code.

TABLE 1

DIGITAL WORK STATE INFORMATION		
Property	**Value**	**Use**
Copies-in-Use	Number	A counter of the number of copies of a work that are in use. Incremented when another copy is used; decremented when use is completed.
Loan-Period	Time-Units	Indicator of the maximum number of time-units that a document can be loaned out
Loaner-Copy	Boolean	Indicator that the current work is a loaned out copy of an authorized digital work.
Remaining-Time	Time-Units	Indicator of the remaining time of use on a metered document right.
Document-Descr	String	A string containing various identifying information about a document. The exact format of this is not specified, but it can include information such as a publisher name, author name, ISBN number, and so on.
Revenue-Owner	RO-Descr	A handle identifying a revenue owner for a digital work. This is used for reporting usage fees.
Publication-Date	Date-Descr	The date that the digital work was published.
History-list	History-Rec	A list of events recording the repostories and dates for operations that copy, transfer, backup, or restore a digital work.

The approach for representing digtal works by separating description data from content assumes that parts of a file are contiguous but takes no position on the actual representation of content. In particular, it is neutral to the question of whether content representation may take an object oriented approach. It would be natural to represent content as objects. In principle, it may be convenient to have content objects that include the billing structure and rights information that is represented in the d-blocks. Such variations in the design of the representation are possible and are viable alternatives but may introduce processing overhead, e.g. the interpretation of the objects.

Digital works are stored in a repository as part of a hierarchical file system. Folders (also termed directories and sub-directories) contain the digital works as well as other folders. Digital works and folders in a folder are ordered in alphabetical order. The digital works are typed to reflect how the files are used. Usage rights can be attached to folders so that the folder itself is treated as a digital work. Access to the folder would then be handled in the same fashion as any other digital work As will be described in more detail below, the contents of the folder are subject to their own rights. Moreover, file management rights may be attached to the folder which define how folder contents can be managed.

ATTACHING USAGE RIGHTS TO A DIGITAL WORK

It is fundamental to the present invention that the usage rights are treated as part of the digital work. As the digital work is distributed, the scope of the granted usage rights will remain the same or may be narrowed. For example, when a digital work is transferred from a document server to a repository, the usage rights may include the right to loan a copy for a predetermined period of time (called the original rights). When the repository loans out a copy of the digital work, the usage rights in the loaner copy (called the next set of rights) could be set to prohibit any further rights to loan out the copy. The basic idea is that one cannot grant more rights than they have.

The attachment of usage rights into a digital work may occur in a variety of ways. If the usage rights will be the same for an entire digital work, they could be attached when the digital work is processed for deposit in the digital work server. In the case of a digital work having different usage rights for the various components, this can be done as the digital work is being created. An authoring tool or digital work assembling tool could be utilized which provides for an automated process of attaching the usage rights.

As will be described below, when a digital work is copied, transferred or loaned, a "next set of rights" can be specified. The"next set of rights" will be attached to the digital work as it is transported.

Resolving Conflicting Rights

Because each part of a digital work may have its own usage rights, there will be instances where the rights of a "contained part" are different from its parent or container part. As a result, conflict rules must be established to dictate when and how a right may be exercised. The hierarchical structure of a digital work facilitates the enforcement of such rules. A "strict" rule would be as follows: a right for a part in a digital work is sanctioned if and only if it is sanctioned for the part, for ancestor d-blocks containing the part and for all descendent d-blocks. By sanctioned, it is meant that (1) each of the respective parts must have the right, and (2) any conditions for exercising the right are satisfied.

It also possible to implement the present invention using a more lenient rule. In the more lenient rule, access to the part may be enabled to the descendent parts which have the right, but access is denied to the descendents which do not.

An example of applying both the strict rule and lenient is illustrated with reference to Figure 11. Referring to Figure 11, a root d-block 1101 has child d-blocks 1102-1105. In this case, root d-block represents a magazine, and each of the child d-blocks 1102-1105 represent articles in the magazine. Suppose that a request is made to PRINT the digital work represented by root d-block 1101 wherein the strict rule is followed. The rights for the root d-block 1101 and child d-blocks 1102-1105 are then examined. Root d-block 1101 and child d-blocks 1102 and 1105 have been granted PRINT rights. Child d-block 1103 has not been granted PRINT rights and child d-block 1104 has PRINT rights conditioned on payment of a usage fee.

Under the strict rule the PRINT right cannot be exercised because the child d-block does not have the PRINT right. Under the lenient rule, the result would be different. The digital works represented by child d-blocks 1102 and 1105 could be printed and the digital work represented by d-block 1104 could be printed so long as the usage fee is paid. Only the digital work represented by d-block 1103 could not be printed. This same result would be accomplished under the strict rule if the requests were directed to each of the individual digital works.

The present invention supports various combinations of allowing and disallowing access. Moreover, as will be described below, the usage rights grammar permits the owner of a digital work to specify if constraints may be imposed on the work by a container part. The manner in which digital works may be sanctioned because of usage rights conflicts would be implementation specific and would depend on the nature of the digital works.

REPOSITORIES

In the description of Figure 2, it was indicated that repositories come in various forms. All repositories provide a core set of services for the transmission of digital works. The manner in which digital works are exchanged is the basis for all transaction between repositories. The various repository types differ in the ultimate functions that they perform. Repositories may be devices themselves, or they may be incorporated into other systems. An example is the rendering repository 203 of Figure 2.

A repository will have associated with it a repository identifier. Typically, the repository identifier would be a unique number assigned to the repository at the time of manufacture. Each repository will also be classified as being in a particular security class. Certain communications and transactions may be conditioned on a repository being in a particular security class. The various security classes are described in greater detail below.

As a prerequisite to operation, a repository will require possession of an identification certificate. Identification certificates are encrypted to prevent forgery and are issued by a Master repository. A master repository plays the role of an authorization agent to enable repositories to receive digital works. Identification certificates must be updated on a periodic basis. Identification certificates are described in greater detail below with respect to the registration transaction.

A repository has both a hardware and functional embodiment. The functional embodiment is typically software executing on the hardware embodiment. Alternatively, the functional embodiment may be embedded in the hardware embodiment such as an Application Specific Integrated Circuit (ASIC) chip.

The hardware embodiment of a repository will be enclosed in a secure housing which if compromised, may cause the repository to be disabled. The basic components of the hardware embodiment of a repository are described with reference to Figure 12. Referring to Figure 12, a repository is comprised of a processing means 1200, storage system 1207, clock 1205 and external interface 1206. The processing means 1200 is comprised of a processor element 1201 and processor memory 1202. The processing means 1201 provides controller, repository transaction and usage rights transaction functions for the repository. Various functions in the operation of the repository such as decryption and/or decompression of digital works and transaction messages are also performed by the processing means 1200. The processor element 1201 may be a microprocessor or other suitable computing component. The processor memory 1202 would typically be further comprised of Read Only Memories (ROM) and Random Access Memories (RAM). Such memories would contain the software instructions utilized by the processor element 1201 in performing the functions of the repository.

The storage system 1207 is further comprised of descriptor storage 1203 and content storage 1204. The description tree storage 1203 will store the description tree for the digital work and the content storage will store the associated content. The description tree storage 1203 and content storage 1204 need not be of the same type of storage medium, nor are they necessarily on the same physical device. So for example, the descriptor storage 1203 may be stored on a solid state storage (for rapid retrieval of the description tree information), while the content storage 1204 may be on a high capacity storage such as an optical disk.

The clock 1205 is used to time-stamp various time based conditions for usage rights or for metering usage fees which may be associated with the digital works. The clock 1205 will have an uninterruptable power supply, e.g. a battery, in order to maintain the integrity of the time-stamps. The external interface means 1206 provides for the signal connection to other repositories and to a credit server. The external interface means 1206 provides for the exchange of signals via such standard interfaces such as RS-232 or Personal Computer Manufacturers Card Industry Association (PCMCIA) standards, or FDDI. The external interface means 1206 may also provide network connectivity.

The functional embodiment of a repository is described with reference to Figure 13. Referring to Figure 13, the functional embodiment is comprised of an operating system 1301, core repository services 1302, usage transaction handlers 1303, repository specific functions, 1304 and a user interface 1305. The operating system 1301 is specific to the repository and would typically depend on the type of processor being used. The operating system 1301 would also provide the basic services for controlling and interfacing between the basic components of the repository.

The core repository services 1302 comprise a set of functions required by each and every repository. The core repository services 1302 include the session initiation transactions which are defined in greater detail below. This set of services also includes a generic ticket agent which is used to "punch" a digital ticket and a generic authorization server for processing authorization specifications. Digital tickets and authorizations are specific mechanisms for controlling the distribution and use of digital works and are described in more detail below. Note that coupled to the core repository services are a plurality of identification certificates 1306. The identification certificates 1306 are required to enable the use of the repository.

The usage transactions handlers 1303 comprise functionality for processing access requests to digital works and for billing fees based on access. The usage transactions supported will be different for each repository type. For example, it may not be necessary for some repositories to handle access requests for digital works.

The repository specific functionality 1304 comprises functionality that is unique to a repository. For example, the master repository has special functionality for issuing digital certificates and maintaining encryption keys. The repository specific functionality 1304 would include the user interface implementation for the repository.

Repository Security Classes

For some digital works the losses caused by any individual instance of unauthorized copying is insignificant and the chief economic concern lies in assuring the convenience of access and low-overhead billing. In such cases, simple and inexpensive handheld repositories and network-based workstations may be suitable repositories, even though the measures and guarantees of security are modest.

At the other extreme, some digital works such as a digital copy of a first run movie or a bearer bond or stock certificate would be of very high value so that it is prudent to employ caution and fairly elaborate security measures to ensure that they are not copied or forged. A repository suitable for holding such a digital work could have elaborate measures for ensuring physical integrity and for verifying authorization before use.

By arranging a universal protocol, all kinds of repositories can communicate with each other in principle. However, creators of some works will want to specify that their works will only be transferred to repositories whose level of security is high enough. For this reason, document repositories have a ranking system for classes and levels of security. The security classes in the currently preferred embodiment are described in Table 2.

TABLE 2

REPOSITORY SECURITY LEVELS	
Level	Description of Security
0	Open system. Document transmission is unencrypted. No digital certificate is required for identification. The security of the system depends mostly on user honesty, since only modest knowledge may be needed to circumvent the security measures. The repository has no provisions for preventing unauthorized programs from running and accessing or copying files. The system does not prevent the use of removable storage and does not encrypt stored files.
1	Minimal security. Like the previous class except that stored files are minimally encrypted, including ones on removable storage.
2	Basic security. Like the previous class except that special tools and knowledge are required to compromise the programming, the contents of the repository, or the state of the clock. All digital communications are encrypted. A digital certificate is provided as identification. Medium level encryption is used. Repository identification number is unforgeable.

TABLE 2 (continued)

REPOSITORY SECURITY LEVELS	
Level	Description of Security
3	General security. Like the previous class plus the requirement of special tools are needed to compromise the physical integrity of the repository and that modest encryption is used on all transmissions. Password protection is required to use the local user interface. The digital clock system cannot be reset without authorization. No works would be stored on removable storage. When executing works as programs, it runs them in their own address space and does not give them direct access to any file storage or other memory containing system code or works. They can access works only through the transmission transaction protocol.
4	Like the previous class except that high level encryption is used on all communications. Sensors are used to record attempts at physical and electronic tampering. After such tampering, the repository will not perform other transactions until it has reported such tampering to a designated server.
5	Like the previous class except that if the physical or digital attempts at tampering exceed some preset threshold that threaten the physical integrity of the repository or the integrity of digital and cryptographic barriers, then the repository will save only document description records of history but will erase or destroy any digital identifiers that could be misused if released to an unscrupulous party. It also modifies any certificates of authenticity to indicate that the physical system has been compromised. It also erases the contents of designated documents.
6	Like the previous class except that the repository will attempt wireless communication to report tampering and will employ noisy alarms.
10	This would correspond to a very high level of security. This server would maintain constant communications to remote security systems reporting transactions, sensor readings, and attempts to circumvent security.

The characterization of security levels described in Table 2 is not intended to be fixed. More important is the idea of having different security levels for different repositories. It is anticipated that new security classes and requirements will evolve according to social situations and changes in technology.

Repository User Interface

A user interface is broadly defined as the mechanism by which a user interacts with a repository in order to invoke transactions to gain access to a digital work, or exercise usage rights. As described above, a repository may be embodied in various forms. The user interface for a repository will differ depending on the particular embodiment. The user interface may be a graphical user interface having icons representing the digital works and the various transactions that may be performed. The user interface may be a generated dialog in which a user is prompted for information.

The user interface itself need not be part of the repository. As a repository may be embedded in some other device, the user interface may merely be a part of the device in which the repository is embedded. For example, the repository could be embedded in a "card" that is inserted into an available slot in a computer system. The user interface may be a combination of a display, keyboard, cursor control device and software executing on the computer system.

At a minimum, the user interface must permit a user to input information such as access requests and alpha numeric data and provide feedback as to transaction status. The user interface will then cause the repository to initiate the suitable transactions to service the request. Other facets of a particular user interface will depend on the functionality that a repository will provide.

CREDIT SERVERS

In the present invention, fees may be associated with the exercise of a right. The requirement for payment of fees is described with each version of a usage right in the usage rights language. The recording and reporting of such fees is performed by the credit server. One of the capabilities enabled by associating fees with rights is the possibility of supporting a wide range of charging models. The simplest model, used by conventional software, is that there is a single fee at the time of purchase, after which the purchaser obtains unlimited rights to use the work as often and for as long as he or she wants. Alternative models, include metered use and variable fees. A single work can have different fees for different uses. For example, viewing a photograph on a display could have different fees than making a hardcopy or including it in a newly created work. A key to these alternative charging models is to have a low overhead means of establishing fees and accounting for credit on these transactions.

A credit server is a computational system that reliably authorizes and records these transactions so that fees are billed and paid. The credit server reports fees to a billing clearinghouse. The billing clearinghouse manages the financial transactions as they occur. As a result, bills may be generated and accounts reconciled. Preferably, the credit server would store the fee transactions and periodically communicate via a network with the billing clearinghouse for reconciliation. In such an embodiment, communications with the billing clearinghouse would be encrypted for integrity and security reasons. In another embodiment, the credit server acts as a "debit card" where transactions occur in "real-time" against a user account.

A credit server is comprised of memory, a processing means, a clock, and interface means for coupling to a repository and a financial institution (e.g. a modem). The credit server will also need to have security and authentication functionality. These elements are essentially the same elements as those of a repository. Thus, a single device can be both a repository and a credit server, provided that it has the appropriate processing elements for carrying out the corresponding functions and protocols. Typically, however, a credit server would be a card-sized system in the possession of the owner of the credit. The credit server is coupled to a repository and would interact via financial transactions as described below. Interactions with a financial institution may occur via protocols established by the financial institutions themselves.

In the currently preferred embodiment credit servers associated with both the server and the repository report the financial transaction to the billing clearinghouse. For example, when a digital work is copied by one repository to another for a fee, credit servers coupled to each of the repositories will report the transaction to the billing clearinghouse. This is desirable in that it insures that a transaction will be accounted for in the event of some break in the communication between a credit server and the billing clearinghouse. However, some implementations may embody only a single credit server reporting the transaction to minimize transaction processing at the risk of losing some transactions.

USAGE RIGHTS LANGUAGE

The present invention uses statements in a high level "usage rights language" to define rights associated with digital works and their parts. Usage rights statements are interpreted by repositories and are used to determine what transactions can be successfully carried out for a digital work and also to determine parameters for those transactions. For example, sentences in the language determine whether a given digital work can be copied, when and how it can be used, and what fees (if any) are to be charged for that use. Once the usage rights statements are generated, they are encoded in a suitable form for accessing during the processing of transactions.

Defining usage rights in terms of a language in combination with the hierarchical representation of a digital work enables the support of a wide variety of distribution and fee schemes. An example is the ability to attach multiple versions of a right to a work. So a creator may attach a PRINT right to make 5 copies for $10.00 and a PRINT right to make unlimited copies for $100.00. A purchaser may then choose which option best fits his needs. Another example is that rights and fees are additive. So in the case of a composite work, the rights and fees of each of the components works is used in determining the rights and fees for the work as a whole.

The basic contents of a right are illustrated in Figure 14. Referring to Figure 14, a right 1450 has a transactional component 1451 and a specifications component 1452. A right 1450 has a label (e.g. COPY or PRINT) which indicates the use or distribution privileges that are embodied by the right. The transactional component 1451 corresponds to a particular way in which a digital work may be used or distributed. The transactional component 1451 is typically embodied in software instructions in a repository which implement the use or distribution privileges for the right. The specifications components 1452 are used to specify conditions which must be satisfied prior to the right being exercised or to designate various transaction related parameters. In the currently preferred embodiment, these specifications include copy count 1453, Fees and Incentives 1454, Time 1455, Access and Security 1456 and Control 1457. Each of these specifications will be described in greater detail below with respect to the language grammar elements.

The usage rights language is based on the grammar described below. A grammar is a convenient means for defining valid sequence of symbols for a language. In describing the grammar the notation "[a|b|c]" is used to indicate distinct choices among alternatives. In this example, a sentence can have either an "a", "b" or "c". It must include exactly one of them. The braces {} are used to indicate optional items. Note that brackets, bars and braces are used to describe the language of usage rights sentences but do not appear in actual sentences in the language.

In contrast, parentheses are part of the usage rights language. Parentheses are used to group items together in lists. The notation (x^*) is used to indicate a variable length list, that is, a list containing one or more items of type x. The notation $(x)^*$ is used to indicate a variable number of lists containing x.

Keywords in the grammar are words followed by colons. Keywords are a common and very special case in the language. They are often used to indicate a single value, typically an identifier. In many cases, the keyword and the parameter are entirely optional. When a keyword is given, it often takes a single identifier as its value. In some cases, the keyword takes a list of identifiers.

In the usage rights language, time is specified in an hours:minutes:seconds (or hh:mm:ss) representation. Time zone indicators, e.g. PDT for Pacific Daylight Time, may also be specified. Dates are represented as year/ month/day (or YYYY/MMM/DD). Note that these time and date representations may specify moments in time or units of time Money units are specified in terms of dollars.

Finally, in the usage rights language, various "things" will need to interact with each other. For example, an instance of a usage right may specify a bank account, a digital ticke etc.. Such things need to be identified and are specified herein using the suffix "-ID."

The Usage Rights Grammar is listed in its entirety in Figure 15 and is described below.

Grammar element 1501 **"Digital Work Rights:= (Rights*)"** define the digital wor rights as a set of rights. The set of rights attached to a digital work define how that digital work may be transferred, used, performed or played. A set of rights will attach to the entire digital work and in the case of compound digital works, each of the components of the digita work. The usage rights of components of a digital may be different.

Grammar element 1502 **"Right: = (Right-Code {Copy-Count} {Control-Spec} {Time-Spec} {Access-Spec} {Fee-Spec})"** enumerates the content of a right. Each usage right must specify a right code. Each right may also optionally specify conditions which must be satisfied before the right can be exercised. These conditions are copy count, control, time, access and fee conditions. In the currently preferred embodiment, for the optional elements, the following defaults apply: copy count equals 1, no time limit on the use of the right, no access tests or a security level required to use the right and no fee is required. These conditions will each be described in greater detail below.

It is important to note that a digital work may have multiple versions of a right, each having the same right code. The multiple version would provide alternative conditions and fees for accessing the digital work.

Grammar element 1503 **"Right-Code : = Render-Code I Transport-Code I File-Management-CodeI Derivative-Works- Code Configuration-Code"** distinguishes each of the specific rights into a particular right type (although each right is identified by distinct right codes). In this way, the grammar provides a catalog of possible rights that can be associated with parts of digital works. In the following, rights are divided into categories for convenience in describing them.

Grammar element 1504 **"Render-Code : = [Play:{Player:Player-ID}I Print: {Printer: Printer-ID}]"** lists a category of rights all involving the making of ephemeral, transitory, or non-digital copies of the digital work. After use the copies are erased.

- Play A process of rendering or performing a digital work on some processor. This includes such things as playing digital movies, playing digital music, playing a video game, running a computer program, or displaying a document on a display.
- Print To render the work in a medium that is not further protected by usage rights, such as printing on paper.

Grammar element 1505 **"Transport-Code : = [Copy I Transfer I Loan (Remaining-Rights: Next-Set-of-Rights}] {(Next-Copy-Rights: Next-Set of Rights)}"** lists a category of rights involving the making of persistent, usable copies of the digital work on other repositories. The optional Next-Copy-Rights determine the rights on the work after it is transported. If this is not specified, then the rights on the transported copy are the same as on the original. The optional Remaining-Rights specify the rights that remain with a digital work when it is loaned out. If this is not specified, then the default is that no rights can be exercised when it is loaned out.

- Copy Make a new copy of a work
- Transfer Moving a work from one repository to another.
- Loan Temporarily loaning a copy to another repository for a specified period of time.

Grammar element 1506 **"File-Management-Code: = Backup {Back-Up-Copy-Rights: Next-Set -of Rights}I Restore I Delete I Folder I Directory {Name:Hide-Local I Hide - Remote}{Parts:Hide-Local I Hide-Remote}"** lists a category of rights involving operations for file management, such as the making of backup copies to protect the copy owner against catastrophic equipment failure.

Many software licenses and also copyright law give a copy owner the right to make backup copies to protect against catastrophic failure of equipment. However, the making of uncontrolled backup copies is inherently at odds with the ability to control usage, since an uncontrolled backup copy can be kept and then restored even after the authorized copy was sold.

The File management rights enable the making and restoring of backup copies in a way that respects usage rights, honoring the requirements of both the copy owner and the rights grantor and revenue owner. Backup copies of work descriptions (including usage rights and fee data) can be sent under appropriate protocol and usage rights control to other document repositories of sufficiently high security. Further rights permit organization of digital works into folders which themselves are treated as digital works and whose contents may be "hidden" from a party seeking to determine the contents of a repository.

- Backup To make a backup copy of a digital work as protection against media failure.
- Restore To restore a backup copy of a digital work.
- Delete To delete or erase a copy of a digital work.
- Folder To create and name folders, and to move files and folders between folders.
- Directory To hide a folder or its contents.

Grammar element 1507 "**Derivative-Works-Code: [Extract I Embed I Edit {Process: Process-ID}] {Next-Copy-Rights : Next-Set-of Rights}**" lists a category of rights involving the use of a digital work to create new works.

- Extract To remove a portion of a work, for the purposes of creating a new work.
- Embed To include a work in an existing work.
- Edit To alter a digital work by copying, selecting and modifying portions of an existing digital work.

Grammar element 1508 "**Configuration-Code: = Install I Uninstall**" lists a category of rights for installing and uninstalling software on a repository (typically a rendering repository.) This would typically occur for the installation of a new type of player within the rendering repository.

- Install: To install new software on a repository.
- Uninstall: To remove existing software from a repository.

Grammar element 1509 "**Next-Set-of-Rights: = {(Add: Set-Of-Rights)} {(Delete: Set-Of-Rights)} {(Replace: Set-Of-Rights)} {(Keep: Set-Of-Rights)}**" defines how rights are carried forward for a copy of a digital work. If the Next-Copy-Rights is not specified, the rights for the next copy are the same as those of the current copy. Otherwise, the set of rights for the next copy can be specified. Versions of rights after Add: are added to the current set of rights. Rights after Delete: are deleted from the current set of rights. If only right codes are listed after Delete:, then all versions of rights with those codes are deleted. Versions of rights after Replace: subsume all versions of rights of the same type in the current set of rights.

If Remaining-Rights is not specified, then there are no rights for the original after all Loan copies are loaned out. If Remaining-Rights is specified, then the Keep: token can be used to simplify the expression of what rights to keep behind. A list of right codes following keep means that all of the versions of those listed rights are kept in the remaining copy. This specification can be overridden by subsequent Delete: or Replace: specifications.

Copy Count Specification

For various transactions, it may be desirable to provide some limit as to the number of "copies" of the work which may be exercised simultaneously for the right. For example, it may be desirable to limit the number of copies of a digital work that may be loaned out at a time or viewed at a time.

Grammar element 1510 "**Copy-Count : = (Copies: positive-integer I 0 I unlimited)**" provides a condition which defines the number of "copies" of a work subject to the right . A copy count can be 0, a fixed number, or unlimited. The copy-count is associated with each right, as opposed to there being just a single copy-count for the digital work. The Copy-Count for a right is decremented each time that a right is exercised. When the Copy-Count equals zero, the right can no longer be exercised. If the Copy-Count is not specified, the default is one.

Control Specification

Rights and fees depend in general on rights granted by the creator as well as further restrictions imposed by later distributors. Control specifications deal with interactions between the creators and their distributors governing the imposition of further restrictions and fees. For example, a distributor of a digital work may not want an end consumer of a digital work to add fees or otherwise profit by commercially exploiting the purchased digital work.

Grammar element 1511 "**Control-Spec : = (Control: {Restrictable I Unrestrictable} {Unchargeable I Chargeable})**" provides a condition to specify the effect of usage rights and fees of parents on the exercise of the right. A digital work is restrictable if higher level d-blocks can impose further restrictions (time specifications and access specifications) on the right. It is unrestrictable if no further restrictions can be imposed. The default setting is restrictable. A right is unchargeable if no more fees can be imposed on the use of the right. It is chargeable if more fees can be imposed. The default is chargeable.

Time Specification

It is often desirable to assign a start date or specify some duration as to when a right may be exercised. Grammar element 1512 "**Time-Spec : = ({Fixed-Interval I Sliding-Interval I Meter-Time} Until: Expiration-Date)**" provides for specification of time conditions on the exercise of a right. Rights may be granted for a specified time. Different kinds of time specifications are appropriate for different kinds of rights. Some rights may be exercised during a fixed and predetermined duration. Some rights may be exercised for an interval that starts the first time that the right is invoked by some transaction. Some rights may be exercised or are charged according to some kind of metered time, which may be split into separate intervals. For example, a right to view a picture for an hour might be split into six ten minute viewings or four fifteen minute viewings or twenty three minute viewings.

The terms "time" and "date" are used synonymously to refer to a moment in time. There are several kinds of time specifications. Each specification represents some limitation on the times over which the usage right applies. The Expiration-Date specifies the moment at which the usage right ends. For example, if the Expiration-Date is "Jan 1, 1995," then the right ends at the first moment of 1995. If the Expiration-Date is specified as "forever", then the rights are interpreted as continuing without end. If only an expiration date is given, then the right can be exercised as often as desired until the expiration date.

Grammar element 1513 "**Fixed-Interval : = From: Start-Time**" is used to define a predetermined interval that runs from the start time to the expiration date.

Grammar element 1514 "**Sliding-Interval : = Interval: Use-Duration**" is used to define an indeterminate (or "open") start time. It sets limits on a continuous period of time over which the contents are accessible. The period starts on the first access and ends after the duration has passed or the expiration date is reached, whichever comes first. For example, if the right gives 10 hours of continuous access, the use-duration would begin when the first access was made and end 10 hours later.

Grammar element 1515 "**Meter-Time: = Time-Remaining: Remaining-Use**" is used to define a "meter time," that is, a measure of the time that the right is actually exercised. It differs from the Sliding-Interval specification in that the time that the digital work is in use need not be continuous. For example, if the rights guarantee three days of access, those days could be spread out over a month. With this specification, the rights can be exercised until the meter time is exhausted or the expiration date is reached, whichever comes first.

Remaining-Use: = Time-Unit
Start-Time: = Time-Unit
Use-Duration: = Time-Unit
All of the time specifications include time-unit specifications in their ultimate instantiation.

Security Class and Authorization Specification

The present invention provides for various security mechanisms to be introduced into a distribution or use scheme. Grammar element 1516 "**Access-Spec : ({SC: Security-Class} {Authorization: Authorization-ID*} {Other-Authorization: Authorization-ID*} {Ticket: Ticket-ID})**" provides a means for restricting access and transmission. Access specifications can specify a required security class for a repository to exercise a right or a required authorization test that must be satisfied.

The keyword "**SC:**" is used to specify a minimum security level for the repositories involved in the access. If "**SC:** " is not specified, the lowest security level is acceptable.

The optional "**Authorization:**" keyword is used to specify required authorizations on the same repository as the work. The optional "**Other-Authorization:**" keyword is used to specify required authorizations on the other repository in the transaction.

The optional "**Ticket:**" keyword specifies the identity of a ticket required for the transaction. A transaction involving digital tickets must locate an appropriate digital ticket agent who can "punch" or otherwise validate the ticket before the transaction can proceed. Tickets are described in greater detail below.

In a transaction involving a repository and a document server, some usage rights may require that the repository have a particular authorization, that the server have some authorization, or that both repositories have (possibly different) authorizations. Authorizations themselves are digital works (hereinafter referred to as an authorization object) that can be moved between repositories in the same manner as other digital works. Their copying and transferring is subject to the same rights and fees as other digital works. A repository is said to have an authorization if that authorization object is contained within the repository.

In some cases, an authorization may be required from a source other than the document server and repository. An authorization object referenced by an Authorization-ID can contain digital address information to be used to set up a communications link between a repository and the authorization source. These are analogous to phone numbers.

For such access tests, the communication would need to be established and authorization obtained before the right could be exercised.

For one-time usage rights, a variant on this scheme is to have a digital ticket. A ticket is presented to a digital ticket agent, whose type is specified on the ticket. In the simplest case, a certified generic ticket agent, available on all repositories, is available to "punch" the ticket. In other cases, the ticket may contain addressing information for locating a "special" ticket agent. Once a ticket has been punched, it cannot be used again for the same kind of transaction (unless it is unpunched or refreshed in the manner described below.) Punching includes marking the ticket with a timestamp of the date and time it was used. Tickets are digital works and can be copied or transferred between repositories according to their usage rights.

In the currently preferred embodiment, a "punched" ticket becomes "unpunched" or "refreshed" when it is copied or extracted. The Copy and Extract operations save the date and time as a property of the digital ticket. When a ticket agent is given a ticket, it can simply check whether the digital copy was made after the last time that it was punched. Of course, the digital ticket must have the copy or extract usage rights attached thereto.

The capability to unpunch a ticket is inportant in the following cases:

- A digital work is circulated at low cost with a limitation that it can be used only once.
- A digital work is circulated with a ticket that can be used once to give discounts on purchases of other works.
- A digital work is circulated with a ticket (included in the purchase price and possibly embedded in the work) that can be used for a future upgrade.

In each of these cases, if a paid copy is made of the digital work (including the ticket) the new owner would expect to get a fresh (unpunched) ticket, whether the copy seller has used the work or not. In contrast, loaning a work or simply transferring it to another repository should not revitalize the ticket.

Usage Fees and Incentives Specification

The billing for use of a digital work is fundamental to a commercial distribution system. Grammar Element 1517 **"Fee-Spec: = {Scheduled-Discount} Regular-Fee-Spec I Scheduled-Fee-Spec I Markup-Spec"** provides a range of options for billing for the use of digital works.

A key feature of this approach is the development of low-overhead billing for transactions in potentially small amounts. Thus, it becomes feasible to collect fees of only a few cents each for thousands of transactions.

The grammar differentiates between uses where the charge is per use from those where it is metered by the time unit. Transactions can support fees that the user pays for using a digital work as well as incentives paid by the right grantor to users to induce them to use or distribute the digital work.

The optional scheduled discount refers to the rest of the fee specification--discounting it by a percentage over time. If it is not specified, then there is no scheduled discount. Regular fee specifications are constant over time. Scheduled fee specifications give a schedule of dates over which the fee specifications change. Markup specifications are used in d-blocks for adding a percentage to the fees already being charged.

Grammar Element 1518 **"Scheduled-Discount: = (Scheduled-Discount: (Time-Spec Percentage)*)"** A Scheduled-Discount is a essentially a scheduled modifier of any other fee specification for this version of the right of the digital work. (It does not refer to children or parent digital works or to other versions of rights.). It is a list of pairs of times and percentages. The most recent time in the list that has not yet passed at the time of the transaction is the one in effect. The percentage gives the discount percentage. For example, the number 10 refers to a 10% discount.

Grammar Element 1519 **"Regular-Fee-Spec : = ({Fee: I Incentive:} [Per-Use-Spec I Metered-Rate-Spec I Best-Price-Spec I Call-For-Price-Spec] {Min: Money-Unit Per: Time-Spec}{Max: Money-Unit Per: Time-Spec} To: Account-ID)"** provides for several kinds of fee specifications.

Fees are paid by the copy-owner/user to the revenue-owner if **Fee:** is specified. Incentives are paid by the revenue-owner to the user if **Incentive:** is specified. If the **Min:** specification is given, then there is a minimum fee to be charged per time-spec unit for its use. If the **Max:** specification is given, then there is a maximum fee to be charged per time-spec for its use. When **Fee:** is specified, **Account-ID** identifies the account to which the fee is to be paid. When **Incentive:** is specified, **Account-ID** identifies the account from which the fee is to be paid.

Grammar element 1520 **"Per-Use-Spec: = Per-Use: Money-unit"** defines a simple fee to be paid every time the right is exercised, regardless of how much time the transaction takes.

Grammar element 1521 **"Metered-Rate-Spec : = Metered: Money-Unit Per: Time-Spec"** defines a metered-rate fee paid according to how long the right is exercised. Thus, the time it takes to complete the transaction determines the fee.

Grammar element 1522 **"Best-Price-Spec : = Best-Price: Money-unit Max: Money-unit"** is used to specify a best-price that is determined when the account is settled. This specification is to accommodate special deals, rebates, and pricing that depends on information that is not available to the repository. All fee specifications can be combined with tickets or authorizations that could indicate that the consumer is a wholesaler or that he is a preferred customer, or that the seller be authorized in some way. The amount of money in the **Max:** field is the maximum amount that the use will cost. This is the amount that is tentatively debited from the credit server. However, when the transaction is ultimately reconciled, any excess amount will be returned to the consumer in a separate transaction.

Grammar element 1523 "**Call-For-Price-Spec: = Call-For-Price** " is similar to a "**Best-Price-Spec**" in that it is intended to accommodate cases where prices are dynamic. A **Call-For-Price Spec** requires a communication with a dealer to determine the price. This option cannot be exercised if the repository cannot communicate with a dealer at the time that the right is exercised. It is based on a secure transaction whereby the dealer names a price to exercise the right and passes along a deal certificate which is referenced or included in the billing process.

Grammar element 1524 "**Scheduled-Fee-Spec: = (Schedule: (Time-Spec Regular-Fee-Spec)*)**" is used to provide a schedule of dates over which the fee specifications change. The fee specification with the most recent date not in the future is the one that is in effect. This is similar to but more general than the scheduled discount. It is more general, because it provides a means to vary the fee agreement for each time period.

Grammar element 1525 "**Markup-Spec: = Markup: percentage To: Account-ID**" is provided for adding a percentage to the fees already being charged. For example, a 5% markup means that a fee of 5% of cumulative fee so far will be allocated to the distributor. A markup specification can be applied to all of the other kinds of fee specifications. It is typically used in a shell provided by a distributor. It refers to fees associated with d-blocks that are parts of the current d-block. This might be a convenient specification for use in taxes, or in distributor overhead.

REPOSITORY TRANSACTIONS

When a user requests access to a digital work, the repository will initiate various transactions. The combination of transactions invoked will depend on the specifications assigned for a usage right. There are three basic types of transactions, Session Initiation Transactions, Financial Transactions and Usage Transactions. Generally, session initiation transactions are initiated first to establish a valid session. When a valid session is established, transactions corresponding to the various usage rights are invoked. Finally, request specific transactions are performed.

Transactions occur between two repositories (one acting as a server), between a repository and a document playback platform (e.g. for executing or viewing), between a repository and a credit server or between a repository and an authorization server. When transactions occur between more than one repository, it is assumed that there is a reliable communication channel between the repositories. For example, this could be a TCP/IP channel or any other commercially available channel that has built-in capabilities for detecting and correcting transmission errors. However, it is not assumed that the communication channel is secure. Provisions for security and privacy are part of the requirements for specifying and implementing repositories and thus form the need for various transactions.

Message Transmission

Transactions require that there be some communication between repositories. Communication between repositories occurs in units termed as messages. Because the communication line is assumed to be unsecure, all communications with repositories that are above the lowest security class are encrypted utilizing a public key encryption technique. Public key encryption is a well known technique in the encryption arts. The term key refers to a numeric code that is used with encryption and decryption algorithms. Keys come in pairs, where "writing keys" are used to encrypt data and "checking keys" are used to decrypt data. Both writing and checking keys may be public or private. Public keys are those that are distributed to others. Private keys are maintained in confidence.

Key management and security is instrumental in the success of a public key encryption system. In the currently preferred embodiment, one or more master repositories maintain the keys and create the identification certificates used by the repositories.

When a sending repository transmits a message to a receiving repository, the sending repository encrypts all of its data using the public writing key of the receiving repository. The sending repository includes its name, the name of the receiving repository, a session identifier such as a nonce (described below), and a message counter in each message.

In this way, the communication can only be read (to a high probability) by the receiving repository, which holds the private checking key for decryption. The auxiliary data is used to guard against various replay attacks to security. If messages ever arrive with the wrong counter or an old nonce, the repositories can assume that someone is interfering with communication and the transaction terminated.

The respective public keys for the repositories to be used for encryption are obtained in the registration transaction described below.

Session Initiation Transactions

A usage transaction is carried out in a session between repositories. For usage transactions involving more than one repository, or for financial transactions between a repository and a credit server, a registration transaction is performed. A second transaction termed a login transaction, may also be needed to initiate the session. The goal of the registration transaction is to establish a secure channel between two repositories who know each others identities. As it is assumed that the communication channel between the repositories is reliable but not secure, there is a risk that a non-repository may mimic the protocol in order to gain illegitimate access to a repository.

The registration transaction between two repositories is described with respect to Figures 16 and 17. The steps described are from the perspective of a "repository-1" registering its identity with a "repository-2". The registration must be symmetrical so the same set of steps will be repeated for repository-2 registering its identity with repository-1. Referring to Figure 16, repository-1 first generates an encrypted registration identifier, step 1601 and then generates a registration message, step 1602. A registration message is comprised of an identifier of a master repository, the identification certificate for the repository-1 and an encrypted random registration identifier. The identification certificate is encrypted by the master repository in its private key and attests to the fact that the repository (here repository-1) is a bona fide repository. The identification certificate also contains a public key for the repository, the repository security level and a timestamp (indicating a time after which the certificate is no longer valid.) The registration identifier is a number generated by the repository for this registration. The registration identifier is unique to the session and is encrypted in repository-1's private key. The registration identifier is used to improve security of authentication by detecting certain kinds of communications based attacks. Repository-1 then transmits the registration message to repository-2, step 1603.

Upon receiving the registration message, repository-2 determines if it has the needed public key for the master repository, step 1604. If repository-2 does not have the needed public key to decrypt the identification certificate, the registration transaction terminates in an error, step 1618.

Assuming that repository-2 has the proper public key the identification certificate is decrypted, step 1605. Repository-2 saves the encrypted registration identifier, step 1606, and extracts the repository identifier, step 1607. The extracted repository identifier is checked against a "hotlist" of compromised document repositories, step 1608. In the currently preferred embodiment, each repository will contain "hotlists" of compromised repositories. If the repository is on the "hotlist', the registration transaction terminates in an error per step 1618. Repositories can be removed from the hotlist when their certificates expire, so that the list does not need to grow without bound. Also, by keeping a short list of hotlist certificates that it has previously received, a repository can avoid the work of actually going through the list. These lists would be encrypted by a master repository. A minor variation on the approach to improve efficiency would have the repositories first exchange lists of names of hotlist certificates, ultimately exchanging only those lists that they had not previously received. The "hotlists" are maintained and distributed by Master repositories.

Note that rather than terminating in error, the transaction could request that another registration message be sent based on an identification certificate created by another master repository. This may be repeated until a satisfactory identification certificate is found, or it is determined that trust cannot be established.

Assuming that the repository is not on the hotlist, the repository identification needs to be verified. In other words, repository-2 needs to validate that the repository on the other end is really repository-1. This is termed performance testing and is performed in order to avoid invalid access to the repository via a counterfeit repository replaying a recording of a prior session initiation between repository-1 and repository-2. Performance testing is initiated by repository-2 generating a performance message, step 1609. The performance message consists of a nonce, the names of the respective repositories, the time and the registration identifier received from repository-1. A nonce is a generated message based on some random and variable information (e.g. the time or the temperature.) The nonce is used to check whether repository-1 can actually exhibit correct encrypting of a message using the private keys it claims to have, on a message that it has never seen before. The performance message is encrypted using the public key specified in the registration message of repository-1. The performance message is transmitted to repository-1, step 1610, where it is decrypted by repository-1 using its private key, step 1611. Repository-1 then checks to make sure that the names of the two repositories are correct, step 1612, that the time is accurate, step 1613 and that the registration identifier corresponds to the one it sent, step 1614. If any of these tests fails, the transaction is terminated per step 1616. Assuming that the tests are passed, repository-1 transmits the nonce to repository-2 in the clear, step 1615. Repository-2 then compares the received nonce to the original nonce, step 1617. If they are not identical, the registration transaction terminates in an error per step 1618. If they are the same, the registration transaction has successfully completed.

At this point, assuming that the transaction has not terminated, the repositories exchange messages containing session keys to be used in all communications during the session and synchronize their clocks. Figure 17 illustrates the session information exchange and clock synchronization steps (again from the perspective of repository-1.) Referring to Figure 17, repository-1 creates a session key pair, step 1701. A first key is kept private and is used by repository-1 to encrypt messages. The second key is a public key used by repository-2 to decrypt messages. The second key is encrypted using the public key of repository-2, step 1702 and is sent to repository-2, step 1703. Upon receipt, repository-2 decrypts the second key, step 1704. The second key is used to decrypt messages in subsequent communications. When each repository has completed this step, they are both convinced that the other repository is bona fide and that they are communicating with the original. Each repository has given the other a key to be used in decrypting further communications during the session. Since that key is itself transmitted in the public key of the receiving repository only it will be able to decrypt the key which is used to decrypt subsequent messages.

After the session information is exchanged, the repositories must synchronize their clocks. Clock synchronization is used by the repositories to establish an agreed upon time base for the financial records of their mutual transactions. Referring back to Figure 17, repository-2 initiates clock synchronization by generating a time stamp exchange message, step 1705, and transmits it to repository- 1, step 1706. Upon receipt, repository-1 generates its own time stamp message, step 1707 and transmits it back to repository-2, step 1708. Repository-2 notes the current time, step 1709 and stores the time received from repository-1, step 1710. The current time is compared to the time received from repository-1, step 1711. The difference is then checked to see if it exceeds a predetermined tolerance (e.g. one minute), step 1712. If it does, repository-2 terminates the transaction as this may indicate tampering with the repository, step 1713. If not repository-2 computes an adjusted time delta, step 1714. The adjusted time delta is the difference between the clock time of repository-2 and the average of the times from repository-1 and repository-2.

To achieve greater accuracy, repository-2 can request the time again up to a fixed number of times (e.g. five times), repeat the clock synchronization steps, and average the results.

A second session initiation transaction is a Login transaction. The Login transaction is used to check the authenticity of a user requesting a transaction. A Login transaction is particularly prudent for the authorization of financial transactions that will be charged to a credit server. The Login transaction involves an interaction between the user at a user interface and the credit server associated with a repository. The information exchanged here is a login string supplied by the repository/credit server to identify itself to the user, and a Personal Identification Number (PIN) provided by the user to identify himself to the credit server. In the event that the user is accessing a credit server on a repository different from the one on which the user interface resides, exchange of the information would be encrypted using the public and private keys of the respective repositories.

Billing Transactions

Billing Transactions are concerned with monetary transactions with a credit server. Billing Transactions are carried out when all other conditions are satisfied and a usage fee is required for granting the request. For the most part, billing transactions are well understood in the state of the art. These transactions are between a repository and a credit server, or between a credit server and a billing clearinghouse. Briefly, the required transactions include the following:

- Registration and LOGIN transactions by which the repository and user establish their bona fides to a credit server. These transactions would be entirely internal in cases where the repository and credit server are implemented as a single system.
- Registration and LOGIN transactions, by which a credit server establishes its bona fides to a billing clearinghouse.
- An Assign-fee transaction to assign a charge. The information in this transaction would include a transaction identifier, the identities of the repositories in the transaction, and a list of charges from the parts of the digital work. If there has been any unusual event in the transaction such as an interruption of communications, that information is included as well.
- A Begin-charges transaction to assign a charge. This transaction is much the same as an assign-fee transaction except that it is used for metered use. It includes the same information as the assign-fee transaction as well as the usage fee information. The credit-server is then responsible for running a clock.
- An End-charges transaction to end a charge for metered use. (In a variation on this approach, the repositories would exchange periodic charge information for each block of time.)
- A report-charges transaction between a personal credit server and a billing clearinghouse. This transaction is invoked at least once per billing period. It is used to pass along information about charges. On debit and credit cards, this transaction would also be used to update balance information and credit limits as needed.

All billing transactions are given a transaction ID and are reported to the credit severs by both the server and the client. This reduces possible loss of billing information if one of the parties to a transaction loses a banking card and provides a check against tampering with the system.

Usage Transactions

After the session initiation transactions have been completed, the usage request may then be processed. To simplify the description of the steps carried out in processing a usage request, the term requester is used to refer to a repository in the requester mode which is initiating a request, and the term server is used to refer to a repository in the server mode and which contains the desired digital work. In many cases such as requests to print or view a work, the requester and server may be the same device and the transactions described in the following would be entirely internal. In such instances, certain transaction steps, such as the registration transaction, need not be performed.

There are some common steps that are part of the semantics of all of the usage rights transactions. These steps are referred to as the common transaction steps. There are two sets -- the "opening" steps and the "closing" steps. For simplicity, these are listed here rather than repeating them in the descriptions of all of the usage rights transactions.

Transactions can refer to a part of a digital work, a complete digital work, or a Digital work containing other digital works. Although not described in detail herein, a transaction may even refer to a folder comprised of a plurality of digital works. The term "work" is used to refer to what ever portion or set of digital works is being accessed.

Many of the steps here involve determining if certain conditions are satisfied. Recall that each usage right may have one or more conditions which must be satisfied before the right can be exercised. Digital works have parts and parts have parts. Different parts can have different rights and fees. Thus, it is necessary to verify that the requirements are met for ALL of the parts that are involved in a transaction For brevity, when reference is made to checking whether the rights exist and conditions for exercising are satisfied, it is meant that all such checking takes place for each of the relevant parts of the work.

Figure 18 illustrates the initial common opening and closing steps for a transaction. At this point it is assumed that registration has occurred and that a "trusted" session is in place. General tests are tests on usage rights associated with the folder containing the work or some containing folder higher in the file system hierarchy. These tests correspond to requirements imposed on the work as a consequence of its being on the particular repository, as opposed to being attached to the work itself. Referring to Figure 18, prior to initiating a usage transaction, the requester performs any general tests that are required before the right associated with the transaction can be exercised, step, 1801. For example, install, uninstall and delete rights may be implemented to require that a requester have an authorization certificate before the right can be exercised. Another example is the requirement that a digital ticket be present and punched before a digital work may be copied to a requester. If any of the general tests fail, the transaction is not initiated, step, 1802. Assuming that such required tests are passed, upon receiving the usage request, the server generates a transaction identifier that is used in records or reports of the transaction, step 1803. The server then checks whether the digital work has been granted the right corresponding to the requested transaction, step 1804. If the digital work has not been granted the right corresponding to the request, the transaction terminates, step 1805. If the digital work has been granted the requested right, the server then determines if the various conditions for exercising the right are satisfied. Time based conditions are examined, step 1806. These conditions are checked by examining the time specification for the the version of the right. If any of the conditions are not satisfied, the transaction terminates per step 1805.

Assuming that the time based conditions are satisfied, the server checks security and access conditions, step 1807. Such security and access conditions are satisfied if: 1) the requester is at the specified security class, or a higher security class, 2) the server satisfies any specified authorization test and 3) the requester satisfies any specified authorization tests and has any required digital tickets. If any of the conditions are not satisfied, the transaction terminates per step 1805.

Assuming that the security and access conditions are all satisfied, the server checks the copy count condition, step 1808. If the copy count equals zero, then the transaction cannot be completed and the transaction terminates per step 1805.

Assuming that the copy count does not equal zero, the server checks if the copies in use for the requested right is greater than or equal to any copy count for the requested right (or relevant parts), step 1809. If the copies in use is greater than or equal to the copy count, this indicates that usage rights for the version of the transaction have been exhausted. Accordingly, the server terminates the transaction, step 1805. If the copy count is less than the copies in use for the transaction the transaction can continue, and the copies in use would be incremented by the number of digital works requested in the transaction, step 1810.

The server then checks if the digital work has a "Loan" access right, step 1811. The "Loan" access right is a special case since remaining rights may be present even though all copies are loaned out. If the digital work has the "Loan" access right, a check is made to see if all copies have been loaned out, step 1812. The number of copies that could be loaned is the sum of the Copy-Counts for all of the versions of the loan right of the digital work. For a composite work, the relevant figure is the minimal such sum of each of the components of the composite work. If all copies have been loaned out, the remaining rights are determined, step 1813. The remaining-rights is determined from the remaining rights specifications from the versions of the Loan right. If there is only one version of the Loan right, then the determination is simple. The remaining rights are the ones specified in that version of the Loan right, or none if Remaining-Rights: is not specified. If there are multiple versions of the Loan right and all copies of all of the versions are loaned out, then the remaining rights is taken as the minimum set (intersection) of remaining rights across all of the versions of the loan right. The server then determines if the requested right is in the set of remaining rights, step 1814. If the requested right is not in the set of remaining rights, the server terminates the transaction, step 1805.

If Loan is not a usage right for the digital work or if all copies have not been loaned out or the requested right is in the set of remaining rights, fee conditions for the right are then checked, step 1815. This will initiate various financial transactions between the repository and associated credit server. Further, any metering of usage of a digital work will commence. If any financial transaction fails, the transaction terminates per step 1805.

It should be noted that the order in which the conditions are checked need not follow the order of steps 1806-1815.

At this point, right specific steps are now performed and are represented here as step 1816. The right specific steps are described in greater detail below.

The common closing transaction steps are now performed. Each of the closing transaction steps are performed by the server after a successful completion of a transaction. Referring back to Figure 18, the copies in use value for the requested right is decremented by the number of copies involved in the transaction, step 1817. Next, if the right had a metered usage fee specification, the server subtracts the elapsed time from the Remaining-Use-Time associated with the right for every part involved in the transaction, step 1818. Finally, if there are fee specifications associated with the right, the server initiates End-Charge financial transaction to confirm billing, step 1819.

Transmission Protocol

An important area to consider is the transmission of the digital work from the server to the requester. The transmission protocol described herein refers to events occurring after a valid session has been created. The transmission protocol must handle the case of disruption in the communications between the repositories. It is assumed that interference such as injecting noise on the communication channel can be detected by the integrity checks (e.g., parity, checksum, etc.) that are built into the transport protocol and are not discussed in detail herein.

The underlying goal in the transmission protocol is to preclude certain failure modes, such as malicious or accidental interference on the communications channel. Suppose, for example, that a user pulls a card with the credit server at a specific time near the end of a transaction. There should not be a vulnerable time at which "pulling the card" causes the repositories to fail to correctly account for the number of copies of the work that have been created. Restated, there should be no time at which a party can break a connection as a means to avoid payment after using a digital work.

If a transaction is interrupted (and fails), both repositories restore the digital works and accounts to their state prior to the failure, modulo records of the failure itself.

Figure 19 is a state diagram showing steps in the process of transmitting information during a transaction. Each box represents a state of a repository in either the server mode (above the central dotted line 1901) or in the requester mode (below the dotted line 1901). Solid arrows stand for transitions between states. Dashed arrows stand for message communications between the repositories. A dashed message arrow pointing to a solid transition arrow is interpreted as meaning that the transition takes place when the message is received. Unlabeled transition arrows take place unconditionally. Other labels on state transition arrows describe conditions that trigger the transition.

Referring now to Figure 19, the server is initially in a state 1902 where a new transaction is initiated via start message 1903. This message includes transaction information including a transaction identifier and a count of the blocks of data to be transferred. The requester, initially in a wait state 1904 then enters a data wait state 1905.

The server enters a data transmit state 1906 and transmits a block of data 1907 and then enters a wait for acknowledgement state 1908. As the data is received, the requester enters a data receive state 1909 and when the data blocks are completely received it enters an acknowledgement state 1910 and transmits an Acknowledgement message 1911 to the server.

If there are more blocks to send, the server waits until receiving an Acknowledgement message from the requester. When an Acknowledgement message is received it sends the next block to the requester and again waits for acknowledgement. The requester also repeats the same cycle of states.

If the server detects a communications failure before sending the last block, it enters a cancellation state 1912 wherein the transaction is cancelled. Similarly, if the requester detects a communications failure before receiving the last block it enters a cancellation state 1913.

If there are no more blocks to send, the server commits to the transaction and waits for the final Acknowledgement in state 1914. If there is a communications failure before the server receives the final Acknowledgement message, it still commits to the transaction but includes a report about the event to its credit server in state 1915. This report serves two purposes. It will help legitimize any claims by a user of having been billed for receiving digital works that were not completely received. Also it helps to identify repositories and communications lines that have suspicious patterns of use and interruption. The server then enters its completion state 1916.

On the requester side, when there are no more blocks to receive, the requester commits to the transaction in state 1917. If the requester detects a communications failure at this state, it reports the failure to its credit server in state 1918, but still commits to the transaction. When it has committed, it sends an acknowledgement message to the server. The server then enters its completion state 1919.

The key property is that both the server and the requester cancel a transaction if it is interrupted before all of the data blocks are delivered, and commits to it if all of the data blocks have been delivered.

There is a possibility that the server will have sent all of the data blocks (and committed) but the requester will not have received all of them and will cancel the transaction. In this case, both repositories will presumably detect a communications failure and report it to their credit server. This case will probably be rare since it depends on very precise timing of the communications failure. The only consequence will be that the user at the requester repository may want to request a refund from the credit services -- and the case for that refund will be documented by reports by both repositories.

To prevent loss of data, the server should not delete any transferred digital work until receiving the final acknowledgement from the requester. But it also should not use the file. A well known way to deal with this situation is called "two-phase commit" or 2PC.

Two-phase commit works as follows. The first phase works the same as the method described above. The server sends all of the data to the requester. Both repositories mark the transaction (and appropriate files) as uncommitted. The server sends a ready-to-commit message to the requester. The requester sends back an acknowledgement. The server then commits and sends the requester a commit message. When the requester receives the commit message, it commits the file.

If there is a communication failure or other crash, the requester must check back with the server to determine the status of the transaction. The server has the last word on this. The requester may have received all of the data, but if it did not get the final message, it has not committed. The server can go ahead and delete files (except for transaction records) once it commits, since the files are known to have been fully transmitted before starting the 2PC cycle.

There are variations known in the art which can be used to achieve the same effect. For example, the server could use an additional level of encryption when transmitting a work to a client. Only after the client sends a message acknowledging receipt does it send the key. The client then agrees to pay for the digital work. The point of this variation is that it provides a clear audit trail that the client received the work. For trusted systems, however, this variation adds a level of encryption for no real gain in accountability.

The transaction for specific usage rights are now discussed.

The Copy Transaction

A Copy transaction is a request to make one or more independent copies of the work with the same or lesser usage rights. Copy differs from the extraction right discussed later in that it refers to entire digital works or entire folders containing digital works. A copy operation cannot be used to remove a portion of a digital work.

- The requester sends the server a message to initiate the Copy Transaction. This message indicates the work to be copied, the version of the copy right to be used for the transaction, the destination address information (location in a folder) for placing the work, the file data for the work (including its size), and the number of copies requested.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the client according to the transmission protocol. If a Next-Set-Of-Rights has been provided in the version of the right, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted. In any event, the Copy-Count field for the copy of the digital work being sent right is set to the number-of-copies requested.
- The requester records the work contents, data, and usage rights and stores the work. It records the date and time that the copy was made in the properties of the digital work.
- The repositories perform the common closing transaction steps.

The Transfer Transaction

A Transfer transaction is a request to move copies of the work with the same or lesser usage rights to another repository. In contrast with a copy transaction, this results in removing the work copies from the server.

- The requester sends the server a message to initiate the Transfer Transaction. This message indicates the work to be transferred, the version of the transfer right to be used in the transaction, the destination address information for placing the work, the file data for the work, and the number of copies involved.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted. In either case, the Copy-Count field for the transmitted rights are set to the number-of-copies requested.

- The requester records the work contents, data, and usage rights and stores the work.
- The server decrements its copy count by the number of copies involved in the transaction.
- The repositories perform the common closing transaction steps.
- If the number of copies remaining in the server is now zero, it erases the digital work from its memory.

The Loan Transaction

A loan transaction is a mechanism for loaning copies of a digital work. The maximum duration of the loan is determined by an internal parameter of the digital work. Works are automatically returned after a predetermined time period.

- The requester sends the server a message to initiate the Transfer Transaction. This message indicates the work to be loaned, the version of the loan right to be used in the transaction, the destination address information for placing the work, the number of copies involved, the file data for the work, and the period of the loan.
- The server checks the validity of the requested loan period, and ends with an error if the period is not valid. Loans for a loaned copy cannot extend beyond the period of the original loan to the server.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted, as modified to reflect the loan period.
- The requester records the digital work contents, data, usage rights, and loan period and stores the work.
- The server updates the usage rights information in the digital work to reflect the number of copies loaned out.
- The repositories perform the common closing transaction steps.
- The server updates the usage rights data for the digital work. This may preclude use of the work until it is returned from the loan. The user on the requester platform can now use the transferred copies of the digital work. A user accessing the original repository cannot use the digital work , unless there are copies remaining. What happens next depends on the order of events in time.

  Case 1. If the time of the loan period is not yet exhausted and the requester sends the repository a Return message.

  - The return message includes the requester identification, and the transaction ID.
  - The server decrements the copies-in-use field by the number of copies that were returned. (If the number of digital works returned is greater than the number actually borrowed, this is treated as an error.) This step may now make the work available at the server for other users.
  - The requester deactivates its copies and removes the contents from its memory.

  Case 2. If the time of the loan period is exhausted and the requester has not yet sent a Return message.

  - The server decrements the copies-in-use field by the number digital works that were borrowed.
  - The requester automatically deactivates its copies of the digital work. It terminates all current uses and erases the digital work copies from memory. One question is why a requester would ever return a work earlier than the period of the loan, since it would be returned automatically anyway. One reason for early return is that there may be a metered fee which determines the cost of the loan. Returning early may reduce that fee.

The Play Transaction

A play transaction is a request to use the contents of a work. Typically, to "play" a work is to send the digital work through some kind of transducer, such as a speaker or a display device. The request implies the intention that the contents will not be communicated digitally to any other system. For example, they will not be sent to a printer, recorded on any digital medium, retained after the transaction or sent to another repository.

This term "play" is natural for examples like playing music, playing a movie, or playing a video game. The general form of play means that a "player" is used to use the digital work. However, the term play covers all media and kinds of recordings. Thus one would "play" a digital work, meaning, to render it for reading, or play a computer program, meaning to execute it. For a digital ticket the player would be a digital ticket agent.

- The requester sends the server a message to initiate the play transaction. This message indicates the work to be played, the version of the play right to be used in the transaction, the identity of the player being used, and the file data for the work.

- The server checks the validity of the player identification and the compatibility of the player identification with the player specification in the right. It ends with an error if these are not satisfactory.
- The repositories perform the common opening transaction steps.
- The server and requester read and write the blocks of data as requested by the player according to the transmission protocol. The requester plays the work contents, using the player.
- When the player is finished, the player and the requester remove the contents from their memory.
- The repositories perform the common closing transaction steps.

The Print Transaction

A Print transaction is a request to obtain the contents of a work for the purpose of rendering them on a "printer." We use the term "printer" to include the common case of writing with ink on paper. However, the key aspect of "printing" in our use of the term is that it makes a copy of the digital work in a place outside of the protection of usage rights. As with all rights, this may require particular authorization certificates.

Once a digital work is printed, the publisher and user are bound by whatever copyright laws are in effect. However, printing moves the contents outside the control of repositories. For example, absent any other enforcement mechanisms, once a digital work is printed on paper, it can be copied on ordinary photocopying machines without intervention by a repository to collect usage fees. If the printer to a digital disk is permitted, then that digital copy is outside of the control of usage rights. Both the creator and the user know this, although the creator does not necessarily give tacit consent to such copying, which may violate copyright laws.

- The requester sends the server a message to initiate a Print transaction. This message indicates the work to be played, the identity of the printer being used, the file data for the work, and the number of copies in the request.
- The server checks the validity of the printer identification and the compatibility of the printer identification with the printer specification in the right. It ends with an error if these are not satisfactory.
- The repositories perform the common opening transaction steps.
- The server transmits blocks of data according to the transmission protocol.
- The requester prints the work contents, using the printer.
- When the printer is finished, the printer and the requester remove the contents from their memory.
- The repositories perform the common closing transaction steps.

The Backup Transaction

A Backup transaction is a request to make a backup copy of a digital work, as a protection against media failure. In the context of repositories, secure backup copies differ from other copies in three ways: (1) they are made under the control of a Backup transaction rather than a Copy transaction, (2) they do not count as regular copies, and (3) they are not usable as regular copies. Generally, backup copies are encrypted.

Although backup copies may be transferred or copied, depending on their assigned rights, the only way to make them useful for playing, printing or embedding is to restore them.

The output of a Backup operation is both an encrypted data file that contains the contents and description of a work, and a restoration file with an encryption key for restoring the encrypted contents. In many cases, the encrypted data file would have rights for " printing" it to a disk outside of the protection system, relying just on its encryption for security. Such files could be stored anywhere that was physically safe and convenient. The restoration file would be held in the repository. This file is necessary for the restoration of a backup copy. It may have rights for transfer between repositories.

- The requester sends the server a message to initiate a backup transaction. This message indicates the work to be backed up, the version of the backup right to be used in the transaction, the destination address information for placing the backup copy, the file data for the work.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, a set of default rights for backup files of the original are transmitted by the server.
- The requester records the work contents, data, and usage rights. It then creates a one-time key and encrypts the contents file. It saves the key information in a restoration file.
- The repositories perform the common closing transaction steps.

In some cases, it is convenient to be able to archive the large, encrypted contents file to secure offline storage, such as a magneto-optical storage system or magnetic tape. This creation of a non-repository archive file is as secure as the encryption process. Such non-repository archive storage is considered a form of "printing" and is controlled by a print right with a specified "archive-printer." An archive-printer device is programmed to save the encrypted contents file (but not the description file) offline in such a way that it can be retrieved.

The Restore Transaction

A Restore transaction is a request to convert an encrypted backup copy of a digital work into a usable copy. A restore operation is intended to be used to compensate for catastrophic media failure. Like all usage rights, restoration rights can include fees and access tests including authorization checks.

- The requester sends the server a message to initiate a Restore transaction. This message indicates the work to be restored, the version of the restore right for the transaction, the destination address information for placing the work, and the file data for the work.
- The server verifies that the contents file is available (i.e. a digital work corresponding to the request has been backed-up.) If it is not, it ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The server retrieves the key from the restoration file. It decrypts the work contents, data, and usage rights.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, a set of default rights for backup files of the original are transmitted by the server.
- The requester stores the digital work.
- The repositories perform the common closing transaction steps.

The Delete Transaction

A Delete transaction deletes a digital work or a number of copies of a digital work from a repository. Practically all digital works would have delete rights.

- The requester sends the server a message to initiate a delete transaction. This message indicates the work to be deleted , the version of the delete right for the transaction.
- The repositories perform the common opening transaction steps.
- The server deletes the file, erasing it from the file system.
- The repositories perform the common closing transaction steps.

The Directory Transaction

A Directory transaction is a request for information about folders, digital works, and their parts. This amounts to roughly the same idea as protection codes in a conventional file system like TENEX, except that it is generalized to the full power of the access specifications of the usage rights language.

The Directory transaction has the important role of passing along descriptions of the rights and fees associated with a digital work. When a user wants to exercise a right, the user interface of his repository implicitly makes a directory request to determine the versions of the right that are available. Typically these are presented to the user -- such as with different choices of billing for exercising a right. Thus, many directory transactions are invisible to the user and are exercised as part of the normal process of exercising all rights.

- The requester sends the server a message to initiate a Directory transaction. This message indicates the file or folder that is the root of the directory request and the version of the di rectory right used for the transaction.
- The server verifies that the information is accessible to the requester. In particular, it does not return the names of any files that have a HIDE-NAME status in their directory specifications, and it does not return the parts of any folders or files that have HIDE-PARTS in their specification. If the information is not accessible, the server ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The server sends the requested data to the requester according to the transmission protocol.
- The requester records the data.
- The repositories perform the common closing transaction steps.

The Folder Transaction

A Folder transaction is a request to create or rename a folder, or to move a work between folders. Together with Directory rights, Folder rights control the degree to which organization of a repository can be accessed or modified from another repository.

- The requester sends the server a message to initiate a Folder transaction. This message indicates the folder that is the root of the folder request, the version of the folder right for the transaction, an operation, and data. The operation can be one of create, rename, and move file. The data are the specifications required for the operation, such as a specification of a folder or digital work and a name.
- The repositories perform the common opening transaction steps.
- The server performs the requested operation -- creating a folder, renaming a folder, or moving a work between folders.
- The repositories perform the common closing transaction steps.

The Extract Transaction

A extract transaction is a request to copy a part of a digital work and to create a new work containing it. The extraction operation differs from copying in that it can be used to separate a part of a digital work from d-blocks or shells that place additional restrictions or fees on it. The extraction operation differs from the edit operation in that it does not change the contents of a work, only its embedding in d-blocks. Extraction creates a new digital work.

- The requester sends the server a message to initiate an Extract transaction. This message indicates the part of the work to be extracted, the version of the extract right to be used in the transaction, the destination address information for placing the part as a new work, the file data for the work, and the number of copies involved.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the new work. Otherwise, the rights of the original are transmitted. The Copy-Count field for this right is set to the number-of-copies requested.
- The requester records the contents, data, and usage rights and stores the work. It records the date and time that new work was made in the properties of the work.
- The repositories perform the common closing transaction steps.

The Embed Transaction

An embed transaction is a request to make a digital work become a part of another digital work or to add a shell d-block to enable the adding of fees by a distributor of the work.

- The requester sends the server a message to initiate an Embed transaction. This message indicates the work to be embedded, the version of the embed right to be used in the transaction, the destination address information for placing the part as a a work, the file data for the work, and the number of copies involved.
- The server checks the control specifications for all of the rights in the part and the destination. If they are incompatible, the server ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the new work. Otherwise, the rights of the original are transmitted. The Copy-Count field for this right is set to the number-of-copies requested.
- The requester records the contents, data, and usage rights and embeds the work in the destination file.
- The repositories perform the common closing transaction steps.

The Edit Transaction

An Edit transaction is a request to make a new digital work by copying, selecting and modifying portions of an existing digital work. This operation can actually change the contents of a digital work. The kinds of changes that are permitted depend on the process being used. Like the extraction operation, edit operates on portions of a digital work. In contrast with the extract operation, edit does not affect the rights or location of the work. It only changes the contents. The kinds of changes permitted are determined by the type specification of the processor specified in the rights. In the currently preferred embodiment, an edit transaction changes the work itself and does not make a new work. However, it would be a reasonable variation to cause a new copy of the work to be made.

- The requester sends the server a message to initiate an Edit transaction. This message indicates the work to be edited, the version of the edit right to be used in the transaction, the file data for the work (including its size), the process-ID for the process, and the number of copies involved.
- The server checks the compatibility of the process-ID to be used by the requester against any process-ID specification in the right. If they are incompatible, it ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The requester uses the process to change the contents of the digital work as desired. (For example, it can select and duplicate parts of it; combine it with other information; or compute functions based on the information. This can amount to editing text, music, or pictures or taking whatever other steps are useful in creating a derivative work.)
- The repositories perform the common closing transaction steps.

The edit transaction is used to cover a wide range of kinds of works. The category describes a process that takes as its input any portion of a digital work and then modifies the input in some way. For example, for text, a process for editing the text would require edit rights. A process for "summarizing" or counting words in the text would also be considered editing. For a music file, processing could involve changing the pitch or tempo, or adding reverberations, or any other audio effect. For digital video works, anything which alters the image would require edit rights. Examples would be colorizing, scaling, extracting still photos, selecting and combining frames into story boards, sharpening with signal processing, and so on.

Some creators may want to protect the authenticity of their works by limiting the kinds of processes that can be performed on them. If there are no edit rights, then no processing is allowed at all. A processor identifier can be included to specify what kind of process is allowed. If no process identifier is specified, then arbitrary processors can be used. For an example of a specific process, a photographer may want to allow use of his photograph but may not want it to be colorized. A musician may want to allow extraction of portions of his work but not changing of the tonality.

Authorization Transactions

There are many ways that authorization transactions can be defined. In the following, our preferred way is to simply define them in terms of other transactions that we already need for repositories. Thus, it is convenient sometimes to speak of "authorization transactions," but they are actually made up of other transactions that repositories already have.

A usage right can specify an authorization-ID, which identifies an authorization object (a digital work in a file of a standard format) that the repository must have and which it must process. The authorization is given to the generic authorization (or ticket) server of the repository which begins to interpret the authorization.

As described earlier, the authorization contains a server identifier, which may just be the generic authorization server or it may be another server. When a remote authorization server is required, it must contain a digital address. It may also contain a digital certificate.

If a remote authorization server is required, then the authorization process first performs the following steps:

- The generic authorization server attempts to set up the communications channel. (If the channel cannot be set up, then authorization fails with an error.)
- When the channel is set up, it performs a registration process with the remote repository. (If registration fails, then the authorization fails with an error.)
- When registration is complete, the generic authorization server invokes a "Play" transaction with the remote repository, supplying the authorization document as the digital work to be played, and the remote authorization server (a program) as the "player." (If the player cannot be found or has some other error, then the authorization fails with an error.)
- The authorization server then "plays" the authorization. This involves decrypting it using either the public key of the master repository that issued the certificate or the session key from the repository that transmitted it. The authorization server then performs various tests. These tests vary according to the authorization server. They include such steps as checking issue and validity dates of the authorization and checking any hot-lists of known invalid authorizations. The authorization server may require carrying out any other transactions on the repository as well, such as checking directories, getting some person to supply a password, or playing some other digital work. It may also invoke some special process for checking information about locations or recent events. The "script" for such steps is contained within the authorization server.
- If all of the required steps are completed satisfactorily, the authorization server completes the transaction normally, signaling that authorization is granted.

The Install Transaction

An Install transaction is a request to install a digital work as runnable software on a repository. In a typical case, the requester repository is a rendering repository and the software would be a new kind or new version of a player. Also in a typical case, the software would be copied to file system of the requester repository before it is installed.

- The requester sends the server an Install message. This message indicates the work to be installed, the version of the Install right being invoked, and the file data for the work (including its size).
- The repositories perform the common opening transaction steps.
- The requester extracts a copy of the digital certificate for the software. If the certificate cannot be found or the master repository for the certificate is not known to the requester, the transaction ends with an error.
- The requester decrypts the digital certificate using the public key of the master repository, recording the identity of the supplier and creator, a key for decrypting the software, the compatibility information, and a tamper-checking code. (This step certifies the software.)
- The requester decrypts the software using the key from the certificate and computes a check code on it using a 1-way hash function. If the check-code does not match the tamper-checking code from the certificate, the installation transaction ends with an error. (This step assures that the contents of the software, including the various scripts, have not been tampered with.)
- The requester retrieves the instructions in the compatibility-checking script and follows them. If the software is not compatible with the repository, the installation transaction ends with an error. (This step checks platform compatibility.)
- The requester retrieves the instructions in the installation script and follows them. If there is an error in this process (such as insufficient resources), then the transaction ends with an error. Note that the installation process puts the runnable software in a place in the repository where it is no longer accessible as a work for exercising any usage rights other than the execution of the software as part of repository operations in carrying out other transactions.
- The repositories perform the common closing transaction steps.

The Uninstall Transaction

An Uninstall transaction is a request to remove software from a repository. Since uncontrolled or incorrect removal of software from a repository could compromise its behavioral integrity, this step is controlled.

- The requester sends the server an Uninstall message. This message indicates the work to be uninstalled, the version of the Uninstall right being invoked, and the file data for the work (including its size).
- The repositories perform the common opening transaction steps.
- The requester extracts a copy of the digital certificate for the software. If the certificate cannot be found or the master repository for the certificate is not known to the requester, the transaction ends with an error.
- The requester checks whether the software is installed. If the software is not installed, the transaction ends with an error.
- The requester decrypts the digital certificate using the public key of the master repository, recording the identity of the supplier and creator, a key for decrypting the software, the compatibility information, and a tamper-checking code. (This step authenticates the certification of the software, including the script for uninstalling it.)
- The requester decrypts the software using the key from the certificate and computes a check code on it using a 1-way hash function. If the check-code does not match the tamper-checking code from the certificate, the installation transaction ends with an error. (This step assures that the contents of the software, including the various scripts, have not been tampered with.)
- The requester retrieves the instructions in the uninstallation script and follows them. If there is an error in this process (such as insufficient resources), then the transaction ends with an error.
- The repositories perform the common closing transaction steps.

Claims

1. A distribution system for distributing digital works, said digital works having one or more usage rights attached thereto, said distribution system comprising:

   a grammar for creating instances of usage rights indicating a manner by which a possessor of an associated digital work may transport said associated digital work;

means for creating usage rights from said grammar;
means for attaching created usage rights to a digital work;
a requester repository for accessing digital works, said requester repository having means for generating usage transactions, each said usage transaction specifying a usage right;
a server repository for storing digital works with attached created usage rights, said server repository having means for processing usage transactions from said requester repository to determine if access to a digital work may be granted.

2. The distribution system as recited in Claim 1 wherein said grammar further specifies a default plurality of conditions for an instance of a usage right, wherein said one or more conditions must be satisfied before said usage right may be exercised.

3. The distribution system as recited in Claim 2 wherein said means for creating usage rights from said grammar is further comprised of means for changing said default plurality of conditions for an instance of a usage right.

4. The distribution system as recited in Claim 1 wherein said digital work is a software program.

5. The distribution system as recited in Claim 1 wherein said grammar is further for creating a first version of a usage right having a first set of conditions and a second version of said usage right having a second set of conditions.

6. A computer based system for controlling distribution and use of digital works comprising:

a usage rights grammar for creating instances of usages rights which define how a digital work may be used or distributed, said usage rights grammar comprising a first plurality of grammar elements for defining transport usage rights and a second plurality of grammar elements for defining rendering usage rights;
means for attaching usage rights to digital works;
a plurality of repositories for storing and exchanging digital works, each of said plurality of repositories comprising :
means for storing digital works and their attached usage rights;
transaction processing means having a requester mode of operation for requesting access to a requested digital work, said request specifying a usage right, and a server mode of operation for processing requests to access said requested digital work based on said usage right specified in said request and the usage rights attached to said requested digital work; and
a coupling means for coupling to another of said plurality of repositories across a communications medium.

7. The computer based system for controlling distribution and use of digital works as recited in Claim 6 wherein said first plurality of grammar elements is comprised of:

a loan grammar element for enabling a digital work to be loaned to another repository;
a copy grammar element for enabling a copy of a digital work to be made and transported to another repository; and
a transfer grammar element for enabling a digital work to be transferred to another repository.

8. The computer based system for controlling distribution and use of digital works as recited in Claim 6 or Claim 7 wherein said second plurality of grammar elements is comprised of:

a play grammar element for enabling a digital work to be rendered on a specified class of player device; and
a print grammar element for enabling a digital work to be printed on a specified class of printer device.

9. The computer based system for controlling distribution and use of digital works as recited in any one of Claims 6 to 8 wherein said grammar comprises one or more further pluralities of grammar elements, for defining file management usage rights, for enabling a digital work to be used in the creation of a new digital work,for enabling the secure installation and uninstallation of digital works comprising of software programs, or for providing a set of creator specified conditions which must be satisfied for each instantiation of a usage right defined by a grammar element.

10. A method for controlling distribution and use of digital works comprising the steps of:

a) creating a set of usage rights from a usage rights grammar, each of said usage rights defining a specific instance of how a digital work may be used or distributed, each of said usage rights specifying one or more conditions which must be satisfied in order for said usage right to be exercised;
b) attaching said set of usage rights to a digital work;
c) storing said digital work and its attached usage rights in a first repository;
d) a second repository initiating a request to access said digital work in said first repository, said request specifying a usage right;
e) said first repository receiving said request from said second repository;
f) said first repository determining if said specified usage right is attached to said digital work;
g) said first repository denying access to said digital work if said identified usage right is not attached to said digital work;
h) if said identified usage right is attached to said digital work, said first repository determining if conditions specified by said usage right are satisfied;
i) if said conditions are not satisfied, said first repository denying access to said digital work;
j) if said conditions are satisfied, said first repository transmitting said digital work to said second repository.

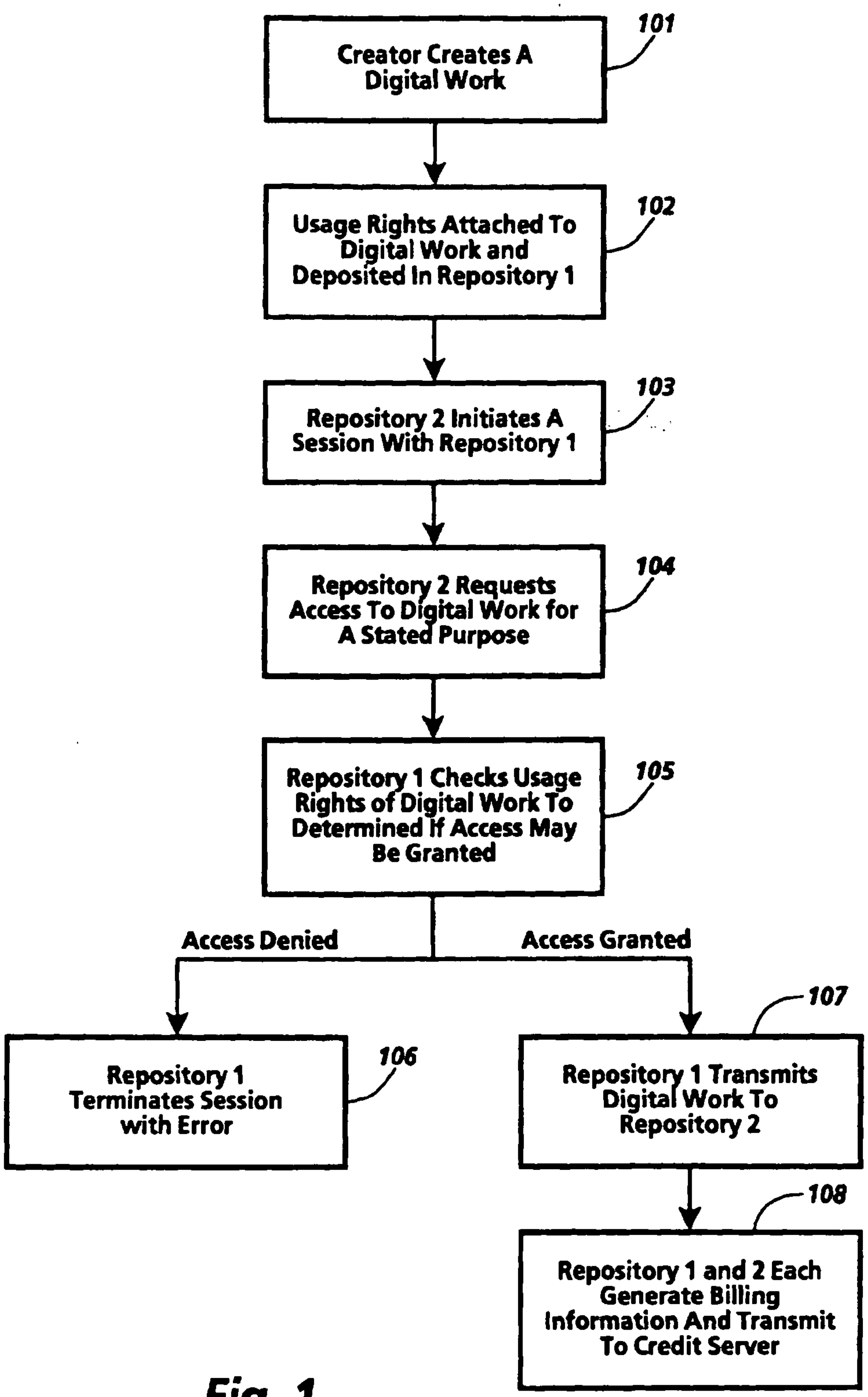

Creator Creates A Digital Work
101
Usage Rights Attached To Digital Work and Deposited In Repository 1
102
Repository 2 Initiates A Session With Repository 1
103
Repository 2 Requests Access To Digital Work for A Stated Purpose
104
Repository 1 Checks Usage Rights of Digital Work To Determined If Access May Be Granted
105
Access Denied
Access Granted
107
Repository 1 Terminates Session with Error
106
Repository 1 Transmits Digital Work To Repository 2
108
Repository 1 and 2 Each Generate Billing Information And Transmit To Credit Server

Fig. 1

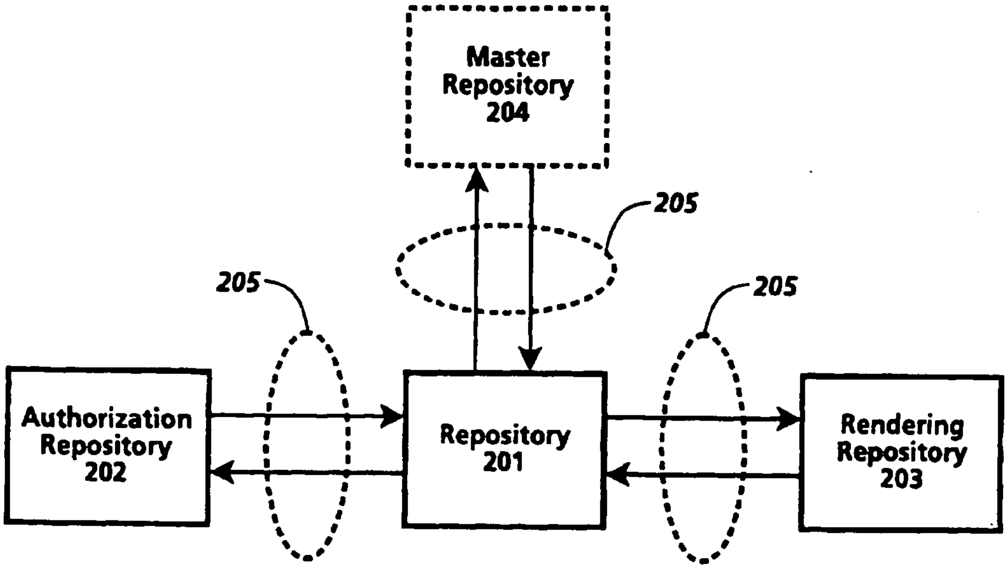

Master Repository 204
205
205
205
Authorization Repository 202
Repository 201
Rendering Repository 203

Fig. 2

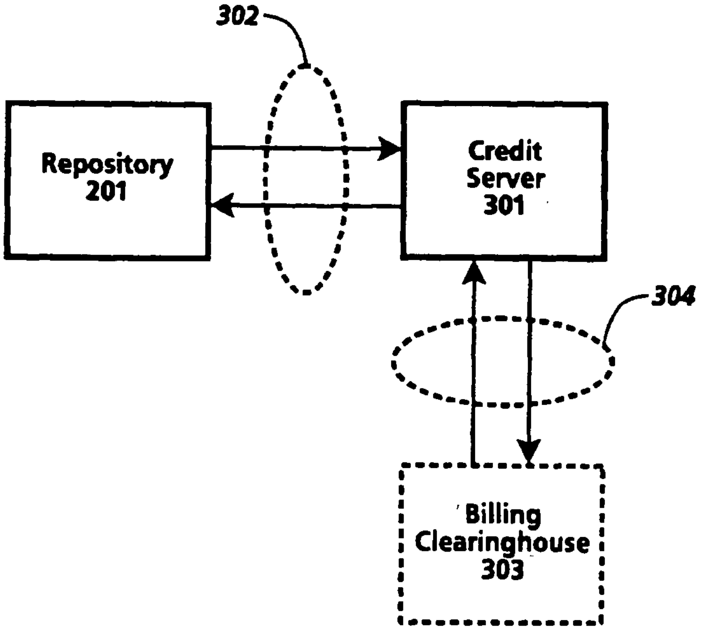

302
Repository 201
Credit Server 301
304
Billing Clearinghouse 303

Fig. 3

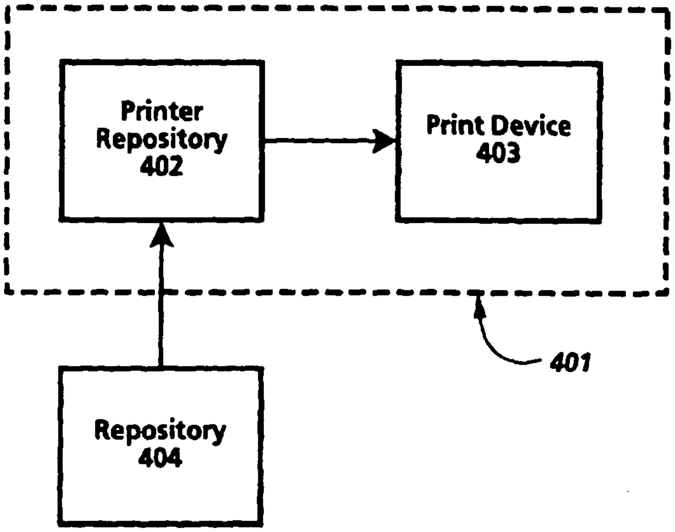

Printer Repository 402
Print Device 403
401
Repository 404

Fig. 4a

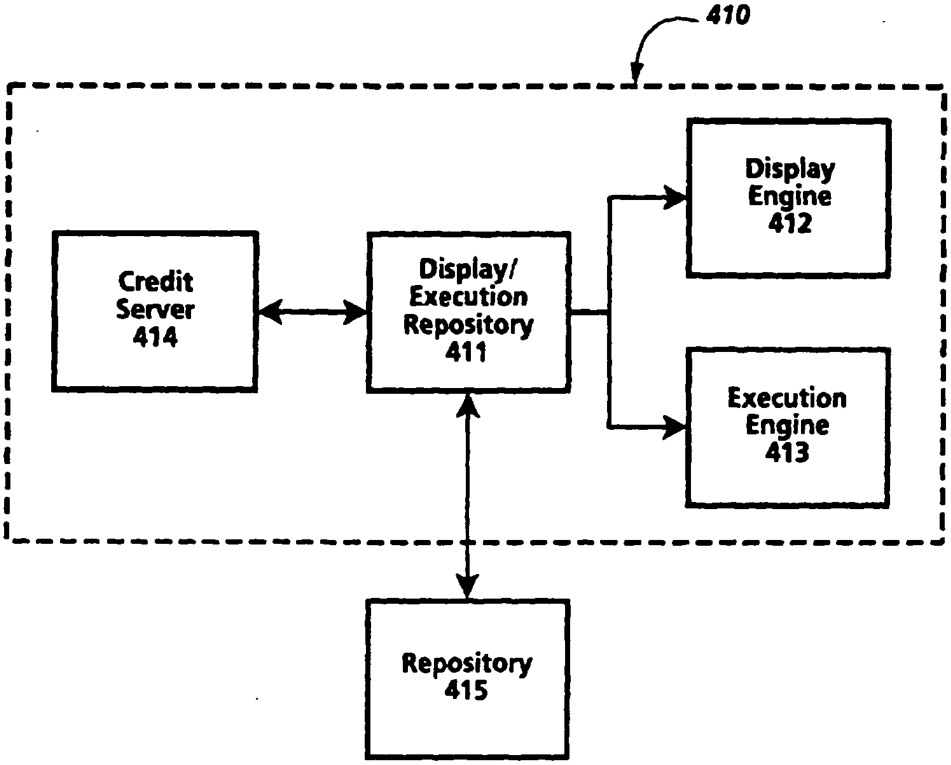

410
Credit Server 414
Display/ Execution Repository 411
Display Engine 412
Execution Engine 413
Repository 415

Fig. 4b

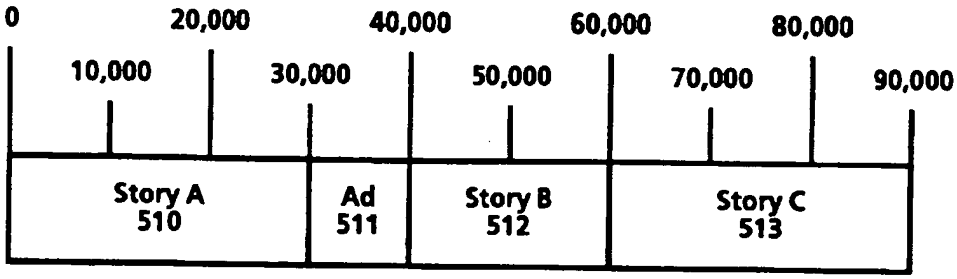

0
10,000
20,000
30,000
40,000
50,000
60,000
70,000
80,000
90,000
Story A
510
Ad
511
Story B
512
Story C
513

Fig. 5

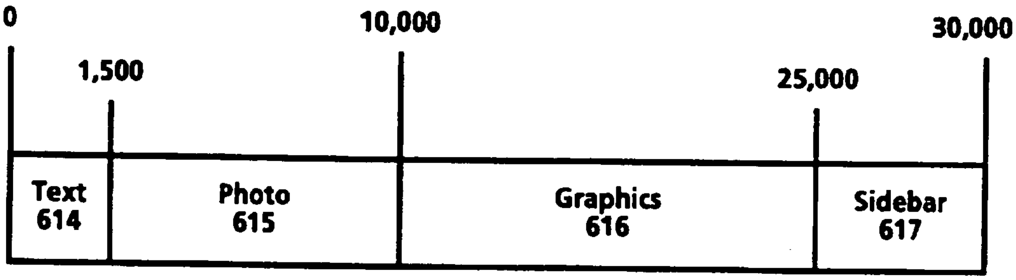

0
1,500
10,000
25,000
30,000
Text
614
Photo
615
Graphics
616
Sidebar
617

Fig. 6

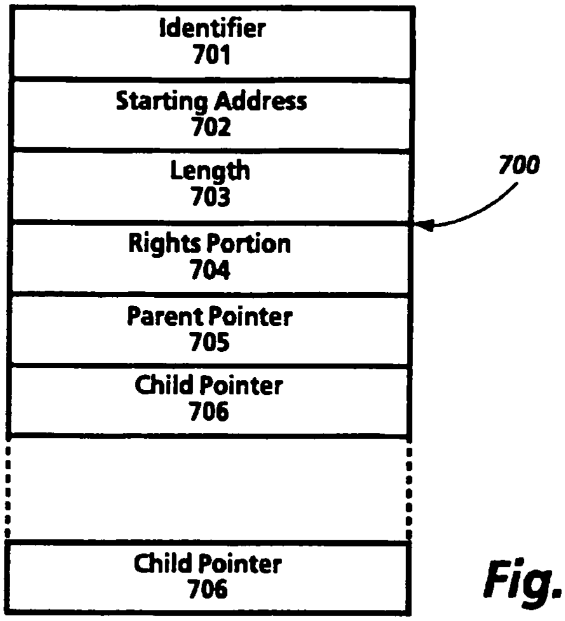

Identifier
701
Starting Address
702
Length
703
Rights Portion
704
Parent Pointer
705
Child Pointer
706
Child Pointer
706
700

Fig. 7

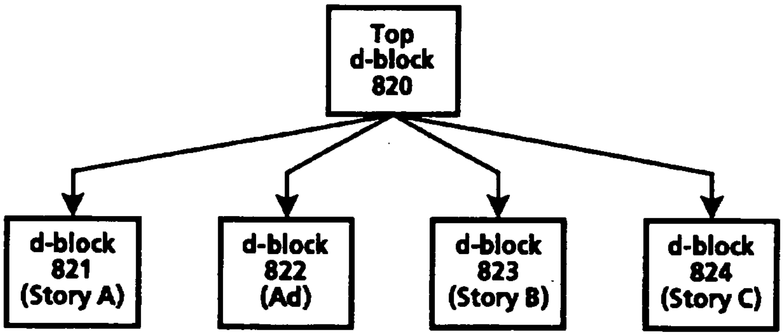

Top
d-block
820
d-block
821
(Story A)
d-block
822
(Ad)
d-block
823
(Story B)
d-block
824
(Story C)

Fig. 8

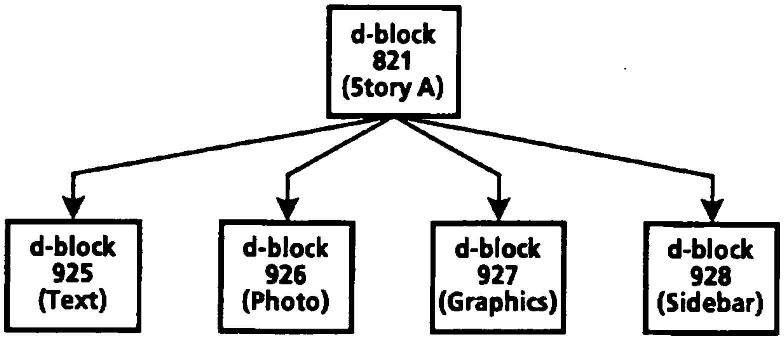

d-block
821
(Story A)
d-block
925
(Text)
d-block
926
(Photo)
d-block
927
(Graphics)
d-block
928
(Sidebar)

Fig. 9

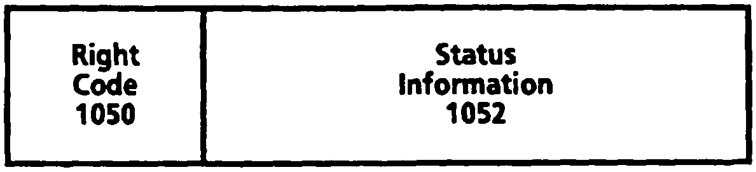

Right
Code
1050
Status
Information
1052

Fig.10

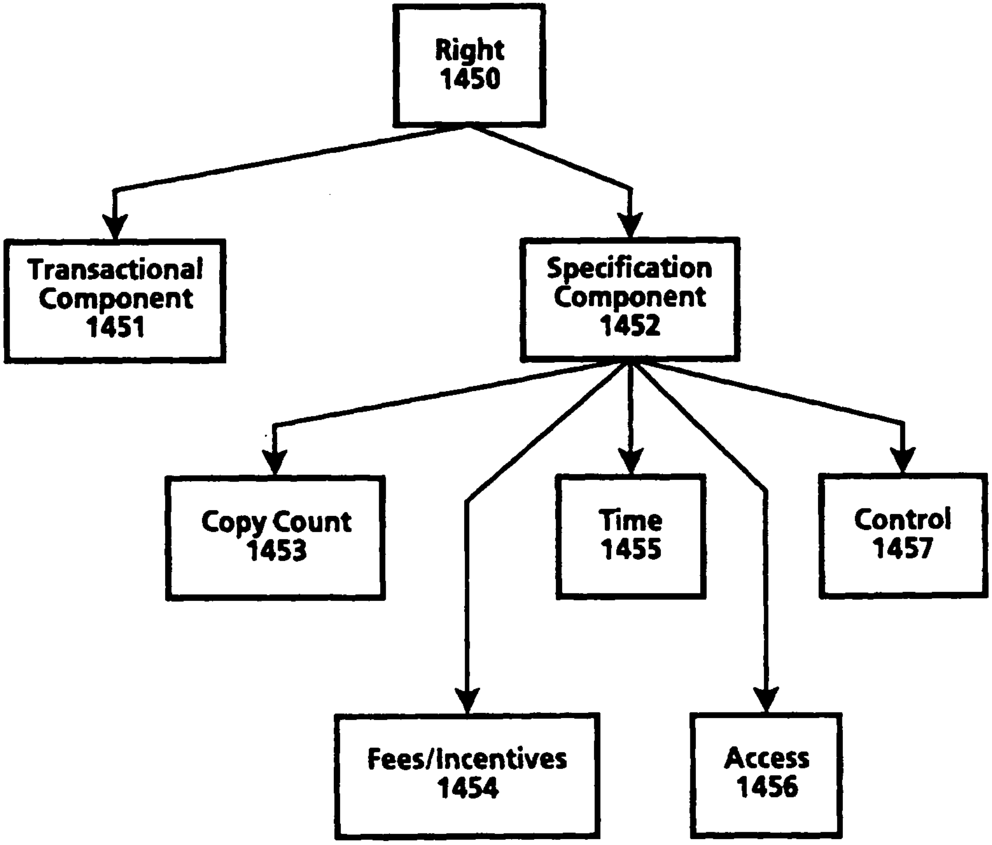

Right
1450
Transactional
Component
1451
Specification
Component
1452
Copy Count
1453
Time
1455
Control
1457
Fees/Incentives
1454
Access
1456

Fig.14

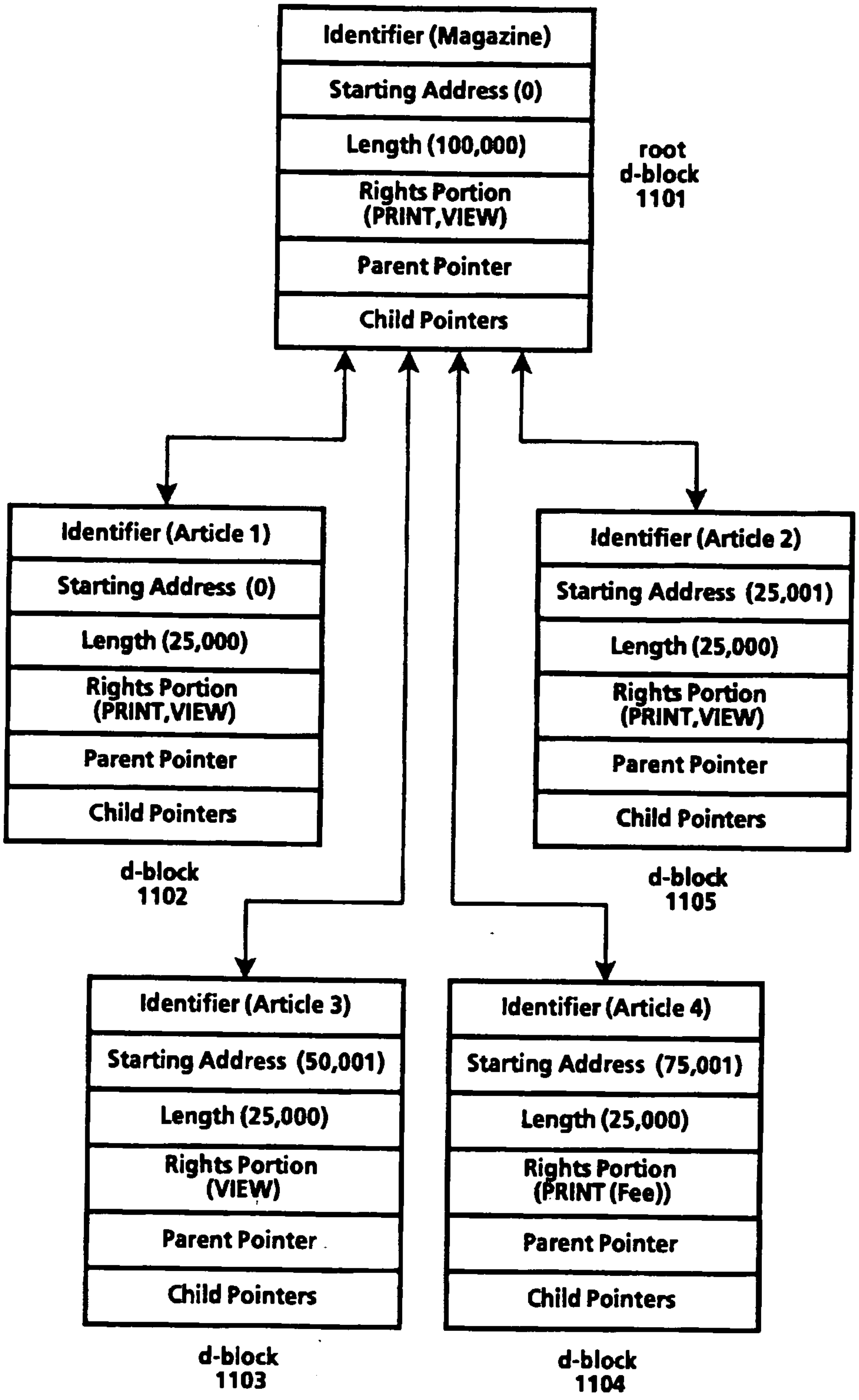

Identifier (Magazine)
Starting Address (0)
Length (100,000)
Rights Portion (PRINT,VIEW)
Parent Pointer
Child Pointers
root d-block 1101
Identifier (Article 1)
Starting Address (0)
Length (25,000)
Rights Portion (PRINT,VIEW)
Parent Pointer
Child Pointers
d-block 1102
Identifier (Article 2)
Starting Address (25,001)
Length (25,000)
Rights Portion (PRINT,VIEW)
Parent Pointer
Child Pointers
d-block 1105
Identifier (Article 3)
Starting Address (50,001)
Length (25,000)
Rights Portion (VIEW)
Parent Pointer
Child Pointers
d-block 1103
Identifier (Article 4)
Starting Address (75,001)
Length (25,000)
Rights Portion (PRINT (Fee))
Parent Pointer
Child Pointers
d-block 1104

Fig.11

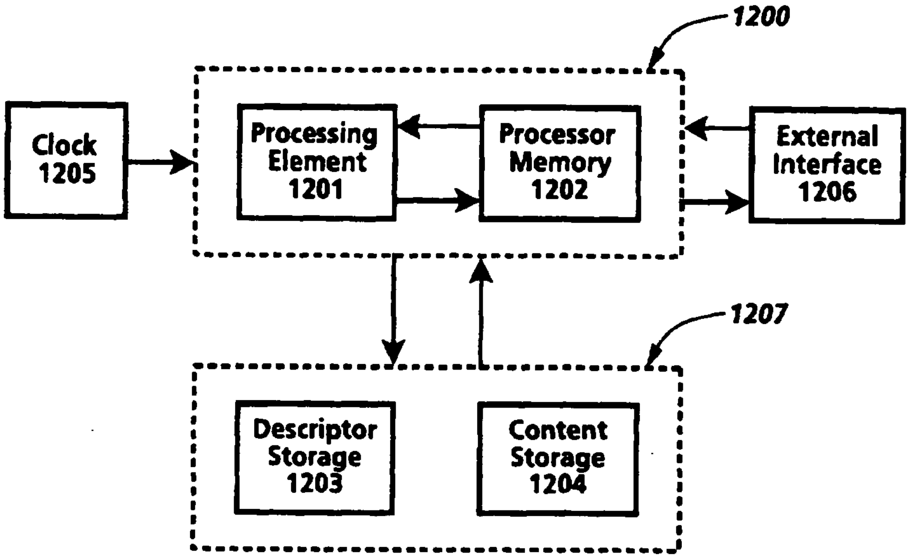

1200
Clock
1205
Processing
Element
1201
Processor
Memory
1202
External
Interface
1206
1207
Descriptor
Storage
1203
Content
Storage
1204

Fig.12

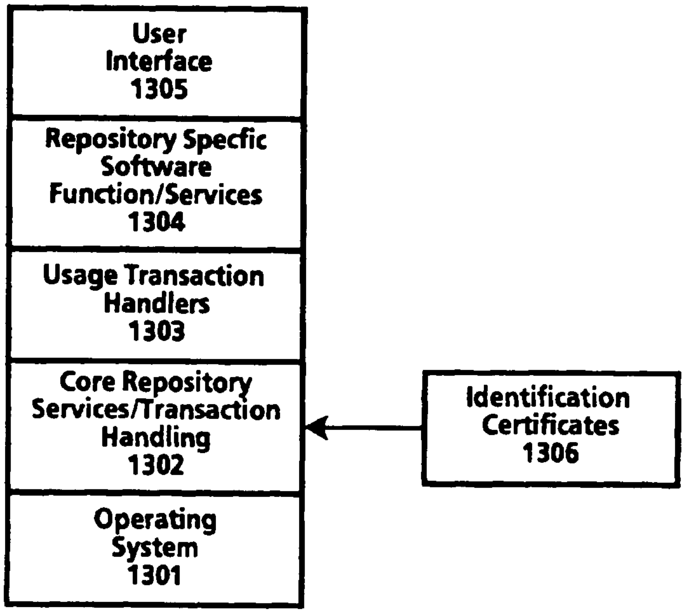

User
Interface
1305
Repository Specfic
Software
Function/Services
1304
Usage Transaction
Handlers
1303
Core Repository
Services/Transaction
Handling
1302
Operating
System
1301
Identification
Certificates
1306

Fig.13

```
1501~Digital Work Rights: = (Rights*)
 1502~Right := (Right-Code {Copy-Count} {Control-Spec} {Time-Spec }
          {Access-Spec} {Fee-Spec})
   1503~Right-Code := Render-Code | Transport-Code | File-Management-
            Code| Derivative-Works- Code | Configuration-Code
     1504~Render-Code := [ Play : {Player: Player-ID} | Print: {Printer: Printer-ID}]
     1505~Transport-Code := [Copy | Transfer | Loan {Remaining-Rights:
              Next-Set-of-Rights}]{(Next-Copy-Rights: Next-Set-of-Rights)}
     1506~File-Management-Code    := Backup {Back-Up-Copy-Rights:
                                     Next-Set-of-Rights} | Restore | Delete | Folder
                                     | Directory {Name: Hide-Local | Hide-Remote}
                                     {Parts: Hide-Local | Hide-Remote}
     1507~Derivative-Works-Code :=    [Extract | Embed | Edit{Process:
                                       Process-ID}] {Next-Copy-Rights :
                                       Next-Set-of Rights}
     1508~Configuration-Code := Install | Uninstall
     1509~Next-Set-of-Rights := {(Add: Set-Of-Rights)} {(Delete:
             Set-Of-Rights)} {(Replace: Set-Of-Rights )}{(Keep: Set-Of-Rights )}
1510~Copy-Count := (Copies:positive-integer | 0 | Unlimited)
1511~Control-Spec := (Control: {Restrictable | Unrestrictable}
                      {Unchargeable | Chargeable})
1512~Time-Spec := ({Fixed-Interval  | Sliding-Interval | Meter-Time}
                   Until: Expiration-Date)
  1513~ Fixed-Interval := From: Start-Time
  1514~Sliding-Interval := Interval: Use-Duration
  1515~Meter-Time: = Time-Remaining: Remaining-Use
1516~ Access-Spec :=  ({SC: Security-Class} {Authorization: Authorization-ID*}
        {Other-Authorization: Authorization-ID*} {Ticket: Ticket-ID})
1517~Fee-Spec: = {Scheduled-Discount} Regular-Fee-Spec | Scheduled-Fee-Spec |
                  Markup-Spec
  1518~Scheduled-Discount: = Scheduled-Discount: (Scheduled-Discount:
                               (Time-Spec Percentage)*)
  1519~Regular-Fee-Spec := ({Fee:  | Incentive: } [Per-Use-Spec | Metered-Rate-
                           Spec | Best-Price-Spec | Call-For-Price-Spec]
                           {Min: Money-Unit Per: Time-Spec}{Max:
                           Money-Unit Per: Time-Spec} To: Account-ID)
    1520~Per-Use-Spec: = Per-Use: Money-unit
    1521~Metered-Rate-Spec := Metered: Money-Unit Per: Time-Spec
    1522~Best-Price-Spec := Best-Price: Money-unit Max: Money-unit
    1523~Call-For-Price-Spec := Call-For -Price
  1524~ Scheduled-Fee-Spec: = (Schedule: (Time-Spec Regular-Fee-Spec)* )
  1525~Markup-Spec: = Markup: percentage To: Account-ID
```

Fig. 15

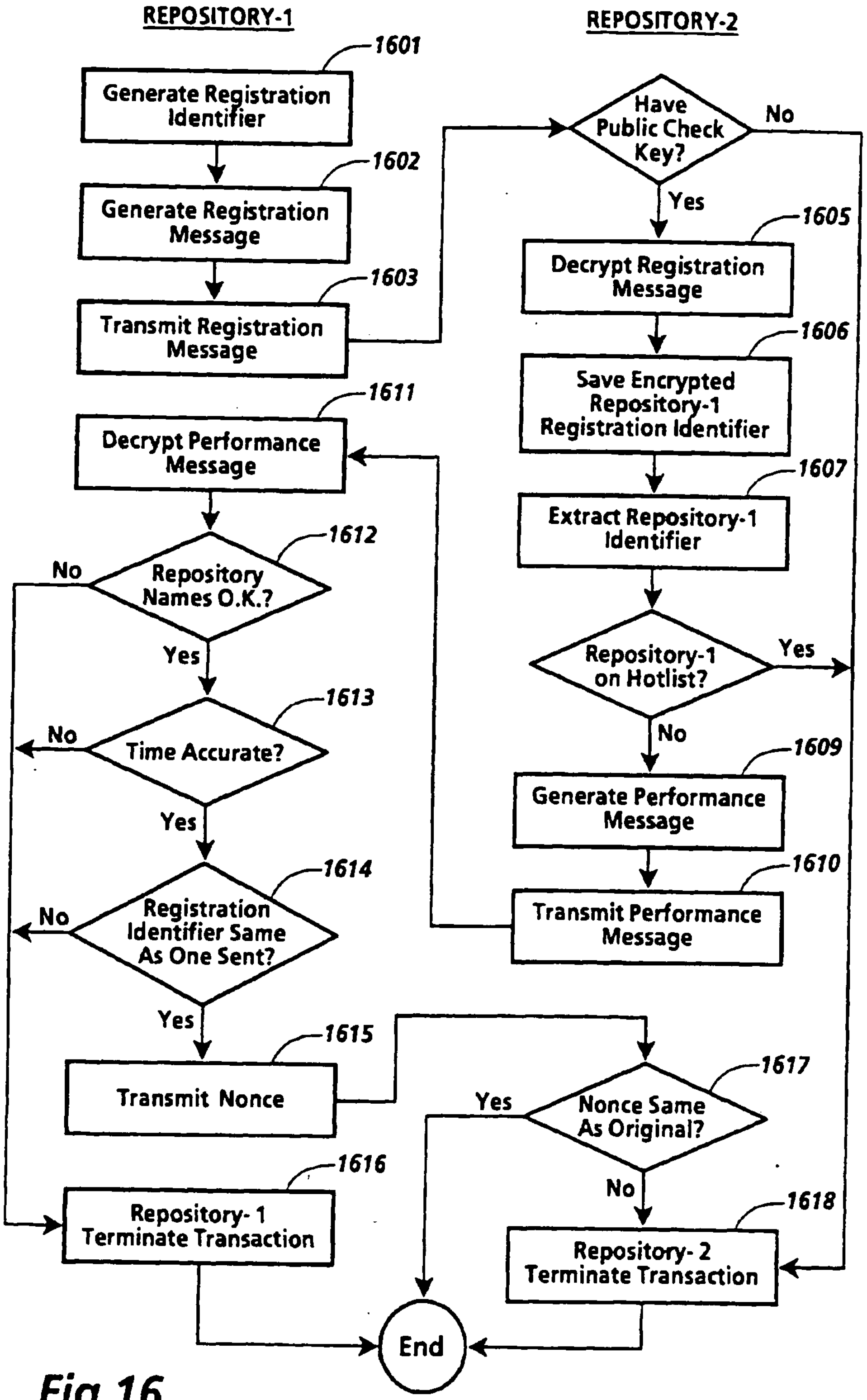

REPOSITORY-1
REPOSITORY-2
1601
Generate Registration Identifier
1602
Generate Registration Message
1603
Transmit Registration Message
Have Public Check Key?
No
Yes
1605
Decrypt Registration Message
1606
Save Encrypted Repository-1 Registration Identifier
1607
Extract Repository-1 Identifier
Repository-1 on Hotlist?
Yes
No
1609
Generate Performance Message
1610
Transmit Performance Message
1611
Decrypt Performance Message
1612
Repository Names O.K.?
No
Yes
1613
Time Accurate?
No
Yes
1614
Registration Identifier Same As One Sent?
No
Yes
1615
Transmit Nonce
1617
Nonce Same As Original?
Yes
No
1616
Repository- 1 Terminate Transaction
1618
Repository- 2 Terminate Transaction
End

Fig.16

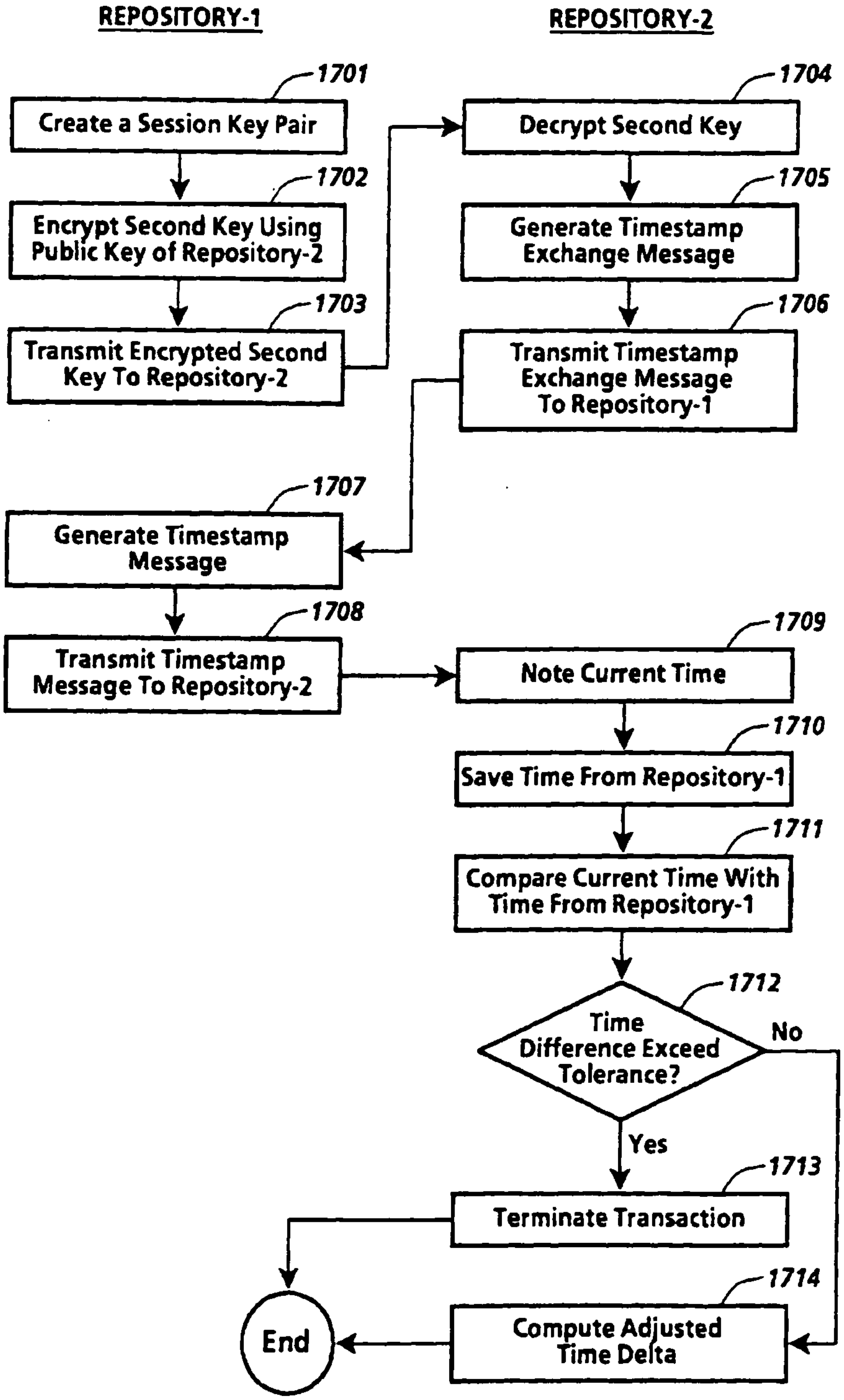

REPOSITORY-1
REPOSITORY-2
1701
Create a Session Key Pair
1702
Encrypt Second Key Using Public Key of Repository-2
1703
Transmit Encrypted Second Key To Repository-2
1704
Decrypt Second Key
1705
Generate Timestamp Exchange Message
1706
Transmit Timestamp Exchange Message To Repository-1
1707
Generate Timestamp Message
1708
Transmit Timestamp Message To Repository-2
1709
Note Current Time
1710
Save Time From Repository-1
1711
Compare Current Time With Time From Repository-1
1712
Time Difference Exceed Tolerance?
No
Yes
1713
Terminate Transaction
1714
Compute Adjusted Time Delta
End

Fig.17

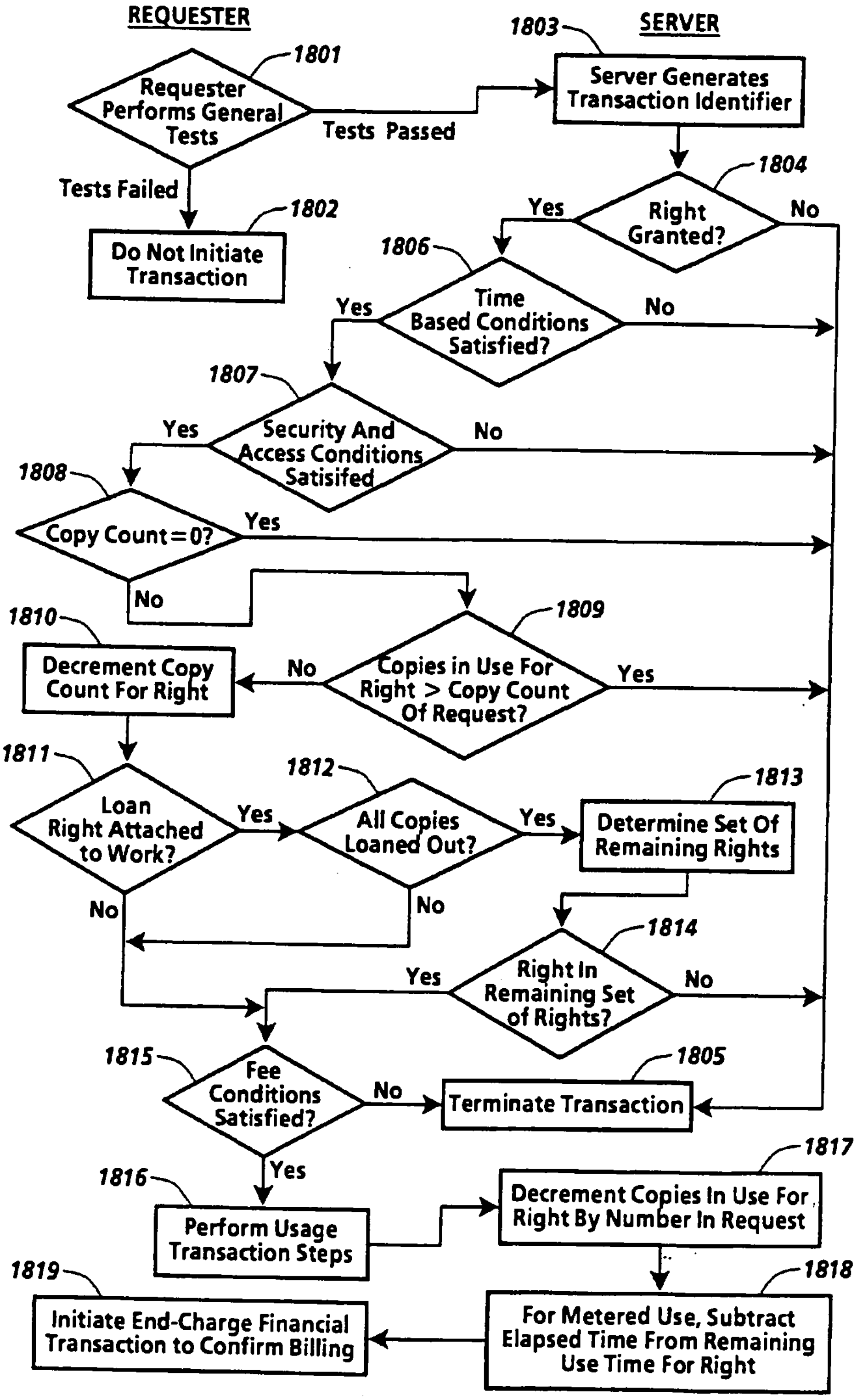

REQUESTER
SERVER
1801
Requester Performs General Tests
Tests Passed
Tests Failed
1802
Do Not Initiate Transaction
1803
Server Generates Transaction Identifier
1804
Right Granted?
Yes
No
1806
Time Based Conditions Satisfied?
Yes
No
1807
Security And Access Conditions Satisifed
Yes
No
1808
Copy Count=0?
Yes
No
1809
Copies in Use For Right > Copy Count Of Request?
No
Yes
1810
Decrement Copy Count For Right
1811
Loan Right Attached to Work?
Yes
No
1812
All Copies Loaned Out?
Yes
No
1813
Determine Set Of Remaining Rights
1814
Right In Remaining Set of Rights?
Yes
No
1815
Fee Conditions Satisfied?
No
Yes
1805
Terminate Transaction
1816
Perform Usage Transaction Steps
1817
Decrement Copies In Use For Right By Number In Request
1818
For Metered Use, Subtract Elapsed Time From Remaining Use Time For Right
1819
Initiate End-Charge Financial Transaction to Confirm Billing

Fig.18

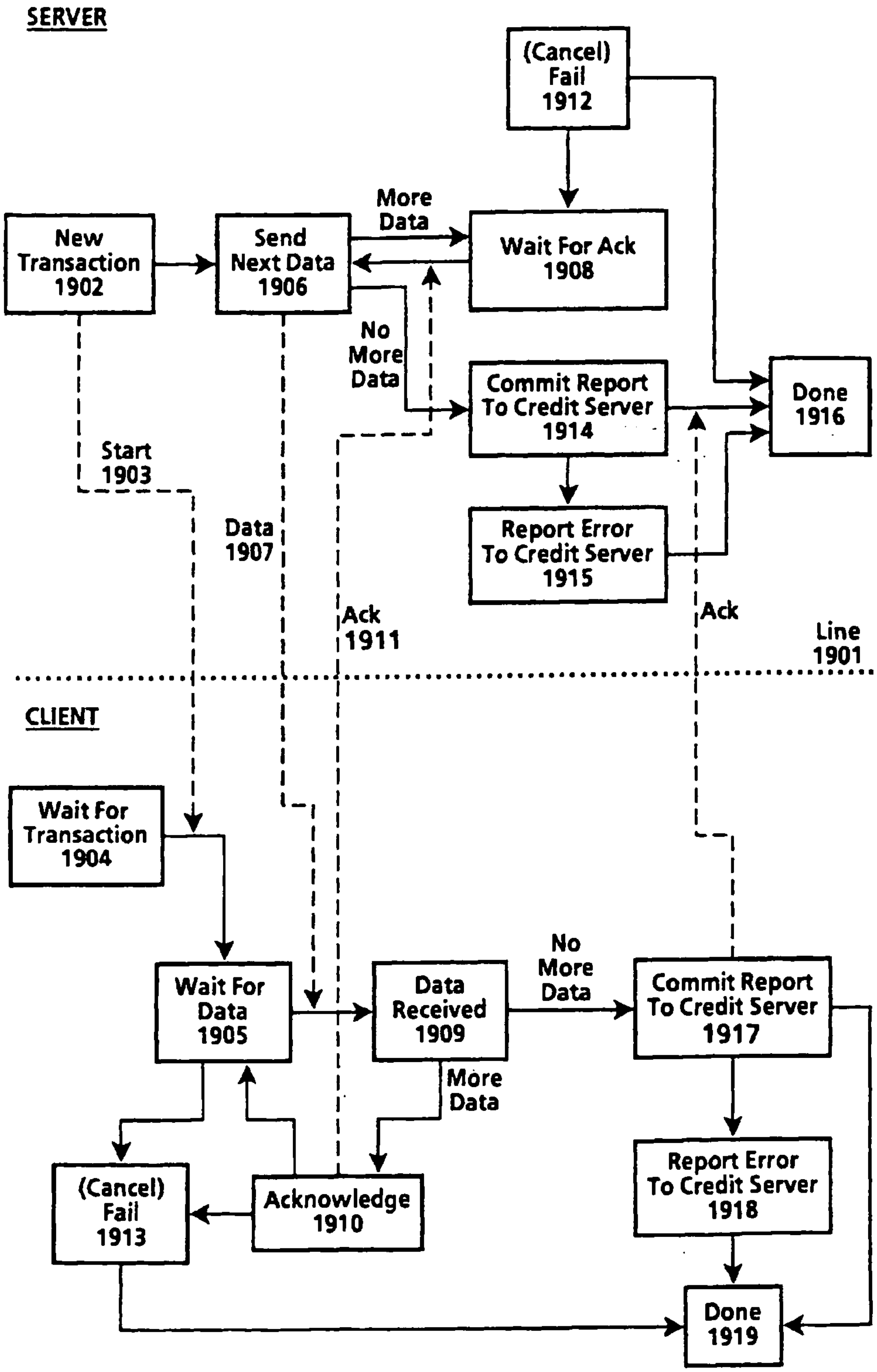

SERVER
(Cancel) Fail 1912
New Transaction 1902
Send Next Data 1906
More Data
Wait For Ack 1908
No More Data
Commit Report To Credit Server 1914
Done 1916
Report Error To Credit Server 1915
Start 1903
Data 1907
Ack 1911
Ack
Line 1901
CLIENT
Wait For Transaction 1904
Wait For Data 1905
Data Received 1909
No More Data
Commit Report To Credit Server 1917
More Data
(Cancel) Fail 1913
Acknowledge 1910
Report Error To Credit Server 1918
Done 1919

Fig.19

European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 95 30 8417

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	WO-A-92 20022 (DIGITAL EQUIPMENT CORP.) * page 45, line 10 - page 80, line 19; figures 1-43 * ---	1,6,10	G06F1/00
A	US-A-5 291 596 (MIITA) * the whole document * ---	1,6,10	
A	GB-A-2 236 604 (SUN MICROSYSTEMS INC) * page 9, line 11 - page 20, line 15 * -----	1,6,10	
			TECHNICAL FIELDS SEARCHED (Int.Cl.6) G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	1 April 1996	Moens, R

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 715 245 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
05.06.1996 Bulletin 1996/23

(51) Int Cl.6: G06F 1/00

(21) Application number: 95308420.9

(22) Date of filing: 23.11.1995

(84) Designated Contracting States:
DE FR GB

(30) Priority: 23.11.1994 US 344042

(71) Applicant: **XEROX CORPORATION**
Rochester New York 14644 (US)

(72) Inventors:
- **Stefik, Mark J.**
  Woodside, California 94062 (US)
- **Casey, Michalene M.**
  Morgan Hill, California 95037 (US)

(74) Representative: **Goode, Ian Roy**
Rank Xerox Ltd
Patent Department
Parkway
Marlow Buckinghamshire SL7 1YL (GB)

(54) System for controlling the distribution and use of digital works

(57) A system for controlling use and distribution of digital works, in which the owner of a digital work (101) attaches usage rights (102) to that work. Usage rights are granted by the "owner" of a digital work to "buyers" of the digital work. The usage rights define how a digital work may be used and further distributed by the buyer. Each right has associated with it certain optional specifications which outline the conditions and fees upon which the right may be exercised. Digital works are stored in a repository. A repository will process each request (103,104) to access a digital work by examining the corresponding usage rights (105). Digital work playback devices, coupled to the repository containing the work, are used to play, display or print the work. Access to digital works for the purposes of transporting between repositories (e.g. copying, borrowing or transfer) is carried out using a digital work transport protocol. Access to digital works for the purposes of replay by a digital work playback device(e.g. printing, displaying or executing) is carried out using a digital work playback protocol. Access is denied (106) or granted (107) depending whether the requesting repository has the required usage rights.

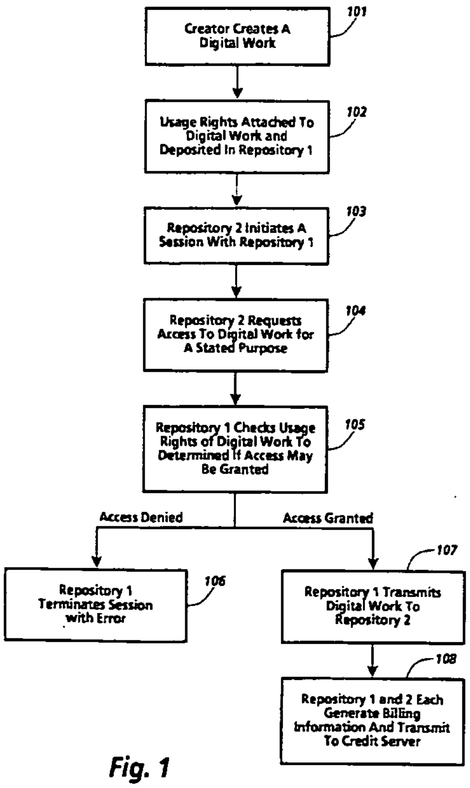

Creator Creates A Digital Work
101
Usage Rights Attached To Digital Work and Deposited In Repository 1
102
Repository 2 Initiates A Session With Repository 1
103
Repository 2 Requests Access To Digital Work for A Stated Purpose
104
Repository 1 Checks Usage Rights of Digital Work To Determined If Access May Be Granted
105
Access Denied
Access Granted
107
Repository 1 Terminates Session with Error
106
Repository 1 Transmits Digital Work To Repository 2
108
Repository 1 and 2 Each Generate Billing Information And Transmit To Credit Server

Fig. 1

EP 0 715 245 A1

Printed by Jouve, 75001 PARIS (FR)

Description

The present invention relates to the field of distribution and usage rights enforcement for digitally encoded works.

A fundamental issue facing the publishing and information industries as they consider electronic publishing is how to prevent the unauthorized and unaccounted distribution or usage of electronically published materials. Electronically published materials are typically distributed in a digital form and recreated on a computer based system having the capability to recreate the materials. Audio and video recordings, software, books and multimedia works are all being electronically published. Companies in these industries receive royalties for each accounted for delivery of the materials, e.g. the sale of an audio CD at a retail outlet. Any unaccounted distribution of a work results in an unpaid royalty (e.g. copying the audio recording CD to another digital medium.)

The ease in which electronically published works can be "perfectly" reproduced and distributed is a major concern. The transmission of digital works over networks is commonplace. One such widely used network is the Internet. The Internet is a widespread network facility by which computer users in many universities, corporations and government entities communicate and trade ideas and information. Computer bulletin boards found on the Internet and commercial networks such as CompuServ and Prodigy allow for the posting and retrieving of digital information. Information services such as Dialog and LEXIS/NEXIS provide databases of current information on a wide variety of topics. Another factor which will exacerbate the situation is the development and expansion of the National Information Infrastructure (the NII). It is anticipated that, as the NII grows, the transmission of digital works over networks will increase many times over. It would be desirable to utilize the NII for distribution of digital works without the fear of widespread unauthorized copying.

The most straightforward way to curb unaccounted distribution is to prevent unauthorized copying and transmission. For existing materials that are distributed in digital form, various safeguards are used. In the case of software, copy protection schemes which limit the number of copies that can be made or which corrupt the output when copying is detected have been employed. Another scheme causes software to become disabled after a predetermined period of time has lapsed. A technique used for workstation based software is to require that a special hardware device must be present on the workstation in order for the software to run, e.g., see US-A-4,932,054 entitled "Method and Apparatus for Protecting Computer Software Utilizing Coded Filter Network in Conjunction with an Active Coded Hardware Device." Such devices are provided with the software and are commonly referred to as dongles.

Yet another scheme is to distribute software, but which requires a "key" to enable its use. This is employed in distribution schemes where "demos" of the software are provided on a medium along with the entire product. The demos can be freely used, but in order to use the actual product, the key must be purchased. These schemes do not hinder copying of the software once the key is initially purchased.

It is an object of the present invention to provide an improved system and method for controlling the use and distribution of digital works.

The invention accordingly provides a system and method as claimed in the accompanying claims.

A system for controlling use and distribution of digital works is disclosed. A digital work is any written, aural, graphical or video based work including computer programs that has been translated to or created in a digital form, and which can be recreated using suitable rendering means such as software programs. The present invention allows the owner of a digital work to attach usage rights to the work. The usage rights for the work define how it may be used and distributed. Digital works and their usage rights are stored in a secure repository. Digital works may only be accessed by other secure repositories.

Usage rights for a digital work are embodied in a flexible and extensible usage rights grammar. Conceptually, a right in the usage rights grammar is a label attached to a predetermined behavior and conditions to exercising the right. For example, a COPY right denotes that a copy of the digital work may be made. A condition to exercising the right is the requester must pass certain security criteria. Conditions may also be attached to limit the right itself. For example, a LOAN right may be defined so as to limit the duration of which a work may be LOANed. Conditions may also include requirements that fees be paid.

A repository is comprised of a storage means for storing a digital work and its attached usage rights, an external interface for receiving and transmitting data, a processor and a clock. A repository has two primary operating modes, a server mode and a requester mode. When operating in a server mode, the repository is responding to requests to access digital works. When operating in requester mode, the repository is requesting access to a digital work.

Generally, a repository will process each request to access a digital work by examining the work's usage rights. For example, in a request to make a copy of a digital work, the digital work is examined to see if rights have been granted which would allow copies to be given out. If such a right has been granted, then conditions to exercise of the right are checked (e.g. a right to make 2 copies). If conditions associated with the right are satisfied, the copy can be made. Before transporting the digital work, any specified changes to the set of usage rights in the copy are attached to the copy of the digital work.

Repositories communicate utilizing a set of repository transactions. The repository transactions embody a set of protocols for establishing secure sessions connections between repositories, and for processing access requests to the digital works.

Digital works are recreated on rendering systems. A rendering system is comprised of at least a rendering repository and a rendering device (e.g. a printer, display or audio system.) Rendering systems are internally secure. Access to digital works not contained within the rendering repository is accomplished via repository transactions with an external repository containing the desired digital work.

A system and method in accordance with the invention will now be described, by way of example, with reference to the accompanying drawings, in which:-

Figure 1 is a flowchart illustrating a simple instantiation of the operation of the currently preferred embodiment of the present invention.

Figure 2 is a block diagram illustrating the various repository types and the repository transaction flow between them in the currently preferred embodiment of the present invention.

Figure 3 is a block diagram of a repository coupled with a credit server in the currently preferred embodiment of the present invention.

Figures 4a and 4b are examples of rendering systems as may be utilized in the currently preferred embodiment of the present invention.

Figure 5 illustrates a contents file layout for a digital work as may be utilized in the currently preferred embodiment of the present invention.

Figure 6 illustrates a contents file layout for an individual digital work of the digital work of Figure 5 as may be utilized in the currently preferred embodiment of the present invention.

Figure 7 illustrates the components of a description block of the currently preferred embodiment of the present invention.

Figure 8 illustrates a description tree for the contents file layout of the digital work illustrated in Figure 5.

Figure 9 illustrates a portion of a description tree corresponding to the individual digital work illustrated in Figure 6.

Figure 10 illustrates a layout for the rights portion of a description block as may be utilized in the currently preferred embodiment of the present invention.

Figure 11 is a description tree wherein certain d-blocks have PRINT usage rights and is used to illustrate "strict" and "lenient" rules for resolving usage rights conflicts.

Figure 12 is a block diagram of the hardware components of a repository as are utilized in the currently preferred embodiment of the present invention.

Figure 13 is a block diagram of the functional (logical) components of a repository as are utilized in the currently preferred embodiment of the present invention.

Figure 14 is diagram illustrating the basic components of a usage right in the currently preferred embodiment of the present invention.

Figure 15 lists the usage rights grammar of the currently preferred embodiment of the present invention.

Figure 16 is a flowchart illustrating the steps of certificate delivery, hotlist checking and performance testing as performed in a registration transaction as may be performed in the currently preferred embodiment of the present invention.

Figure 17 is a flowchart illustrating the steps of session information exchange and clock synchronization as may be performed in the currently preferred embodiment of the present invention, after each repository in the registration transaction has successfully completed the steps described in Figure 16.

Figure 18 is a flowchart illustrating the basic flow for a usage transaction, including the common opening and closing step, as may be performed in the currently preferred embodiment of the present invention.

Figure 19 is a state diagram of server and client repositories in accordance with a transport protocol followed when moving a digital work from the server to the client repositories, as may be performed in the currently preferred embodiment of the present invention.

OVERVIEW

A system for controlling use and distribution of digital works is disclosed. The present invention is directed to supporting commercial transactions involving digital works.

Herein the terms "digital work", "work" and "content" refer to any work that has been reduced to a digital representation. This would include any audio, video, text, or multimedia work and any accompanying interpreter (e.g. software) that may be required for recreating the work. The term composite work refers to a digital work comprised of a collection of other digital works. The term "usage rights" or "rights" is a term which refers to rights granted to a recipient of a digital work. Generally, these rights define how a digital work can be used and if it can be further distributed. Each usage right may have one or more specified conditions which must be satisfied before the right may be exercised.

Figure 1 is a high level flowchart omitting various details but which demonstrates the basic operation of the present