PTO/SB/05 (03-01)
Approved for use through 10/31/2002. OMB 0651-0032

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.	111325-113
First Inventor	Xin Wang et al.
Title	***Method And Apparatus Managing The Transfer Of Rights***
Express Mail Label No.	

APPLICATION ELEMENTS

See *MPEP chapter 600 concerning utility patent application contents.*

ADDRESS TO: Commissioner for Patents
Box Patent Application
Washington, DC 20231

1. ☐ Fee Transmittal Form (e.g., PTO/SB/17)
(Submit an original and a duplicate for fee processing)
2. ☐ Applicant claims small entity status.
See 37 CFR 1.27.
3. ☒ Specification [Total Pages **23**]
(preferred arrangement set forth below)
   - Descriptive title of the invention
   - Cross Reference to Related Applications *(if applicable)*
   - Statement Regarding Fed sponsored R & D *(if applicable)*
   - Reference to sequence listing, a table, or a computer program listing appendix *(if applicable)*
   - Background of the Invention
   - Brief Summary of the Invention
   - Brief Description of the Drawings *(if filed)*
   - Detailed Description
   - Claim(s)
   - Abstract of the Disclosure
4. ☒ Drawing(s) *(35 U.S.C. 113)* [*Total Sheets* **07**]
5. Oath or Declaration [*Total Pages* ☐]
   a. ☐ Newly executed (original or copy)
   b. ☐ Copy from a prior application (37 CFR 1.63(d))
   (for continuation/divisional with Box 18 completed)
      i. ☐ DELETION OF INVENTOR(S)
      Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b)
6. ☐ Application Data Sheet. See 37 CFR 1.76
7. ☐ CD-ROM or CD-R in duplicate, large table or Computer Program (Appendix)
8. Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
   a. ☐ Computer Readable Form (CRF)
   b. Specification Sequence Listing on:
      i. ☐ CD-ROM or CD-R (2 copies; or
      ii. ☐ paper
   c. ☐ Statements verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

9. ☐ Assignment Papers (cover sheet & document(s))
10. ☐ 37 CFR 3.73(b) Statement ☐ Power of Attorney *(when there is an assignee)*
11. ☐ English Translation Document *(if applicable)*
12. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations
13. ☐ Preliminary Amendment
14. ☒ Return Receipt Postcard (MPEP 503) *(Should be specifically itemized)*
15. ☐ Certified Copy of Priority Document(s) *(if foreign priority is claimed)*
16. ☐ Nonpublication request under 35 U.S.C. 122(b)(2)(B)(i). Applicant must attach form PTO/SB/35 or its equivalent.
17. ☐ Other: _____

18. If a CONTINUING APPLICATION, *check appropriate box, and supply the requisite information below and in a preliminary amendment, or in an Application Data Sheet under 37 CFR 1.76:*

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____ / _____

Prior application information: Examiner _____ Group / Art Unit: _____

For CONTINUATION OR DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

19. CORRESPONDENCE ADDRESS

☒ *Customer Number or Bar Code Label* 22204 or ☐ *Correspondence address below*

Name	Marc S. Kaufman				
Address					
City		State		Zip Code	
Country		Telephone		Fax	

Name (Print/Type)	Marc S. Kaufman	Registration No. (Attorney/Agent)	35,212
Signature		Date	June 6, 2002

NVA228768.1

METHOD AND APPARATUS MANAGING THE TRANSFER OF RIGHTS

RELATED APPLICATION DATA

[0001] This application claims benefit from U.S. provisional applications Ser. Nos. 60/331,624, 60/331,623, and 60/331,621 filed on November 20, 2001, the disclosures of which are incorporated herein by reference. This application also claims benefit of U.S. provisional applications Ser. Nos. 60/296,113, 60/296,117, and 60/296,118 filed on June 7, 2001, the disclosures of which are incorporated herein by reference.

COPYRIGHT NOTICE

[0002] A portion of the disclosure of this patent document contains material, which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

BACKGROUND

[0003] One of the most important issues impeding the widespread distribution of digital works (i.e. documents or other content in forms readable by computers), via electronic means, and the Internet in particular, is the current lack of ability to enforce the intellectual property rights of content owners during the distribution and use of digital works. Efforts to resolve this problem have been termed "Intellectual Property Rights Management" ("IPRM"), "Digital Property Rights Management" ("DPRM"), "Intellectual Property Management" ("IPM"), "Rights Management" ("RM"), and "Electronic Copyright Management" ("ECM"), collectively referred to as "Digital Rights Management (DRM)" herein.

There are a number of issues to be considered in effecting a DRM System. For example, authentication, authorization, accounting, payment and financial clearing, rights specification, rights verification, rights enforcement, and document protection issues should be addressed. U.S. patents 5,530,235, 5,634,012, 5,715,403, 5,638,443, and 5,629,980, the disclosures of which are incorporated herein by reference, disclose DRM systems addressing these issues.

[0004] Two basic DRM schemes have been employed, secure containers and trusted systems. A "secure container" (or simply an encrypted document) offers a way to keep document contents encrypted until a set of authorization conditions are met and some copyright terms are honored (e.g., payment for use). After the various conditions and terms are verified with the document provider, the document is released to the user in clear form. Commercial products such as CRYPTOLOPES™ and DIGIBOXES™ fall into this category. Clearly, the secure container approach provides a solution to protecting the document during delivery over insecure channels, but does not provide any mechanism to prevent legitimate users from obtaining the clear document and then using and redistributing it in violation of content owners' intellectual property.

[0005] In the "trusted system" approach, the entire system is responsible for preventing unauthorized use and distribution of the document. Building a trusted system usually entails introducing new hardware such as a secure processor, secure storage and secure rendering devices. This also requires that all software applications that run on trusted systems be certified to be trusted. While building tamper-proof trusted systems is a real challenge to existing technologies, current market trends suggest that open and untrusted systems, such as PC's and workstations using browsers to access the Web, will be the dominant systems used to access digital works. In this sense, existing computing environments such as PC's and workstations equipped with popular

operating systems (e.g., Windows™, Linux™, and UNIX) and rendering applications, such as browsers, are not trusted systems and cannot be made trusted without significantly altering their architectures. Of course, alteration of the architecture defeats a primary purpose of the Web, i.e. flexibility and compatibility.

[0006] As an example, U.S. patent 5,634,012, the disclosure of which is incorporated herein by reference, discloses a system for controlling the distribution of digital documents. Each rendering device has a repository associated therewith. A predetermined set of usage transaction steps define a protocol used by the repositories for enforcing usage rights. Usage rights define one or more manners of use of the associated document content and persist with the document content. The usage rights can permit various manners of use such as, viewing only, use once, distribution, and the like. Usage rights can be contingent on payment or other conditions. Further, a party may grant usage rights to others that are a subset of usage rights possessed by the party.

[0007] DRM systems have facilitated distribution of digital content by permitting the content owner to control use of the content. However, known business models for creating, distributing, and using digital content and other items involve a plurality of parties. For example, a content creator may sell content to a publisher who then authorizes a distributor to distribute content to an on-line storefront who then sells content to end-users. Further, the end users may desire to share or further distribute the content. In such a business model, usage rights can be given to each party in accordance with their role in the distribution chain. However, the parties do not have control over downstream parties unless they are privy to any transaction with the downstream parties in some way. For example, once the publisher noted above provides content to the distributor, the publisher cannot readily control rights granted to downstream parties, such as the first or subsequent users unless the

publisher remains a party to the downstream transaction. This loss of control combined with the ever increasing complexity of distribution chains results in a situation, which hinders the distribution of digital content and other items. Further, the publisher may want to prohibit the distributor and/or the storefront from viewing or printing content while allowing an end user receiving a license from the storefront to view and print. Accordingly, the concept of simply granting rights to others that are a subset of possessed rights is not adequate for multi-party, i.e. multi-tier, distribution models.

SUMMARY OF THE INVENTION

[0008] A first aspect of the invention is a method for transferring rights adapted to be associated with items from a rights supplier to a rights consumer. The method comprises obtaining a set of rights associated with an item, said set of rights including meta-rights specifying derivable rights that can be derived therefrom by the rights consumer, and determining whether the rights consumer is entitled to derive the derivable rights specified by the meta-rights, and at least one of deriving the derivable rights, and generating a license including the derived rights with the rights consumer designated as a principal if the rights consumer is entitled to derive the derivable rights specified by the meta-rights.

[0009] A second aspect of the invention is a license associated with an item and adapted to be used within a system for managing the transfer of rights to the item from a rights supplier to a rights consumer. The license comprises a set of rights including meta-rights specifying derivable rights that can be derived therefrom by the rights consumer, a principal designating at least one rights consumer who is authorized to derive the derivable rights, and a mechanism for providing access to the item in accordance with the set of rights.

[0010] A third aspect of the invention is a method for deriving rights adapted to be associated with items from meta-rights. The method comprises obtaining a set of rights associated with an item, said set of rights including meta-rights specifying derivable rights that can be derived therefrom by the rights consumer, and generating a license associated with said item and including the derived rights.

BRIEF DESCRIPTION OF THE DRAWING

[0011] The invention will be described through a preferred embodiment and the attached drawing in which:

[0012] Fig. 1 is a schematic illustration of a rights management system in accordance with the preferred embodiment;

[0013] Fig. 2 is a block diagram of an example distribution chain showing the derivation of rights from meta-rights;

[0014] Fig. 3 is a schematic illustration of a license in accordance with the preferred embodiment;

[0015] Fig. 4 is an example of a license expressed with an XML based rights language in accordance with the preferred embodiment;

[0016] Fig. 5 is a block diagram of the license server of the system of Fig. 1;

[0017] Fig. 6 is a block diagram of a rights label in accordance with the preferred embodiment; and

[0018] Fig. 7 is a flow chart of the procedure for transferring and deriving rights in accordance with the preferred embodiment.

DETAILED DESCRIPTION

[0019] A DRM system can be utilized to specify and enforce usage rights for specific content, services, or other items. Fig. 1 illustrates DRM System 10 that can be used in connection with the preferred embodiment. DRM System 10 includes a user activation component, in the form of activation server 20, that issues public and private key pairs to content users in a protected fashion, as is well known. During an activation process, some information is exchanged between activation server 20 and client environment 30, a computer or other device associated with a content recipient, and client component 60 is downloaded and installed in client environment 30. Client component 60 preferably is tamper resistant and contains the set of public and private keys issued by activation server 20 as well as other components, such as any component necessary for rendering content 42.

[0020] Rights label 40 is associated with content 42 and specifies usage rights and possibly corresponding conditions that can be selected by a content recipient. License Server 50 manages the encryption keys and issues licenses for protected content. These licenses embody the actual granting of usage rights to an end user. For example, rights label 40 may include usage rights permitting a recipient to view content for a fee of five dollars and view and print content for a fee of ten dollars. License 52 can be issued for the view right when the five dollar fee has been paid, for example. Client component 60 interprets and enforces the rights that have been specified in license 52.

[0021] Fig. 6 illustrates rights label 40 in accordance with the preferred embodiment. Rights label 40 includes plural rights offers 44 each including usage rights 44a, conditions 44b, and content specification 44c. Content specification 44c can include any mechanism for calling, referencing, locating, linking or otherwise specifying content 42 associated with offer 44. Clear (unprotected) content can be prepared with document preparation application 72 installed on computer 70 associated with a content publisher, a content distributor, a content service provider, or any

other party. Preparation of content consists of specifying the rights and conditions under which content 42 can be used, associating rights label 40 with content 42 and protecting content 42 with some crypto algorithm. A rights language such as XrML™ can be used to specify the rights and conditions. However, the rights can be specified in any manner. Also, the rights can be in the form of a pre-defined specification or template that is merely associated with the content. Accordingly, the process of specifying rights refers to any process for associating rights with content. Rights label 40 associated with content 42 and the encryption key used to encrypt the content can be transmitted to license server 50. As discussed in detail below, rights 44a can include usage rights, which specify a manner of use, and meta-rights, which permit other rights to be derived.

[0022] In some case, license 52 includes conditions that must be satisfied in order to exercise a specified right. For, example a condition may be the payment of a fee, submission of personal data, or any other requirement desired before permitting exercise of a manner of use. Conditions can also be "access conditions" for example, access conditions can apply to a particular group of users, say students in a university, or members of a book club. In other words, the condition is that the user is a particular person or member of a particular group. Rights and conditions can exist as separate entities or can be combined.

[0023] Labels, offers, usage rights, and conditions can be stored together with content 42 or otherwise associated with content 42 through content specification 44c or any other mechanism. A rights language such as XrML™ can be used to specify the rights and conditions. However, the rights can be specified in any manner. Also, the rights can be in the form of a pre-defined specification or template that is merely associated with content 42.

[0024] A typical workflow for DRM system 10 is described below. A recipient operating within client environment 30 is activated for receiving

content 42 by activation server 20. This results in a public-private key pair (and possibly some user/machine specific information) being downloaded to client environment 30 in the form of client software component 60 in a known manner. This activation process can be accomplished at any time prior to the issuing of a license.

[0025] When a recipient wishes to obtain specific content 42, the recipient makes a request for content 42. For example, a user, as a recipient, might browse a Web site running on Web server 80, using a browser installed in client environment 30, and request content 42. During this process, the user may go through a series of steps possibly including a fee transaction (as in the sale of content) or other transactions (such as collection of information). When the appropriate conditions and other prerequisites, such as the collection of a fee and verification that the user has been activated, are satisfied, Web server 80 contacts license server 50 through a secure communications channel, such as a channel using a Secure Sockets Layer (SSL). License server 50 then generates license 52 for content 42 and Web server 80 causes both the content and license 52 to be downloaded. License 52 includes the appropriate rights, such as usage rights and/or meta-rights, and can be downloaded from license server 50 or an associated device. Content 42 can be downloaded from computer 70 associated with a vendor, distributor, or other party.

[0026] Client component 60 in client environment 30 will then proceed to interpret license 52 and allow use of content 42 based on the usage rights and conditions specified in license 52. The interpretation and enforcement of usage rights are well known generally and described in the patents referenced above, for example. The steps described above may take place sequentially or approximately simultaneously or in various orders.

[0027] DRM system 10 addresses security aspects of content 42. In particular, DRM system 10 may authenticate license 52 that has been

issued by license server 50. One way to accomplish such authentication is for application 60 to determine if license 52 can be trusted. In other words, application 60 has the capability to verify and validate the cryptographic signature, or other identifying characteristic of license 52. Of course, the example above is merely one way to effect a DRM system. For example, license 52 and content 42 can be distributed from different entities. Clearinghouse 90 can be used to process payment transactions and verify payment prior to issuing a license.

[0028] As noted above, typical business models for distributing digital content include plural parties, such as owners, publishers, distributors, and users. Each of these parties can act as a supplier granting rights to a consumer downstream in the distribution channel. The preferred embodiment extends the known concepts of usage rights, such as the usage rights and related systems disclosed in U.S. patents 5,629,980, 5,634,012, 5,638,443, 5,715,403 and 5,630,235, to incorporate the concept of "meta-rights." Meta-rights are the rights that one has to generate, manipulate, modify, dispose of or otherwise derive other rights. Meta-rights can be thought of as usage rights to usage rights (or other meta-rights). This concept will become clear based on the description below.

[0029] Meta-rights can include derivable rights to offer rights, grant rights, negotiate rights, obtain rights, transfer rights, delegate rights, expose rights, archive rights, compile rights, track rights, surrender rights, exchange rights, and revoke rights to/from others. Meta-rights can include the rights to modify any of the conditions associated with other rights. For example, a meta-right may be the right to extend or reduce the scope of a particular right. A meta-right may also be the right to extend or reduce the validation period of a right. Meta-rights can be hierarchical and can be structured as objects within objects. For example, a distributor may have a meta-right permitting the distributor to grant a meta-right to a retailer which permits the retailer to grant users rights to view content. Just as

rights can have conditions, meta-rights can also have conditions. Meta-rights can also be associated with other meta-rights.

[0030] The concept of meta-rights can be particularly useful because distribution models may include entities that are not creators or owners of digital content, but are in the business of manipulating the rights associated with the content. For example, as noted above, in a multi-tier content distribution model, intermediate entities (e.g., distributors) typically will not create or use the content but will be given the right to issue rights for the content they distribute. In other words, the distributor or reseller will need to obtain rights (meta-rights) to issue rights. For the sake of clarity, the party granting usage rights or meta-rights is referred to as "supplier" and the party receiving and/or exercising such rights is referred to as "consumer" herein. It will become clear that any party can be a supplier or a consumer depending on their relationship with the adjacent party in the distribution chain. Note that a consumer "consumes" , i.e. exercises, rights and does not necessarily consume, i.e. use, the associated content.

[0031] Fig. 2 schematically illustrates an example of a multi-tier distribution model 200. Publisher 210 publishes content for distribution, by distributor 220 for example. Distributor 220 distributes content to retailers, such as retailer 230 and retailer 230 sells content to users, such as user 240. In model 200, publisher 210 could negotiate business relationships with distributor 220 and distributor 220 could negotiate business relationships with retailer 230. Also, retailer 230 may desire usage rights that are beyond usage rights granted to distributor 220. However, keep in mind that, in a distribution chain that utilizes a DRM system to control use and distribution of content or other items, content can travel from publisher 210 to user 240 through any digital communication channel, such a network or transfer of physical media. When user 240 wishes to use content, a license is obtained, in the manner described above for example. Accordingly, the negotiated relationships can become difficult, if not impossible, to manage.

[0032] In model 200 of Fig. 2, retailer 230 will only grant rights to user 240 that have been predetermined and authorized by the distributor 220, publisher 210 and potentially other parties upstream of the transaction, such as the content creator or owner. The rights are predetermined through, and derived from, meta-rights granted to retailer 230 by distributor 220. Of course, there can be any number of parties in the distribution chain. For example, distributor 220 may sell directly to the public in which case retailer 230 is not necessary. Also, there may be additional parties. For example user 240 can distribute to other users.

[0033] In model 200 publisher grants to distributor 220 usage rights 212 permitting distribution of content, and meta-rights 214. Meta-rights 214 permit distributor 220 to grant to retailer 230 the usage right 214' (derived from meta-rights 214) to distribute or possibly sell content and meta-rights 216 which permit retailer 230 to grant user 240 the right to use content. For example, publisher 210 may specify, through meta-rights 214, that meta-right 216 granted to retailer 230 permits retailer 230 to grant only 500 licenses and usage rights 216' that retailer 230 can grant to a user can only be "view" and "print-once". In other words, distributor 220 has granted meta-rights to retailer 230. Similarly, publisher 210 issues meta-rights 214 to the distributor that will govern what type, and how many , rights distributor 220 can grant to retailer 230. Note that these entities could be divisions, units or persons that are part of a larger enterprise, which also has other roles. For example, an enterprise might create, distribute, and sell content and carry out those activities using different personnel or different business units within the enterprise. The principles of meta-rights can be applied to an enterprise to determine content usage within that enterprise. Also, retailer 230 could grant meta-rights 218 to user 240 permitting user 240 to share rights or grant usage rights to achieve a super-distribution model. It can be seen that meta-rights of a party are derived from meta-rights granted by an upstream party in the distribution chain.

[0034] For example, a person's medical records can be in digital form managed by a first hospital as publisher 230. In this scenario, the person, as supplier, grants usage rights to the hospital, as consumer, to access and update the medical records. Should that person require treatment at a second hospital and desires to transfer their records to the second hospital, the person can grant to the first hospital the right to transfer the access rights to the new hospital through meta-rights. In other words, the person has specified meta-rights and granted the meta-rights to the first hospital. The meta-rights permit the first hospital to grant rights, as a supplier, to the second hospital, as a consumer. In another example, a person's last will and testament can be in digital form and managed by a law firm as publisher 210. If the person wishes to allow a third party to review the will. The person can grant meta-rights to the law firm permitting the law firm to grant access rights to this third party.

[0035] At a high level the process of enforcing and exercising meta-rights are the same as for usage rights. However, the difference between usage rights and meta-rights are the result from exercising the rights. When exercising usage rights, actions to content result. For example usage rights can be for viewing, printing, or copying digital content. When meta-rights are exercised, new rights are created from the meta-rights or existing rights are disposed as the result of exercising the meta-rights. The recipient of the new rights may be the same principal (same person, entity, or machine, etc), who exercises the meta-rights. Alternatively, the recipient of meta-rights can be a new principal. The principals who receive the derived rights may be authenticated and authorized before receiving/storing the derived rights. Thus, the mechanism for exercising and enforcing a meta-right can be the same as that for a usage right. For example, the mechanism disclosed in U.S. Patent 5,634,012 can be used.

[0036] Meta-rights can be expressed by use of a grammar or rights language including data structures, symbols, elements, or sets of rules. For example, the XrML™ rights language can be used. As illustrated in

Fig. 3, the structure of license 52 can consist of one or more grants 300 and one or more digital signatures 310. Each grant 300 includes specific granted meta-rights 302 such as rights to offer usage rights, grant usage rights, obtain usage rights, transfer usage rights, exchange usage rights, transport usage rights, surrender usage rights, revoke usage rights, reuse usage rights, or management meta-rights such as the rights to backup rights, restore rights, recover rights, reissue rights, or escrow the rights for management of meta-rights and the like.

[0037] Grant 300 can also specify one or more principals 304 to whom the specified meta-rights are granted. Also grants 300 can include conditions 306 and state variables 308. Like usage rights, access and exercise of the granted meta-rights are controlled by any related conditions 306 and state variables 308. The integrity of license 52 is ensured by the use of digital signature 310, or another identification mechanism. Signature 310 can include a crypto-algorithm, a key, or another mechanism for providing access to content 42 in a known manner. The structure of digital signature 310 includes the signature itself, the method of how the code is computed, the key information needed to verify the code and issuer identification.

[0038] State variables track potentially dynamic states conditions. State variables are variables having values that represent status of rights, or other dynamic conditions. State variables can be tracked, by clearinghouse 90 or another device, based on identification mechanisms in license 52. Further, the value of state variables can be used in a condition. For example, a usage right can be the right to print content 42 for and a condition can be that the usage right can be exercised three times. Each time the usage right is exercised, the value of the state variable is incremented. In this example, when the value of the state variable is three, the condition is no longer satisfied and content 42 cannot be printed. Another example of a state variable is time. A condition of license 52 may require that content 42 is printed within thirty days. A state

variable can be used to track the expiration of thirty days. Further, the state of a right can be tracked as a collection of state variables. The collection of the change is the state of a usage right represents the usage history of that right.

[0039] Fig. 4 is an example of license 52 encoded in XrML™. The provider grants the distributor a meta right to issue a usage right (i.e., play) to the content (i.e., a book) to any end user. With this meta right, the distributor may issue the right to play the book within the U.S. region and subject to some additional conditions that the distributor may impose upon the user, as long as the distributor pays $1 to the provider each time the distributor issues a license for an end user. The XrML™ specification is published and thus well known.

[0040] Fig. 5 illustrates the primary modules of license server 50 in accordance with the preferred embodiment. License interpreter module 502 validates and interprets license 52 and also provides the functions to query any or all fields in the license such as meta–rights 302, conditions 306, state variables 308, principle 304, and/or digital signature 310. License manager module 503 manages all license repositories for storing licenses 52, and also provides functions to create licenses 52 for derived rights, verify licenses, store licenses, retrieve licenses and transfer licenses. State of rights module 504 manages the state and history of rights and meta-rights. The current value and history of the state variables together with the conditions controls the permission to exercise given meta-rights for a given authenticated principal. Condition validator 506 verifies conditions associated with the meta-rights. Together with the state variables, conditions associated with meta-rights define variables whose values may change over the lifetime of the meta-rights. Values of state variables used in conditions can affect the meta-rights at the time and during the time the rights are exercised.

[0041] Authorization module 508 authorizes the request to exercise meta-rights and to store the newly created rights or derived rights as the result of exercising the meta-rights. Authorization module 508 accesses both state of rights manager module 504 and condition validator module 506. Authorization module 508 interacts with license manager module 503 and the list of state variables and conditions and then passes the state variables to state of rights manager module 504 and condition list to condition validator module 506 for authorization.

[0042] A request for exercising a meta-right is passed to meta-rights manager module 510. Assuming that the requesting device has been authenticated, meta-rights manager module 510 requests the license manager module 504 to verify the license for exercising the requested meta-rights. License manager module 504 verifies the digital signature of the license and the key of the signer. If the key of the signer is trusted and the digital signature is verified then license manager module 504 returns "verified" to the meta-rights manager module 510. Otherwise "not verified" is returned.

[0043] Authorization module 508 instructs license manager 503 to fetch state variable 308 and conditions 306 of license 52. Authorization manager 508 then determines which state variables are required to enforce to enforce license 52. State of rights manager 504 then supplies the current value of each required state variable to authorization module 508. Authorization module 508 then passes conditions 306 and the required state variables to condition validator 506. If all conditions 306 are satisfied, authorization module 508 returns "authorized" to meta-rights manager module 510.

[0044] Meta-rights manager module 510 verifies license 52 and meta-rights 302 therein, to authorize the request to exercise meta-rights 302, to derive new rights from meta-rights 302, and to update the state of rights and the current value of the conditions. Rights manager module 512, on

the other hand, manages the new rights created or the derived rights as the result of exercising the meta-rights. Rights manager module 512 uses authorization module 508 to verify that recipient of the newly created rights or derived rights is intended principal 304. If the recipient are authorized then the rights manager module 512 directs license manager 504 to store the newly created rights in a repository associated with the consumer. This is discussed in greater detail below with reference to Fig. 7.

[0045] The authorization process is not limited to the sequence or steps described above. For example, a system could be programmed to allow authorization module 508 to request the state conditions from license manager 504 prior to verification of the digital signature. In such a case it would be possible to proceed subject to a verified license. Further, the various modules need not reside in the license server or related devices. The modules can be effected through hardware and/or software in any part of the system and can be combined or segregated in any manner.

[0046] Once a request to exercise a meta-rights has been authorized, the meta-right can be exercised. Meta-rights manager module 510 informs state of rights module 504 that it has started exercising the requested meta-rights. State of rights module 504 then records the usage history and changes its current value of the state variables. Meta-rights manager module 510 exercises the requested meta-rights in a manner similar to known procedures for usage rights. If new rights are derived, then meta-rights manager module 510 invokes license manager module 504 to create new rights as the result of exercising the target meta-rights. Each new right is then sent to the corresponding rights manager module 512 of the consumer and stored in a repository associated with the consumer. Rights manager module 512 of the consumer will authenticate and authorize the consumer before receiving and storing the newly created right. New rights can be derived from meta-rights in accordance with a set of rules or other logic. For example, one rule can dictate that a consumed right to offer a license for use will result in the consumer having

the right to offer a usage right and grant a license to that usage right to another consumer.

[0047] Fig. 7 illustrates the workflow for transferring meta-rights and deriving new rights from the meta-rights in accordance with the preferred embodiment. All steps on the left side of Fig. 7 relate to the supplier of rights and all steps on the right side of Fig. 7 relate to the consumer of rights. In step 702, principal 304 of license 52 is authenticated in a known manner. In other words, it is determined if the party exercising meta-right 302 has the appropriate license to do so. If the principal is not authorized, the procedure terminates in step 704. If the principal is authorized, the procedures advances to step 706 in which meta right 302 is exercised and transmitted to the consumer in the form of license 52 having derived rights in the manner set forth above. In step 708 the principal of this new license is authenticated. In other words, it is determined if the party exercising the derived rights has the appropriate license to do so. If the principal is not authorized, the procedure terminates in step 710. If the principal is authorized, the procedures advances to step 712 in which the derived right is stored. The procedure then returns to step 708 for each additional right in the license and terminates in step 714 when all rights have been processed.

[0048] The preferred embodiments is not limited to situations where resellers, distributors or other "middlemen" are used. For example, the preferred embodiment can be applied within enterprises or other organizations, which create and/or distribute digital content or other items to control use of the content within the enterprise or other organization. Meta-rights can also be issued to end-users, when the grant of a right relates to another right. For example, the right to buy or sell securities as it is in the case of trading options and futures. Meta-rights can be assigned or associated with goods services, resources, or other items.

[0049] The invention can be implemented through any type of devices, such as computers and computer systems. The preferred embodiment is implemented in a client server environment. However, the invention can be implemented on a single computer or other device. Over a network using dumb terminals, thin clients, or the like, or through any configuration of devices. The various modules of the preferred embodiment have been segregated and described by function for clarity. However, the various functions can be accomplished in any manner through hardware and/or software. The various modules and components of the preferred embodiment have separate utility and can exist as distinct entities. Various communication channels can be used with the invention. For example, the Internet or other network can be used. Also, data can be transferred by moving media, such as a CD, DVD, memory stick or the like, between devices. Devices can include, personal computers, workstations, thin clients, PDA's and the like.

[0050] The invention has been described through a preferred embodiment. However, various modifications can be made without departing from the scope of the invention as defined by the appended claims and legal equivalents.

What is claimed:

1. A method for transferring rights adapted to be associated with items from a rights supplier to a rights consumer, said method comprising:

obtaining a set of rights associated with an item, said set of rights including meta-rights specifying derivable rights that can be derived therefrom by the rights consumer; and

determining whether the rights consumer is entitled to derive the derivable rights specified by the meta-rights, and at least one of deriving the derivable rights, and generating a license including the derived rights with the rights consumer designated as a principal if the rights consumer is entitled to derive the derivable rights specified by the meta-rights.

2. A method recited in claim 1, further comprising, transmitting the set of rights, in the form of a license to the item, from the rights supplier to the rights consumer.

3. A method as recited in claim 1, wherein the derived rights are rights disposal rights.

4. A method as recited in claim 1, wherein the items are content.

5. A method as recited in claim 1, wherein the derived rights include usage rights.

6. A method as recited in claim 1, wherein the derived rights include meta-rights that the rights consumer may transfer to another rights consumer in the form of a license.

7. A method as recited in claim 4, wherein the consumer is a content distributor.

8. A method as recited in claim 4, wherein the consumer is a content retailer.

9. A method as recited in claim 4, wherein the consumer is a content publisher.

10. A license associated with an item and adapted to be used within a system for managing the transfer of rights to the item from a rights supplier to a rights consumer, said license comprising:

a set of rights including meta-rights specifying derivable rights that can be derived therefrom by the rights consumer;

a principal designating at least one rights consumer who is authorized to derive the derivable rights; and

a mechanism for providing access to the item in accordance with the set of rights.

11. A license as recited in claim 10, wherein the derived rights are rights disposal rights.

12. A license as recited in claim 10, wherein the items are content.

13. A license as recited in claim 10, wherein the derived rights include usage rights.

14. A license as recited in claim 10, wherein the derived rights include meta-rights that the rights consumer may transfer to another rights consumer in the form of a license.

15. A license as recited in claim 12, wherein the consumer is a content distributor.

16. A license as recited in claim 12, wherein the consumer is a content retailer.

17. A license as recited in claim 12, wherein the consumer is a content publisher.

18. A license as recited in claim 10, further comprising a digital signature corresponding to a party issuing the license.

19. A license as recited in claim 10, further comprising at least one condition that must be satisfied to exercise at least on of said meta-rights.

20. A license as recited in claim 19, further comprising at least one state variable related to said at least one condition.

21. A method for deriving rights adapted to be associated with items from meta-rights, said method comprising:

obtaining a set of rights associated with an item, said set of rights including meta-rights specifying derivable rights that can be derived therefrom by the rights consumer; and

generating a license associated with said item and including the derived rights.

22. A method as recited in claim 21, wherein the derived rights are rights disposal rights.

23. A method as recited in claim 21, wherein the items are content.

24. A method as recited in claim 21, wherein the derived rights include usage rights.

10/162701

ABSTRACT

[0051] A method and apparatus for managing the transfer of rights associated with items from a rights supplier to a rights consumer. A set of rights is associated with an item and includes meta-rights specifying derivable rights that can be derived therefrom by the rights consumer. The set of rights is transferred, in the form of a license to the item, from the rights supplier to the rights consumer. If it is determined that the rights consumer is entitled to derive the derivable rights specified by the meta-rights, the derivable rights are derived and a license including the derived rights is generated with the rights consumer designated as a principal.

Fig. 1

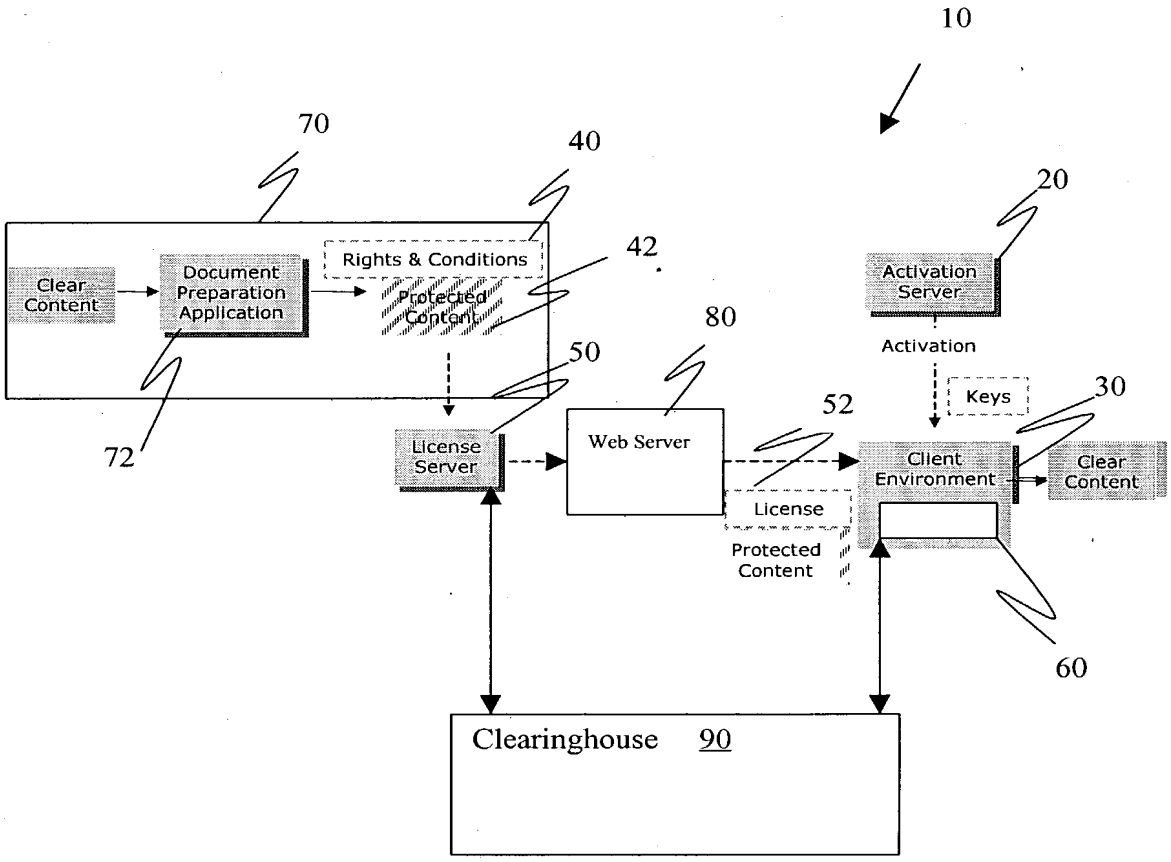


Fig. 2

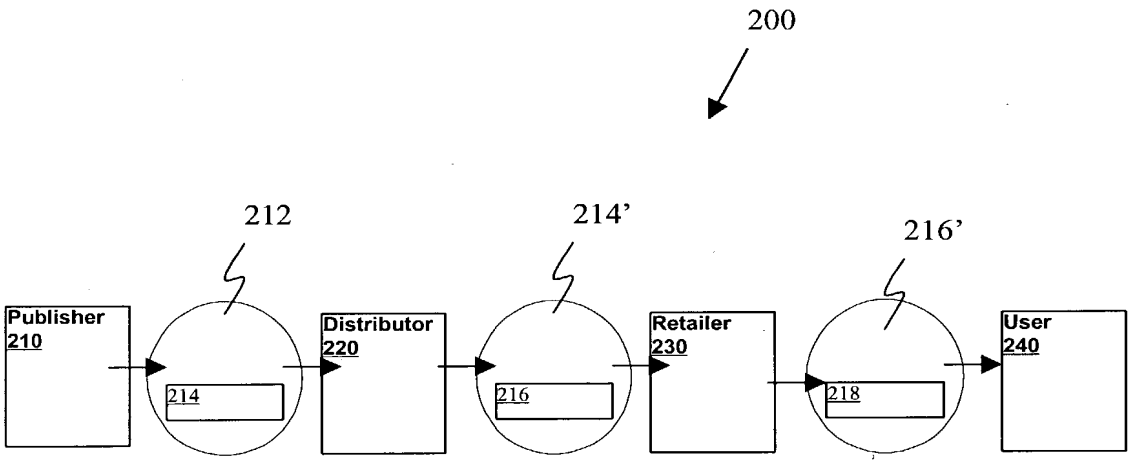


Fig. 3

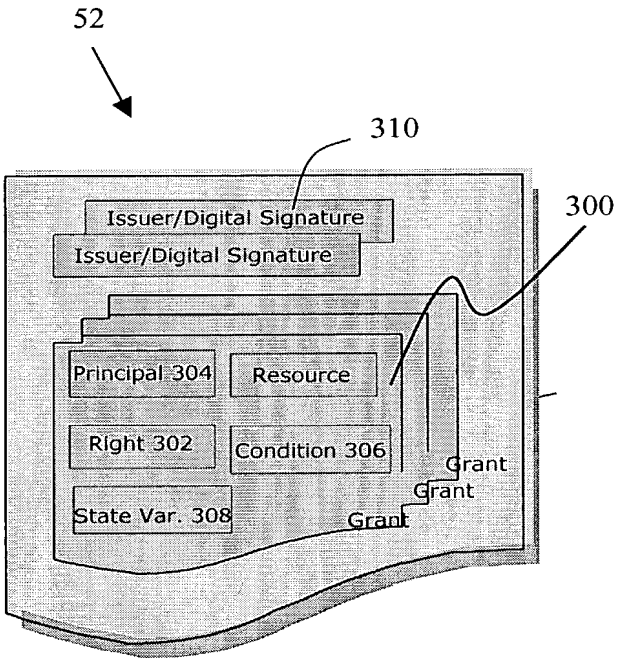


Fig. 4

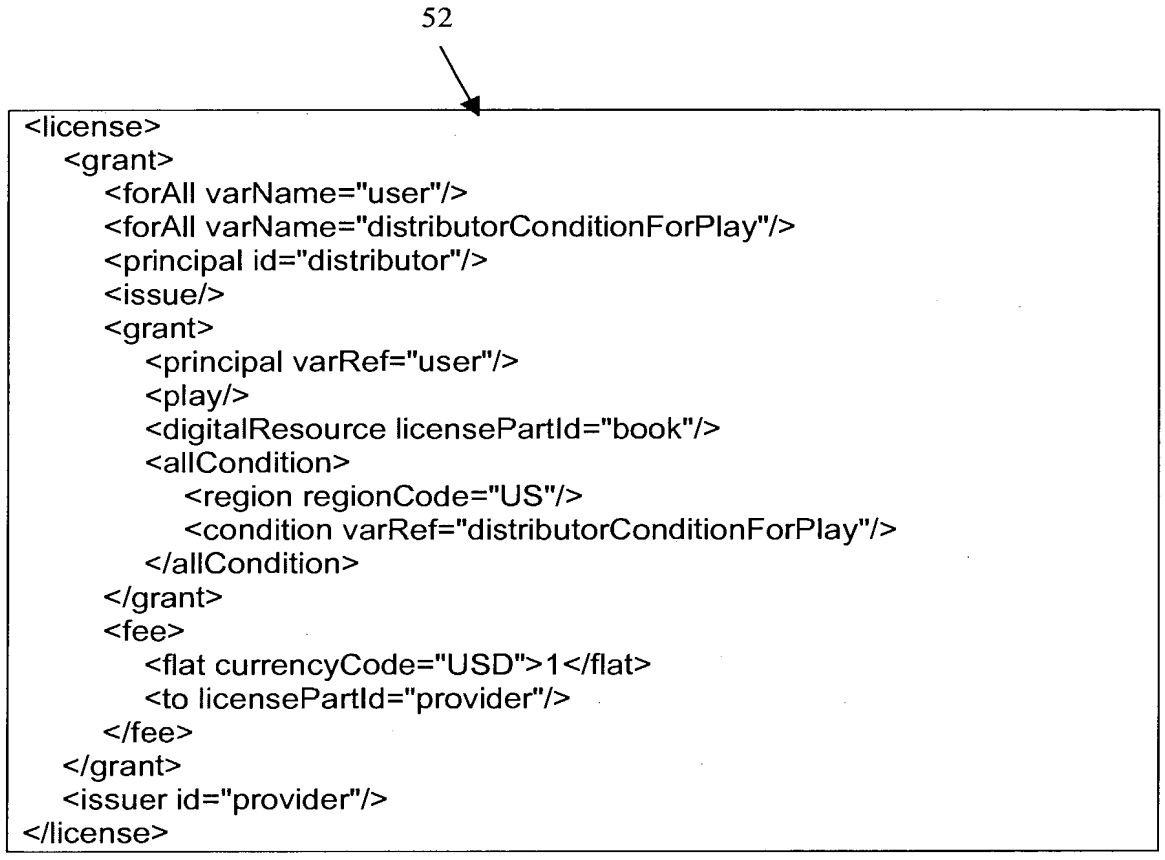


Fig. 5

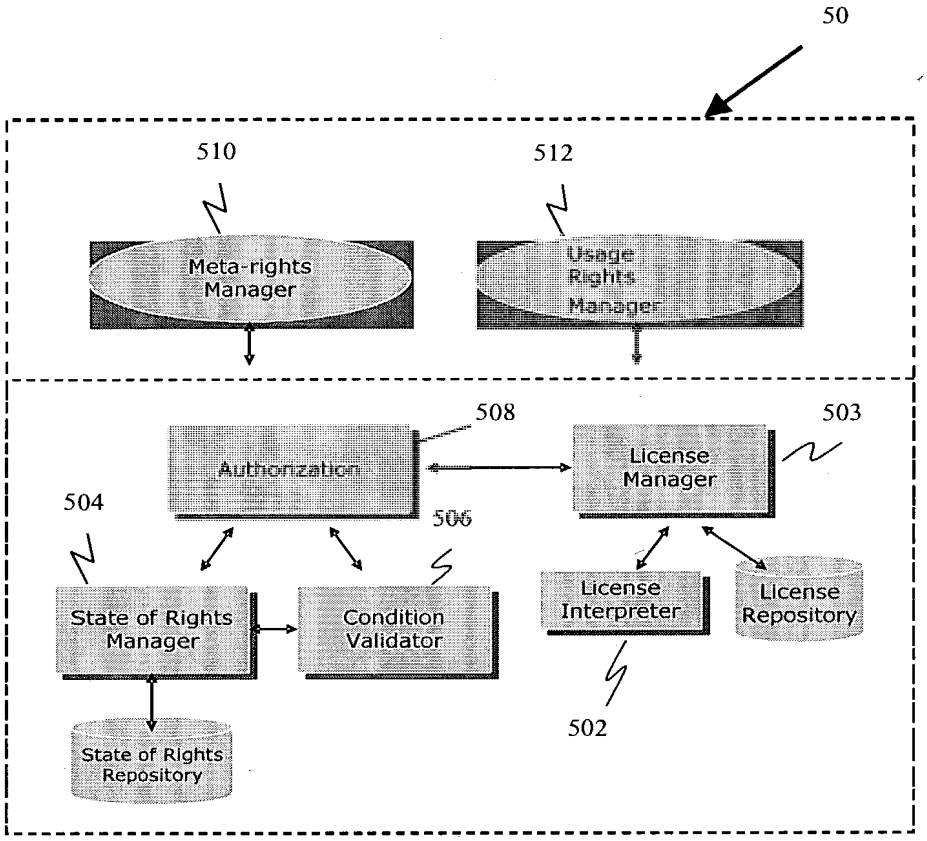


Fig. 6

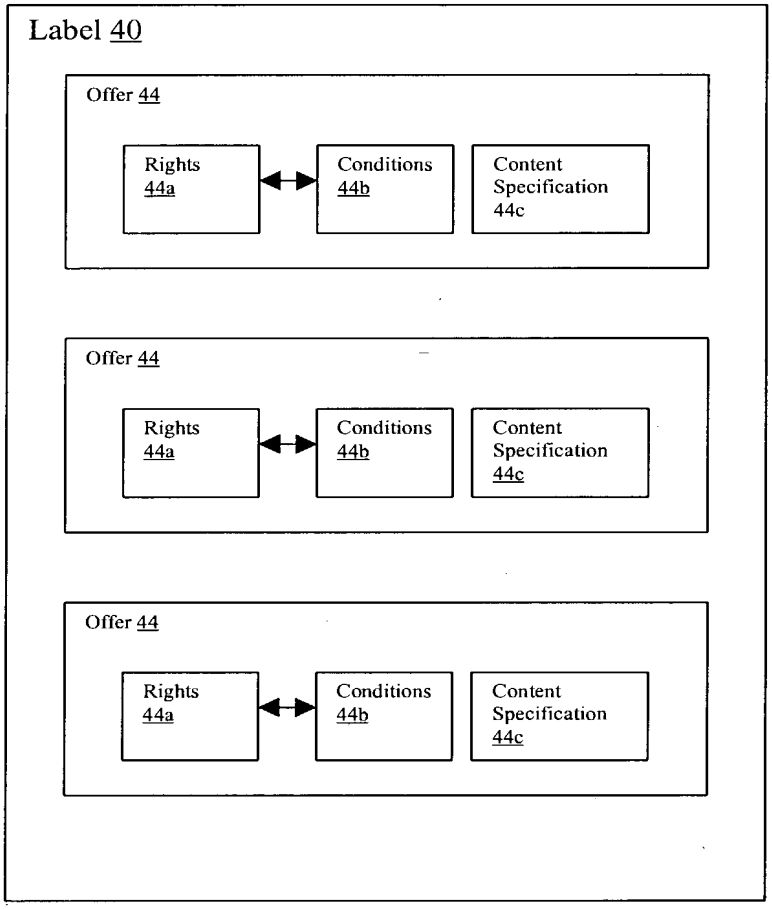
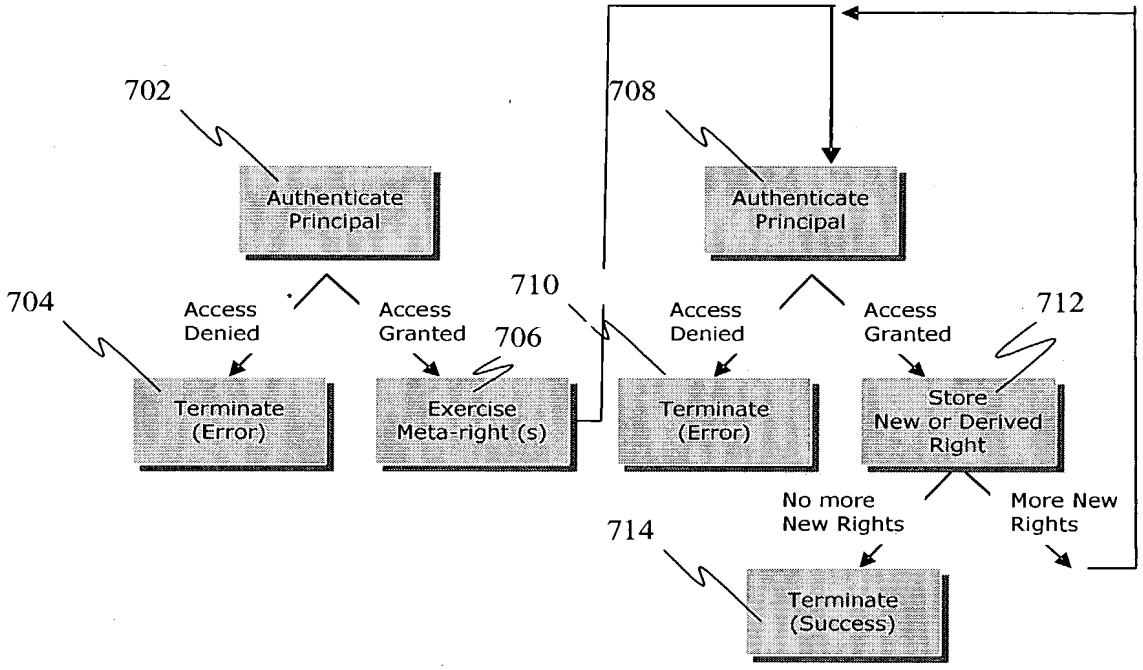


Fig. 7



SCANNED, #18

United States Patent & Trademark Office

Office of Initial Patent Examination -- Scanning Division

Application deficiencies found during scanning:

☐ Page(s)_____ of _____ were not present for scanning. (Document title)

☐ Page(s)_____ of _____ were not present for scanning. (Document title)

■ ***Scanned copy is best available.*** Some drawings are dark

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

APPLICATION NUMBER	FILING/RECEIPT DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NUMBER
10/162,701	06/06/2002	Xin Wang	111325-113

CONFIRMATION NO. 6475

22204
NIXON PEABODY, LLP
8180 GREENSBORO DRIVE
SUITE 800
MCLEAN, VA 22102

FORMALITIES LETTER

OC000000008490560

Date Mailed: 07/22/2002

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

Items Required To Avoid Abandonment:

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items and pay any fees required below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The statutory basic filing fee is missing.
  Applicant must submit $ **740** *to complete the basic filing fee for a non-small entity. If appropriate, applicant may make a written assertion of entitlement to small entity status and pay the small entity filing fee (37 CFR 1.27).*
- The oath or declaration is missing.
  A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.
- To avoid abandonment, a late filing fee or oath or declaration surcharge as set forth in 37 CFR 1.16(l) of $130 for a non-small entity, must be submitted with the missing items identified in this letter.

Items Required To Avoid Processing Delays:

The item(s) indicated below are also required and should be submitted with any reply to this notice to avoid further processing delays.

- Additional claim fees of **$72** as a non-small entity, including any required multiple dependent claim fee, are required. Applicant must submit the additional claim fees or cancel the additional claims for which fees are due.

SUMMARY OF FEES DUE:

Total additional fee(s) required for this application is **$942** for a Large Entity

- **$740** Statutory basic filing fee.
- **$130** Late oath or declaration Surcharge.
- Total additional claim fee(s) for this application is **$72**
  - **$72** for **4** total claims over 20.

A copy of this notice <u>MUST</u> be returned with the reply.

Customer Service Center
Initial Patent Examination Division (703) 308-1202

PART 3 - OFFICE COPY

0360

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: Xin WANG	Examiner: Unassigned
Serial No. 10/162,701	Group Art Unit:
Filed: 06/06/2002	
For: METHOD AND APPARATUS MANAGING THE TRANSFER OF RIGHTS	

INFORMATION DISCLOSURE STATEMENT

Commissioner of Patents
Washington, D.C. 20231

Sir:

In accordance with the duty of disclosure as set forth in 37 C.F.R. §1.56, Applicants hereby submit the following information in conformance with 37 C.F.R. §§ 1.97 and 1.98. Pursuant to 37 C.F.R. § 1.98, a copy of each of the documents cited is enclosed.

The documents are being submitted within three (3) months of the filing of this application or entry into the national stage of this application, or before the first Office Action on the merits, whichever is later, therefore no fee or certification is required under 37 C.F.R § 1.97(b).

It is requested that the accompanying information disclosure statement be considered and made of record in the above-captioned application. To assist the Examiner, the documents are listed on the attached form PTO-1449. It is respectfully requested that an Examiner initialed copy of this form be returned to the undersigned.

The Commissioner is hereby authorized to charge any fees connected with this filing which may be required now, or credit any overpayment to Deposit Account No. 19-2380.

Respectfully submitted,

By: Marc S. Kaufman, Esq.
Registration No. 35,212

NIXON PEABODY LLP
8180 Greensboro Drive, Suite 800
McLean, Virginia 22102
Telephone: (703) 770-9300

Form PTO-1449 (Rev. 8-83)	U.S. Department of Commerce Patent and Trademark Office	Atty Docket 111325-113	Serial No. 10/162,701
INFORMATION DISCLOSURE STATEMENT		Applicants: Xin WANG	
		Filing Date: June 06, 2002	Group Art Unit:

U.S. PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date (if appropriate)
	4,713,753	12/15/1987	Beobert et al.			
	5,052,040	09/24/1991	Preston et al.			
	5,301,231	04/05/1994	Abraham et al.			
	5,502,766	03/26/1996	Boebert et al.			
	5,649,013	07/15/1997	Stuckey et al.			
	5,737,413	04/07/1998	Akiyama et al.			
	5,737,416	04/07/1998	Cooper et al.			
	5,757,907	05/26/1998	Cooper et al.			
	6,253,193	06/26/2001	Ginter et al.			
	6,301,660	10/09/2001	Benson			
	6,327,652	12/04/2001	England et al.			
	6,330,670	12/11/2001	England et al.			

Examiner	Date Considered

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this formwith next communication to applicant.

2135
10-1-02

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: Xin WANG	)	Examiner: Unassigned
Serial No. 10/162,701	)	Group Art Unit:
Filed: 06/06/2002	)	
For: METHOD AND APPARATUS MANAGING THE TRANSFER OF RIGHTS	)	

INFORMATION DISCLOSURE STATEMENT

Commissioner of Patents
Washington, D.C. 20231

Sir:

In accordance with the duty of disclosure as set forth in 37 C.F.R. §1.56, Applicants hereby submit the following information in conformance with 37 C.F.R. §§ 1.97 and 1.98. Pursuant to 37 C.F.R. § 1.98, a copy of each of the documents cited is enclosed.

The documents are being submitted within three (3) months of the filing of this application or entry into the national stage of this application, or before the first Office Action on the merits, whichever is later, therefore no fee or certification is required under 37 C.F.R § 1.97(b).

The submitted documents are patents issued to a company known to be developing related technology.

It is requested that the accompanying information disclosure statement be considered and made of record in the above-captioned application. To assist the Examiner, the documents are listed on the attached form PTO-1449. It is respectfully requested that an Examiner initialed copy of this form be returned to the undersigned.

The Commissioner is hereby authorized to charge any fees connected with this filing which may be required now, or credit any overpayment to Deposit Account No. 19-2380.

Respectfully submitted,

By: Marc S. Kaufman, Esq.
Registration No. 35,212

NIXON PEABODY LLP
8180 Greensboro Drive, Suite 800
McLean, Virginia 22102
Telephone: (703) 770-9300

Form PTO-1449 (Rev. 8-83)

U.S. Department of Commerce Patent and Trademark Office

INFORMATION DISCLOSURE STATEMENT

Atty Docket 1113[redacted]13	**Serial No.** 10/162,701
Applicants: Xin WANG	
Filing Date: June 06, 2002	**Group Art Unit:**

U.S. PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date (if appropriate)
	4,827,508	05/02/1989	Shear			
	4,977,594	12/11/1990	Shear			
	5,050,213	09/17/1991	Shear			
	5,410,598	04/25/1995	Shear			
	5,892,900	04/06/1999	Ginter et al.			
	5,910,987	06/08/1999	Ginter et al.			
	5,915,019	06/22/1999	Ginter et al.			
	5,917,912	06/29/1999	Ginter et al.			
	5,920,861	07/06/1999	Hall et al.			
	5,940,504	08/17/1999	Griswold			
	5,943,422	08/24/1999	Van Wie et al.			
	5,949,876	09/07/1999	Ginter et al.			
	5,982,891	11/09/1999	Ginter et al.			
	5,999,949	12/07/1999	Crandall			
	6,112,181	08/29/2000	Shear et al.			
	6,138,119	10/24/2000	Hall et al.			
	6,157,721	12/05/2000	Shear et al.			
	6,185,683	02/06/2001	Ginter et al.			
	6,237,786	05/29/2001	Ginter et al.			
	6,240,185	05/29/2001	Van Wie et al.			
	6,253,193	06/26/2001	Ginter et al.			
	6,292,569	09/18/2001	Shear et al.			
	6,363,488	05/26/2002	Ginter et al.			
	6,389,402	05/14/2002	Ginter et al.			

Examiner	Date Considered

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this formwith next communication to applicant.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: Xin WANG	Examiner: Unassigned
Serial No. 10/162,701	Group Art Unit:
Filed: 06/06/2002	
For: METHOD AND APPARATUS MANAGING THE TRANSFER OF RIGHTS	

INFORMATION DISCLOSURE STATEMENT

Commissioner of Patents
Washington, D.C. 20231

Sir:

In accordance with the duty of disclosure as set forth in 37 C.F.R. §1.56, Applicants hereby submit the following information in conformance with 37 C.F.R. §§ 1.97 and 1.98. Pursuant to 37 C.F.R. § 1.98, a copy of each of the documents cited is enclosed.

The documents are being submitted within three (3) months of the filing of this application or entry into the national stage of this application, or before the first Office Action on the merits, whichever is later, therefore no fee or certification is required under 37 C.F.R § 1.97(b).

It is requested that the accompanying information disclosure statement be considered and made of record in the above-captioned application. To assist the Examiner, the documents are listed on the attached form PTO-1449. It is respectfully requested that an Examiner initialed copy of this form be returned to the undersigned.

The Commissioner is hereby authorized to charge any fees connected with this filing which may be required now, or credit any overpayment to Deposit Account No. 19-2380.

Respectfully submitted,

By: Marc S. Kaufman
Registration No. 35,212

NIXON PEABODY LLP
8180 Greensboro Drive, Suite 800
McLean, Virginia 22102
Telephone: (703) 770-9300

Form PTO-1449 (Rev. 8-83)	Department of Commerce Patent and Trademark Office	Atty Docket 11132[redacted]3	Serial No. 10/162,701
INFORMATION DISCLOSURE STATEMENT		Applicants: Xin WANG	
		Filing Date: June 06, 2002	Group Art Unit:

U.S. PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date (if appropriate)
	3,263,158	07/01/1966	Janis			
	3,609,697	09/28/1971	Blevins et al.			
	3,790,700	02/05/1974	Callais et al.			
	3,798,605	03/19/1974	Feistel			
	4,159,468	06/26/1979	Barnes et al.			
	4,220,991	09/02/1980	Hamano et al.			
	4,278,837	07/14/1981	Best			
	4,323,921	04/06/1982	Guillou			
	4,442,486	04/10/1984	Mayer			
	4,529,870	07/16/1985	Chaum			
	4,558,176	12/10/1985	Arnold et al.			
	4,593,376	06/03/1986	Volk			
	4,614,861	09/30/1986	Pavlov et al.			
	4,644,493	02/17/1987	Chandra et al.			
	4,658,093	04/14/1987	Hellman			
	4,817,140	03/28/1989	Chandra et al.			
	4,868,376	09/19/1989	Lessin et al.			
	4,891,838	01/02/1990	Faber			
	4,924,378	05/08/1990	Hershey et al.			
	4,932,054	06/05/1990	Chou et al.			
	4,937,863	06/26/1990	Robert et al.			
	4,949,187	08/14/1990	Cohen			
	4,953,209	08/28/1990	Ryder, Sr. et al.			
	4,961,142	10/02/1990	Elliott et al.			
	4,975,647	12/04/1990	Downer et al.			
	4,999,806	03/12/1991	Chernow et al.			
	5,010,571	04/23/1991	Katznelson			
	5,014,234	05/07/1991	Edwards, Jr.			
	5,023,907	06/11/1991	Johnson et al.			
	5,047,928	09/10/1991	Wiedemer			
	5,058,164	10/15/1991	Elmer et al.			

Examiner	Date Considered

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this formwith next communication to applicant.

Form PTO-1449 (Rev. 8-83)	Department of Commerce Patent and Trademark Office	Atty Docket 11132[illegible]3	Serial No. 10/162,701
INFORMATION DISCLOSURE STATEMENT		Applicants: Xin WANG	
		Filing Date: June 06, 2002	Group Art Unit:

U.S. PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date (if appropriate)
	5,103,476	04/07/1992	Waite et al.			
	5,113,519	05/12/1992	Johnson et al.			
	5,136,643	08/04/1992	Fischer			
	5,138,712	08/11/1992	Corbin			
	5,146,499	09/08/1992	Geffrotin			
	5,148,481	09/15/1992	Abraham et al.			
	5,159,182	10/27/1992	Eisele			
	5,183,404	02/02/1993	Aldous et al.			
	5,191,193	03/02/1993	Le Roux			
	5,204,897	04/20/1993	Wyman			
	5,222,134	06/22/1993	Waite et al.			
	5,235,642	08/10/1993	Wobber et al.			
	5,247,575	09/21/1993	Sprague et al.			
	5,255,106	10/19/1993	Castro			
	5,260,999	11/09/1993	Wyman			
	5,263,157	11/16/1993	Janis			
	5,263,158	11/16/1993	Janis			
	5,276,444	01/04/1994	McNair			
	5,276,735	01/04/1994	Boebert et al.			
	5,291,596	03/01/1994	Mita			
	5,311,591	05/10/1994	Fischer			
	5,319,705	06/07/1994	Halter et al.			
	5,337,357	08/09/1994	Chou et al.			
	5,339,091	08/16/1994	Yamazaki et al.			
	5,341,429	08/23/1994	Stringer et al.			
	5,347,579	09/13/1994	Blandford			
	5,381,526	01/10/1995	Ellson			
	5,394,469	02/28/1995	Nagel et al.			
	5,412,717	05/02/1995	Fischer			
	5,428,606	06/27/1995	Moskowitz			
	5,432,849	07/11/1995	Johnson et al.			

Examiner	Date Considered

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this formwith next communication to applicant.

Form PTO-1449 (Rev. 8-83)	Department of Commerce Patent and Trademark Office	Atty Docket 11132[illegible]3	Serial No. 10/162,701
INFORMATION DISCLOSURE STATEMENT		Applicants: Xin WANG	
		Filing Date: June 06, 2002	Group Art Unit:

U.S. PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date (if appropriate)
	5,438,508	08/01/1995	Wyman			
	5,444,779	08/22/1995	Daniele			
	5,453,601	09/26/1995	Rosen			
	5,455,953	10/03/1995	Russell			
	5,457,746	10/10/1995	Dolphin			
	5,473,687	12/05/1995	Lipscomb et al.			
	5,473,692	12/05/1995	Davis			
	5,499,298	03/12/1996	Narasimhalu et al.			
	5,504,814	04/02/1996	Miyahara			
	5,504,818	04/02/1996	Okano			
	5,504,837	04/02/1996	Griffeth et al.			
	5,509,070	04/16/1996	Schull			
	5,530,235	06/25/1996	Stefik et al.			
	5,532,920	07/02/1996	Hartrick et al.			
	5,534,975	07/09/1996	Stefik et al.			
	5,539,735	07/23/1996	Moskowitz			
	5,563,946	10/08/1996	Cooper et al.			
	5,568,552	10/22/1996	Davis			
	5,621,797	04/15/1997	Rosen			
	5,629,980	05/13/1997	Stefik et al.			
	5,633,932	05/27/1997	Davis et al.			
	5,634,012	05/27/1997	Stefik et al.			
	5,638,443	06/10/1997	Stefik et al.			
	5,655,077	08/05/1997	Jones et al.			
	5,708,717	01/13/1998	Alasia			
	5,734,823	03/31/1998	Saigh et al.			
	5,734,891	03/31/1998	Saigh			
	5,745,569	04/28/1998	Moskowitz et al.			
	5,748,783	05/05/1998	Rhoads			
	5,761,686	06/02/1998	Bloomberg			
	5,765,152	06/09/1998	Erickson			

Examiner	Date Considered

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this formwith next communication to applicant.

Form PTO-1449 (Rev. 8-83)	Department of Commerce Patent and Trademark Office	Atty Docket 11132[redacted]23	Serial No. 10/162,701
INFORMATION DISCLOSURE STATEMENT		Applicants: Xin WANG	
		Filing Date: June 06, 2002	Group Art Unit:

U.S. PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date (if appropriate)
	5,768,426	06/16/1998	Rhoads			
	5,825,892	10/20/1998	Braudaway et al.			
	6,047,067	04/04/2000	Rosen			
	6,115,471	09/05/2000	Oki et al.			
	6,233,684	05/15/2001	Stefik et al.			
	6,266,618	05/01/2001	Downs et al.			
	6,345,256	02/05/2002	Milsted et al.			

FOREIGN PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Country	Class	Subclass	Translation Yes	Translation No
	0 084 441	07/27/1983	EP			Full Eng	
	0 180 460	05/07/1986	EP			Full Eng	
	0 332 707	09/01/1989	EP			Full Eng	
	0 651 554	05/03/1995	EP			Full Eng	
	0 668 695	08/23/1995	EP			Full Eng	
	0 725 376	08/07/1996	EP			Full Eng	
	2 136 175	09/12/1984	GB			Full Eng	
	2 236 604	04/10/1991	GB			Full Eng	
	WO 01/63528	08/30/2001	PCT			Full Eng	
	WO 92/20022	11/12/1992	PCT			Full Eng	
	WO 93/01550	01/21/1993	PCT			Full Eng	
	WO 99/49615	09/30/1999	PCT			Full Eng	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

Examiner Initial	
	"National Semiconductor and EPR Partner for Information Metering/Data Security Cards" March 4, 1994, Press Release from Electronic Publishing Resources, Inc.
	Weber, R., "Digital Rights Management Technology" October 1995
	Flasche, U. et al., "Decentralized Processing of Documents", pp. 119-131, 1986, Comput. & Graphics, Vol. 10, No. 2
	Mori, R. et al., "Superdistribution: The Concept and the Architecture", pp. 1133-1146, 1990, The Transactions of the IEICE, Vol. E 73, No. 7, Tokyo, JP

Examiner	Date Considered

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this formwith next communication to applicant.

Form PTO-1449 (Rev. 8-83)	Department of Commerce Patent and Trademark Office	Atty Docket 11132[redacted]3	Serial No. 10/162,701
INFORMATION DISCLOSURE STATEMENT		Applicants: Xin WANG	
		Filing Date: June 06, 2002	Group Art Unit:

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

Examiner Initial	
	Weber, R., "Metering Technologies for Digital Intellectual Property", pp. 1-29, Oct. 1994, A Report to the International Federation of Reproduction Rights Organizations
	Clark, P.C. et al., "Bits: A Smartcard protected Operating System", pp. 66-70 and 94, November 1994, Communications of the ACM, Vol. 37, No. 11
	Ross, P.E., "Data Guard", pp. 101, June 6, 1994, Forbes
	Saigh, W.K., "Knowledge is Sacred", 1992, Video Pocket/Page Reader Systems, Ltd.
	Kahn, R.E., "Deposit, Registration and Recordation in an Electronic Copyright Management System", pp. 1-19, August 1992, Corporation for National Research Initiatives, Virginia
	Hilts, P. et al., "Books While U Wait", pp. 48-50, January 3, 1994, Publishers Weekly
	Strattner, A, "Cash Register on a Chip may Revolutionize Software Pricing and Distribution; Wave Systems Corp.", pp. 62, April 1994, Computer Shopper, Vol. 14, No. 4, ISSN 0886-0556
	O'Conner, M., "New Distribution Option for Electronic Publishers; iOpener Data Encryption and Metering System for CD-ROM use; Column", pp. 134, March 1994, CD-ROM Professional, Vol. 7, No. 2, ISSN: 1409-0833
	Willett, S., "Metered PCs: Is Your System Watching You? Wave System beta tests new technology", pp. 84, May 2, 1994, InfoWorld
	Linn, R., "Copyright and Information Services in the Context of the National Research and Education Network ", pp. 9-20, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1
	Perrit, Jr., H., "Permission Headers and Contract Law", pp. 27-48, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1
	Upthegrove, L., "Intellectual Property Header Descriptors: A Dynamic Approach", pp. 63-66, January 1994, IMA Intellectual Property Proceedings, Vol. 1, Issue 1
	Sirbu, M., "Internet Billing Service Design and prototype Implementation", pp. 67-80, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1
	Simmel, S. et al., "Metering and Licensing of Resources: Kala's General Purpose Approach", pp. 81-110, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1
	Kahn, R., "Deposit, Registration and Recordation in an Electronic Copyright Management System", pp. 111-120, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1
	Tygar, J. et al., "Dyad: A System for Using Physically Secure Coprocessors", pp. 121-152, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1
	Griswold, G., "A Method for Protecting Copyright on Networks", pp. 169-178, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1
	Nelson, T., "A Publishing and Royalty Model for Networked Documents", pp. 257-259, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1
	Robinson, E., "Redefining Mobile Computing", pp. 238-240, 247-248 and 252, July 1993, PC Computing
	Abadi, M. et al., "Authentication and Delegation with Smart-cards", 1990, Research Report DEC Systems Research Center
	Mark Stefik, "Letting Loose the Light: Igniting Commerce in Electronic Publication", pp. 219-253, 1996, Internet Dreams: Archetypes, Myths, and Metaphors, IDSN 0-262-19373-6
	Mark Stefik, "Letting Loose the Light: Igniting Commerce in Electronic Publication", pp. 2-35, February 8, 1995, Internet Dreams: Archetypes, Myths and Metaphors,

Examiner	Date Considered

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this formwith next communication to applicant.

Form PTO-1449 (Rev. 8-83) U.S. Department of Commerce Patent and Trademark Office	Atty Docket 11132[illegible]13	Serial No. 10/162,701
INFORMATION DISCLOSURE STATEMENT	**Applicants:** Xin WANG	
	Filing Date: June 06, 2002	**Group Art Unit:**

Examiner	Date Considered

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this formwith next communication to applicant.

Requested Patent: EP0084441A2

Title:

METHOD AND APPARATUS FOR THE PROTECTION OF PROPRIETARY COMPUTER SOFTWARE. ;

Abstracted Patent: EP0084441 ;

Publication Date: 1983-07-27 ;

Inventor(s): POOLE TERENCE ERIC CYRIL;; ROGERS DAVID NIGEL ;

Applicant(s): TABS LIMITED (GB) ;

Application Number: EP19830300178 19830113 ;

Priority Number(s): GB19820001353 19820119 ;

IPC Classification: G06F13/00 ;

Equivalents: ;

ABSTRACT:

Apparatus for protecting proprietary computer software against unauthorised use comprises a store (11) for selected data, means (10) for comparing data successively communicated by a program running on a computer (1) with data from the storage means, means such as an indelible memory (15) associated with a microprocessor (14) for storing identifying data, and transmitting means (14) for sending stored identifying data to the computer. When a match is detected by the comparator, the identifying data are sent to the computer, which requires this data for continued normal running. A copy of the software cannot run on a computer without associated protection apparatus and unauthorised copies will therefore be unusable unless the protection apparatus can be obtained. For a great degree of protection, a sequence of matches and identifying data messages may be required to allow continued normal running of a program.

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

(11) Publication number: 0 084 441
A2

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 83300178.7

(22) Date of filing: 13.01.83

(51) Int. Cl.³: G 06 F 13/00

(30) Priority: 19.01.82 GB 8201353

(43) Date of publication of application:
27.07.83 Bulletin 83/30

(84) Designated Contracting States:
AT DE FR GB NL SE

(71) Applicant: TABS LIMITED
Sopers House Chantry Way
Andover Hampshire SP10 1PE(GB)

(72) Inventor: Rogers, David Nigel
The Lodge 118 Ringwood Road
Verwood Wimborne Dorset(GB)

(72) Inventor: Poole, Terence Eric Cyril
The Old Rectory
Blackford Yeovil Somerset(GB)

(74) Representative: Pritchard, Colin Hubert et al,
Mathys & Squire 10 Fleet Street
London EC4Y 1AY(GB)

(54) Method and apparatus for the protection of proprietary computer software.

(57) Apparatus for protecting proprietary computer software against unauthorised use comprises a store (11) for selected data, means (10) for comparing data successively communicated by a program running on a computer (1) with data from the storage means, means such as an indelible memory (15) associated with a microprocessor (14) for storing identifying data, and transmitting means (14) for sending stored identifying data to the computer. When a match is detected by the comparator, the identifying data are sent to the computer, which requires this data for continued normal running. A copy of the software cannot run on a computer without associated protection apparatus and unauthorised copies will therefore be unusable unless the protection apparatus can be obtained. For a great degree of protection, a sequence of matches and identifying data messages may be required to allow continued normal running of a program.

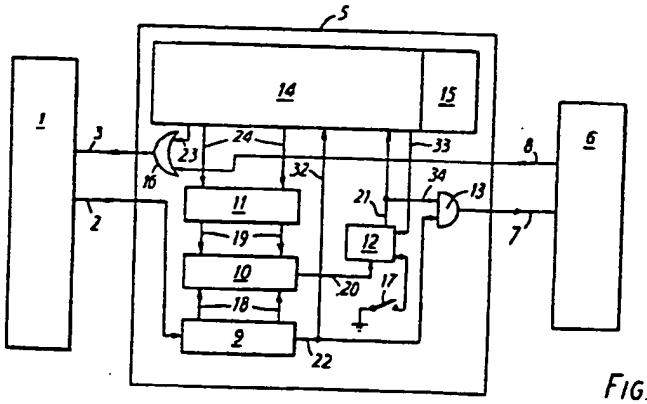


FIG. 1

EP 0 084 441 A2

Croydon Printing Company Ltd.

Method and Apparatus for the Protection of Proprietary Computer Software

The present invention relates to apparatus for the protection of proprietary computer software, and in particular for protecting such software against unauthorised use.

The large amount of time and skill which is frequently involved in the writing of computer programs means that software is very often expensive. When a hardware and software package is sold to users of computer systems, a significant part of the cost will be attributable to the software. Currently, hardware costs are falling as the number of users increases and mass production techniques can be used. For software, reproduction costs are low, but initial development costs remain high. The market has become very competitive and the unscrupulous, instead of commissioning their own programs, may be tempted to take the much cheaper alternative of copying without authority the programs of others. For sophisticated computer systems which are tailored to the particular user's requirements it is relatively straightforward to monitor the system and avoid unauthorised adoption by others of the associated software. For standard systems, on the other hand, software products are frequently shipped in volume, often through distributors, to a large buying public, giving the opportunity for copying.

One proposal for protecting software is for the software package to be sold with a hardware key. The key could, for example, be physically connected to the computer data bus. With

such an arrangement, the software would be written to read a code, such as the serial number of the particular software package, embedded in the hardware. Provided this code matched counterparts held at suitable places in the software, the running of the program would be allowed to continue, but if there was no match, the program would be stopped from running. A person who copied the program and attempted to use it with a different hardware key or with no key at all would not be able to do so.

The above proposal has two disadvantages: one is that the key is very simple and can quite easily be copied and the other is that the key is specific to each type of hardware. Software is often written in a form which allows it to be easily transferred from one basic hardware system to another. This enables the user to upgrade his hardware without having to invest in new software at the same time and also enables the supplier to offer his software for use on the latest equipment at little extra expense to him. The simple key proposed would not therefore be of much value.

The present invention provides apparatus for protecting software which is adaptable for use on all kinds of computer hardware conforming to certain standards and which cannot easily be copied.

Accordingly, the invention provides apparatus capable of protecting proprietary computer software against unauthorised use, comprising storage means for storing selected data, comparison means for comparing data successively communicated by a program

running on a computer with data from the storage means, means for storing identifying data and transmitting means for sending stored identifying data to the computer, whereby on detection of a match in use, the transmitting means send said identifying data to the computer, said data being require for continued normal running of a proprietary program.

If the program were to be run on a computer without associated protection apparatus, the program would cease running normally as it would not receive the required identifying data.

The apparatus may be provided with communications means which enable data to be transferred to and from a computer associated with the apparatus. The apparatus may also be adapted for communication with a peripheral device. The communications means can consist of simple and standard communications hardware, making the apparatus very versatile and usable with many kinds of computer hardware.

In one form, the apparatus comprises blocking means to prevent transfer of data to the peripheral device after detection of a match and until after said identifying data has been transmitted by said transmitting means. It may also include switching means for preventing actuation of the blocking means, to allow the running of a non-proprietary program without interruption by the blocking means.

In a preferred form of the apparatus which provides particularly effective protection, said selected data comprises a code of data bits and said identifying data comprises a message

of data bits and a plurality of the data codes and data messages are stored in said respective storage means for the data codes to be successively compared and said data messages to be successively transmitted, whereby a proprietary program on an associated computer will cease running normally unless a predetermined sequence of identifying data messages is received by the program from the apparatus. The apparatus can be programmed to alter in a predeteremined manner the selected group of data bits which are to be matched and to send a sequence of messages to the computer, these being required by the computer in order to allow continued normal running of the program.

The invention will now be described, by way of example, with reference to the accompanying drawing which is designated Figure 1.

The Figure shows, in block schematic form, particular protection apparatus constructed in accordance with the invention, in combination with other apparatus.

Referring to the Figure, a programmed computer 1 is connected to a peripheral device 6, which could be for example a printer, a visual display unit, or another computer. Connected between the computer 1 and peripheral device 6 is protection apparatus 5 according to the invention. Communications means 2, 3 and 7, 8 are standard RS-223-C interfaces for transmission of data to and from the computer and peripheral device.

Protection apparatus 5 comprises a shift register 9 which receives groups of data bits from the computer 1 via line 2. A further register 11 stores data in the form of an eight digit code number for example the particular code of the proprietor of the software, and is under the control of a microprocessor 14 which can, via lines 24, alter the data stored in the register 11 if required. Registers 9 and 11 are connected respectively via lines 18 and 19 to a comparator 10. The output of the comparator is connected via line 20 to a bistable 12 which is also controlled by a switch 17 and a line 33 from microprocessor 14. The output of the bistable 12 is connected via line 21 to the microprocessor 14 and line 34 to an AND-gate 13. The output of shift register 9 is directly connected with the AND-gate 13 via line 22 and also has a line 32 to the microprocessor 14. Data from gate 13 passes via line 7 to peripheral device 6 and data from device 6 may be transmitted to the computer 1 via line 8, OR-gate 16 and line 3. The microprocessor 14 is also connected to OR-gate 16 via line 23.

It will be apparent from the foregoing that microprocessor 14 receives data from and supplies data to various components of the protection apparatus 5. The microprocessor is associated with a non-volatile memory 15 which stores the multi-digit serial number of the software, or other identifying data such as a serialised message of coded characters. A program running on the computer requires transmission of the serial number (or identifying data) from the protection apparatus 5 in order to continue running.

The operation of the apparatus 5 is as follows. A program running on computer 1 sends data via line 2 as a sequence of bits to the shift register 9, where the bits are held a byte (8 bits) at a time for the purposes of comparison with the 8 digit code in register 11. Bits are successively passed to register 9 for comparison and, in the absence of a match with the code in register 11, the data passes via line 22 to AND-gate 13. As long as there is no match between the contents of registers 9 and 11, bistable 12 continues to enable AND-gate 13 and the data passes via line 7 to the peripheral device 6. All data passing via line 22 to the peripheral device 6 can be read by microprocessor 14 which is connected to the output of register 9 via line 32.

The bistable 12 and AND-gate 13 function as a temporary blocking means. If the comparator 10 detects a match, the state of bistable 12 is changed and the AND-gate 13 is disabled. This prevents peripheral 6, which may for example be a VDU or a printer, from displaying or printing data which is part of the checking operation and in unrelated to the purpose of the program. Bistable 12 is connected via line 21 to the microprocessor 14 and the bistable 12 prompts the microprocessor to transmit the serial number stored in memory 15 to the computer via line 23, OR-gate 16 and line 3.

The proprietary program running on the computer is so written as to require transmission of the serial number to keep on running normally, that is to say to continue to run and perform its intended tasks instead of carrying out checks. The program

receives and checks the serial number from apparatus 5. If the number is incorrect, the program will cease running. In an alternative form, the program may be compiled to send appropriate messages to the peripheral device 6 if the correct serial number is not received, telling the user to check that the correct apparatus and program are being used in conjunction, or warning that the program in use is an unauthorised copy.

If the serial number is found to be correct, the program is allowed to continue running normally and data passes to the apparatus via line 2, and register 9. The data is read by the microprocessor 14 via line 32, the microprocessor causes a signal to be sent to bistable 12 via line 33 which enables AND-gate 13, previously disabled on detection of a match, allowing data to pass via line 7 to peripheral device 6 which can resume its display or printing of relevant data.

If it is desired to run a non-proprietary program on computer 1, the apparatus 5 need not be disconnected as the program is permitted to run and is unaffected by the apparatus. Data from the computer 1 passes via line 2 to register 9. It is unlikely that a match will be detected by comparator 10 but to prevent a spurious match disabling gate 13, switch 17 may be closed to cause bistable 12 to keep the gate 13 enabled, allowing data to pass directly from the register 9 to gate 13 via line 22 and thence to peripheral device 6 via line 7. Data from the peripheral device may pass via line 8 through the protection apparatus, emerging

from gate 16 on line 3 to reach computer 1.

A producer of proprietary computer software can protect the software from unauthorised copying by selling the software together with apparatus according to the invention, suitably a piece of hardware in a "black box". The program is compiled so as to require specific data for continued normal running and the apparatus of the present invention provides that data. In the embodiment described above, the program is written to include the code required by register 11 of apparatus 5 to provide a match, and the match causes microprocessor 14 to send the serial number of the program to the computer 1, this being the data required to cause the program to continue running. An unauthorised person who has copied the program but has not been supplied with the apparatus 5 cannot run the program past a certain point because the serial number will not be transmitted, thus causing the running to cease. A manufacturer need not go to considerable trouble and expense, as has been necessary in the past, to stop his programs from falling into the hands of the unscrupulous because copies of proprietary programs are unusable without associated protection apparatus according to the invention.

The apparatus described and illustrated is one example of an embodiment of the invention, which has been shown for convenience as comprising a number of hardware components. It will be appreciated that these could be replaced by fixed firmware in a dedicated microcomputer to perform similar functions. The code or

other data store which in the embodiment described above was register 11 could comprise a PROM. The incoming data from the computer could be stored for comparison purposes in a memory register and the comparison carried out by the firmware through a sequence of logical operations under the instructions of the microcomputer.

The apparatus of Figure 1 is a particularly simple example, and more sophisticated forms could provide a greater degree of software protection. For example, microprocessor 14, which can control register 11 via lines 24, may be programmed to modify the contents of the register once or several times after comparator 10 has detected a first match, the program running on the computer being programmed to supply the appropriate codes via line 2. Memory 15 may store a sequence of serial numbers or messages to be released in turn, after detection of successive matches by comparator 10, to the program which requires these numbers for the performance of a sequence of tasks or checks, necessary to allow continued running of the program and for the sending of a message via line 2 for the enabling of gate 13.

Gate 13, or an equivalent device, is optional and may be included if a peripheral device is running in series with the protection apparatus. If a peripheral device is running on a different line or there is no such device, lines 2 and 3 will communicate solely with apparatus 5, and gate 13, which serves to prevent data present purely for the purpose of checking from

reaching the peripheral device, would be redundant.

The versatility of the apparatus may be increased by carrying out the initial comparison on an arbitrary time base with repeated sampling so as to determine the baud rate of the data prior to checking the pattern of bits for a match. Once a match has been detected the apparatus may allow all data unconnected with the checking operation to be transmitted to a peripheral device running on the computer.

Where the apparatus is connected in an on-line mode on a communications port on a computer, it may act as a terminal concentrator for more than one output channel. The apparatus would then be provided with several interfaces such as interface 7, 8 with associated logic in the microprocessor. The apparatus is versatile because it is connected via a standard interface to the particular user's hardware and can still be used if the hardware is updated, in association with suitably modified software.

CLAIMS:

1. Apparatus capable of protecting proprietary computer software against unauthorised use, comprising storage means for storing selected data, comparison means for comparing data successively communicated by a program running on a computer with data from the storage means, means for storing identifying data and transmitting means for sending stored identifying data to the computer, whereby on detection of a match in use, the transmitting means send said identifying data to the computer, said data being required for continued normal running of a proprietary program.

2. Apparatus as claimed in claim 1, wherein said selected data comprises a code of data bits and said identifying data comprises a message of data bits and a plurality of the data codes and data messages are stored in said respective storage means for the data codes to be successively compared and said data messages to be successively transmitted, whereby a proprietary program on an associated computer will cease running normally unless a predetermined sequence of identifying data messages is received by the program from the apparatus.

3. Apparatus as claimed in claim 1 or claim 2, including a combined storage means for storing said selected data and said identifying data.

4. Apparatus as claimed in any preceding claim, including communications means for the transfer of data to and from a computer associated with the apparatus.

5. Apparatus as claimed in any preceding claim wherein the apparatus is adapted for communication with a peripheral device.

6. Apparatus as claimed in claim 5, comprising blocking means to prevent transfer of data to the peripheral device after detection of a match and until after said identifying data has been transmitted by said transmitting means.

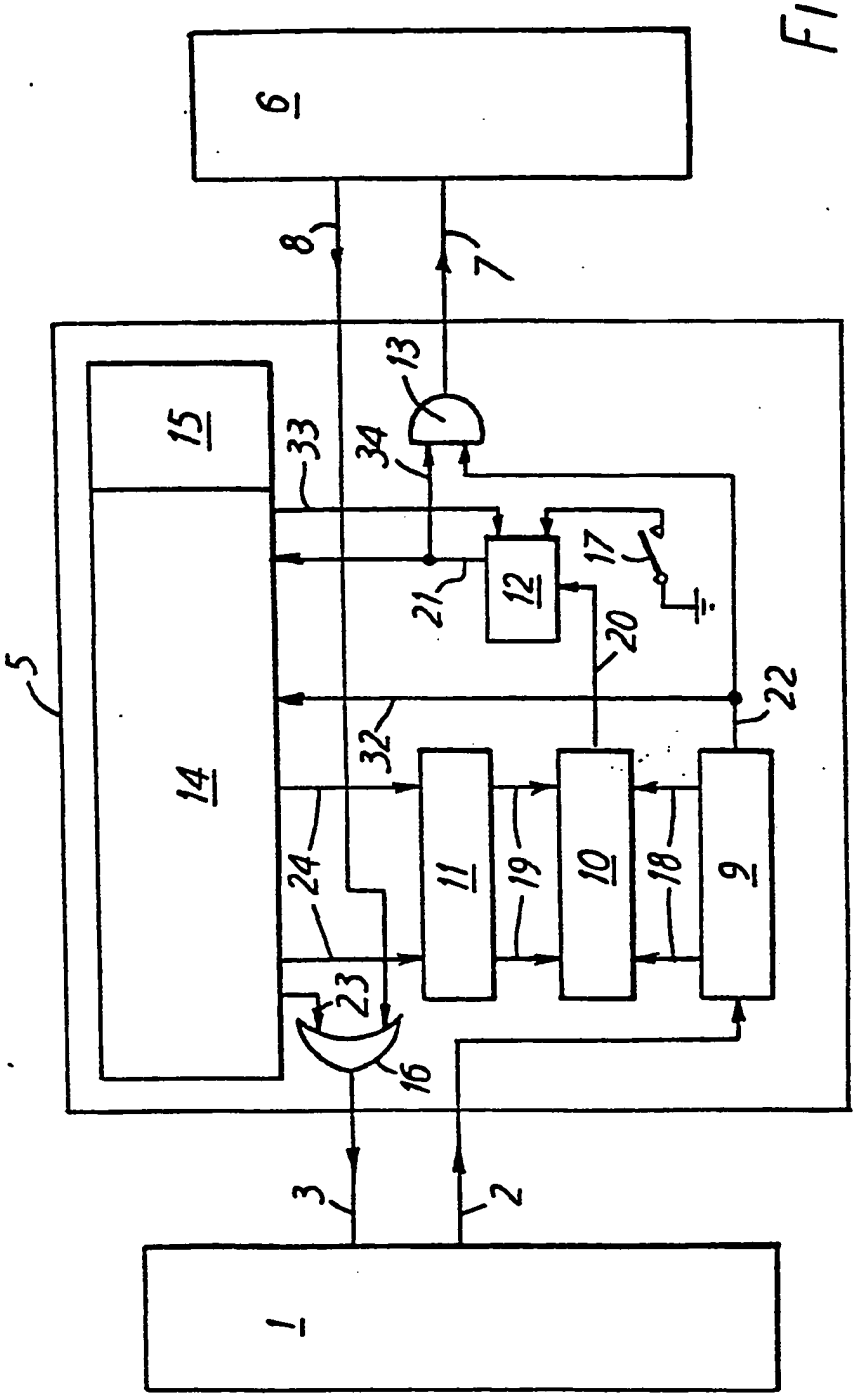
7. Apparatus as claimed in claim 6, including switching means for preventing actuation of the blocking means, to allow the running of a non-proprietary program without interruption by the blocking means.

8. Apparatus as claimed in any preceding claim comprising a programmed microcomputer, wherein the storage means for the selected data and the identifying data comprise programmable read only memories and the comparison means is adapted to operate under the instructions of the programmed microcomputer by carrying out a sequence of logical operations to detect a match.

9. A method for protecting proprietary computer software against unauthorised use, comprising storing selected data and identifying data in protection apparatus associated with a programmed computer, successively communicating data from the computer to the apparatus, comparing said communicated data with said selected data and, on detection of a match, transmitting said identifying data to the computer, the identifying data being required by a proprietary program for continued normal running.

1/1

FIG. 1



Requested Patent: EP0180460A1

Title: DECODER FOR A PAY TELEVISION SYSTEM ;

Abstracted Patent: US4759060 ;

Publication Date: 1988-07-19 ;

Inventor(s): HAYASHI TOSHIHIDE (JP); KANNO MASAYOSHI (JP) ;

Applicant(s): SONY CORP (JP) ;

Application Number: US19850793554 19851031 ;

Priority Number(s): JP19850110001 19850521; JP19840229348 19841031 ;

IPC Classification: ;

Equivalents: CA1261960, DE3579785D ;

ABSTRACT:

A decoder for a pay television system incorporates a receiver for receiving scrambled programming and control data transmitted from a remote location, with the control data including data corresponding to the program fee and the program status, and a manual switch for selectively descrambling the program data when the program status is a pay-per-view program. The decoder has a storage unit for storing a credit value, transmitted from the remote location, and the decoder has timer apparatus for measuring the time during which a pay-per-view program is being received and for subtracting a program fee periodically from the value stored in the storage unit, during the time of reception of the selected program, provided the switch is operated to descramble the program.

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

(11) Publication number: **0 180 460**
A1

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 85307832.7

(22) Date of filing: 29.10.85

(51) Int. Cl.⁴: H 04 N 7/16

(30) Priority: 31.10.84 JP 228348/84
21.05.85 JP 110001/85

(43) Date of publication of application: 07.05.86
Bulletin 86/19

(84) Designated Contracting States: DE FR GB NL

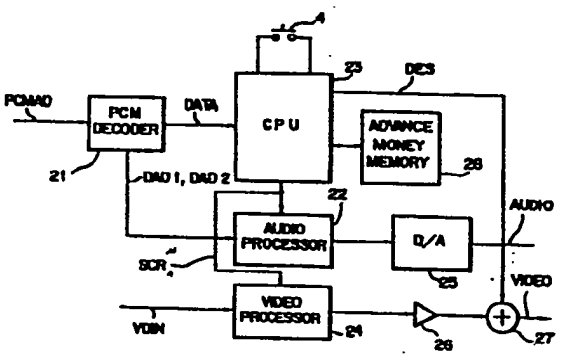
(71) Applicant: SONY CORPORATION,
7-35 Kitashinagawa 6-Chome Shinagawa-ku,
Tokyo 141 (JP)

(72) Inventor: Hayashi, Toshihide c/o Patents Division, Sony Corporation 6-7-35 Kitashinagawa, Shinagawa-ku
Tokyo 141 (JP)
Inventor: Kanno, Masayoshi c/o Patents Division, Sony Corporation 6-7-35 Kitashinagawa, Shinagawa-ku
Tokyo 141 (JP)

(74) Representative: Thomas, Christopher Hugo et al, D Young & Co 10 Staple Inn, London WC1V 7RD (GB)

(54) Decoders for pay television systems.

(57) A decoder control circuit for a pay television system includes a descrambler (22, 24) for descrambling a selected video programme transmitted from a broadcast centre (9), a decoder (21) for receiving and decoding control data transmitted from the centre (9), the control data including at least programme feed data and programme status data, an advance money memory (28) for storing money data transmitted from the centre (9), and a central processing unit (23) for subtracting a programme fee from the money data stored in the advance money memory (28) upon reception of the selected programme.



EP 0 180 460 A1

DECODERS FOR PAY TELEVISION SYSTEMS

This invention relates to decoders for pay television systems.

Broadcasting systems include cable television (CATV) systems and direct-broadcasting satellite (DBS) systems. These are frequently used for pay television systems, using an appropriate converter.

In a pay television system of this type, a conventional decoder for decoding a programme of a desired channel has two main status modes; a free mode and a pay mode. The free mode allows free reception of a television programme, and the pay mode represents a status mode in which a subscriber is charged for reception of a television programme.

More particularly, the pay mode is further classified into a flat fee mode, a tier level pay mode and a pay-per-view mode. In the tier level pay mode, a tier level representing the rank of programmes to be received by the decoder is predetermined. A user subscribes to a desired tier level and pays fees to a broadcast station or centre, corresponding to the subscribed tier level. In the pay-per-view mode, the user reserves desired programmes, and pays the programme fee only for the reserved programmes to a broadcast centre.

In the pay-per-view mode, user procedures are cumbersome. When the user wishes to watch a pay programme in the pay-per-view mode according to a first known procedure, the broadcast centre posts a programme schedule to each user. The user telephones the broadcast centre by a predetermined date before a desired programme is to be broadcast (say a week or a day beforehand) so as to reserve the desired programme. At this time, the user pays the fee for the programme. When the desired programme is on the air, the broadcast centre sends a reservation confirmation signal, hereinafter referred to as an identification (ID) signal, to a user who has reserved the corresponding programme. When a user's receiver or a decoder in the receiver receives the ID signal, a scrambled programme signal is descrambled by the decoder, so that the user can watch the programme.

In the pay-per-view system described above;

(A) the user must telephone the broadcast centre to reserve the desired programme, which is time-consuming and may be inconvenient,

(B) each programme has a reservation due date, so that the user cannot reserve the desired programme when the reservation due date has passed,

(C) fees cannot be refunded even if the user does not watch the reserved programme, and

(D) an idle time is required to send the ID signal to all reserved users at the beginning of every reserved programme.

According to a second, improved procedure, a pay-per-view status signal is sent from the broadcast centre to each user. When the user wishes to watch a programme represented by the pay-per-view station signal, he depresses a pay-per-view switch located on his tuner. The scrambled programme is then descrambled, and the user can watch the programme. When the user actually watches the programme, charge data transmitted with the subscribed programme is stored in an account memory of the decoder. The centre periodically checks the contents of the account memory of each decoder, using a telephone line, and collects fees or bills the applicable charge.

This effectively solves the drawbacks of the first procedure. However, since the broadcast centre must periodically check the account memories of all users, the checking system is complicated. Moreover, since fee collection is performed by use of a telephone line, an auto-dial unit and a modem (modulator/demodulator) are required, so that the required user unit and centre unit are complex and expensive.

According to the present invention there is provided a decoder control circuit for a pay television system, the decoder control circuit comprising:

means for descrambling a selected video programme transmitted from a centre at a remote location;

characterised by:

means for receiving and decoding control data transmitted from said centre, said control data including at least programme feed data and programme status data;

an advance money memory for storing money data transmitted from said centre; and

control means for subtracting a programme fee from the money data stored in said advance money memory upon reception of said selected programme.

According to the present invention there is also provided a decoder for a pay television system, the decoder comprising an advance account memory for storing deposit data sent from a broadcast centre, means for subtracting programme fees from the contents of the advance account memory for every reception of a pay programme, and means for displaying a warning display when the contents of the advance account memory equals less than a predetermined value.

The pay-per-view mode may be classified into two sub-modes to prevent unreasonable payment. A first sub-mode constitutes a time pay-per-view for determining the charge per unit of time, and the second sub-mode constitutes a programme pay-per-view mode for determining a charge per programme. The tuner decodes a programme status code signal sent from the broadcast centre and selects either the time pay-per-view sub-mode or the programme pay-per-view sub-mode.

The broadcast centre can determine which sub-mode applies to programmes of different sorts, without interfering with the user's ability to view programmes. For instance, the programme pay-per-view mode may be employed for a special sports programme such as a boxing match programme, while other programming employs the other sub-mode. In this way a reasonable charge for any programme can be made.

The invention will now be described by way of example with reference to the accompanying drawings, throughout which like parts are referred to by like references, and in which:

Figure 1 is a schematic diagram of a pay-per-view system applied to a DBS utilizing a broadcast satellite;

Figure 2 is an illustration of a data format of a pulse code modulation (PCM) data signal used in the DBS;

Figure 3 is a block diagram of a tuner unit shown in Figure 1;

Figure 4 is a flow chart illustrating the operational mode of the tuner shown in Figure 1;

Figure 5 is a flow chart illustrating the pay-per-view mode of the tuner shown in Figure 1; and

Figure 6 is a flow chart illustrating a control programme continuously executed by the central processing unit (CPU) of the tuner.

In the system of Figure 1, a broadcast wave from a broadcast satellite 13 is directly received by a receiving parabolic antenna 2 and supplied to a DBS tuner 1. Audio and video signals reproduced by the DBS tuner 1 are supplied to a television monitor 14. An unscrambled broadcast signal is reproduced by a decoder in the tuner 1 in the free mode. However, in the pay mode or the pay-per-view mode, a scrambled broadcast signal is descrambled and reproduced on the television screen.

When the pay-per-view mode is utilized by the user, the user deposits funds using a cash card 8 or the like, to advance money from an account within a bank 6 to a broadcast centre 9. The cash card 8 is conveniently used with an automatic teller machine (ATM) (not shown). The deposit data is transmitted from a computer 7 of the bank 6 to a computer 10 of the broadcast centre 9. The desired programme is transmitted from the broadcast centre 9 to the user through a satellite broadcast link. The deposit data is inserted in the data coded in a PCM data signal transmitted during a vertical blanking period of the video signal. The deposit data is transmitted from a transmitter 11 to the antenna 2 of each user through a broadcast parabolic antenna 12 by way of the satellite 13.

The tuner 1 of each user has an advance money memory, and the deposit data is stored therein. The content of the storage can be displayed at any time on a display 5.

The broadcast centre 9 transmits a television signal having a PCM data signal PCMAD with the format shown in Figure 2, during a vertical blanking period of the video signal. The PCM audio signal PCMAD incorporates a plurality of data slots for data of different types. Two slots correspond to two (alternately used) channels of audio data DAD1 and DAD2. Also included are a slot for the deposit money data DCIN (representing the amount of the deposited fees) and a slot for address data DADD, including the user identification (ID) code. A data code field DDCD is added to the above data string, and an error check code field DBCH is also added to the end of the data string.

The data code slot DDCD contains a frame sync code FSYN as the initial data thereof. The following data consist of the mode data word MODE, a programme status code word PROM, a channel code CHCD representing a broadcast channel number, first and second range bits REG1 and REG2 used to expand the compressed audio data, a scramble sync code SCRM used for decrambling the signal, a charge code CHAG representing the programme fees, a data code DATE, and user bits USBT.

The programme status code PROM comprises a 4-bit code signal representing the pay mode assigned to the current broadcast programme. For example, in one embodiment the programme status code PROM is set to be "0000" in the free mode and "0001" in the pay mode.

The pay-per-view mode is further classified into a time pay-per-view mode represented by a code "0101" and a programme pay-per-view mode represented by a code "0110".

The free and pay mode are, respectively, a mode for allowing free reception of programmes and a mode for subscription programmes on a monthly basis in the same manner as in the known system. The time pay-per-view sub-mode is set to charge for the length of listening time, and the programme pay-per-view sub-mode is set to charge predetermined fees for the programme regardless of reception time.

The broadcast centre 9 transmits with each programme the programme status code PROM added to the data code DDCD. A proper pay mode is predetermined in accordance with the contents of the programme by the broadcast centre 9, so that the appropriate programme status code PROM is added to the data code DDCD, and the resulting broadcast signal is transmitted to the user. Thus, fees can be charged in accordance with charging levels suitable for the respective programmes.

In the tuner 1 which receives the broadcast signal, the PCM data signal PCMAD is supplied to a PCM decoder 21 (Figure 3), located within the tuner 1 (Figure 1). The PCM decoder 21 decodes the PCM audio signal PCMAD to extract the first and second channel audio data DAD1 and DAD2 (Figure 2) which are supplied to an audio processor 22. The other data DATA are supplied to a central processing unit (CPU) 23 which is preferably a microcomputer. A video signal VDIN (which may be scrambled) in the broadcast signal is supplied to the input of a video processor 24, which is arranged to unscramble the video signal if necessary.

As shown in Figure 4, a step SP1 inspects the code word PROM and controls the subsequent operation accordingly. When the programme status code PROM represents the time pay-per-view mode or the programme pay-per-view mode (step SP1) a pay-per-view programme SP2 runs under the control of the CPU 23 to perform a processing step SP3.

However, when the programme status code PROM represents the free mode, a free mode programme SP4 is executed. In this case, the CPU 23 causes the audio processor 22 to decode the audio data DAD1 and DAD2 and the video processor 24 to decode the video input signal VDIN.

When the programme status code PROM represents the pay mode, a pay programme SP5 is executed under control of the CPU 23. The CPU 23 determines in a step SP6 whether or not the user ID signal included in the data DATA is the same as a unique user ID assigned to the tuner 1. If YES in the step SP6, the CPU 23 supplies the scramble sync code SCRM to the audio and video processors 22 and 24, so as to cause them to perform descrambling of the audio data DAD1 and DAD2 in a step SP7. Therefore, the audio and video signals are both reproduced. However, if the result is NO in the step SP6, the CPU 23 does not supply the scramble sync code SCRM to the audio and video processors 22 and 24, so as not to cause them to perform descrambling of the audio and data DAD1 and DAD2.

The audio signal reproduced by the audio processor 22 is supplied to a digital-to-analog (D/A) converter 25. The D/A converter 25 generates an audio output signal AUDIO. The video signal reproduced by the video processor 24 is supplied to a buffer amplifier 26. A video output signal VIDEO is generated as the sum output of an adder 27.

In the pay-per-view mode, the CPU 23 performs the processing step SP3, as shown in Figure 5. When the processing step SP3 is started, the CPU 23 checks in a step SP11 whether or not a pay-per-view switch 4 is turned on.

As shown in Figure 1, the pay-per-view switch 4 is mounted on the operation panel of the tuner 1. After the user has entered a desired channel with a ten-key pad 3, he depresses the pay-per-view switch 4, if the programme is a pay-per-view programme, which sets the tuner 1 in the pay-per-view mode. The reception channel number is displayed on the display 5. The above sequential operations are performed under the control of the CPU 23.

If the result of the step SP11 is NO (Figure 5), the CPU 23 determines that the user has to set the tuner 1 in the pay-per-view mode. Control then advances to a step SP12, and no descrambling is performed. Thereafter, the flow returns repeatedly to the step SP11. The CPU 23 thus waits until the user turns on the pay-per-view switch 4.

When the pay-per-view switch 4 is turned on by the user, the result of the step SP11 is YES. The CPU 23 then supplies the scramble sync code SCRM to the audio and video processors 22 and 24 for descrambling. The CPU 23 thus controls the generation of both the audio and video output signals AUDIO and VIDEO.

When the content of the programme status code PROM represents the time pay-per-view mode, the CPU 23 fetches this data in a step SP14, and the flow advances to a step SP15.

The CPU 23 checks in the step SP15 the length of viewing time in the time pay-per-view mode. More specifically, the CPU 23 checks whether or not a unit time has been counted by a timer incorporated in the CPU 23. If the result is NO in the step SP15, the control flow returns to the step SP11. The CPU 23 waits until counting of the unit time by the timer is completed by a loop of the steps SP11, SP13, SP14, SP15 and SP11.

When the unit time has elapsed, the result of the step SP15 is YES. The control flow then advances to a charge step SP16, in which one unit time fee is subtracted from the contents of an advance money memory 28 (Figure 3). This may be a fixed quantity, or a quantity designated by the CHAG field of the PCM data. The control flow returns to the step SP11 again to reset the timer. The CPU 23 waits for the predetermined unit time to elapse by the loop of steps SP11, SP13, SP14, SP15 and SP11.

In the same manner as described above, the CPU 23 sends out the audio and video output signals AUDIO and VIDEO which are descrambled and reproduced by the audio and video processors 22 and 24 while the time pay-per-view mode programme is being received by the user. A fee corresponding to the viewing time of the programme is subtracted from the contents of the advance money memory 28 in the step SP16.

When the user wishes to stop receiving the programme in the time pay-per-view mode, he merely turns off the pay-per-view switch 4. In this case, the step SP11 is determined by the CPU 23 to be NO, and the CPU 23 then causes the audio and video processors 22 and 24 to disable descrambling by the loops of steps SP11, SP12, SP11 ... , etc.

When the user wishes to watch programmes in the programme pay-per-view mode, the user simply turns on the time pay-per-view switch 4 of the tuner 1 in the same manner as in the time pay-per-view mode. In this case, after the step SP11 has been determined by the CPU 23 to be YES, the CPU 23 controls the audio and video processors 22 and 24 to perform descrambling in the step SP13.

In this case, the programme status code PROM included in the data DATA represents the programme pay-per-view mode detected by the CPU 23 in a step SP21. The control flow advances to a decision step SP22. The CPU 23 determines in the step SP22 whether the current programme being received is the same as the programme being received when the step SP22 last had control or whether the programme being received has been changed. If the result is NO in the step SP22, the CPU 23 determines that the same programme is being received. Then the control flow returns to the step SP11. The user can continuously watch the programme in the programme pay-per-view mode by a loop of the steps SP11, SP13, SP21, SP22 and SP11.

When the user wishes to change the programme being received, the step SP22 is determined by the CPU 23 to be YES. Then a charge is subtracted from the contents of the advance money memory 28 at the step SP23, and the flow returns to the step SP11.

Once the programme has changed, the mode is set to continue the changed programme. The CPU 23 monitors this programme by the loop of the steps SP11, SP13, SP21, SP22 and SP11.

Every time one programme has been completed or finished, the CPU 23 performs charge processing, so that the user is charged per programme, irrespective of the elapsed time.

When the user operates the pay-per-view switch 4, the mode of the tuner 1 is set to be either the time pay-per-view mode or the programme pay-per-view mode. The CPU 23 supplies to an input of the adder 27 (Figure 3) a display signal DES representing that the current programme is set in the time or programme pay-per-view mode. For example, numerical value "1" representing the time pay-per-view mode or numerical value "2" representing the programme pay-per-view mode is displayed on the screen of the monitor 14. The user can easily see the charging mode being used for the current programme.

Figure 6 illustrates a flow chart of operations continuously checked by the CPU 23. Normally, control stays in one of the loops illustrated in Figure 5. These loops are represented in Figure 6 by a fetch charge code step SP31, which passes control through a return step SP38. The return step SP38 returns control to a step SP30 which normally passes control to the fetch charge code step SP31, to define the loop of Figure 5 which is active (for the appropriate sub-mode). Periodically, the check routine step SP30 sends control to two other paths illustrated in Figure 5. Preferably this is accomplished by a signal from a timer of the CPU 23 which interrupts the normal control loop, to check the balance data maintained in the advance money memory 28 at periodic intervals, or to update the data stored in the advance money memory 28 in response to detection of data within the DCIN slot of PCMAD (Figure 2). In the loops represented by the step SP31, the charge code data CHAG superimposed on the programme by PCMAD is detected by the CPU 23, and, as previously described, when the programme or channel is changed, the step SP31 is executed.

A step SP32 receives control periodically by a timer interrupt. Alternatively, execution of the charge step SP16 or SP23 (Figure 5) may include the setting of a flag which causes the step SP30 to pass control to the step SP32. In the step SP32, the CPU 23 compares the balance of the advance money memory 28 with the charge code CHAG fetched by the step SP31. When the CPU 23 determines that the balance of the advance money memory 28 is smaller than the data represented by the charge code CHAG, the CPU 23 causes the monitor 14 to display a message representing a request for deposit in a step SP37. In this case, the CPU 23 inhibits the SCRM signal, so that the programme cannot be watched. The deposit data is checked in a step SP33. The step SP30 passes control to the step SP33 periodically, by a timer interrupt, so that the control data PCMAD may be checked for deposit information. Alternatively, the step SP30 may check each incoming control word PCMAD for deposit data and branch to the deposit routine step SP33 whenever data is detected within the DCIN field. When deposit money data DCIN is detected as present in the step SP33 (Figure 6), the CPU 23 checks the ID code included in the address data DADD. In this case, if the result is YES in a step SP34, the deposit money data DCIN is added to the content of the advance money memory 28 in a step 36. When all the routines have been completed in Figure 6, the flow

returns from the step SP38. Preferably, the advance money memory 38 comprises a non-volatile memory device such as an MNOS semiconductor RAM.

With the above arrangement, the pay-per-view mode for designating a pay programme is classified into time and programme pay-per-view sub-modes. Charging is determined in units of time or programmes. Therefore, a reasonable charging system suitable for the purposes of users can be established.

A desired programme need not be reserved by telephone, and the user can conveniently enjoy the DBS system. If the user does not watch a reserved programme, no charge is made. Moreover, when the pay-per-view switch 4 is depressed, the user can watch any programme at any time, provided the sum stored in the advance money memory 28 is enough.

Since the user is not charged through a telephone line (bidirectional communication system), the configuration of the user's receiver and the broadcast centre system can be simplified and made less expensive. The user simply receives a message via the step SP37 representing a request for a deposit when the value of the advance account memory is less than the predetermined value which is needed for viewing a particular programme. The user pays a fee in accordance with the message, or else has it automatically deducted from the advance money memory 28, so that a simple efficient system can be provided. In the pay-per-view system of this embodiment, the transmission line from each user to the centre comprises a link through a bank or the like. Such a link may be located at the user's home, or a public link may be used such as the link of a bank's ATM. Programme fee data, deposit amount data, and the like are sent from the broadcast centre 9 to the user through a transmission line using the regular broadcast channel as shown in Figure 1. Shortage of a deposit amount is automatically displayed by the user's monitor 14. Therefore, a conversational two-way communication between the user and the broadcast 9 is provided, although only a one-way line terminating at the user's home is actually used.

The broadcast centre 9 may, but need not, send the ID signal to all users when payment data is obtained. Many control words relative to payment data can be sent to users within a limited time, and additional information such as a text or message can be sent, if desired.

The above embodiment exemplifies a DBS system. However, the present invention can be applied to other pay broadcast systems such as a CATV system. The invention can also be applied to an audio pay broadcast system.

Alternatively, in the above embodiment, the fee corresponding to the actual viewing time detected by the timer incorporated in the CPU 23 can be subtracted from the contents of the advance account memory periodically, by means of a timer interrupt procedure, which is enabled during the course of a pay mode.

CLAIMS

1. A decoder control circuit for a pay television system, the decoder control circuit comprising:

means (22,24) for descrambling a selected video programme transmitted from a centre (9) at a remote location;

characterised by:

means (21) for receiving and decoding control data transmitted from said centre (9), said control data including at least programme feed data and programme status data;

an advance money memory (28) for storing money data transmitted from said centre (9); and

control means (23) for subtracting a programme fee from the money data stored in said advance money memory (28) upon reception of said selected programme.

2. A decoder control circuit according to claim 1 wherein said control means (23) includes a manual switch (4) for selectively allowing descrambling of said programme when said programme is a pay-per-view programme, as represented by pay-per-view status data in said control data.

3. A decoder control circuit according to claim 1 wherein said control means (23) includes means to inhibit the reception of said programme when said programme fee data exceeds the money data stored in said advance money memory (28).

4. A decoder control circuit according to claim 3 wherein said control means (23) further includes display means (14) for displaying a message giving reasons for inhibiting the reception of said programme.

5. A decoder control circuit according to claim 1 wherein said advance money memory (28) includes means for verifying an identification code transmitted with said control data and for selectively prohibiting the storing of said money data when said identification data transmitted in association with said money data does not coincide with the individual identification code of the decoder with which said control circuit is associated.

6. A decoder control circuit according to claim 2 wherein said control means (23) includes timer means arranged to measure the time during which said programme is received, and for controlling said subtraction based on the measure of reception time.

5

7. A decoder control circuit according to claim 6 including means for activating said timer means when said manual switch (4) is turned on, and for deactivating said scrambling when said manual switch (4) is turned off during the period of reception of a time pay-per-view programme.

FIG. 1

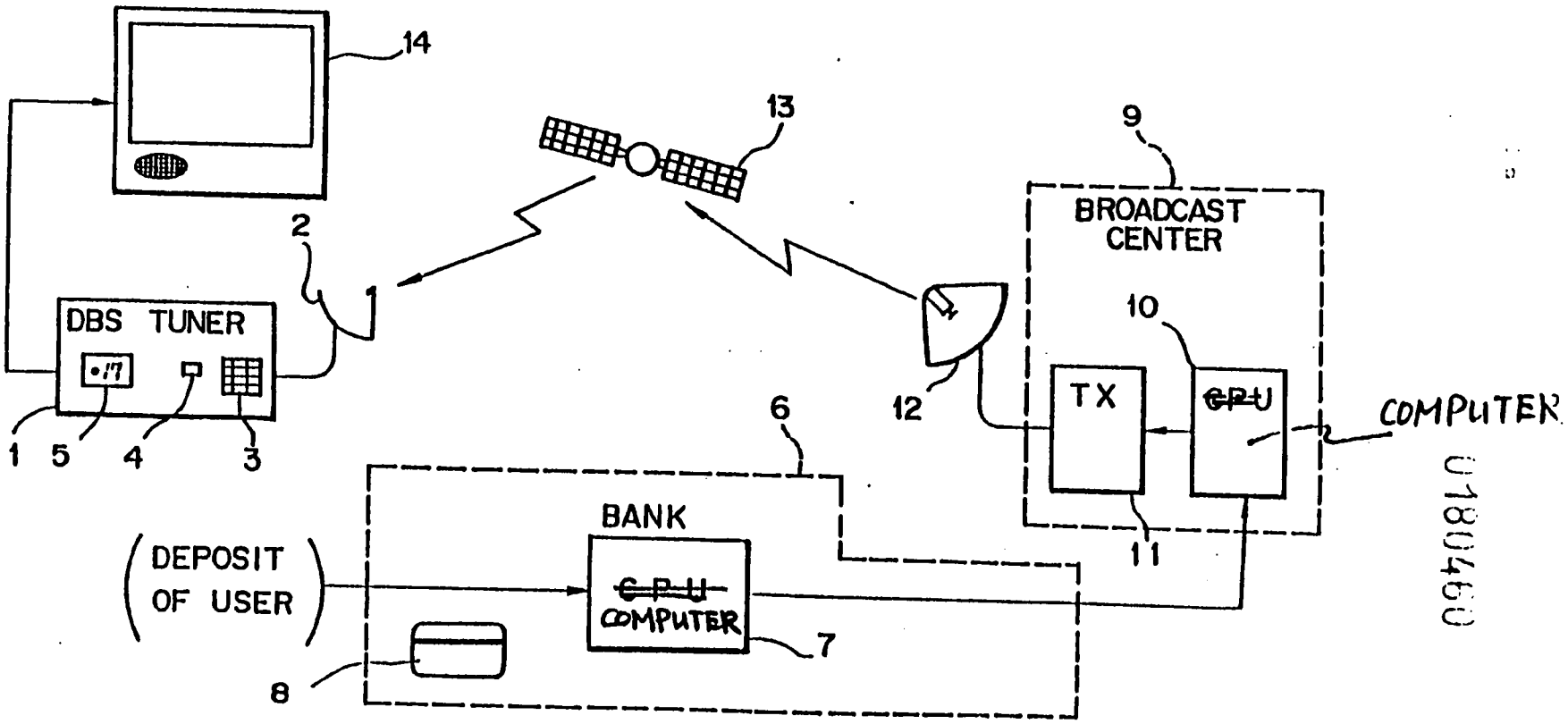


FIG. 2

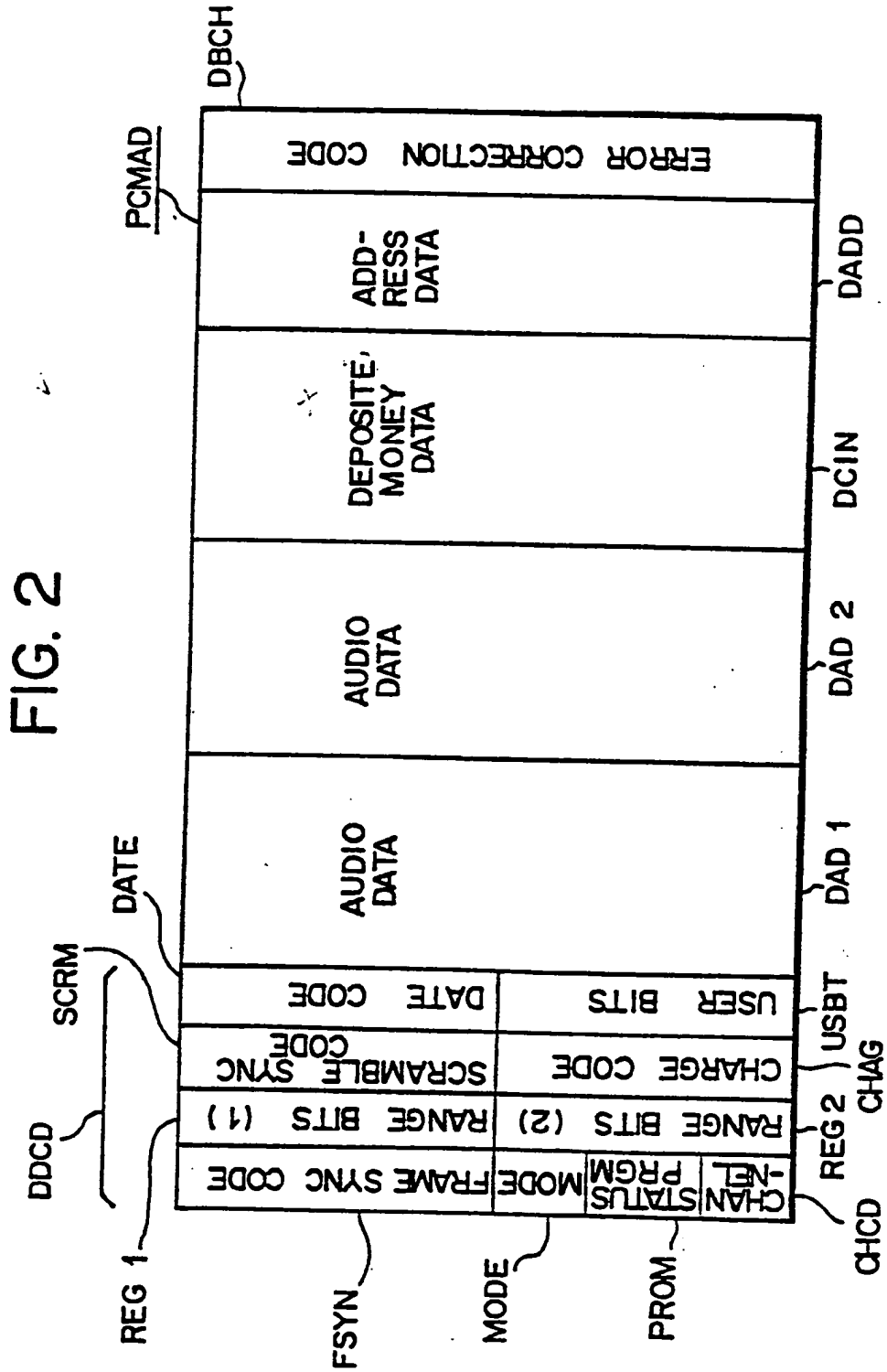
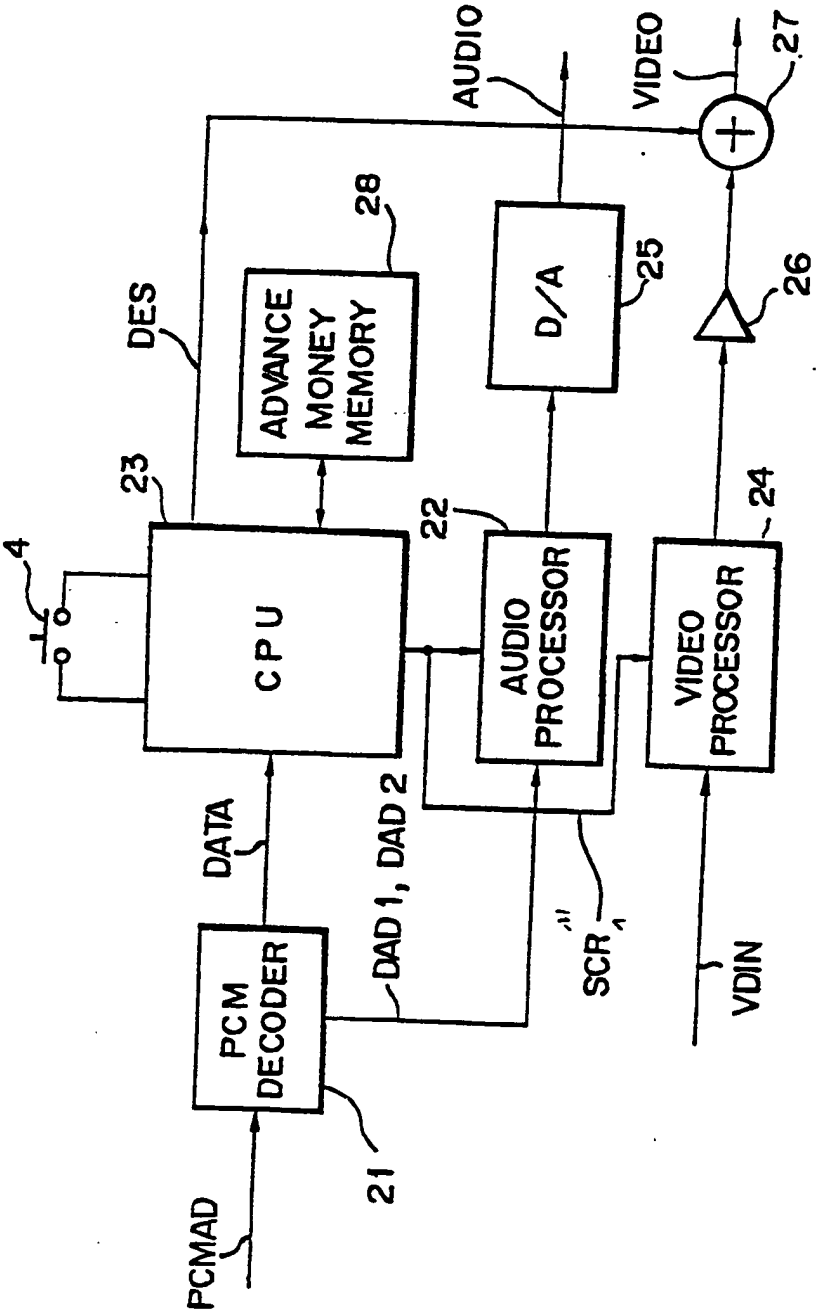

DBCH
PCMAD
ERROR CORRECTION CODE
ADD-RESS DATA
DADD
DEPOSITE MONEY DATA
DCIN
AUDIO DATA
DAD 2
AUDIO DATA
DAD 1
DATE
SCRM
USER BITS
USBT
DATE CODE
SCRAMBLE SYNC CODE
CHARGE CODE
CHAG
RANGE BITS (2)
REG 2
RANGE BITS (1)
DDCD
FRAME SYNC CODE
MODE
STATUS PRGM
CHAN-NEL
CHCD
REG 1
FSYN
MODE
PROM

FIG. 3


AUDIO
VIDEO
27
28
DES
ADVANCE MONEY MEMORY
D/A
25
26
23
22
24
4
CPU
AUDIO PROCESSOR
VIDEO PROCESSOR
DATA
DAD 1, DAD 2
SCR
PCM DECODER
VDIN
21
PCMAD

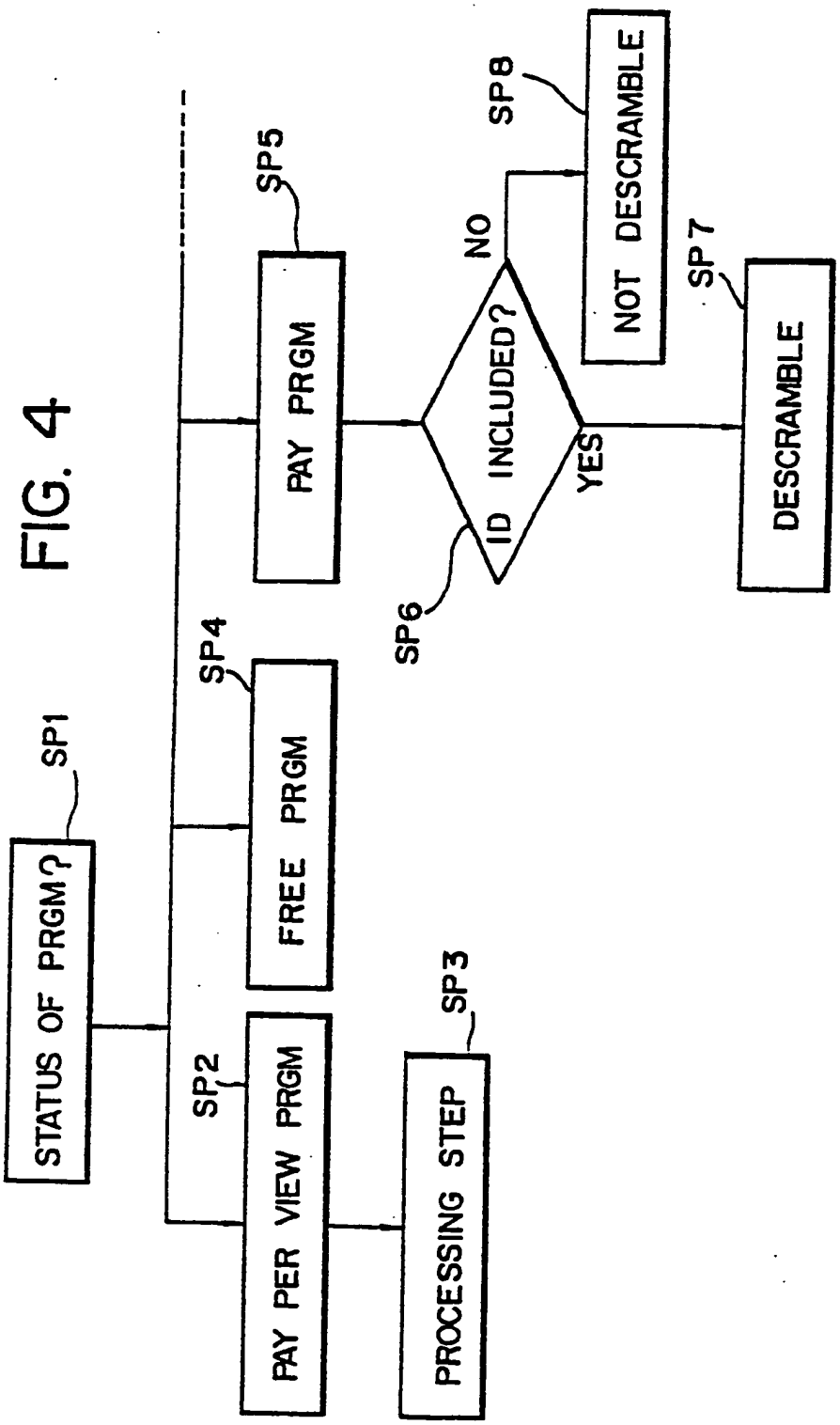
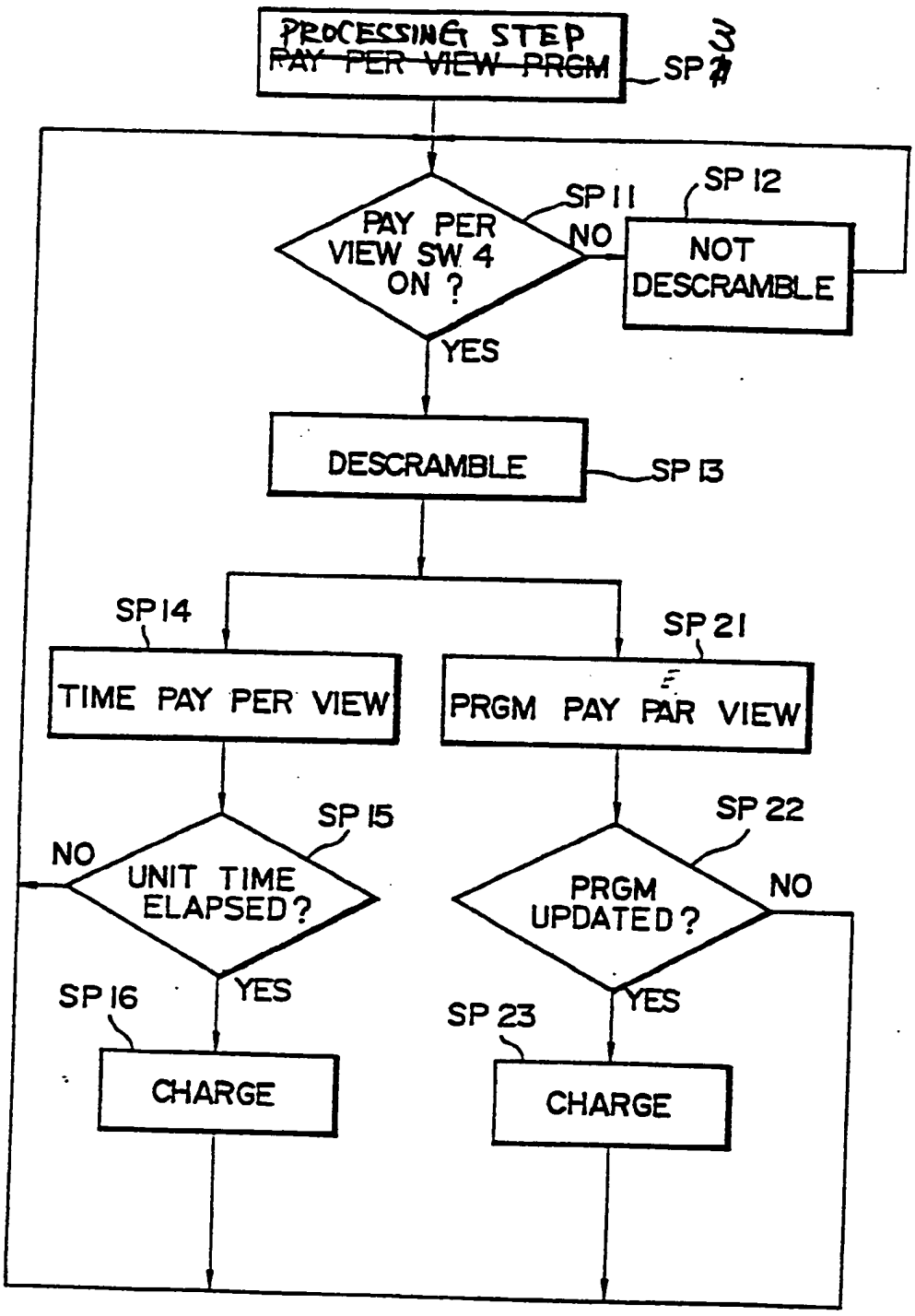

STATUS OF PRGM?
SP1
FREE PRGM
SP4
PAY PER VIEW PRGM
SP2
PROCESSING STEP
SP3
PAY PRGM
SP5
ID INCLUDED?
SP6
NO
YES
NOT DESCRAMBLE
SP8
DESCRAMBLE
SP7

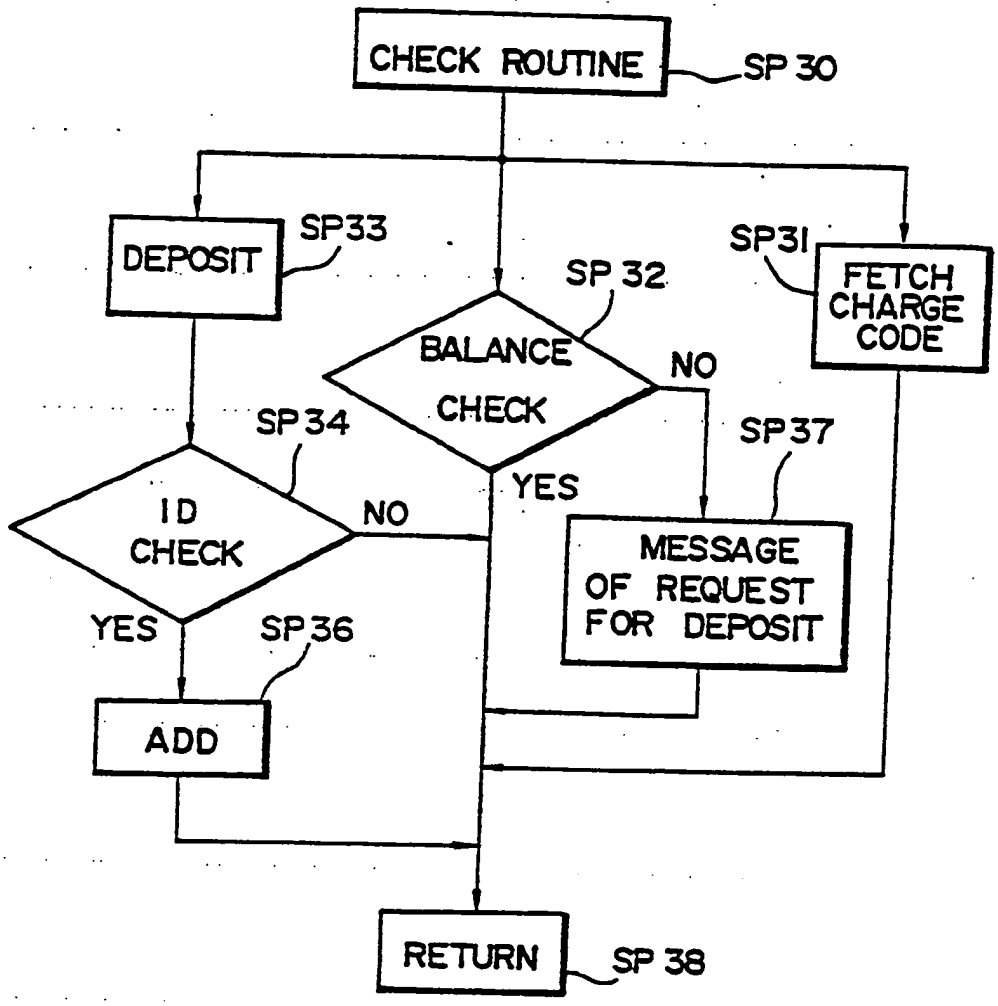
FIG. 4

FIG. 5


PROCESSING STEP
PAY PER VIEW PRGM
SP 3
PAY PER VIEW SW 4 ON ?
SP 11
NO
SP 12
NOT DESCRAMBLE
YES
DESCRAMBLE
SP 13
SP 14
TIME PAY PER VIEW
SP 21
PRGM PAY PAR VIEW
NO
UNIT TIME ELAPSED?
SP 15
YES
SP 16
CHARGE
PRGM UPDATED ?
SP 22
NO
YES
SP 23
CHARGE

0180460

FIG. 6



European Patent Office

EUROPEAN SEARCH REPORT

Application number

DOCUMENTS CONSIDERED TO BE RELEVANT

EP 85307832.7

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.4)
Y	WO - A1 - 83/04 154 (TELEASE)	1-4	H 04 N 7/16
A	* Abstract; fig. 1-4; page 9, line 12 - page 33, line 22; page 35, lines 3-14 *	6	
Y	EP - A2 - 0 052 822 (AVM SCHMELTER)	1-4	
	* Totality *		
A	US - A - 4 460 922 (ENSINGER)		

TECHNICAL FIELDS SEARCHED (Int. Cl.4)

H 04 N 1/00
H 04 N 7/00

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
VIENNA	24-01-1986	BENISCHKA

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

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

(11) EP 0 332 707 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
06.11.1996 Bulletin 1996/45

(21) Application number: **88907376.3**

(22) Date of filing: **22.08.1988**

(51) Int Cl.[6]: **G06F 17/30**

(86) International application number:
PCT/JP88/00832

(87) International publication number:
WO 89/02118 (09.03.1989 Gazette 1989/06)

(54) DATA PROCESSING APPARATUS AND EDITING APPARATUS USING THE SAME

DATENVERARBEITUNGSANORDNUNG UND DAMIT AUSGERÜSTETES AUSGABEGERÄT

SYSTEME DE TRAITEMENT DE DONNEES ET DISPOSITIF D'EDITION L'UTILISANT

(84) Designated Contracting States:
DE FR GB

(30) Priority: **28.08.1987 JP 216233/87**
28.08.1987 JP 216234/87
08.09.1987 JP 224276/87

(43) Date of publication of application:
20.09.1989 Bulletin 1989/38

(60) Divisional application: **94119237.9 94119239.5**

(73) Proprietor: **HONDA GIKEN KOGYO KABUSHIKI KAISHA**
Minato-ku Tokyo 107 (JP)

(72) Inventors:
- **MIYOSHI, Akito**
  Iruma-gun Saitama 350-02 (JP)
- **TERAI, Hiromitsu**
  Sakado-shi Saitama 350-02 (JP)

(74) Representative: **Lehn, Werner, Dipl.-Ing. et al**
Hoffmann, Eitle & Partner,
Patentanwälte,
Postfach 81 04 20
81904 München (DE)

(56) References cited:
EP-A- 0 110 676
WO-A-86/05294
JP-A- 5 995 645
JP-A-59 214 966

- **I.E.E.E. SOFTWARE, vol. 4, no. 2, March 1987, New York, USA, pp. 4-14; S. GIBBS et al.: "MUSE: A Multimedia Filing System"**
- **I.E.E.E. COMPUTER SOCIETY, OFFICE AUTOMATION SYMPOSIUM, 29 April 1987, NAT. B. OF STANDARDS, GAITHERSBURG, pages 180-189; D. WOELK et al.: "Multimedia Applications and Database Requirements"**
- **COMPUTER & GRAPHICS, vol. 10, no. 2, 1986, Great Britain, pages 119-131; U. FLASCHE et al.: "Decentralized Processing of Documents"**
- **ENCYCLOPEDIA OF COMPUTER SCIENCE AND ENGINEERING by Anthony Ralston, 1983, page 496 and 786-787**
- **SMART TEXTVERARBEITUNG by Innovative Software, 1897, chapter "Passwort-1" to "Passwort-2" and "Kapitel1-3" to "Kapitel 1-4".**

Printed by Jouve, 75001 PARIS (FR)

EP 0 332 707 B1

Description

Technical Field

This invention relates to a data processing system. More particularly this invention relates to a data processing system which is capable of easy and efficient retrieval of data.

This invention also relates to an editing apparatus aided by the data processing system. More particularly this invention relates to an editing apparatus possessing protective functions controlled by Security No. and enabling handling manuals such as for automobiles and electric appliances, books in general, magazines, and newspapers (hereinafter referred to collectively as "books") to be edited quickly and easily.

Background Art

In recent years, the practice of performing the work of editing a book by the use of a personal computer or a work station (editing device) has been disseminated. In this case, the work of editing a book is carried out by newly feeding complete data on documents, illustrations, etc. for each of the pages, with the whole information of an edited book memorized as separated in units of page. With the conventional editing apparatus, however, those documents, illustrations, etc. which appear in two or more places cannot be utilized in common but must be repeatedly fed in all over. The conventional editing apparatus, therefore, takes up much time and labor in the production of a book. The production of a book, accordingly, is very expensive.

An editing apparatus may be characterized by attaching an element data key possessing a format common to various kinds of element data such as documents, illustrations, and photographs which jointly form one page of the book, to each of the various kinds of element data thereby enabling the data concerning the elements to be handled as common data, causing data of an edited book to be decomposed into data concerning layout and other data concerning elements for storage, and composing data for one page with the data concerning the layout and the other data concerning desired elements.

When an editing apparatus is so constructed as to operate on the principle just described, since the data for one edited page are enabled to be again decomposed into data concerning the layout and other data concerning the elements and put to storage. By again composing data for one page with the data concerning the layout and those concerning desired elements, therefore, the element data common to the plural pages can be used in common without requiring these data to be introduced newly or repeatedly.

When the editing of a book is enabled to be easily carried out as described above, however, there arises the possibility that an operator not authorized performs the work of editing a book or revises the contents of an already edited book. Further, since the various kinds of data forming a book are not only retained in the edited book but also memorized as element data, there ensues the possibility that the element data not yet disclosed to the public will be laid open to the public inspection without leave or even will be altered.

From Smart Textverarbeitung, SmartWare from Inovative Software, copyright 1984/1985/1986/1987, pages "passwort-1" to "passwort-2", page "Kapitel 1-3" to "Kapitel 1-4", a word processor is known in which a password can be allocated to a document. This password will always be requested by the program if an attempt is made to access a file through the commands of LOAD, PRINT, or READ.

From the Encyclopedia of Computer Science and Engineering, copyright 1983, editor Anthony Ralston, page 496 and pages 786-787, it is known that private files may be labelled with the name and the system identification number of a user and may even contain in their labels a password that must be matched against one provided by the user at the time of issuing the access request. Furthermore, a log-in procedure is described in which the system will request that the user enter an ID and a non-printing password.

Disclosure of Invention

This invention aims to solve the problem of alteration mentioned above.

This object is achieved by an editing apparatus having the features described in claim 1. An advantageous embodiment is given in the subclaim.

The security of a book and element data can be warranted by checking the security No. of an operator seeking access to the editing apparatus against the memorized security Nos. to determine whether this operator should be allowed to do so.

This description further provides processing means capable of efficient retrieval of data necessary for a new book from an already accumulated data base.

Generally, in the retrieval of data by a data processing apparatus, the wanted data are obtained by assigning titles (items) corresponding to various kinds of data registered in advance and, at the time of actual retrieval, introducing an item corresponding to a specific kind of data (hereinafter referred to as "data item") wanted into the data processing apparatus.

In a typical conventional data retrieval means, the items mentioned above are registered in advance in a hierarchical structure, the concepts of the items are sequentially defined by the use of input means for data items, and a particular data item which conforms to the item finally defined from a major concept is singled out to permit retrieval of the data required.

To be specific, when the retrieval of data is started, the items as major headlines appear on such a display as CRT. When the operator selects from among the dis-

played items one having the contents of a pertinent data item, the second items resulting from further division of the contents of the selected item appear as medial headlines on the display.

When the operator selects from among the newly displayed items one specific item having the contents of the data item, the third items resulting from further division of the contents of the selected item are exhibited as minor headlines.

Thereafter, the processing described above is repeated until the item defined sequentially from the major concept conforms to the data item and the item so conforming is selected and used for the retrieval of necessary data.

Another typical conventional data retrieval means permits direct introduction of a pertinent data item to be used for retrieval of necessary data. To be specific, indexes are allocated one each to all the species of data registered in the data processing apparatus. Thus, the data processing apparatus permits retrieval of necessary data by exhibiting all the items and corresponding indexes on the display and consequently enabling the operator to inject the index corresponding to the necessary data through such input means as a keyboard.

Now, the various conventional techniques will be described further in detail as applied to the retrieval of such information as population, area, etc. of the cities in Japan.

For example, the use of the conventional technique in finding the population of Shinjuku Ward, Tokyo will be explained.

(1) In the typical conventional data retrieval means which defines the major concept sequentially until conformity with a pertinent data item, when the data processing is started, the names of 47 prefectures (including Metropolis and Hokkaido) are exhibited as items of major headlines on the display. At this point the operator selects from the 47 names Tokyo Metropolis as an item embracing Shinjuku Ward being sought.

Subsequently, on the display, the names of 23 wards, cities, towns, and villages forming Tokyo Metropolis are exhibited as a medial headline. At this point, the operator selects from the displayed names Shinjuku Ward as an item being sought.

Then, the information on population, area, and number of households is exhibited as a minor headline on the display. At this point, the population in the items obtained by the sequential definition of the major concept and exhibited as a minor headline conforms to the population of Shinjuku Ward as the desired data item. By singling out the population from the minor headline mentioned above, therefore, the data on the population of Shinjuku Ward are obtained.

(2) In the other typical conventional data retrieval means which has indexes allocated one each to all the species of data, when the data processing is executed, all the items and corresponding indexes are exhibited on the display and, at the same time, a message announcing that the index corresponding to the percinent data item should be introduced is exhibited. At this point, the operator is able to get the data on the population of Shinjuku Ward by introducing the index corresponding to the population of Shinjuku Ward. Where the desired item of data happens to be the population of Shibuya Ward, the operator is allowed to take the data on the population of Shibuya Ward by introducing the index corresponding to the population of Shibuya Ward.

The conventional techniques described above, however, have the following problems.

(1) In the conventional data processing apparatus which sequentially define the major concept until conformity with the pertient data item, the retrieval of data is invariably started with the display of the item of the most significant concept. Where the number of steps to be taken until the major concept sequentially defined conforms to the pertinent data item is large, the efficiency of the operation of the apparatus is diluted because the time and labor spent for these steps is increased. Moreover, since the items to be selected are exhibited on the display in all steps to be involved, an operator of small experience is readily accustomed to the operation of the apparatus but an operator of large experience finds only a small part to manifest the outcome of his learning. An increase in the accumulation of the operator's experience, therefore, cannot be expected to bring about any improvement in the efficiency of the operation of the apparatus.

(2) In the second typical conventional data processing apparatus mentioned above, since indexes are allocated one each to all the species of data, the retrieval of data compels the operator to introduce the pertinent index corresponding to the particular species of data. The search for the pertinent item in the items exhibited on the display is an irksome task. Where the data are so voluminous as to require the items and indexes to be exhibited as split into a plurality of sections the size of the screen of the display, the operator is compelled to scroll the screen of the display. Thus, there arises the possibility that the necessary data are not obtained very easily.

To overcome the problems mentioned above, a data processing apparatus is contemplated which is characterized by comprising data retrieval table memory means for memorizing titles and indexes in a hierarchical structure therein, n'th rank index input means for the introduction of the n'th rank index, data selection means for reading out of the data retrieval table memory means

the (n+1)'th rank data corresponding to the n'th rank index, data existence discrimination means for making a judgment as to the presence or absence of the (n+1)'th rank data in the data retrieval table memory means, and element data reading means for reading pertinent data out of the element data memory means when the absence of the (n+1)'th rank data is confirmed.

As the result, when the n'th rank index is introduced through the index input means, the (n+1)'th rank data are exhibited when the presence of the (n+1)'th rank data in the data retrieval table memory means is confirmed and, to the contrary, the data corresponding to the n'th rank index memorized in the element data memory means are displayed on the display means when the absence of the (n+1)'th rank data mentioned above is confirmed.

The introduction of the n'th rank index, therefore, brings about an operational effect of permitting the start of the retrieval of data from a desired rank.

Brief Description of Drawing

Fig. 1 is a functional block diagram for illustrating a basic arrangement of an embodiment ;
Fig. 2 is a schematic block diagram of the first embodiment;
Fig. 3 is a flowchart for representing a first editing method according to the first embodiment ;
Fig. 4 illustrates a format of the basic layout code;
Fig. 5 represents a data format of one data element when image data or document data is newly entered;
Fig. 6 illustrates a data format of one specinformation element among machine sort information;
Fig. 7 illustrates a data format of one data element of service data information among machine sort information;
Fig. 8 indicates a data format of one torque information element among machine sort information;
Fig. 9 illustrates a data format of one image information element which is converted during the editing operation;
Fig. 10 represents only element data in one element data of the service data information shown in Fig. 7;
Fig. 11 represents a file for editing a new book;
Fig. 12 illustrates how the converted layout code is produced from the basic layout code;
Fig. 13 is an illustration of one example of a display screen of the basic layout;
Fig. 14 is an illustration of one example of a display screen during the editing operation;
Fig. 15 is an illustration for illustrating how a temporary key of one element data of newly input image information is converted into an element data key;
Fig. 16 is an illustration for illustrating how a temporary key of one element data of newly entered document information is converted into an element data key;
Fig. 17 is an illustration for illustrating a data format of one document information element which is converted during the editing operation;
Fig. 18 is an illustration for showing a data format of one page which is edited by the editing machine;
Fig. 19 is an illustration for representing a data format of one page which is converted by the host computer so as to be registered into the layout D/B 11;
Fig. 20 is a flowchart for representing a second editing method according to the first embodiment ;
Fig. 21 is an illustration for explaining how the data rewriting at the step S104 of Fig. 20 is performed;
Fig. 22 is a schematic diagram for showing a portion of a table for retrieving document data set in the document D/B 12;
Fig. 23 is a schematic diagram for representing a hierarchical structure of a document retrieval table:
Fig. 24 is a flowchart for illustrating a retrival method according to an embodiment ;
Fig. 25 is a graphic representation for showing relationships of a conversion section, an image section and service data on a CRT;
Fig. 26 is a functional block diagram of the first embodiment;
Fig. 27 is a flow chart illustrating a second procedure of editing.
Fig. 28 is a functional block diagram of an embodiment of the present invention.
Fig. 29 is a flow chart illustrating the first procedure of editing according to the embodiment of Fig. 28.
Fig. 30 is a flow chart illustrating the second procedure of editing according to the embodiment of Fig. 28.
Fig. 31 is a model diagram illustrating the condition of data controlled within the data control No. memory means.
Fig. 32, 33, and 34 are functional diagrams of another embodiment of this invention, respectively.
Fig. 35 is a flow chart illustrating a procedure of data retrieval.

Best Mode for Carrying Out the Invention

Now, the present invention will be described in detail below with reference to the accompanying drawings. Preparatory to the detailed description, the general construction and operation of the editing apparatus embodying the present invention will be described.

In figure 2, a plurality of editing apparatus (work stations) 3-1 to 3-N for inputing/outputing document, photographs, illustrations and the like are connected in an on-line mode to a system controller 2 for controlling these editing apparatus. These editing apparatus receive various information such as document, photographs and illustrations by way of the vector conversion.

A CRT (cathode-ray tube) 6 is connected to the editing apparatus 3-1, a printer 7 is connected to the editing

apparatus 3-2, a scanner 8 is connected to the editing apparatus 3-3, and a personal computer 9 and another CRT 10 are connected to the editing apparatus 3-N. Furthermore, keyboards 5-1 to 5-N are connected to the respective editing apparatus 3-1 to 3-N.

The system controller (a medium-scale relay computer) 2 is furthermore connected to a host computer 1.

A book data base (D/B) 21, an element data base 22 and a layout basic data base 23 are connected to the system controller 2.

The system controller 2 controls these editing apparatus 3-1 to 3-N, and transfers/receives the information in a page unit or an element (document, photograph, illustration) unit constituting one page to and from the respective editing apparatus.

The host computer 1 is a large-scale general purpose computer such as the IBM's computer, IBM 3090 or equivalent. The system controller 2, personal computers 4-1 to 4-N, layout D/B 11, document D/B 12, image D/B 13 and machine sort information D/B 14 are respectively connected to the host computer 1.

It should be noted that the personal computers 4-1 to 4-N may be exclusively used for the editing apparatus, or for the general purpose other than the editing work, which is similar to personal computers 31-1 to 31-N (will be discussed later).

In case that the editing apparatus 3-1 to 3-N possess a function capable of inputting a language (referred to as "a specific language") other than the language input by the personal computers 4-1 to 4-N, a software for inputting and processing the specific language by the personal computers 4-1 to 4-N is provided with, for instance, the system controller 2.

In addition, another software is provided with the system controller 2, by which the document input by the personal computers 4-1 to 4-N is vector-converted, as is similar to the document input by the respective editing apparatus 3-1 to 3-N.

The personal computers 31-1 to 31-N which are employed to carry out the work other than the editing work effected by the editing apparatus, are connected to the host computer 1. In other words, the machines or parts surrounded by a dot line shown in Fig. 2 do not constitute the editing apparatus of this invention.

The host computer 1 constitutes the data of one book from the information which have been stored in the data bases 11 to 14, and transfers this data to the system controller 2. The system controller 2 transfers the information of one page which has been edited in the editing apparatus to the host computer 1.

An operation of the data processing apparatus according to a preferred embodiment of the invention will now be described.

Basically, the editing apparatus employing the constructions as illustrated in Fig. 2 can produce, or edit a new book by way of two editing methods as follows.

(A) When anew book is newly produced at all, all of documents or sentences, illustrations, and photographs etc. to be written on each page of the new book must be newly input.

This will be referred as "a first editing method".

(B) When a new book is produced by utilizing other books which have previously been edited or produced in the editing apparatus, the necessary portions of other books are utilized for the new book and only the minimum portions required to produce the new book are newly input. This will be referred to as "a second editing method".

Referring now to Figs. 3 to 19, a description is made that a service manual (simply referred to as "a book" hereinafter) for a bike (auto-bicycle) is produced in accordance with the above-defined first editing method.

Fig. 3 is a flowchart for indicating a first editing method.

In Fig. 3, first as a step S1, a basic layout code is input into the editing apparatus to which CRT is connected (for instance, the editing apparatus 3-1) by the keyboard 5-1. The basic layout code is to set a size of a book to be newly produced, namely, a size, the number of character, a column number and a column space in one page of the new book.

Fig. 4 illustrates one example of the format of the basic layout code.

The basic layout code is constructed of a layout code having, for example, a 6-byte code length, layout data, and another data representing the data length of the layout data.

It should be noted that the data representing a data length of certain data is simply referred to as a "LEN" in the following description.

The layout code is constructed of the data for representing which product classification a book to be produced belongs to; the data for indicating which location (country) the new book is designated; the data representative of the sort of the new book, e.g., service manual, or shop manual; and the data representative of the layout of each page of the new book, e.g., a single frame or a double frame.

The layout data is constructed of the data representing a size of each page of the book, i.e., A6 vertical, A6 horizontal and so on; and the data indicative of a line number, a line space, the number of character, a point, style, presence of a ruled line.

The data concerning a concrete layout of a page on a screen of CRT is set in the layout data, and the layout code is merely an identification code of the layout data. The layout code is converted into a document identification number and a page number which correspond to the identification data of the new book, as will be described later in connection with the steps S29 and S30.

When entry of the layout data is accomplished, LEN of the layout data is added in front of the layout data.

As previously described in detail, the basic layout code is produced from the layout code, LEN of layout data, and layout data.

Referring back to Fig. 3, when the basic layout code

is generated, as illustrated in a step S2, the editing apparatus 3-1 transfers the basic layout code to the system controller 2 which registers the basic layout code in the layout basic D/B 23.

Upon completion of registering the basic layout code, in the next steps S3 to S24, entry of such information of illustrations, photographs, document and the like which are to be arranged in each page of the book is carried out.

It should be noted that both illustrations and photographs are referred to as an image in the following description.

In the step S3, a judgement is made whether or not the information to be input corresponds to the image. When the image is entered, as indicated in the step S4, a temporary key having, for instance, a 6-byte length is entered in the editing apparatus 3-3 by the keyboard 5-3. The temporary key corresponds to an identification code of image data to be entered which is determined by an operator himself. "D" indicating that this temporary key relates to the image data is entered into the head byte of the temporary key.

In the next step S5, the image is read out by the scanner 8.

In a step S23, the temporary key, LEN of the readout image data, and the image data are registered into the element D/B 22 as one element data.

If a judgement is made in the previous step S3 that no image is entered, another judgement is made in a step S6 whether or not document or sentence is entered. When the document (sentence) is entered, as indicated in a step S7, the temporary key is input in the editing apparatus 3-1 by the keyboard 5-1 or in any of the personal computer 4-1 to 4-N. As previously explained in the step S4, the temporary key is constructed of, for instance, a 6-byte length, and corresponds to an identificatin code of the document to be input which is determined by the operator himself. In this case, "T" indicating that the temporary key relates to the document key is input into the head byte of the temporary key.

In the subsequent step S8, the document or sentence is entered in the keyboard 5-1 or any one of personal computers 4-1 to 4-N.

Then, in the step S23, the temporary key entered in the previous step S7, LEN of the document entered in the step S8, and this document are registered as one element data in the element D/B 22.

In Fig. 5, there is shown one element data registered in the step S23.

In the step S6, if a judgement is made that no document is entered, it is decided that said input is related to the machine sort information.

The machine sort information includes the data on the repairing data, tools to be used and the like for a bike. In accordance with the preferred embodiment of this embodiment, the machine sort information includes the specification, service data and torque, which will be described in detail hereinafter.

(A) Specification is, for instance, the reparing data on the bike, and the data representative of the reparing items and details which are entered into a list indicating the data of the tool to be used, and is constructed of alphanumerical data and KANJI characters. In other words, this specification represents a type of an engine oil, a name of a tool to be used, and repairing items such as an inner diameter of a cylinder, and an outer diameter of a piston diameter, and the like.

(B) Service data is numerical data representative of, for instance, maintenance data which is entered into a list of the maintenance data, and constructed of three different data, i.e., a maximum value, a minimum value, and a limit value of usage (or a center value).

That is, as the maintenance data relating to the inner diameter of the cylinder, there are the typical maximum value and minimum value, and the usable maximum value. Also, as the maintenance data concerning the outer diameter of the piston, there are the typical maximum value and minimum value, and the usable minimum value. As to the data of a capacity of an engine oil tank, a capacity of a coolant and the like which has no discrimination in the maximum and minimum values, the same values as these maximum and minimum values are input, by which these data are registered.

(C) Torque is numerical data indicative of fasten torque of fastening screws for the bike (maximum value, minimum value, and center value), a diameter of a screw, the number of the required screws. That is to say, the torque information is constructed of five pieces of numeral data containing the maximum value, minimum value, center value (or the limit value for usage) and other two informations.

When the judgement is made that the input data corresponds to the machine sort information, the machine sort code is entered which represents that the machine sort information to be input is related to which sort of the bike, as indicated in the step S9, by utilizing any one of the personal computers 4-1 to 4-N (e.g., personal computer 4-1) connected to the host computer 1. This machine sort code is constructed of, for instance, 8 bytes, and "K" for representing that the machine sort code is related to the machine sort information is input into the head byte of the machine sort code.

In a step S10, a judgement is made whether or not the machine sort information to be input is the specification.

If a judgement is affirmative in the step S10, then, an information identification code having, for instant, a 4-byte length is input in a step S11. At this time, "A" for indicating that the information identification code is related to the specification is input at the head byte of the identification code.

In a next step S12, an item having, for example, a 3-byte length is entered.

In a subsequent step S13, an image classification or sort having, for instance, a 2-byte length is input. Data representing a language sort of the specification information (for example, the language is Japanese, or English) is entered into this image classification.

In a step S21, specification information is entered by the personal computer 4-1.

In a subsequent step S22, the machine sort code, information identification code, item and image classification which have been input by the previous steps S9, S11, S12, and S13, the data status (which is automatically set in the host computer 1) for indicating the history of the specification information entered in the step S21, and the specification information are registered as one element data in the machine sort D/B 14 by means of the host computer 1.

A data format of the one element data is illustrated in Fig. 6. As is shown in Fig. 6, the data consisting of the machine sort code, information identification code, item, image classification, and data status will be referred to as an element data key in the following description.

If no specification is input in the previous step S10, another judgement is made whether or not the service data is entered in a step S14.

If the service data is entered, the information identification code is input in a step S15, which is similarly done in the preceding step S11. In this case, "B" is input into the head byte of this code.

In the next step S16, the item is input, which is similarly executed in the previous step S12.

In the subsequent step S17, the image section is entered. The function of the image classification information entered at entry of the service data will be discussed later with reference to a step S42.

In the step 21, the service data information is input by means of the personal computer 4-1.

The service data information is constructed of three different data, i.e., the maximum value, minimum value and a limit value of usage (or a central value). When these three data are input in a specific unit in the editing apparatus according to the preferred embodiment, these data are stored and also converted into other units to be stored thereafter.

Fig. 7 illustrates a format of 1-element data (element data key and service data) which is registered in the machine sort information D/B 14 in the step S22.

As illustrated in Fig. 7, the service data is so constructed as to be set in a first data section through a third data section. In the step S21, when the three data are input in the preselected specific unit (for example "mm"), these data are set in the first data section. Although it is not shown in Figs. 3 and 7, just before entry of the service data information, a conversion classification (for example, 2-byte length data) for indicating what other unit three data set in the first data section should be converted into, with the result that the three data entered in the millimeter unit into the first data section are converted into other units (e.g., an inch unit) than millimeter unit, which are designated by the conversion classification, and thereafter stored in the second data section.

Then the three data in millimeter unit stored in the first data section are converted into still other unit designated by the conversion classification and the resultant converted data are transferred to the third data section.

Referring to Fig. 10, the above-mentioned conditions will now be described more in detail. Fig. 10 illustrates only the service data (element data) among the data shown in Fig. 7.

When the information on the inner diameter of the cylinder of the bike is entered as the service data, the minimum value of 56.003, the maximum value of 56.018, and the limit value of usage of 56.08 (illustrated in Fig. 10 as the service limit) are input in the millimeter unit. These numerical data are set in the first data section.

If the conversion classification set before entry of the service data is designated to convert the millimeter unit into the inch unit, each of the data set in the first data section is converted into 2.2048; 2.2054; and 2.207 inches, respectively and the converted data are set in the second data section.

Also in the third data section, each of the data set in the first data section is converted into still other unit in accordance with the designation of the conversion classification.

In other words, the function of the conversion classification is to automatically convert the service data into a certain unit other than the originally input unit when the service data is entered in the certain specific unit, and thereafter to set these converted data into the second and third data sections.

Next, in the step S22 as shown in Fig. 7, the machine sort code, information identification code, item and image classification which have been entered in the previous steps S9, S15, S16 and S17, the data status (which is automatically fixed in the host computer 1) for representing a history of the service data information entered in the step S21, and the service data information consisting of the first to third data sections are registered as the 1-element data in the machine sort information D/B 14 under the control of the host computer 1.

Then, if no service data is entered the judgement of the step S14 is negative and shows that the torque information is entered.

When the entry of the torque information is judged, the information identification code is entered in the step S18, which is similarly executed in the preceding steps S11 and S15. At this time, "C" is input in the head byte of the information identification code.

In the subsequent step S19, the item is input, which is similarly performed in the steps S12 and S16.

In the step S20, the image classification is entered. The function of the image classification which is entered during entry of the torgue information is similar to a func-

tion of an image classification which will be described later relating to a step S42.

In the next step S21, the torque information is input into the personal computer 4-1.

The torque information is constructed of five pieces of data containing the maximum value, minimum value, and center value (or a limit value of usage), and also two pieces of data representative of other information. In the editing apparatus, when these five pieces of data are input in the specific unit similar to the above-mentioned entry method of the service data, these data are stored, and at least three pieces of data other than two pieces of data to represent the other information are firstly converted into other units and secondly stored.

Fig. 8 illustrates a format of 1-element data (element data key and torque) which is registered in the machine sort information D/B 14 in the step S22.

As illustrated in Fig. 8, the torque data is so arranged as to be set in the first to third data sections. In the step S21, when the torque data is entered in some particular unit (for example, "kg"), this torque data is set in the first data section. Although not shown in Figs. 3 and 8, the conversion classification is input, just before entry of the torque information, so as to instruct that five pieces of data set in the first data section should be converted into a certain different unit. As a result, at least three pieces of data input in the "kg" unit into the first data section (i.e., three pieces of data other than two pieces of data representative of the other information among five pieces of data to indicate the torque information) are converted into a different unit (e.g.,"lb" unit) designated by the conversion classification except the "kg" unit. Then, the converted data is set in the second data section.

In addition, at least three pieces of data entered into the first data section in the "kg" unit are converted into still other unit which are designated by the conversion classification, and thereafter the converted data are set in the third data section.

In other words, the conversion classification is to automatically convert the torque data into the unit other than the above-described unit when the torque data is input in the certain specific unit, and also to set these converted data into the second and third data sections, in a manner similar to the case of the service data.

Since the data concerning the diameter and the number are not required to be converted by the unit, no unit conversion is needed for these two pieces of data. It may be possible, of course, to perform the similar unit conversion on these two data as same as in other three data. In this case, however, it is necessary to prohibit from using the converted two data.

In the 1-element data as illustrated in Figs. 6 to 8, a byte length of the respective element data key is identical to each other.

In the step S22, as shown in Fig. 8, the machine sort code, information code, item and image classification which have been input in the steps S9, S18, S19 and S20, the data status (which is automatically set in the host computer 1) representative of the history of the torque data which has been input in the step S21, and the torque data information consisting of the first to third data sections are registered as 1-element data by the host computer 1 into the machine sort information D/B 14.

In the steps S22 or S23, when entry of one-image, one-document, or one piece of the machine sort information (these are referred to as "element data" hereinafter) is accomplished, a judgement is made whether or not the entry of the element data is continued in the next step S24. If the entry of the element data is continued, the control process is returned to the previous step S3. If the entry of the element data is not continued, the editing work of the book is commenced from a step S25. The editing work is performed by one page unit.

In the step S25, a new book producing file is read out from the host computer 1 by any one of the personal computers 4-1 to 4-N connected to the host computer 1.

Fig. 11 illustrates an arrangement of the new book producing file. As shown in Fig. 11, the new book producing file is constructed of material supervision or ID No. information, staff-in-charge information, and starting page information.

The material supervision No. is a title of a book to be newly produced or the supervising number. The staff-in-charge information is the data representative of a staff who is qualified to produce the new book. The starting page information indicates what page the book to be produced is commenced from. In other words, the page information indicates the first page of the new book for the starting page. For instance, the second page corresponds to the starting page of a book distributed in Japan, whereas the eleventh page corresponds to the starting page of a book distributed in U.S.A.

The staff-in-charge information is able to be applied to the security supervision of a newly edited book, for example. That is to say, the information relating to a staff who is permitted to produce a new book is previously registered in the host computer 1, and the new book is edited by the staff only when he has been registered in the new book producing file in the host computer 1. Under this security supervision, it can prevent that the book which is not allowed to be edited is mistakenly edited or the previously edited book is unnecessarily and mistakenly revised by the editing apparatus.

In this case, the staff-in-charge information may be preferably encrypted or stored in a magnetic card.

When the new book producing file is read out, predetermined data is input into this file at a step S26. When the predetermined data is entered, as shown in a step S27, the new book producing file is transferred from the host computer 1 to the system controller 2 and then registered in the book D/B 21.

In the next step S28, the layout code (see Fig. 4) is entered by the keyboard 5-1, so that the basic layout code previously registered in the layout basic D/B 23 is

called up to CRT 6.

When the basic layout code is called up in the editing apparatus, both the document supervision No. and the page number to be edited are input. The document supervision number is a title or the supervision or ID number of the new book, which is similar to the document supervision No. of the new book producing file shown in Fig. 11.

After the document supervision No. and the page number to be edited are input in a step S29, the editing apparatus 3-1 converts the layout code among the basic layout code read in the step S28 into the document supervision No. and the page number which have been input in the step S29, as illustrated in a step 530 and Fig. 12.

Upon completion of the code conversion, the editing apparatus 3-1 stores the document supervision No., the number of page, LEN and layout data in the editing apparatus 3-1 in a step S31. These document supervision No., the number of page, LEN and layout data will be referred as "a converted layout code" in the following description.

In a step S32, the basic layout is displayed on CRT 6 by employing the converted layout code. As illustrated in Fig. 13, the basic layout is basically constructed of a contour 101 of a page to be edited and a document entry region 102. The basic layout shown in Fig. 13 is a double frame.

In a subsequent step S33, a judgement is made whether or not the image is displayed on CRT 6.

To display the image, the temporary key (see Fig. 5) is entered by the keyboard 5-1 in a step S34 so as to call up one element data, i.e., temporary key, LEN of the image data to be called up, and the image data from the element D/B 22, and display the image data called up on CRT 6.

In a step 535, by a mouse (not shown) connected to the keyboard 5-1, the image displayed on CRT 6 is moved to a desirable position on the basic layout.

Fig. 14 illustrates an image of the screen of CRT 6 at that moment, which is similar to the illustration of Fig. 13. As shown in Fig. 14, when the image is moved to the region denoted by "P", the upper left coordinates and lower right coordinates of the image are (X1, Y1) and (X2, Y2), respectively.

When the image is transferred in this way, the upper left coordinates (X1, Y1) and lower right coordinates (X2, Y2) on the basic layout are entered into the editing apparatus 3-1.

It should be noted that while the image is moved, the size of the image may be enlarged or reduced in accordance with the capability of editing apparatus. Only the image can be moved over the document input region 102.

In a step S36, as illustrated in Fig. 15, the temporary key among the 1-element data consisting of the temporary key, LEN of image data to be called up, and image data is converted into the element data key.

As illustrated in Fig. 9, the element data key is constructed of the machine sort code, information identification code, item, image classification, and also data status (which is automatically input in the host computer 1) representative of a history of image data, which is similar to the machine sort information including the specification information, service data information, and torque information.

The conversion operation of the temporary key into the element data key in the step S36, is practically performed when an editor has entered the element data key excepting for the data status. "D" is input in the head byte of the information identification code. A byte length of the element data key shown in Fig. 9 is identical to the respective byte lengths shown in Figs. 6 to 8.

In a step S37, both the 1-element data consisting of the element data key, LEN of the image data corresponding to the element data key and the image data, and the coordinates (X1, Y1), (X2, Y2) are registered in the editing apparatus 3-1.

In a step S45, another judgement is made whether or not the edition of one page is accomplished. If not yet accomplished, then the control process is returned to the step S33.

If it is judged in the step S33, that the image is not displayed on CRT 6, another judgement is made whether or not the document or sentence is displayed in a step S38.

When the document or sentence is displayed, in a step S39, the temporary key (see Fig. 5) is entered by the keyboard 5-1, and the one element data consisting of the temporary key, LEN of the document data to be called up, and the document data themselves is called up from the element D/B 22, and the document data are displayed on CRT 6.

By employing the mouse (not shown) connected to the keyboard 5-1, the document displayed on CRT 6 is moved to the desired position on the basic layout in a step S40.

When the document is moved to the position indicated by the symbol "Q" on the basic layout shown in Fig. 14, the upper left coordinates and the lower right coordinates of the document are (X3, Y3) and (X4, Y4), respectively.

After the document has been moved, both the upper left coordinates (X3, Y3) and the lower right coordinates (X4, Y4) of the document on the basic layout are input into the editing apparatus 3-1 (540).

In a step S41, as illustrated in Fig.16, the temporary key in the 1-element data consisting of the temporary key, LEN of the called up document, and document data is converted into the element key data. Formats of these element data key, LEN of the called up document data and document data are illustrated in Fig. 17.

As shown in Fig. 17, the element data key corresponding to the document data is constructed of the sentence supervision No. having a length of, for instance, 8 bytes; the detailed item code having a length

of, for example, 6 bytes; the language information having a length of, for example, 2 bytes; and the data status (which is automatically added by the host computer 1) indicative of a history of the sentence or document data.

The sentence supervision No. is the data to specify each document, and in the head byte of which "B" is input. The language information indicates what language the document data has been produced in, for example, Japanese or English. The detailed item code is a classification code of the document or sentence.

It should be noted that after the data status of the 1-element data shown in Fig. 17, appropriate number of blanks are added to make the byte length of the element data key equal to that of the respective element data keys as f shown in Figs. 6 to 9. In other words, each byte length of the element data keys corresponding to the document data, image data and machine sort information data is set to be identical with each other.

With employing the detailed item code, the editing apparatus can retrieve the document or sentence. This document retrieval operation will be discussed later in a step S114.

In a step S37, the element data key, LEN of the document data corresponding to the element data key, and document data are registered as the 1-element data together with the coordinates (X3, Y3), (X4, Y4) into the editing apparatus 3-1.

In a step 538, if a judgement is made that no document is displayed on CRT 6, it can be recognized that the machine sort information is to be displayed.

In a step 542, the element data key of the machine sort information to be called up (see Figs. 6 through 8) is input by the keyboard 5-1, whereby the machine sort information having 1-element data, i.e., element data key, specification information, service data information or torque information is called up from the machine sort information D/B 14 to the editing apparatus 3-1, and then the specification information, service data information or torque information is displayed on CRT 6.

In an another embodiment, the element data key is entered by the keyboard 5-1, and the machine sort information is directly called up via the host computer 1 and system controller 2 from the machine sort information D/B 14 to the editing apparatus 3-1. However, alternatively, employing any one of the personal computers 4-1 to 4-N, the machine sort information may be called up from the machine sort information D/B 14 to be temporarily registered into the element D/B 22. Thereafter, the element data key may be input by the keyboard 5-1, and the machine sort information may be called up via the system controller 2 from the element D/B 22 to the editing apparatus 3-1.

It should be noted that display conditions or modes of the service data information and torque information among the machine sort information data are determined in accordance with the image classification information in the element data key and the conversion classification (not shown) described with reference to the step S21.

As to the service data, for instance, as illustrated in Fig. 10, when the data set in the first data section is automatically converted to be set within the second and third data sections in accordance with the conversion classification information previously set, the service data is displayed on CRT 6 in the illustration format of Fig. 25 in accordance with the conversion classification and image classification information.

Fig. 25 is a table for representing one example of a relationship between the conversion classification, image classification and service data, which is displayed on the CRT.

In Fig. 25, if the conversion classification is "10" and the image classification is "10", only the data set in the first data section (see Fig. 10) is displayed on CRT 6 in the millimeter unit.

If the conversion classification is "10" and image classification is "20", the data set in the first data section is displayed in the millimeter unit, and subsequently, the data set in the second data unit is displayed within a parenthesis in the inch unit.

If the conversion classification is "20" and image classification is "10", only the data set in the second data is displayed in the inch unit.

When the conversion classification is "20" and image classification is "20", the data set in the second data is displayed in the inch unit, and subsequently, the data set in the first data unit is displayed within a parenthesis in the millimeter unit.

When the conversion classification is "40", the same data and unit display as in the conversion classification being "10" is performed, and its decimal point is not a period, but a comma.

As is similar to the steps S35 and S40, the data on the machine information is positioned on the basic layout and the coordinates thereof is input in a step S43.

In a subsequent step S44, the element data key and its coordinates information selected from the 1-element data, as illustrated in Figs. 6 to 8, are registered into the editing apparatus 3-1.

Not only the element data key, but also the respective element data together with the coordinates information thereof may be of course registered.

When a judgement is made that the editing work for one page is accomplished in the step S45, the format of one page data is converted into another format shown in Fig. 18 in a step S46. This conversion is executed in the editing apparatus for performing the editing work (the editing apparatus 3-1 in this case).

Fig. 18 illustrates a format of the data having one page information converted in the editing apparatus. In Fig. 18, the one page data converted in the editing apparatus is constructed of the following data:

(A) The converted layout data which has been converted in the step S30 (see Fig. 12)

(B) The coordinates entered in the previous steps

S35, S40 or S43, and the element data key of the 1-element data arranged in the region designated by said coordinates. When a plurality of element data (image data, document data, or machine sort information data) are arranged within one page, plural sets of the coordinates and element data key should be storaged for one page.

The coordinates and element data key are set subsequent to the converted layout data. A set of coordinates and element data key will be referred to as "a layout for the editing work" in the following description.

(C) The element data key of the element data arranged within one page, LEN of the element data, and the element data. When a plurality of element data are inserted in one page, plural sets of the element data key, LEN, and element data should be arranged for one page.

In other words, the number of the 1-element data constituted of the element data key, LEN and element data is same as that of a pair of the coordinates and element data key contained in the layout for the editing work.

Subsequent to the layout for the editing work, the element data key, LEN and element data are set.

When the element data is the machine sort information, as apparent from Figs. 6 to 8, since no LEN is present in front of the element data, of course, only the element data key and element data are set.

These data arranged behind the layout for the editing work will be referred to an element data group in the following description (see Fig. 18).

Although not shown in Fig. 18, an end code for indicating that one page data is completed is added at the end of the final element data.

When one page data is converted as illustrated in Fig. 18, the resultant converted one page data is registered via the system controller 2 in the book D/B 21 in the step 547.

In the next step S48, one page data which has been converted in the previous step S46 is transferred to the host computer 1.

In a step 549, the above-described one page data is duplicated and the format thereof is converted into that illustrated in Fig. 19 in the host computer 1. Fig. 19 illustrates one page layout information which is converted in the host computer 1 so as to be registered in the layout D/B 11.

The one page data transferred to the host computer 1 is distributed and stored in the respective D/B (i.e., layout D/B 11, document D/B 12, image D/B 13 and machine sort information D/B 14) connected to the host computer 1. The data conversion executed in the step S49 is performed so as to store in the layout D/B 11, only the information relating to the layout among the one page data.

As is obvious from the comparison between Figs. 18 and 19, in the step S49, both LEN and the element data have been removed from the element data group of the 1-page data which is set subsequent to the layout for the editing work.

In a next step S50, the data converted in the previous step S49 is registered in the D/B 11.

In a step S51, the element data group is copied by employing the 1-page data which has been transferred to the host computer 1 in the previous step S48.

- In a step S52, the data relating to the document data among the element data group is registered in the document D/B 12.

In a step S53, the data relating to the image data among the element data group is registered in the image D/B 13.

In a subsequent step S54, a judgement is made whether or not the edited 1-page data is to be printed out. If no printing operation is to be carried out, the control process is advanced to a step S57. If the printing operation should be performed, the control process is advanced to a step S55.

In a step S55, both the reference supervision No. and page are entered by using of the keyboard 5-2, so that the 1-page data registered in the book D/B 21 is called up into the editing apparatus 3-2.

In a step S56, the 1-page data is printed out by the printer 7.

In a step S57, a judgement is made whether or not the next page is produced. When the next page is produced, the control process is returned to the step S28. Conversely if the next page is not produced, the control process is accomplished.

In accordance with above-mentioned process (first editing method), a new book is newly produced or edited.

Referring now to Figs. 20 to 24, as is similar to the description on the first editing method, a description will be made of the second editing method applied to produce a service manual of a bike or the like.

Fig. 20 is a flowchart illustrating the second editing method

In the respective steps shown in Fig. 20, the same reference numerals as those in Fig. 3 indicate the same or similar processing operations.

According to the second editing method, in a step S25, a new book producing file is first called up which has been registered in the host computer 1, by employing any one of the personal computers 4-1 to 4-N (for instance, the personal computer 4-1) connected to the host computer 1. Since the arrangement of this new book producing file is identical to that shown in Fig. 11, the description thereof is omitted.

After the new book producing file is called up, predetermined data, i.e., the material supervision No. information, staff-in-charge information and starting page information are entered into the new book producing file in a step S26.

Next, as illustrated in a step S27, the new book producing file is transferred by the host computer 1 to the system controller 2 and registered in the book D/B 21.

In a step S101, referring to a book which has been produced in the editing apparatus (referred to as "an original book"), a selection is made that the pages of this original book are utilized to produce a new book. That is to say, the following page selection is carried out, as one example. The 21st to 30th pages of the first original book are available to produce the 1st to 10th pages of the new book, the 46th to 50th pages of the second original book are utilized to produce the subsequent 11th to 15th pages of the new book, and furthermore, the 31st to 40th pages of the first original book are usable to produce the succeeding 16th to 25th pages thereof.

Such a process is performed while an operator actually observes the original books.

In a next step S102, the material supervision No. and the page number of the original book are input by way of the personal computer 4-1, and 1-page data (see Fig. 19) is copied in the host computer 1 from the layout D/B 11.

In a step S103, a new material supervision No. and a new page number of a book to be newly edited are input into the personal computer 4-1.

In a step S104, both the document supervision No. and the page number in the 1-page data copied in the previous step S102 are rewritten by the new material supervision No. and page number of the new book input in the previous step S103.

Fig. 21 illustrates the data rewriting operation executed in the step S104.

The 1-page data of the original book which has been called up and copied in the step S102 is as shown in the upper portion of Fig. 21. In the step S104, as illustrated in the lower portion of Fig. 21, only the material supervision No. and page number among the above-described data are rewritten by those of the new book.

In the step S105, the 1-page data produced in such a way is stored in the host computer 1.

In a step S106, a judgement is made whether or not the rewriting operation of the material supervision No. and the page number is accomplished for the new book. In accordance with the previous example, a judgement is made whether or not all of the 21st to 30th pages of the first original book, the 46th to 50th pages of the second original book, and the 31st to 40th pages of the first original book have been converted into the 1st to 25th pages of the new book.

If one book is not yet rewritten, the control process is returned to a step S102. To the contrary, if the rewriting operation is accomplished, the control process is advanced to a step S107.

In the step S107, the element data or both LEN of the element data and element data corresponding to the element data key, among the 1-page data converted in the step S104 are called up from the respective D/B 12 to 14. These data are added after the respective element data keys of the 1-page data which are arranged in the rear of the layout for the editing work. That is to say, the 1-page data converted in the stepS104 is rearranged in a complete form as shown in Fig. 18.

In a step S108, a judgement is made whether or not all pages of the new book have been rearranged in a complete form. If all pages of the new complete book are not yet rearranged, the control process is returned to step S107. If all pages are converted, the control process is advanced to a step S109.

In the step S109, the data of the new complete book is transferred from the host computer 1 to the system controller 2.

In a step S110, all pages of one complete book are registered in the book D/B 21 by the system controller 2.

In a subsequent step S111, the new material supervision No. and page number of the new book are entered by the keyboard 5-1 into the editing machine 3-1.

In a step S112, the input page data of the new book is read out from the book D/B 21 to be registered in the editing apparatus 3-1.

Then, in a next step S113, the page called up from the book D/B 21 is displayed on CRT 6.

In a step S130, a judgement is carried out whether or not the editing operation of the 1-page data displayed on CRT 6 is executed. That is to say, a judgement is made whether or not the pages which have been copied from the original books for the new book in the steps S102 to S104 can be merely utilized without any further edition. If no editing work is required, the control process is advanced to a step 546. If the editing work is required, the control process is advanced to a step S114.

In the step S114, a retrieval operation by any one of the personal computers 4-1 to 4-N is performed whether or not the elements of the original book (including the original book selected in the previous step S101) can be utilized for editing the page of the new book displayed on CRT 6. The retrieval method will be discussed later with reference to Figs. 22 to 24.

In a step S115, a judgement is made whetheror not the element usable in the new book editing operation is found in the above-described retrieval operation.

If some relevant elements are found, the element data key of the retrieved relevant elements are input in the personal computer (for example, 4-1) in a step S116.

In a step S117, the 1-element data called up by the above retrieval operation is transferred to the system controller 2, and is registered in the element D/B 22 under the control of the system controller.

In a step S118, a judgement is made whether or not the retrieval operation is continued. If yes, then the control step is returned to the step S114.

If the element usable for editing the new book is not found in the step S115, the 1-element data retrieved in the step S144 is cleared in a step S132. It is, of course, not to clear the data stored in the D/B 12 to 14 at this stage.

In the step S119, a judgement is made whether or not the image data, document data or machine information data, namely element is newly input. If the element is not newly input, the control step is returned to the step

S118.

When the element is newly input, the control process is advanced to the step S3. Since the processes effected in the steps S3 to 523 are same as those denoted by the same reference numerals in Fig. 3, the descriptions thereof are omitted.

If the control process defined by the step 522 or S23 is completed, the control step is returned to the step S119.

In the step S118, if the retrieval operation is not continued, another judgement is made whether or not the image is displayed on CRT 6 in the step S33. If the image is displayed, the temporary key or element data key of the image to be displayed is input by the keyboard 5-1 in the step S120, and the predetermined image is read out from the element D/B 22 and then displayed on CRT 6.

In the next step S35, the image is moved to the desired position and the coordinates thereof are input.

In the step S121, a judgement is judged whether or not the temporary key is input in the previous step S120.

When the temporary key is input, it is converted into the element data key in the step 536. The data conversion has been described with reference to Fig. 3, so that no further description is made here.

If a judgement is done that no temporary key was entered in the step S121, or after the process of the step S36 is accomplished, the element data key, LEN of the image data corresponding to the element data key and the image data itself are registered as one element data into the editing apparatus 3-1 together with the coordinates input in the previous step S35.

In the step S33, if a judgement is made that the image is not displayed, another judgement is done whether or not the document or sentence is displayed in the step S38.

When the document is displayed on CRT 6, the temporary key or element data key to be displayed is entered by the keyboard 5-1 in a step S122, and the predetermined document is called up from the element D/B 22 to be displayed on CRT 6.

In a subsequent step S40, the above document is moved to the desired position and the coordinates thereof are entered.

In a step S123, a judgement is made whether or not the temporary key was input in the previous step S122. When the temporary key is input, it is converted into the element data key in the step S41. This key conversion is carried out as same as in the conversion effected in the preceding step S36.

If a judgement is done-that no temporary key was input in the step S123, or after the process effected in the step S41 is accomplished, the element data key, and both LEN and the sentence data of the document data corresponding to the element data key, are registered as one element data into the editing apparatus 3-1 together with the coordinates input in the preceding step S35.

If a judgement is made that no sentence was displayed in the preceding step S38, it is judged that the machine sort information is displayed, and the control process is advanced to a step S131.

In the step S131, the element data key of the machine sort information to be displayed is entered by the keyboard 5-1 and the predetermined machine sort information is called up from the element D/B 22 to be displayed on CRT 6.

In the next step S43, the above machine sort information is moved to the desired position and the coordinates thereof are input.

In a step S44, the element data key is registered together with the coordinates entered in the step S43 in the editing apparatus 3-1.

When the process in the step S37 or S44 is completed, a judgement is made whether or not the 1-page data displayed on CRT 6 has been edited. If not yet, the process is retumed to the step S33. If completed, then the process is advanced to a step S46.

In a step S46, the edited one page data is converted in the form as illustrated in Fig. 18 in the editing apparatus 3-1.

In the next step S124, the converted one page data is updated in the book D/B 21.

Since the processes defined in the steps S48 through S57 are the same as those denoted by the same reference numerals of Fig. 3, the explanations thereof will be omitted. If in the step S57, the judgement is performed to edit the next page, the control process is returned to the step S111.

It is obvious that although in the explanations with reference to Fig. 20, the retrieval operation was previously executed before the 1-page data was edited in the steps S33 and thereafter, the necessary data were read out of the D/B 12 through 14 to be registered in D/B 22, these processes may be performed during the editing work.

A detailed description will now be made in that the process of the above-described step S114 is carried out for the document data retrieval operation. The retrieval operation of the step S114 is performed by employing any one of the personal computers 4-1 to 4-N.

Fig. 22 is a schematic diagram showing a portion of a table for the document data retrieval which has been previously set in the document D/B 12. A symbol "0" indicated in the respective tables of Fig. 22 represents "null", or that no data is entered.

In the document D/B 12, as illustrated in Fig. 22, a plurality of document data retrieval tables each consisting of an index (referred to as an "ID" simply) and a title have been input and set. A byte length of the ID is set to be equal to that of the detailed item code (6 bytes in the preferred embodiment) in order to correspond to the detailed item code of the element data key which in tum corresponds to the document or sentence data shown in Fig. 17

In an a-table of Fig. 22, the different data such as

A, B, C, and so on, are respectively input only in the head byte (first byte) of ID each having a 6-byte length.

In a b-table of Fig. 22, the common data of "A" is input in the first byte of each ID, and the different data of "A, B, C" are respectively input in the second byte thereof.

In a c-table of Fig. 22, the common data of "B" is entered in the first byte of each ID and the different data of "A, B, C" are entered, respectively, in the second byte thereof.

In a d-table of Fig. 22, the common data of "AA" is input in the first and second bytes of each ID, and also the different data of "A, B, C" are entered, respectively in the third byte.

In an e-table of Fig. 22, the common data of "AB" is input in the first and second bytes of each ID, and the different data of "A, B, C" are entered, respectively, in the third byte thereof.

Similarly, in an f-table of Fig. 22, the common data of "AAA" is input in the first to third bytes of each ID, and the different data of "A, B, C" are entered, respectively, in the fourth byte thereof.

In a g-table of Fig. 22, the common data of "AAB" is input in the first to third bytes of each ID, and the different data of "A, B, C" are entered, respectively, in the fourth byte.

Thus, the respective tables for the document data retrieval operation set in the document D/B 12 are formed in a hierarchical structure as illustrated in Fig. 23.

In practice, each of the tables for the document data retrieval operation are formed in the following method. That is to say, in correspondence to the respective ID's in the table (a-table) of the first hierarchy or rank where the data has been input in only the first byte of each ID, each title (auto-bike, automobile, or special-purpose car etc.) representative of the first highest concept of the document data is input. Also, in correspondence to each of ID's in the tables (b and c-tables) of the second hierarchy or order where the data has been input in only the first and second bytes of each ID, another title (explanation, or notice etc.) representative of the second highest concept is input. Similarly, still other title (structure explanation, operating or handling manual etc.) representative of the third rank concept is input in the tables (d and e-tables) of the third hierarchy, and further title (twelfth month maintenance, or sixth month maintenance, etc.) indicative of the fourth rank concept is input in the tables (f and g-tables) of the fourth hierarchy. Also the data same as the ID representative of the lowest concept is previously input in the detailed item code of the element data key shown in Fig. 17. And then the retrieval operation of the document data is performed as illustrated in Fig. 24.

It should be noted that the lowest hierarchy data need not be set in the table of the sixth hierarchy, but may be set in the table higher than the sixth hierarchy.

Fig. 24 is a flowchart for illustrating the retrieval method indicated in the step S114. As previously described, the retrieval operation is carried out by any one of the personal computers 4-1 to 4-N.

In a step S150 of Fig. 24, first "n" is set to be 1.

In a step S151, ID of the n-th hierarchy or rank is input by, for instance, the personal computer 4-1.

In a step S152, the (n+1)th hierarchy data containing the n-th hierarchy ID in ID thereof is selected from the table for the document data retrieval operation.

In a step S153, a judgement is made whether or not the (n+1)th hierarchy data is present, namely, the (n+1) th hierarchy data is selected in the step S152.

If yes, then all titles of the (n+1)th hierarchy are displayed in the step S154. Then, after 1 is added to "n" in a step S155, the control process is returned to the step S151.

When a judgement is made that the (n+1)th hierarchy data is not present in the step S153, the control process is transferred to the step S156.

In a step S156, the 1-element data containing the n-th hierarchy ID in the datailed item code thereof (i.e., 1-document or 1-sentence data shown in Fig. 17) is called up via the host computer 1 from the document or sentence D/B 12 and displayed in the personal computer 4-1.

Consequently, when the table for the document data retrieval operation is constructed as in Fig. 22, and "A" is input as ID only to the first byte, the b-table of the second hierarchy or rank where "A" is set in the first byte is selected and displayed, as indicated by an arrow "P".

Then, when "AA" is input in the first and second bytes as ID, the d-table of the third hierarchy where "AA" is set in the first and second bytes is displayed, as denoted by an arrow Q.

Similarly, when "AAA" is entered as ID in the first to third bytes, the f-table of the fourth hierarchy where "AAA" is set in the first to third bytes is displayed, as shown by an arrow R.

According to the above-mentioned method, such a retrieval operation can be done to show that such documents or sentences as a twelfth month maintenance manual are registered with respect to the document of the structure explanation on, for example, the auto-bike.

Furthermore, when "AAAA" is input as ID into the first to fourth bytes for the retrieval operation, the list of the fifth hierarchy (not shown) where "AAAA" is set in the first to fourth bytes is displayed as denoted by an arrow S.

When such a retrieval process is performed for the list of the lowest hierarchy, a judgement can be made whether or not therequired document data is registered in the document D/B 12.

When the necessary data is retrieved, the retrieved data can be utilized to edit a new book if the retrieved data is registered in the element D/B 22 as previously described with regards to the step S117.

Although the above explanation was made on the retrieval operation of the document data, the image data or machine sort information may be similarly retrieved.

That is to say, if a plurality of retrieval tables as illustrated in Fig. 22, are set in the image D/B 13 and machine sort information D/B 14, and if ID's of the retrieval tables are input into the items (3-byte length; see Figs. 6 to 9) in the element data key of the image data and machine sort information, the above-described retrieval operation may be performed.

Although in the preceding description, the machine sort information and image were registered into the respective different D/B (i.e., the machine sort information D/B 14 and image D/B 13), they may be registered in a common D/B since the data of the machine sort information shown in Figs. 6 to 8 has the same format as that of the image data shown in Fig. 9.

Also when the image and document data were input in the previous embodiment, the temporary key was input first and converted into the element data key in the actual editing operation. However, the temporary key may not be input in advance but the element data key may be input at the beginning, which is similar to the entry of the machine sort information data.

According to the embodiment described above, since a new book can be formed by making use of an original book formed by the editing apparatus, the new book is produced efficiently.

Where a service manual or a shop manual covering a machine similar to what is covered in an already edited manual is to be newly produced, since the greater part of the former manual is available for the new production, the new manual can be produced very quickly and inexpensively.

Of course, the editing apparatus of the embodiment can be used for editing books in general, magazines, and newspapers as well as for producing service manuals and shopmanuals.

Incidentally, the second procedure of editing illustrated in Fig. 20 has been depicted as what requires pertinent data to be introduced and registered in the file for the production of a new book at the step S25 to S27. Said procedure, however, does not always require introduction of the material ID No. or the starting page of the file for the production of the new book, because the editing apparatus is so constructed that the data of such particulars are to be introduced at the step S103.

The processing at the step S150 illustrated in fig. 24 may be omitted. The retrieval of data can be started by the introduction of ID of a lower rank than the first rank.

Now, the construction of the first embodiment of this invention will be described. Fig. 32 is a functional block diagram illustrating the construction of this invention. In Fig. 32, the reference numerals which have equals in Fig. 2 denote identical or similar parts.

In Fig. 32, a keyboard 106 and 106A are similar in function to the keyboards 5-1 to 5-N and the keyboards of the personal computers 4-1 to 4-N illustrated in Fig. 2 and a scanner 107 to the scanner 8 illustrated in the dame drawing.

Similarly, second element memory means 110 is similar in function to the element D/B 22 of Fig. 2, second layout memory means 111 to the basic layout D/B 23, and page memory means 112 to the book D/B 21 of the same drawing.

In Fig. 32, the keyboard 106 is connected to basic layout code setting means 101, element input means 102, element data key setting means 103, and layout code conversion means 114.

The basic layout code setting means 101 sets the basic layout code (refer to Fig. 4) to be fed in at the step S1 of Fig. 3.

The element data key setting means 103 sets the element data key illustrated in Figs. 6 to 9 and Fig. 17.

The element input means 102 receives convert the document data fed in through the keyboard 106 and the image data fed in through the scanner 107.

The layout code conversion means 114 converts the "layout code" part of the basic layout code introduced through the basic layout code setting means 101 into the material ID No. and page.

The basic layout code setting means 101 is connected to the second layout memory means 111 in the memory means 111. The second layout memory means 111 is connected to the layout code conversion means 114. The element input means 102 and the element data key setting means 103 are connected to the second element memory means 110 in the memory means 113.

The second element memory means 110 is connected to coordinates setting means 104, first page data setting means 105, and second page data setting means 133. The coordinates setting means 104 fulfills the role of feeding the element data key supplies from the second element memory means 110 and the element data corresponding to the element data key to display means 109 and, at the same time, setting the coordinates of the element data to be moved by the of manipulation of a mouse 108, and feeding the information of the coordinates to the display means 109, the first page data setting means 105, and the second page data setting means 133.

The layout code conversion means 114 calls out a desired one of the basic layout codes stores in the second layout memory means 111, converts the basic layout code into such a converted layout code as illustrated in the lower part of Fig. 12, and supplies the converted layout code to the first page data setting means 105 and the display means 109.

The first page data setting means 105 which is actuated when a new book is to be produced wholely newly, serves to prepare such data for one page as illustrated in Fig. 18 by using the element data key and the element data supplied from the second element memory means 110, the converted layout code supplied from the layout code conversion means 114, and the information on the coordinates supplied from the coordinates setting means 104, and supply the produced data for one page to the page memory means 112 in the memory means

113.

The element input means 102 is connected with a broken line to the display means 109, indicating that the data concerning documents, for example, are exhibited on the display means 109 when such documents as letters and sentences are to be introduced through the keyboard 106.

The page memory means 112 is connected to layout conversion means 121 and element conversion means 122.

The layout conversion means 121 prepares such information on layout as illustrated in Fig. 19 by removing the LEN and element data from the group of element data in the one page data (Fig. 18) memorized in the page memory means 112. The layout conversion means 121 is connected to first layout memory means 125 which is in turn connected to information processing means 124. The first layout memory means 125 is a device for memory which corresponds to the layout D/B 11 illustrated in Fig. 2.

The element conversion means 122 copies a group of element data from the data for one page memorized in the page memory means 112 to decompose into units of element. The element conversion means 122 is connected to the first element memory means 123 which is connected to the information processing means 124. The first element memory means 123 is provided with a document D/B 12, an image D/B 13, and a machine sort D/B 14.

The first layout memory means 125 is further connected to page discrimination data alteration means 131. The page discrimination data alteration means 131, when a book is to be produced by utilizing the information on another book already edited by the present editing apparatus and memorized in the first layout memory means 125 and the first element memory means 123, reads necessary part of the layout information memorized in the first layout memory means 125 and converts the material ID No. and page number in the layout information (namely, the page discrimination data for identifying the page of the layout information) into the material ID No. and page number of the new book to be produced. The data necessary for said conversion are introduced through the keyboard 106A.

The page discrimination data alteration means 131 and the first element memory means 123 are connected to element data addition means 132. The element data addition means 132 reads out of the first element memory means 123 the element data corresponding to the element data key being set after the layout on editing in the layout information (Fig. 19) which is supplied from the page discrimination data alteration means 131 and adds the element data to the element data key.

The element data addition means 132 is connected to the page memory means 112 which is further connected to the second page data setting means 133 and the display means 109.

The second page data setting means 133 which is actuated when a new book similar to an already edited book is to be produced based on the edited book, effects modification of a part of the information for one page supplied from the second element memory means 112, based on the data in the element unit supplied from the second element memory means 110 and the positional coordinates of the element data supplied from the coordinates setting means 104. Thereafter, it feeds back the modified information for one page to the page memory means 112. The second page data setting means 133, when the data in element units are not supplied from the second element memory means 110, causes the information for one page conveyed from the page memory means 112 to be returned in their unmodified form to the page memory means 112.

The first element memory means 123 is further connected to the second element memory means 110.

The basic layout code setting means 101, the element input means 102, the element data key setting means 103, the coordinates setting means 104, the first page data setting means 105, the second page data setting means 133, and the layout code conversion means 114 are severally provided for the editing devices 3-1 to 3-N or the system controller 2 illustrated in Fig. 2. The layout conversion means 121, the element conversion means 122, the page discrimination data alteration means 131, and the element data addition means 132 are provided severally for the host computer 1, the system controller 2, or the personal computers 4-1 to 4-N illustrated in the same drawing.

The embodiment of Fig. 32 has a function to be fulfilled in wholely new production of a new book and a function to be fulfilled in the production of a book similar to an already edited book, based on that edited book.

First, the operation of the editing apparatus for the wholely new production of a book will be described in detail below.

In the first place, information for designating a basic layout for one page is introduced through the keyboard 106. In the basic layout code setting means 101, the information is recomposed in such a basic layout code as illustrated in Fig. 4. The basic layout code is memorized in the second layout memory means 111.

The introduction and memorization of the basic layout may be carried out a plurality of times when necessary.

Then, the element data key corresponding to the element data to be introduced is fed in through the keyboard 106. In the element data key setting means 103, the element data key is set in such a form as illustrated in Figs. 6 to 9 and Fig. 17.

After the introduction of the element data key, information on documents or machine sort is fed in through the keyboard 106 or images are taken in through the scanner 107. The data thus introduced are fed out to the element input means 102. In the element input means 102, an LEN for the element data is attached when necessary to the element data. The LEN-attached element

data are fed out to and memorized in the second element memory means 110 in combination with the element data key. The introduction and memorization of the element data key and element data may be carried out a plurality of times as occasion demands.

Then, the necessary part of the basic layout code memorized within the second layout memory means 111, namely the basic layout code in which the layout data for a page to be edited are set, is read out of the second layout memory means 111 by the introduction of the layout code through the keyboard 106 and then fed out to the layout code conversion means 114.

In the layout code conversion means 114, the "layout code" part of the basic layout code is converted into the material ID No. and page introduced through the keyboard 106. As the result, a converted layout code (as illustrated in the lower part of Fig. 12) is produced. The converted layout code is fed out to the first page data setting means 105 and the display means 109.

From the second element memory means 110, either the element data to be used for editing a page or the LEN and the element data are fed out in conjunction with the element data key corresponding to the element data, to the first page data setting means 105 and the coordinates setting means 104. The coordinates setting means 104 supplies the element data to the display means 109.

When the element data exhibited on the display means 109 are moved by the mouse 108, the coordinates corresponding to the position assumed by the movement are set by the coordinates setting means 104. The coordinates are transferred to the first page data setting means 105.

Optionally, a plurality of sets of such element data are displayed at pertinent positions on the basic layout, to effect the editing of a page.

When the work of editing one page is completed, such data for one page as illustrated in Fig. 18 are produced in the first page data setting means 105, based on the output signals of the layout code conversion means 114, the second element memory means 110, and the coordinates setting means 104. To be more specific, the data for one page are composed of the converted layout code fed out of the layout code conversion means 114, the layout on editing consisting of the positional coordinates fed out of the coordinates setting means 104 and the element data key of the element data corresponding to the positional coordinates, and the group of element data consisting of the element data key forming the layout on editing and the element data corresponding the element data key.

The one page data produced in the first page data setting means 105 are memorized in the page memory means 112. In the page memory means 112, as many pages of edited data as are required for completing the book are memorized. The data are fed out to hard copy forming means (not shown).

The data for one page memorized in the page memory means 112 are transmitted to the layout conversion means 121. In the layout conversion means 121, such layout information as illustrated in Fig. 19 is produced from the data for one page. The layout information is transmitted to and memorized in the first layout memory means 125.

The data for one page memorized in the page memory means 112 are transmitted to the element conversion means 122. In the element conversion means 122, the group of element data is copied from the data for one page, decomposed into units of element, and transmitted to the first element memory means 123. Of the decomposed group of element data, those concerning the element of document are memorized in the document D/B 12 in the first element memory means 123, those concerning the image element in the image D/B 13, and those concerning the element of machine sort information in the machine sort information D/B 14, respectively.

The element information memorized in the first element memory means 123 and the layout information memorized in the first layout memory means 125 are called out, when necessary, by the information processing means 124 to be utilized for the various information processing and retrieval to be executed by the information processing means 124.

As the result, a new book is completed. The data on the new book are decomposed into those concerning layout and those concerning element and memorized.

Now, the operation of the editing apparatus in the production of a book similar to the book edited as described above, based on the edited book, will be described below.

The operator personally examines the already edited books to find which books and which pages of the books are available for a new book, feeds the material ID Nos. and pages (page identification data) of the books to be utilized, and reads out the data of the selected pages from the first layout memory means 125. The data thus read out are transmitted to the page discrimination data alteration means 131.

When the material ID No. and pages of the book to be produced are fed in through the keyboard 106, the material ID No. and pages of the data so read out are converted in the page discrimination data alteration means 13 into the material ID No. and pages so introduced.

The element data addition means 132 reads out of the first element memory means 123 the "element data" corresponding to the element data key set after the layout of editing in the converted data for one page and suffixes them to the element data key. This step complete the data for one page.

The data for one page are memorized in the page memory means 112, then transmitted to the display means 109 for display thereon, and also transmitted to the page data setting means 133.

The operator examines the data for one page ex-

hibited on the display means 109 to find whether the data are in need of any alteration or not. When the absence of necessity for data alteration is confirmed, the information announcing this fact is fed out from suitable means (not shown) and the data are transmitted to the page memory means 112. In the layout conversion means 121 and the element conversion means 122, the data are decomposed into those concerning the elements and those concerning the layout. In this case, since the data concerning the elements have been already memorized in the first element memory means 123, only the data concerning the layout are fed out to and memorized in the first layout memory means 125.

When the desired "element data" part of the data for one page exhibited on the display means 109 is to be altered, namely when the data for one page exhibited on the display means 109 are to be revised, the element information comprising the element data key and the element data is read out of the first element memory means 123 and registered in the second element memory means 110 by introducing the element data key corresponding to the required element data.

When the element data required for a new book to be edited are not found in the first element memory means 123, the element data key and the element data are newly introduced and registered in the second memory means 110.

The second element memory means 110 is urged by means (not shown) to supply necessary element data and element data key to the coordinates setting means 104 and the second page data setting means 133. The coordinates setting means 104 supplies the element data to the display means 109, sets the coordinates of the element data fixed by the manipulation of the mouse 108, and supplies the coordinates to the second page data setting means 133.

The operations described above are repeated when necessary. When the one page data exhibited on the display means 109 have been altered (edited), the second page data setting means 133 causes the data for one page transmitted from the page memory means 112 to be reset into edited data for one page. To be more specific, the second page data setting means 133 erases from the data for one page transmitted from the page memory means 112, the element data superposed with the element data set newly on the layout by editing, the element data key corresponding to the element data, and the positional coordinates thereof and, at the same time, supplements the data for one page with the element data set newly on the layout by editing, the element data key corresponding to the element data, and the positional coordinates of the element data.

The data for one page altered as described above are supplied from the second page data setting means 133 to the page memory means 112 and memorized therein. The data for one page are supplied in unit of page or book to the layout conversion means 121 and the element conversion means 122. The layout conversion means 121 and the element conversion means 122 decompose the supplied data into those concerning the elements and those concerning the layout to memory, respectively, in the first layout memory means 125 and the first element memory means 123. In this case, those of the data concerning the elements which have been already memorized in the first element memory means 123 are not doubly memorized.

When the operation described above is performed in the production of a book similar to an already edited book, since common element data can be utilized in their unmodified form in the production of the new book, there is obtained a generous saving in the time required for the production of the book.

The embodiment of Fig. 32 has been depicted as so constructed that the basic layout code is preparatorily set in the basic layout code setting means 101, the basic layout code in which the page layout for the page to be edited is set is called out by the use of the layout code, before the editing work is started, and the "layout code" part of the basic layout code is converted into the material ID No. and page in the layout code conversion means 114. This construction is convenient because the layout of a page to be edited (namely the layout data illustrated in Fig. 4) is not required to be set each time a new page is edited.

The embodiment described above is not specifically restricted to those construction. Optionally, the construction may be modified so that the converted layout code illustrated in the lower part of Fig. 12 may be directly set by introducing the material ID No. and page during the introduction of the layout data. In this case, since the converted layout code fixes the layout of only the page to be edited, the converted layout code is not required to be memorized in the second layout memory means 111. The modified construction is not very suitable for the work of editing a book of a large number of pages because the layout code must be set each time a new page is to be edited. It is nevertheless simple as compared with the construction of the embodiment illustrated in Fig. 32.

The embodiment illustrated in Fig. 32 has been depicted as so constructed that, of the data on the elements decomposed by the element conversion means 122, those concerning the image are memorized in the image D/B 13 and those concerning the information on the machine sort in the machine sort information D/B 14. The present embodiment is not required to be specifically restricted to those construction. The construction may be modified so that either of the image D/B 13 and the machine sort information D/B 14 is installed and the data concerning the image and the data concerning the machine sort are both memorized therein. Since the data concerning the image and the data concerning the machine sort are each furnished with element data keys of one and the same format as illustrated in Fig. 9 and Figs. 6 to 8, these data can be handled in the common identical form.

Further, when the element data key of the data concerning the documents illustrated in Fig. 17 is written in the same format as that of the data concerning the image and the machine sort information, the data concerning the documents can be memorized in the D/B intended for the memorization of the data concerning the image and the machine sort information.

Further the embodiment mentioned above has been depicted as so constructed that the data for one page produced by the first page data setting means 105, the second page data setting means 133, and the element data addition means 132 are memorized in the page memory means 112. When the page memory means 112 is so adapted as to permit memorization of the data for a plurality of pages and the host computer 1 in Fig. 2 is endowed with the functions of the layout conversion means 121, the element conversion means 122, the page discrimination data alteration means 131, and the element data addition means 132, the otherwise inevitable temporary concentration of load upon the host computer 1 is precluded as mentioned below.

When the editing apparatus is not provided with the page memory means 112, since the host computer 1 is required to call out and preserve data each time the data for one page are fed out for the purpose of editing and printing, for example, or after the editing of one page is completed, it must supply data rather frequently to the first element memory means 123 and the first layout memory means 125. As the result, there are times when the host computer 1 is exposed to temporary concentration of load.

In contrast, when the editing apparatus is provided with the page memory means 112, since the data for all the pages of one book can be memorized wholely in the page memory means 112 preparatorily to the editing of the book by virtue of the page discrimination data alteration means 131 and the element data addition means 132, the host computer 1 can be used for processing all the data introduced from the various devices (such as the personal computers 31-1 to 31-N of Fig. 2) connected to the host computer 1 and installed for the purpose other than the editing by the editing apparatus until the processing of the data for the whole book such as editing and printing is completed. As the result, the operation of the host computer 1 with respect to the processing for the editing apparatus is appreciably decreased and, consequently, the load on the host computer 1 is dispersed and alleviated.

When the page memory means 112 has the data of pages of the book to be edited memorized therein, the work of editing the book can be performed even when the office is closed as on a holiday and the operation of the host computer 1 is stopped.

On the condition that inconveniences of the nature suggested above are tolerated, it may be naturally permissible to so modify the present embodiment that the data for one page output from the first and second page data setting means 105 and 133 are directly supplied to the layout conversion means 121 and the element conversion means 122 and the data for one page fed out of the element data addition means 132 are directly supplied to the second page data setting means 133 and the display means 109.

Further, the embodiment has been dipicted as so constructed that the element data key and the element data newly set by the element input means 102 and the element data key setting means 103 and the element data key and the element data supplied from the first element memory means 123 are memorized in the second element memory means 110. The otherwise possible concentration of load upon the host computer 1 can be precluded by installing the second element memory means 110 as described above and consequently allowing the data concerning the elements to be called out abundantly in advance from the first element memory means 123. When the inconveniences are tolerated, the installation of the second element memory means 110 may be omitted.

Further, the installation of the data processing means 124 illustrated in Fig. 32 is not critical.

As is clear from the description given above, the present embodiment brings about the following effect because it is so constructed that the data of a book edited by the editing apparatus itself are decomposed into data concerning the layout and data concerning the elements for storage therein and these data are then utilized for the formation of data for one page. For example, when a new book is to be produced and the new book and a book already edited by the editing apparatus have common documents and illustrations, for example, these common documents and illustrations are not required to be newly fed in and these data may be utilized for the production of the new book. Thus, the production of the new books is simplified and the operation is performed quickly.

With reference to Fig. 1, element memory means 161 memorizes the element data representing such images as documents, photographs, and illustrations and the element data key corresponding to the element data.

Layout memory means 162 memorizes the layout data, the LEN, and the layout code corresponding to the layout data.

Page memory means 163 memorizes the page data composed of material ID No. page, layout data, coordinates, and element data key as illustrated in Fig. 19.

Data memory means 169 which comprises the element memory means 161, the layout means 162, and the page memory means 163 is connected to editing means 160.

Editing information input means 167 is for introducing information necessary for the execution of the editing work. The editing means 160 carries out the editing work in accordance with the information fed in through the editing information input means 167.

The information required by the operator in executing the work of editing is supplied from the editing means

160 to the display means 168.

Security No. input means 166 is used by the operator ready for the operation of the editing means 160 to feed in his own security No.

The security No. memory means 165 memorizes the security Nos. of the operators who are authorized to operate the editing means 160.

Security No. discrimination means 164 examines the security No. fed in through the security No. input means 166 to determine whether said particular security No. is found among the security Nos. memorized in the security No. memory means 165.

Incidentally, the security No. discrimination means 164 and the security No. memory means 165 may be provided indiscriminatingly for any of the host computer 1, the system controller 2, and the editing devices 3-1 to 3-N.

Now, the operation of the second embodiment will be described more specifically with reference to Fig. 1, Fig. 26, and Fig. 27.

Fig. 26 is a flow chart of the second embodiment applied to the first procedure of editing mentioned above.

With reference to Fig. 1 and Fig. 26, when the operator wishes to produce a new book by the first procedure of editing, he is required to feed in the data on elements to register in the element memory means 161 at the step S201, and then input pertinent information serving as his own security No. through the security No. input means 166 such as a keyboard or a magnetic reader at the step S202. The pertinent information may be the data of a staff in charge who is responsible for the formation of a new book illustrated in Fig. 11.

When the security No. is fed in, the security No. discrimination means 164 examines the security No. at the step S203 to define whether said particular security No. is memorized in the security No. memory means 165 as the security No. of the operator authorized to product the new book by the editing apparatus.

When the absence of the security No. in the memory is confirmed, the security No. discriminating means 164 exhibits a message announcing that the operator is not authorized to operate the editing means 160 on the display means 168 formed of CRT, for example, at the step S204. At this point, the processing is immediately terminated.

Conversely when the presence of the security No. in the memory is confirmed, the protection lock controlled by the security No. is released and the operator is allowed to proceed to produce the new book.

When the protection lock is released, the operator begins to produce the new book by feeding in various kinds of data through the editing data input means 167.

The operator feeds in the material ID No. of the book to be newly produced at the step S205, and the page number intended to produced at the step S206.

Subsequently, the operator, for the purpose of setting the layout of the page designated, feeds in the layout code at the step S207.

When the layout code is fed in, the layout information designated by said particular layout code is called out of the layout memory means 162. At the step S208, the basic layout corresponding to the layout information is exhibited on the display as illustrated in Fig. 13.

Then, at the step 5209, the operator feeds in the element data key corresponding to the image he wishes to insert and consequently calls the image from the element memory means 161.

The image, at the step S210, is exhibited on the display. At this time, in which position on the basic layout the image is to be inserted is designated by the use of a mouse (not shown), for example, connected to the editing means 160.

When the position for the insertion of the image is fixed with the mouse, the coordinates (X1, Y1) of the upper left corner of the image and the coordinates (X2, Y2) of the bottom right corner of the image on the basic layout are fed into the editing means 160.

At the step S212, a decision is made as to whether the editing of one page has been completed or not. When the image to be inserted on the same page still remains, the processing retums to the step 209.

When the editing of one page is completed, such page data as illustrated in Fig. 19 are formed and memorized in the page memory means 163 at the step S213. Thereafter, the processing is moved to the step S214.

At the step S14, a decision is made as to whether the editing of one book has been completed or not. When one or more pages to be edited still remain, the processing retums to the step S207. When the editing of one book is completed, the processing is automatically terminated.

Next, the operation of the second embodiment which the operator performes in the production of a new book by the second procedure of editing will be described below. Fig. 27 is a flow chart illustrating the editing operation. In Fig. 27, the reference numerals which have equals in Fig. 26 denote identical of similar parts. The explanation of these parts, therefore, will be omitted here.

With reference to Fig. 1 and Fig. 27, when a new book is to be produce by the second procedure mentioned above, after the processing in the step S203 has been completed, material ID No. of the original book is fed at the step S220 and the page number desired to be copies out of the original book at the step S221, respectively.

At the step S222, the data for one page corresponding to the material ID No. and page number are called out of the page memory means 163 to be supplied to the editing means 160.

Subsequently the material ID No. of the new book is fed in at the step S223 and the page number of the new book at the step 5224, respectively.

At the step S225, the material ID No. and page of the data for one page of the original book memorized in

the editing means 160 at the step 5222 are rewritten into those of the new book fed in at the steps S223 and S224 as illustrated in Fig. 21.

At the step S226, the element data of photographs, illustrations, etc. inserted in the copied page are examined to determine whether they should be replaced with other element data or not. When the absence of necessity for the replacement is confirmed, the processing is moved to the step S213.

Conversely, when the necessity for the replacement is present, the element data are newly fed in at the step S227 and/or the editing processing similar to that in the first procedure of editing at the steps S209 - S212 is executed.

At the step S213, the data for one page rewritten at the step S225 or the data for one page edited at the step S227 are memorized in the page memory means 163 and, at the same time, the necessary part of the data concerning the elements is memorized in the element memory means.

As noted clearly from the description given above, in the second embodiment, the security control is effected by means of the operator's security No.

Now, the third embodiment which permits further elaboration of security control by the secret level of the memorized data and the security No. of the operator will be described below.

Fig. 28 is a functional block diagram illustrating the construction of an embodiment in accordance with the invention. In Fig. 28, the reference numerals which have equals in Fig. 1 denote identical or similar parts. The explanation of these parts, therefore, will be omitted here.

Editing information input means 172 serves for feeding in information necessary for executing the work of editing. Data ID No. input means 173 receives the material ID No. for the control or supervision of books to be edited or the element data key for the control of images such as photographs (hereinafter, the material ID No. and the element data key will be collectively referred to as "data ID No.").

Data ID No. memory means 170 memorizes the security No. of the operator and the data ID No. corresponding to the ng security No. as illustrated in Fig. 31. The operator owning the particular security No. is authorized to utilize the editing apparatus for the work of editing.

The editing means 160 is connected to the editing information input means 172 and adapted to perform the editing work in accordance with the information supplied from the editing information input means 172.

Data ID No. discrimination means 171 examines the data ID No. fed in through the data ID No. means 173 to determine whether it is memorized in the data ID No. memory means 170 as the data ID No. authorizing the operator owning the security No. fed in through the security No. input means or not.

Optionally, the data ID No. memory means 170 and the data ID No. discrimination means 171 may be provided indiscriminatingly in any of the host computer 1, the system controller 2, and the various editing devices 3-1 to 3-N.

Now, the operation of the embodiment will be described in detail below with reference to Fig. 28, Fig. 29, and Fig. 30. Fig. 29 is a flow chart illustrating the embodiment as applied to the first procedure of editing. In Fig. 29, the reference numerals which have equals in Fig. 26 denote identical or similar parts. The explanation of these parts will be omitted here.

With reference to Fig. 28 and Fig. 29, when a new book is to be produced by the first procedure of editing mentioned above, the operator at the step S202 feeds in his own security No. and the security No. discriminating means 164, at the step S230 in response to the input, examines whether or not said particular security No. is memorized in the data ID No. memory means 170 as the security No. for the control of the data ID No. When the absence of the particular security No. is confirmed, the processing is terminated at the end of the step S204.

When the presence of the security No. is confirmed, all of the data ID Nos. controlled by the particular security No., namely the group of data ID Nos. of the books which the operator is authorized to edit and inspect, are selected from among the data ID Nos. memorized in the data ID No. memory means 170 at the step S231.

When "A1234" is fed in as a security No., for example, the data ID No. discriminating means 171 selects the group of a total of five data ID Nos., specifically "B10" to "B16" and "NEW" as the data ID Nos. controlled by "A1234" as shown in Table 2 of Fig. 31.

The term "NEW" as used herein refers to the data ID No. which is assigned to the operator authorized to produce the new book by the use of the editing apparatus. It does not mean the presence of any data with the data ID No. "NEW".

At the step S205, the material ID No. of the book to be newly produced is fed in as the data ID No. thereof.

At the step S232, the decision is made as to whether or not the "NEW" is present among the group of data ID Nos. mentioned above and the material ID No. is present nowhere in the data ID No. memory means 170. When the result of the decision is in the affirmative, the operator is judged as authorized to produce the new book by the editing apparatus.

Conversely, when the "NEW" is not present in the data ID Nos. or when it is present and the corresponding material ID No. is present somewhere in the data ID No. memory means 170, the operator is judged as unauthorized to use the editing apparatus for the production of the new book.

At the step S232, when the operator is not judged as authorized to use the editing apparatus for the production of the new book, the processing is terminated at the end of the step S204 similarly to the second embodiment.

When the operator is judged as authorized, the

processing is carried through the steps S206 to S209, similarly to the first procedure of editing in the second embodiment.

At the step S233, the decision is made as to whether or not the element data key fed in as the data ID No. at the step S209 is found among the group of data ID Nos. selected at the step S231.

When "A1234" is fed in as the security No. and "B12" as the element data key in the case mentioned above, for example, it is judged that "B12" is memorized in the group of data ID Nos. as the element data key corresponding to the security No. "A1234" as shown in Table a of Fig. 31.

Conversely, when "B14" is fed in as the element data key, for example, it is judged that this particular element data key is not memorized in the group of data ID Nos. corresponding to "A1234".

When the absence of the element data key from the group of data ID Nos. is confirmed, the processing is terminated at the end of the step 5204. When the presence of the element data key is confirmed, the processing is carried through the steps S210 to S214, similarly to the first procedure of editing in the second embodiment.

Optionally, the procedure may be modified so that when the absence from the memory is confirmed at the step S233, the processing is retumed to the step S207.

Now, the operation of the embodiment to be performed by the operator in the production of a new book in accordance with the second procedure of editing will be described below. Fig. 30 is a flow chart of the operation of the third embodiment applied to the second procedure of editing. In Fig. 30, the reference numerals which have equals in Fig. 27 and Fig. 29 denote identical or similar parts. The explanation of these parts, therefore, is omitted here.

In the production of a new book by the second procedure of editing, substantially the same processing as that of the second procedure of editing in the second embodiment is carried out as illustrated in Fig. 28 and Fig. 30. Similarly to the first procedure of editing in the third embodiment illustrated in Fig. 29, after the judgment of the step 5230, the group of data ID Nos. corresponding to the security No. are selected at the step S231 and the judgment as to the introduced data ID No. is carried out at the step S232 and the step S233.

It is clear from the description given above that when the secret level of the data ID No. "B10" is higher than that of "B12", "B15", or "B16", namely when the books controlled by "B15" are more important in Fig. 31, the operator whose security No. is "A1234" is authorized to edit or revise all the books controlled by the four data ID Nos., "B10" to "B16" and produce a new book as illustrated in Table a of Fig. 31. The operator assigned the security No. of "A1235" is authorized only to edit the books controlled by the three data ID Nos. "B12", "B15", and "B16" as illustrated in Table b and is not authorized to produce a new book. The present embodiment, therefore, permits security control conforming to the secret level of the data.

The classification of data according to the secret level may be easily supplemented by the classification according to the contents of data to make the security control more effective.

Where the data ID Nos. "B10" and "B14" cover the data on illustrations and the other data ID Nos. that on documents, for example, the operator whose security No. is "A1236" is authorized to edit indiscriminatingly the data concerning the illustrations without reference to the secret level. Thus, the effective control is easily accomplished.

This arrangement proves to be advantageous when the editing system of this invention is used not exclusively by the employees of the owner' company but is used additionally by the employees of the related compainies including contractors for the purpose of inspection of necessary data.

The security control involved in the use of the editing apparatus for the editing work has been described. Optionally, this system may be modified so that the security control is effected when the editing apparatus is started. As the result, that access to the system for the inspection of books and element data can be controlled.

Fig. 33 is a functional block diagram of a data processing device adapted to permit efficient and quick retrieval of accumulated data to be used for editing.

With reference to Fig. 33, the data processing device comprised n'th rank ID input means 200 for the admission of an n'th rank ID, data retrieval table memory means 202 for memorizing tables illustrated in Fig. 22, namely the (n+1)'th rank data (ID and title) corresponding to the n'th rank ID, data selection means 201 for selecting the (n+1)'th rank data from the data retrieval table memory means 202, display means 206, element data memory means 205, data existence discrimination means 203 for making a decision as to whether or not the (n+1)'th rank data are selected by the data selection means 201, namely the data are memorized in the data discrimination table memory means 202 and exhibiting the (n+1)'th rank data on the display means 206 when the presence of the data in the memory means 202 is confirmed or supplying the n'th rank ID to element data read (out) means 204 (which will be described specifically hereinafter) when the absence of the data from the memory means 202 is confirmed, element data read out means 204 for reading the element data corresponding to the n'th rank ID from the element data memory means 205 to supply to the display means 206 and information processing means 207, and information processing means 207 for performing data processing other than retrieval.

First, the data selection means 201 decodes the n'th rank ID fed in through the n'th rank ID input means 200 and calls out of the data retrieval table memory means 202 the (n+1)'th rank data corresponding to the input n'th rank ID, namely the table comprising of the (n+1)'th

rank ID including the n'th rank ID in the ID thereof and the title corresponding to the (n+1)'th rank ID as shown in Figs. 22 and 23.

When the second rank ID "AA" is fed in as an n'th rank ID through the n'th rank ID input means 200, for example, the data selection means 201 reads the table d, shown in Fig. 22, which comprises the third rank ID's "AAA", "AAB", "AAC",... each including the second rank ID "AA" and titles corresponding one to one to said third rank ID's out of the data retrieval memory means 202 as (n+1)'th rank data.

The data existence discrimination means 203 performs the judgment as to whether or not the (n+1)'th rank data are present in the data retrieval table memory means 202.

In other words, the data existence discrimination means 203 performs the judgment as to whether or not the data selection means 201 has called out the (n+1)'th rank ID and the titles corresponding to the (n+1)'th rank ID. When the third rank ID, namely the (n+1)'th rank ID, is called out, the corresponding table d is exhibited on the display means 206.

Conversely when the absence of the ID is confirmed, namely when the absence of the third rank ID is confirmed, the second rank ID which is an n'th rank ID in this case is fed out to the element data read means 204.

The element data read means 204 decodes the n'th rank ID fed out of the data existence discrimination means 203, reads the element data corresponding to the n'th rank ID from the element data memory means 205 to supply to the display means 206 and the information processing means 207.

When the third rank ID which is an (n+1)'th rank ID in this case is absent, namely when the data constituting the table d illustrated in Fig. 22 are absent from the data retrieval table memory means 202, for example, the data existence discrimination means 203 supplies "AA" as the second rank ID to the element data read means 204.

The element data read means 204 decodes the second rank ID "AA", reads out of the element data memory means 205 such element data of sentences, illustrations, etc. as defined by the title "explanation" corresponding to the second rank ID "AA", and supplies the element data to the display means 206 and the information processing means 207.

As described above, the present embodiment permits the data retrieval to be carried out efficiently because the data retrieval can be started by the introduction of a lower ID than the first rank ID.

Now, another version of the data-processing apparatus which allowed further enhancement of the efficiency of data retrieval will be described below with reference to Fig. 2, Fig. 34, and Fig. 35.

Fig. 35 is a flow chart of another typical system of data retrieval according to the present invention. In Fig. 35, the steps with S-numerals which have equals in Fig. 24 denote identical or similar steps of processing.

The n'th rank ID is fed in through the personal computer 4-1, for example, at the step S151. At the step S152, the (n+1)'th rank data including the n'th rank ID in the ID thereof are selected from the data retrieval table by the host computer.

At the step S153, the decision is made as to whether or not the (n+1)'th rank data are present, namely whether or not the (n+1)'th rank data are selected at the step S152, by the host computer 1.

When the presence of the data is confirmed at the step S152, the processing moves to the step S157. When the absence of the data is confirmed, the element data corresponding to the n'th rank ID are called out of the document D/B 12 via the host computer 1 and exhibited on the personal computer 4-1 at the step S156.

At the step S157, the decision is made as to whether or not a plurality of sets each comprising of the (n+1)'th rank ID and the title corresponding to the (n+1)'th rank ID are present in the (n+1)'th rank data.

When the presence of the plurality of the sets mentioned above is conformed, the processing moves to the step S154. When the absence of the plurality of sets is confirmed, the processing moves to the step S158.

At the step S158, the "element data" part of the only one set of data corresponding to the (n+1)'th rank ID is called out of the document D/B 12 via the host computer 1 and exhibited on the personal computer 4-1.

At the step S154, all of the (n+1)'th rank data are displayed. At the step S155, n is increased by 1. Then the processing returns to the step S151.

Now, the procedure for data retrieval will be described more specifically with reference to Fig. 34.

Fig. 34 is a functional block diagram of still another embodiment of the present invention. In Fig. 34, the reference numerals which have equals in Fig. 1 denote identical or similar parts.

The data processing apparatus comprises:

(A) n'th rank ID input means 200 for introducing an n'th rank ID,
(B) data retrieval quality table memory means 202,
(C) data selection means 201 for selecting the (n+1)'th rank data from the data retrieval quality table memory means 202,
(D) display means 206,
(E) element data memory means 205,
(F) data existence discrimination means 203 for making the decision as to whether or not the (n+1)'th rank data are selected by the data selection means 201, namely whether or not the data are present in said data retrieval table memory means 202 and suplying said (n+1)'th rank data to data number discrimination means 208 (being described specifically hereinafter) when the presence of the data is confirmed or supplying the n'th rank ID to first element data read means 209 (being described specifically hereinafter) when the absence of the data is confirmed,

(G) first element data read means 209 for reading element data corresponding to the n'th rank ID from the element data memory means 205 to supply to display means 206 and information processing means 207,

(H) data number discrimination means 208 for making the decision as to whether or not a plurality of sets of data are present in the (n+1)'th rank data fed out of the data existence discrimination means 203 and exhibiting the (n+1)'th rank data on display means 206 when the presence of the plurality of sets of data is confirmed or supplying the (n+1)'th rank ID in only one set of data to second element data read means 210 (being described specifically hereinafter) when the absence of the plurality of sets of data is confirmed,

(I) second element data read means 210 for reading element data corresponding to the (n+1)'th rank ID from the element data memory means 205 to supply to the display means 206 and the information processing means 207, and

(J) information processing means 207 for performing information processing.

The first element data read means 209 has the same function as the element data read means 204 illustrated in Fig. 33.

In Fig. 34, the data selection means 201 decodes the n'th rank ID fed in through the n'th rank ID input means 200 and calls out of the data retrieval table memory means 202 the (n+1)'th rank data corresponding to the n'th rank ID, namely, a table composed of the (n+1)'th rank ID including the n'th rank ID in the ID thereof and the title corresponding to the (n+1)'th rank ID as illustrated in Fig. 22 and Fig. 23.

When the second rank ID "AA" is fed in as an n'th rank ID through the ID input means 200 such as a keyboard, for example, the data selection means 201 reads out of the data retrieval table memory means 202 a table composed of the third rank ID's "AAA", "AAB", "AAC", ... including the second rank ID "AA" and the titles corresponding to the third rank ID's as illustrated in Table d of Fig. 22.

The data existence discrimination means 203 performs the decision as to whether or not the (n+1)'th rank data are present in the data retrieval table memory means 202.

To be more specific, the data existence discrimination means 203 performs the decision as to whether or not the table composed of the (n+1)'th rank ID and the titles corresponding one to one to the (n+1)'th rank ID is called out by the data selection means 201 and supplies the table called out, the table d in this case, to the data number discrimination means 208 when the presence of the calling of the third rank ID corresponding to the (n+1)'th rank ID as described above is confirmed.

Conversely when the absence of the third rank ID is confirmed, the second rank ID "AA" as an n'th rank ID is supplied to the first element data reading means 209.

The data number discrimination means 208 performs the decision as to whether or not a plurality sets of data are present in the (n+1)'th rank data fed out of the data existence discrimination means 203 and exhibits the (n+1)'th rank data on the display means 206 when the presence of the plurality of sets of data is confirmed or supplies the (n+1)'th rank ID to the second element data reading means 210 when the absence of the plurality sets of data is confirmed.

To be more specific, the data number discrimination means 208 performs the decision as to whether or not a plurality sets of data each composed of the (n+1)'th rank ID and the titles corresponding one to one to the (n+1)'th ID are present in the table d whose presence has been confirmed by the data existence discrimination means 203.

Since three sets of third rank ID's "AAA", "AAB", and "AAC" including the second rank ID "AA" are present when the second rank ID "AA" is fed in as the n'th rank ID through the ID input means 200, in this example, the table d composed of the three sets of ID's and the titles corresponding to the three ID's is exhibited on the display means 206.

Conversely, when only one set "AAA" is present, the "AAA" is supplied to the second element data reading means 210.

The first element data reading means 209 decodes the second rank ID "AA" as an n'th rank ID fed from the data existence discrimination means 203, calls out of the element data memory means 205 the element data corresponding to the second rank ID "AA" to supply to the display means 206 and the information processing means 207.

The second element data reading means 210 decodes the third rank ID "AAA" as an (n+1)'th rank ID fed from the data number discrimination means 208, calls out of the element data memory means 205 the element data corresponding to the third rank ID "AAA" to supply to the display means 206 and the information processing means 207.

The data processing apparatus of the present embodiment constructed as described above permits the data retrieval to be carried out more easily because the element data corresponding to the relevant ID are fed out of the element data memory means and displayed when the data selection means 201 selects only one set of (n+1)'th rank data.

Though the present embodiment has been depicted as applied to an editing apparatus, it is naturally applicable to various data processing apparatuses of the ordinary run.

As clearly noted from the description given above, the embodiment of Fig. 33 enables the data retrieval to be made at a desired rank by the introduction of the n'th rank ID. In the case of the data whose index is perfectly remembered by the operator on account of high frequency of use, for example, a necessary part of the data

can be promptly retrieved by the injection of the index in its complete form. Where the index for the necessary part of the data is not completely known, the data can be retrieved from a desired rank by feeding in only the part of the index clearly known to the operator and then keeping under close inspection the titles on the display means.

Thus, the data processing apparatus allows the data retrieval to be carried out very efficiently when the index is perfectly known and even when the index is known only imperfectly.

Further in the case of the embodiment illustrated in Fig. 34 which is provided with the data number discrimination means, the first element data reading means, and the second element data reading means, the element data corresponding to the (n+1)'th rank ID are automatically displayed when there exists only one (n+1)'th rank data. The introduction of the (n+1)'th rank ID through the input means, therefore, is not required to be repeated for display. Thus, the data processing apparatus enjoys improved efficiency.

Industrial Applicability

As clear from the description given above, the first embodiment of the present invention attains the following effect because it enables the data of the book edited with the editing apparatus to be decomposed into data concerning layout and data concerning elements for storage in the D/B and allows data for one page to be composed by using the decomposed and memorized data.

When a new book to be produced and a book already edited with the editing apparatus have common documents, illustrations, etc, the production of the new book can be attained by utilizing such common data without requiring new introduction of the documents, illustrations, etc. Thus, the production of the new book can be accomplished simply and quickly.

In accordance with the second embodiment, the security control can be attained with high reliability because the protect function is manifested not only on such materials as books and pages which have been already edited but also on element data forming such materials. In accordance with the embodiment of the invention, the security control can be carried out more elaborately, depending on the position of the operator or the contents of data. Thus, the security control is attained with high efficiency.

The data processing apparatus permits the data retrieval to be carried out very efficiently when the index of data is perfectly known or even when the index is known imperfectly. The application of the data processing apparatus of this invention to the data retrieval from the data base in the editing apparatus, therefore, warrants highly efficient editing of a new book.

Claims

1. An editing apparatus comprising editing means (160) for editing books, editing information input means (172) for introducing information necessary for the performance of editing, data memory means (169) for memorizing data for editing, and display means (168), which editing apparatus is provided with

   data ID number memory means (170) for memorizing the security numbers of operators and of data ID number groups corresponding to each particular security number, which data ID number groups indicate books which the operator having the particular security number is authorized to edit,

   security number input means (166) for introducing the security numbers,

   security number discrimination means (164) for performing the decision as to whether or not the security number introduced through the security number input means (166) is memorized in the data ID number memory means as a security number corresponding to a data ID number group,

   data ID number input means (173) for introducing the data ID number of a book to be edited, and

   data ID number discrimination means (171) for performing the decision as to whether or not the data ID number introduced through the data ID number input means (173) is registered in the data ID number memory means (170) as being of the data ID number group corresponding to the security number introduced through the security number input means (166),

   and the editing means is adapted so that the editing operation thereof is controlled according to the result of the decision performed by the security number discrimination means (164) and the data reading out of the data memory means (169) is controlled depending on the result of the decision performed by the data ID number discrimination means (171).

2. An editing apparatus according to claim 1, characterized by a specific predetermined data ID number (NEW) being assigned to the security number of the operator authorized to produce a new book.

Patentansprüche

1. Eine Redigiervorrichtung, umfassend eine Redigiereinrichtung (160) zum Redigieren von Büchern, eine Redigierinformations-Eingabevorrichtung (172) zur Eingabe von für die Durchführung des Redigierens notwendigen Informationen, eine Datenspeichervorrichtung (169) zur Speicherung von Daten für das Redigieren und eine Anzeigevorrichtung (168), wobei die Redigiervorrichtung ausgestattet ist mit:

   - einer Daten-ID-Nummer-Speichervorrichtung (170) zur Speicherung der Sicherheitsnummern von Bedienern und von Daten-ID-Nummer-Gruppen, welche jeder bestimmten Sicherheitsnummer entsprechen, wobei die Daten-ID-Nummer-Gruppen Bücher angeben, zu deren Redaktion der Bediener mit der bestimmten Sicherheitsnummer berechtigt ist,

   - einer Sicherheitsnummer-Eingabevorrichtung (166) zur Eingabe der Sicherheitsnummern,

   - einer Sicherheitsnummer-Unterscheidungsvorrichtung (164) zur Entscheidung, ob oder ob nicht die durch die Sicherheitsnummer-Eingabevorrichtung (166) eingegebene Sicherheitsnummer in der Daten-ID-Nummer-Speichervorrichtung als eine Sicherheitnummer gespeichert ist, welche einer Daten-ID-Nummer-Gruppe entspricht,

   - eine Daten-ID-Nummer-Eingabevorrichtung (173) zur Eingabe der Daten-ID-Nummer eines zu redigierenden Buches, und

   - einer Daten-ID-Nummer-Unterscheidungsvorrichtung (171) zur Entscheidung, ob oder ob nicht die durch die Daten-ID-Nummer-Eingabevorrichtung (173) eingegebene Daten-ID-Nummer in der Daten-ID-Nummer-Speichervorrichtung (170) als aus der Daten-ID-Nummer-Gruppe registriert ist, welche der durch die Sicherheitsnummer-Eingabevorrichtung (166) eingegebenen Sicherheitsnummer entspricht,

   - und wobei die Redigiereinrichtung so ausgelegt ist, daß deren Redigierbetrieb gesteuert wird entsprechend dem Ergebnis der Entscheidung, welche durch die Sicherheitsnummer-Unterscheidungsvorrichtung (164) durchgeführt wird, und daß das Auslesen der Daten aus der Datenspeichervorrichtung (169) in Abhängigkeit von dem Ergebnis der durch die Daten-ID-Nummer-Unterscheidungsvorrichtung (171) durchgeführten Entscheidung gesteuert wird.

2. Redigiervorrichtung nach Anspruch 1, **gekennzeichnet** durch das Zuordnen einer spezifischen, vorbestimmten Daten-ID-Nummer (NEW) zur der Sicherheitsnummer des Bedieners, welcher zur Erzeugung eines neuen Buches berechtigt ist.

Revendications

1. Appareil d'édition comprenant un moyen d'édition (160) pour éditer des livres, un moyen d'entrée d'informations d'édition (172) pour introduire les informations nécessaires pour effectuer l'édition, un moyen de mémoire de données (169) pour mémoriser les données pour l'édition, et un moyen d'affichage (168), lequel appareil d'édition est muni de

   un moyen de mémoire de numéro ID de données (170) pour mémoriser les numéros de sécurité des opérateurs et des groupes de numéros ID de données correspondant à chaque numéro de sécurité particulier, lesquels groupes de numéros ID de données indiquent des livres que l'opérateur ayant le numéro de sécurité particulier est autorisé à éditer,
   un moyen d'entrée de numéro de sécurité (166) pour introduire les numéros de sécurité,
   un moyen de discrimination de numéro de sécurité (164) pour décider si oui ou non le numéro de sécurité introduit par l'intermédiaire du moyen d'entrée du numéro de sécurité (166) est mémorisé dans le moyen de mémoire de numéros ID de données comme un numéro de sécurité correspondant à un groupe de numéros ID de données,
   un moyen d'entrée de numéro ID de données (173) pour introduire le numéro ID de données d'un livre qui doit être édité, et
   un moyen de discrimination de numéro ID de données (171) pour décider si oui ou non le numéro ID de données introduit par l'intermédiaire du moyen d'entrée du numéro ID de données (173) est enregistré dans le moyen de mémoire de numéro ID de données (170) comme étant du groupe de numéros ID de données correspondant au numéro de sécurité introduit par l'intermédiaire du moyen d'entrée de numéro de sécurité (166),
   et le moyen d'édition est prévu d'une manière telle que son opération d'édition est contrôlée conformément au résultat de la décision prise par le moyen de discrimination de numéro de sécurité (164) et les données extraites du moyen de mémoire de données (169) sont contrôlées en fonction du résultat de la décision prise par le moyen de discrimination du numéro ID de données (171).

2. Appareil d'édition selon la revendication 1, caractérisé par un numéro ID de données prédéterminé spécifique (NOUVEAU) étant affecté au numéro de sécurité de l'opérateur autorisé à produire un nouveau livre.

FIG. 1

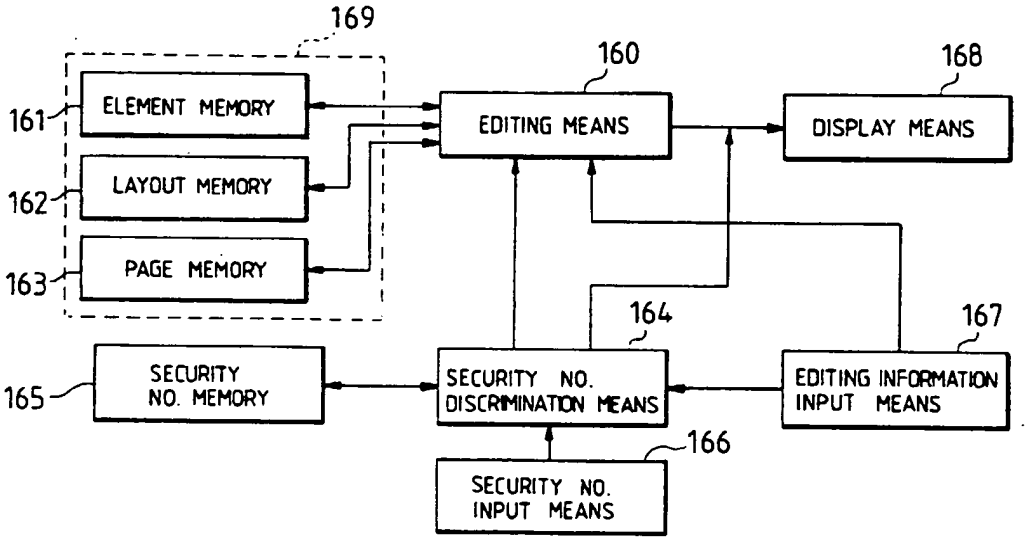


FIG. 2

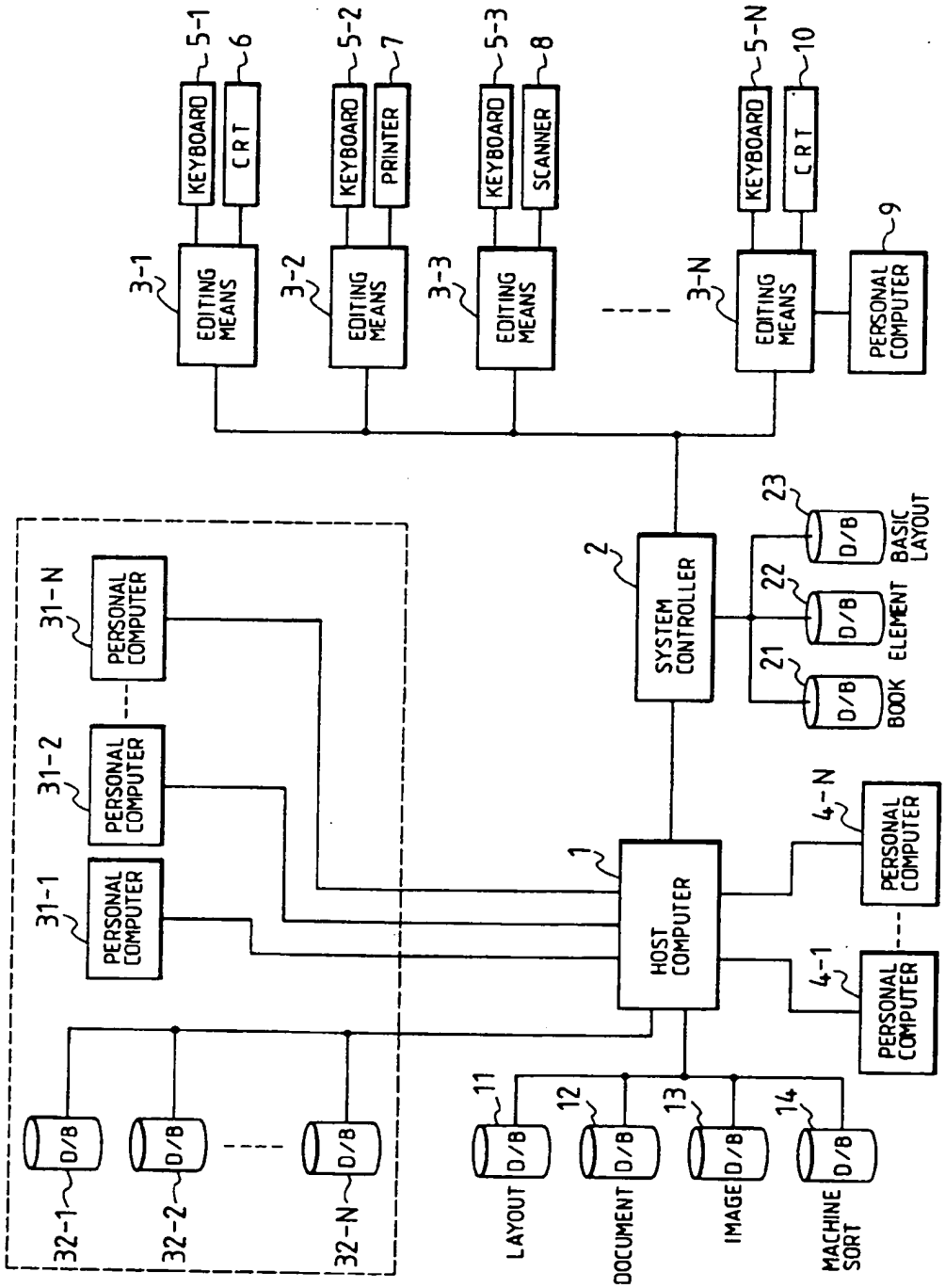


FIG. 3(1/5)

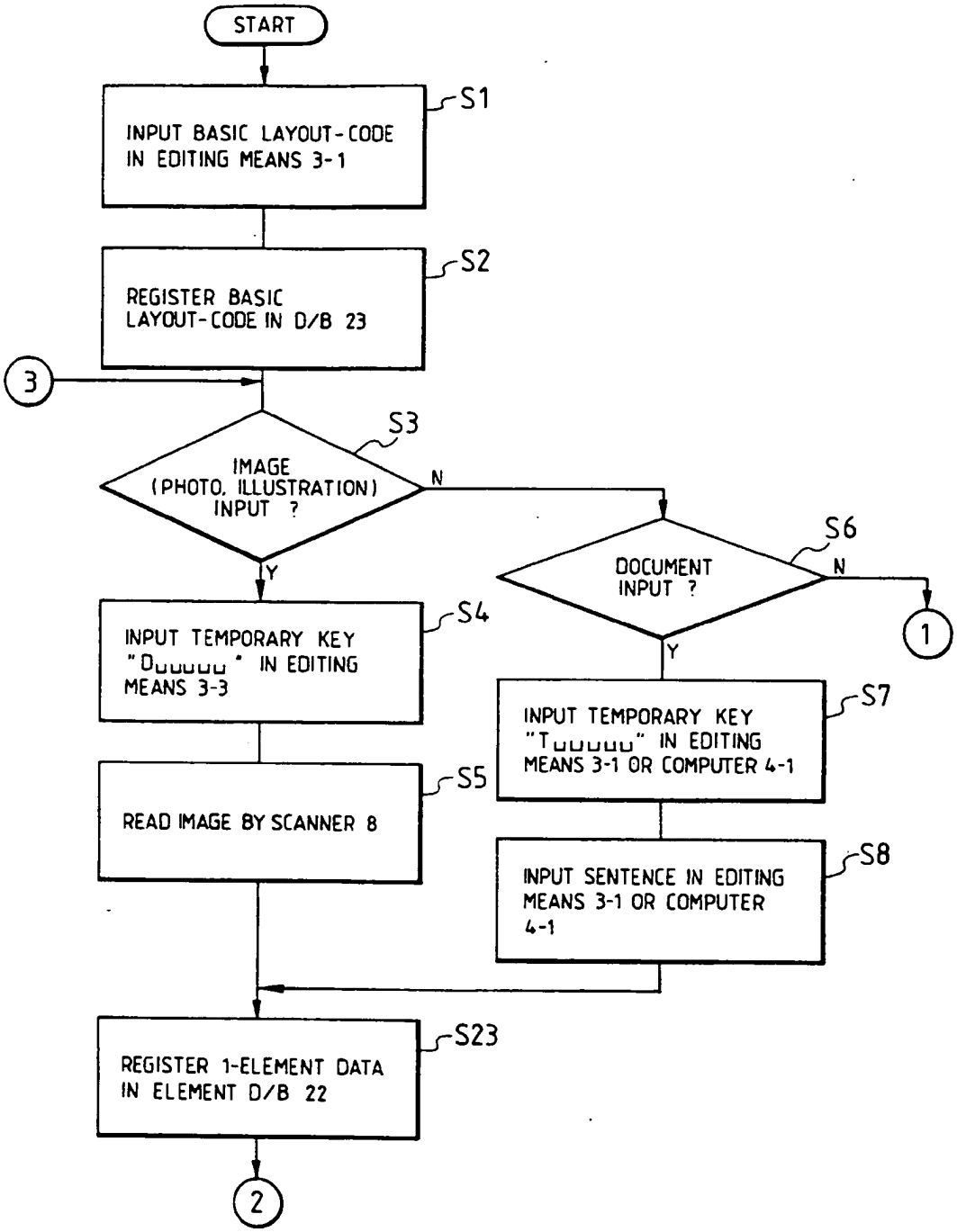


FIG. 3(2/5)

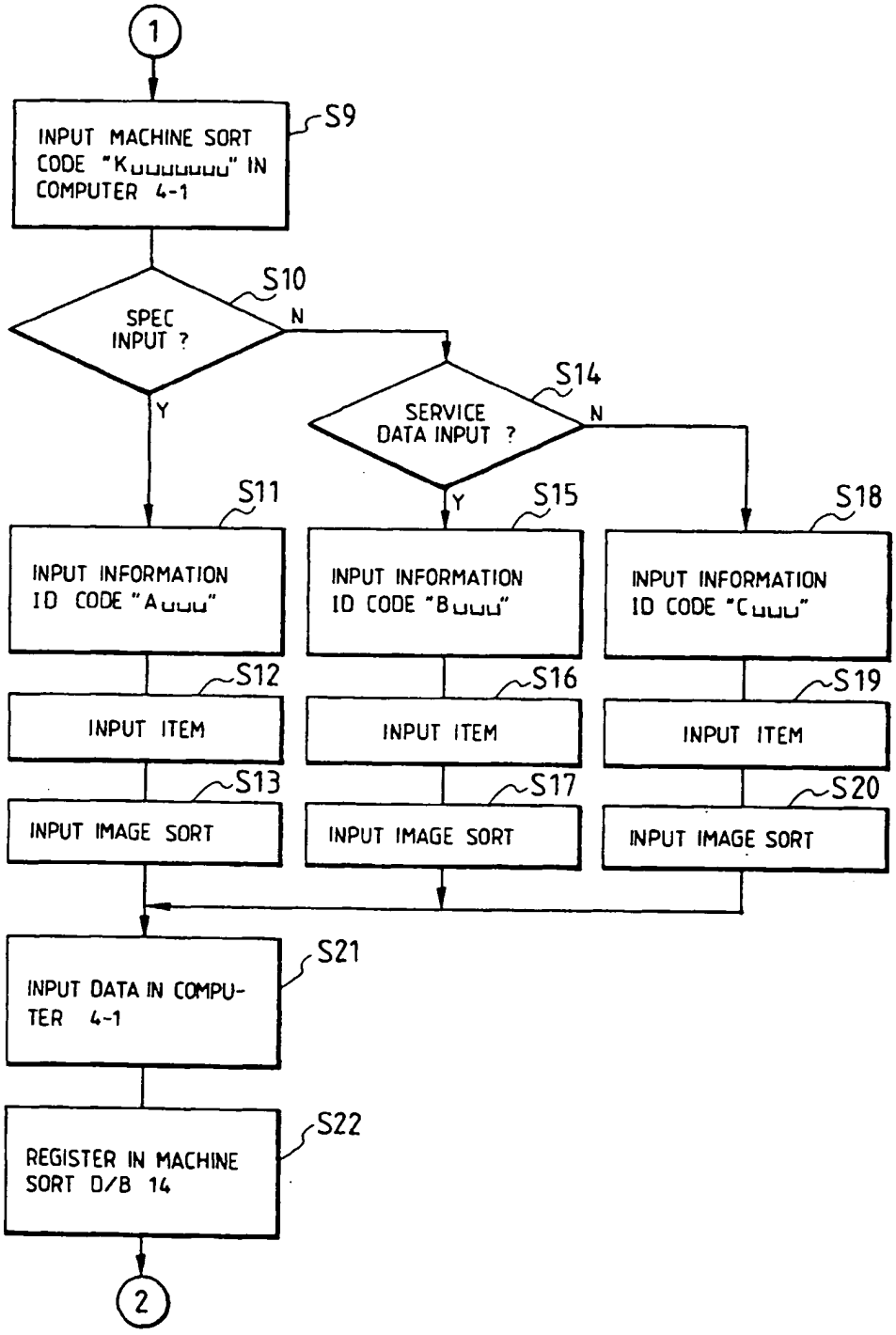


FIG. 3(3/5)

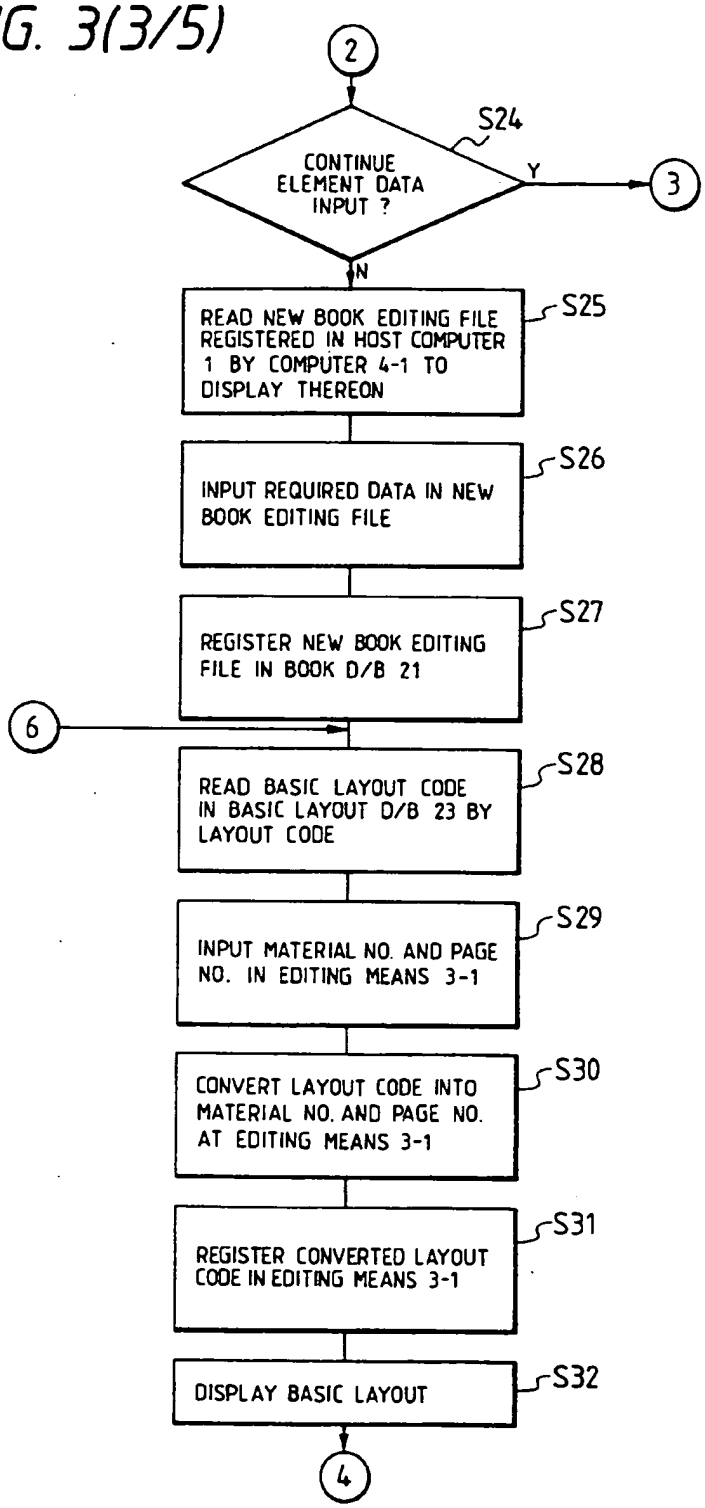


FIG. 3(4/5)

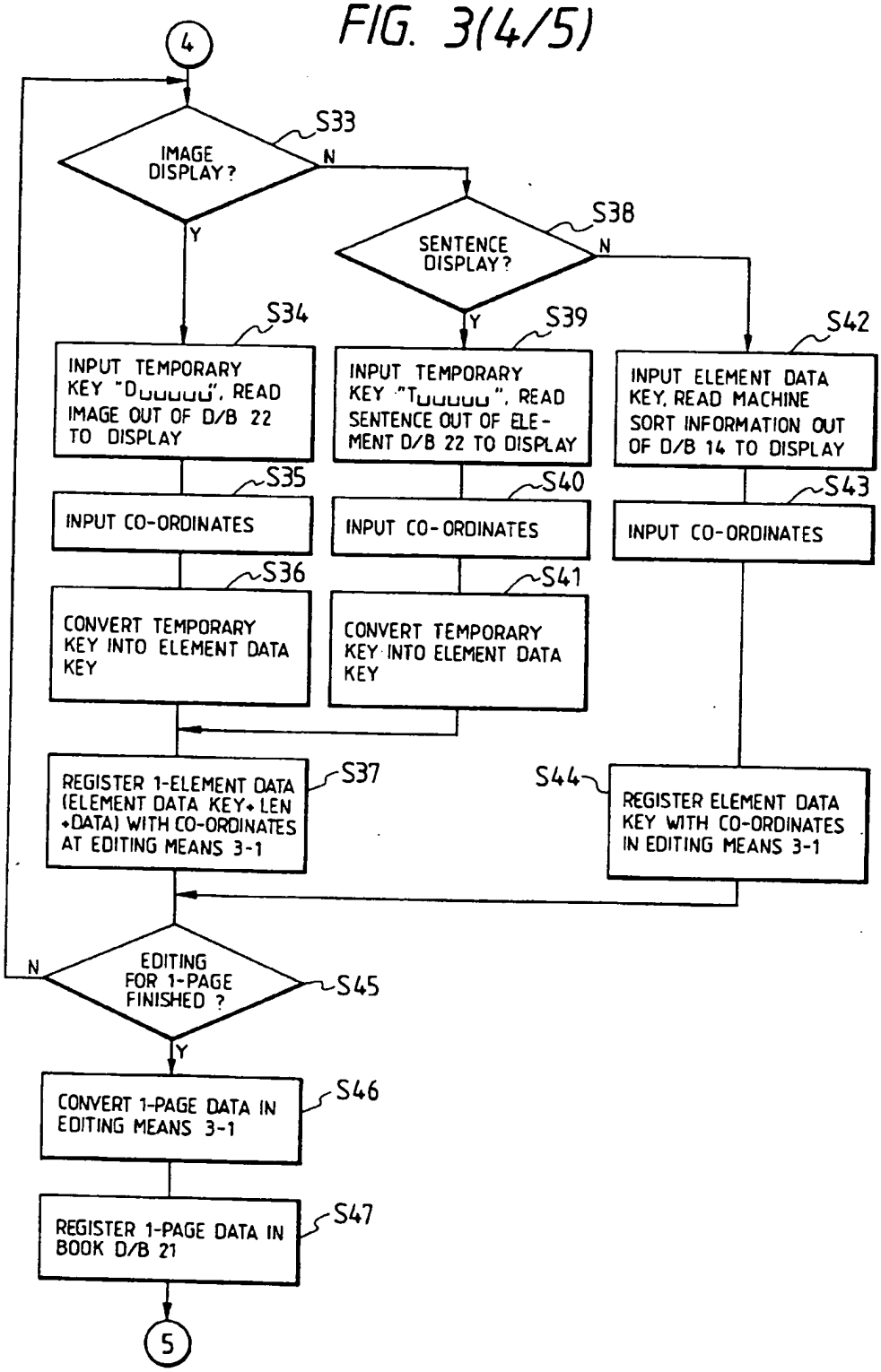


FIG. 3(5/5)

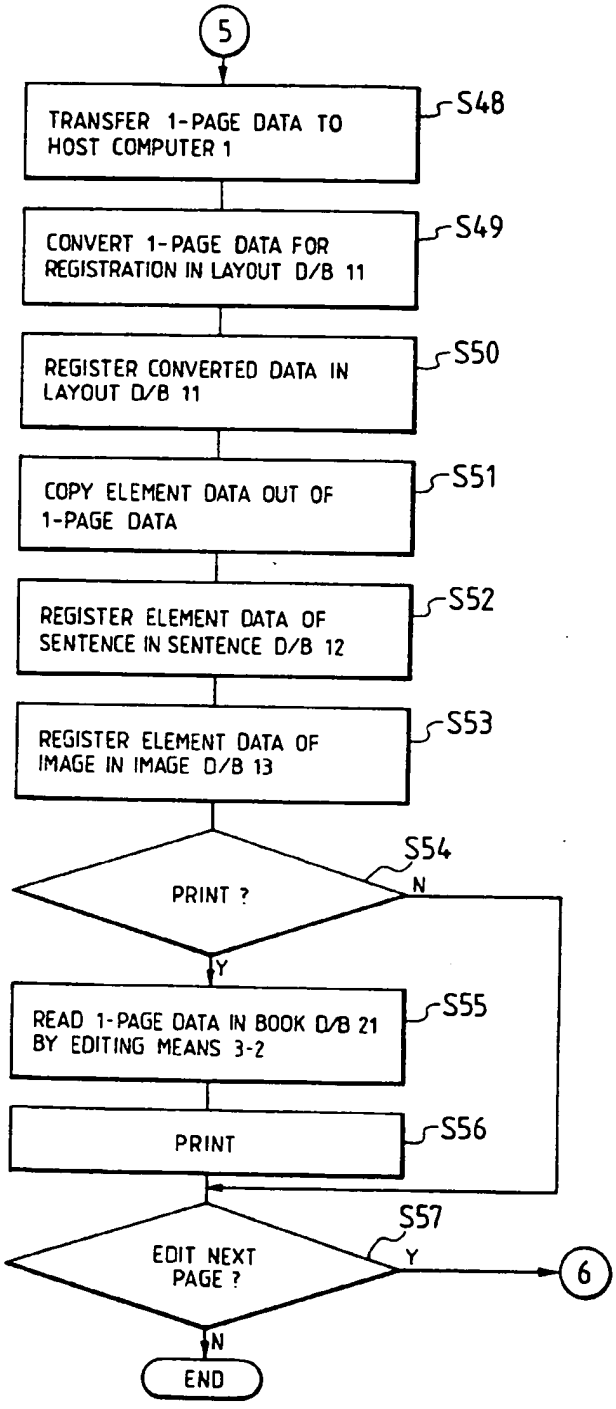


FIG. 4

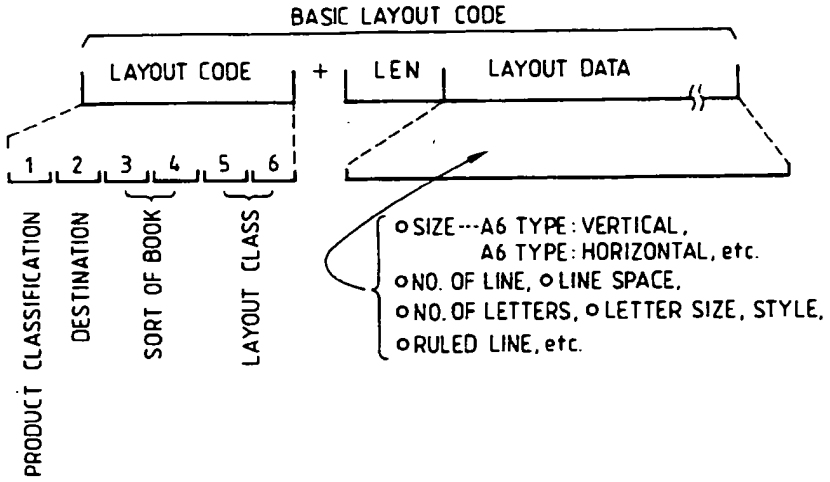


FIG. 5

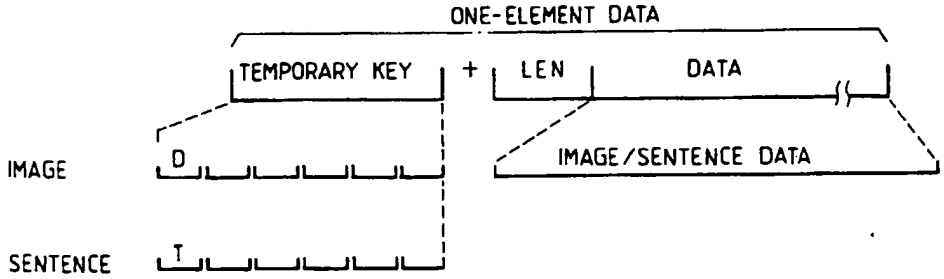


FIG. 6

FIG. 7

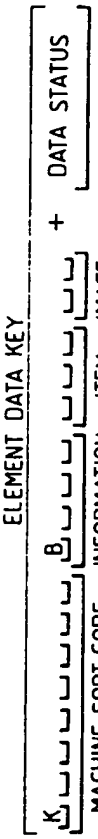


FIG. 8

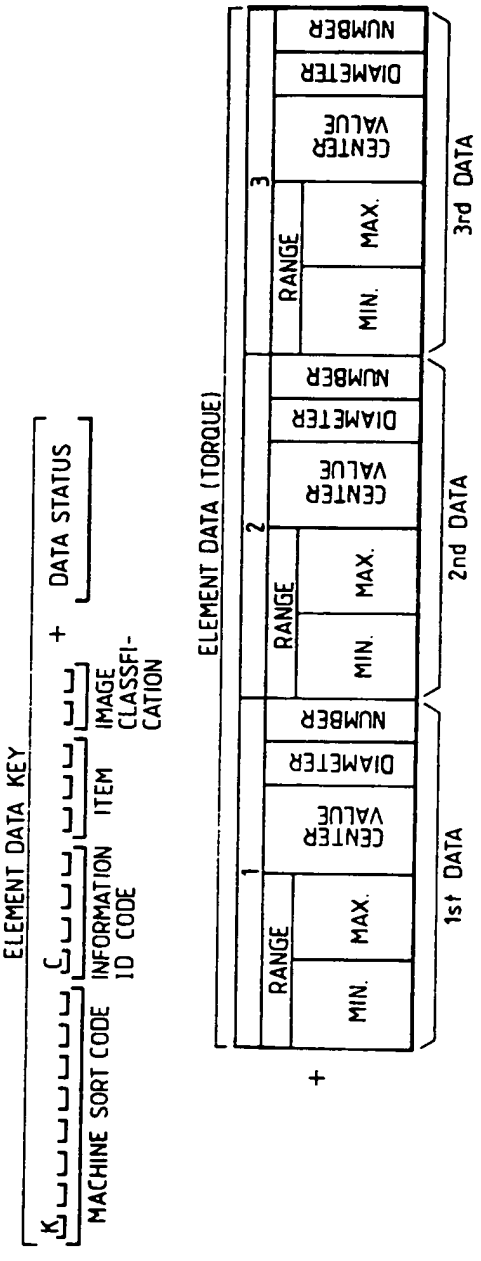


FIG. 9

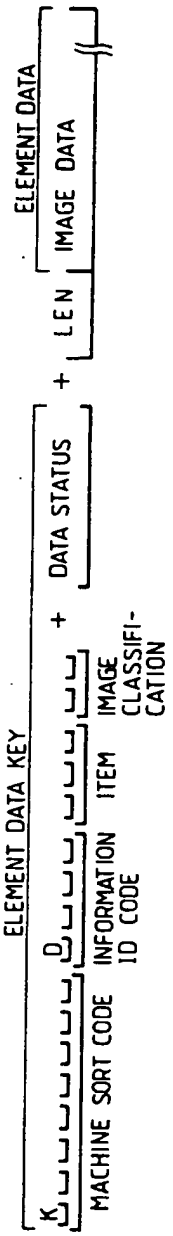


FIG. 10

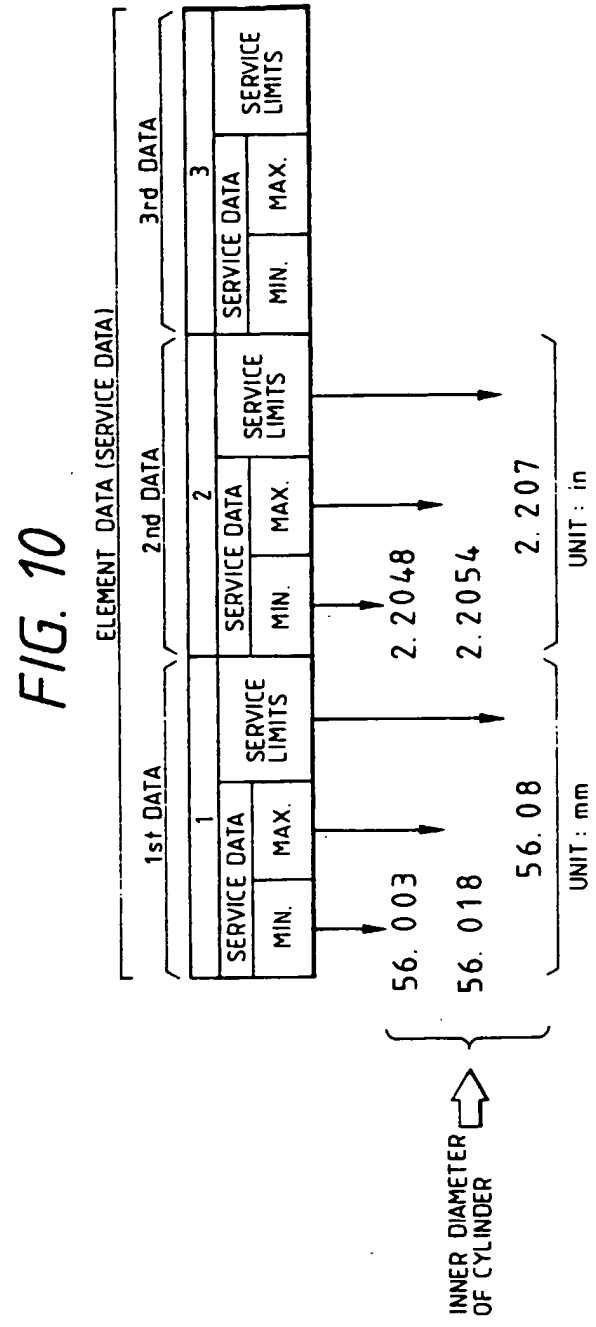


FIG. 11

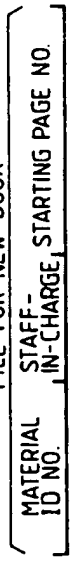


FIG. 12

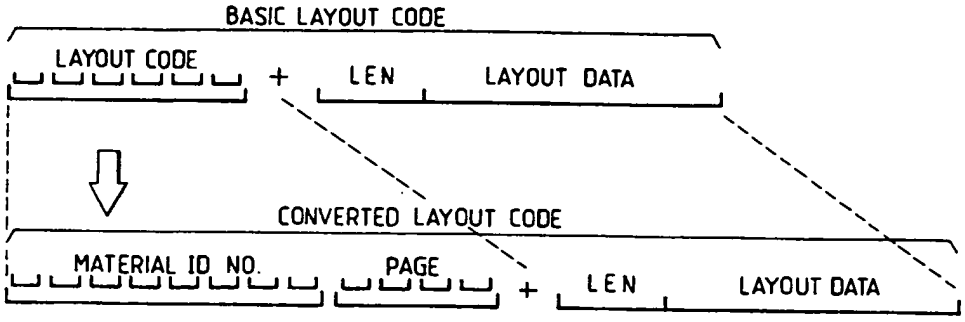


FIG. 13

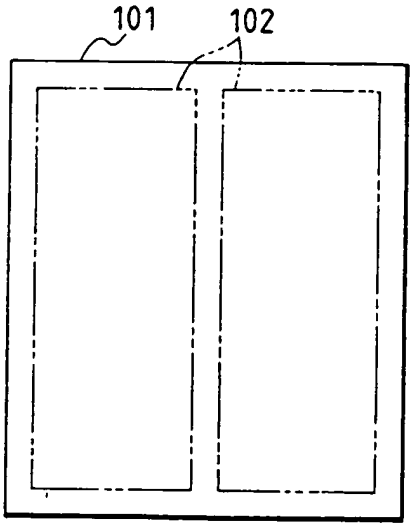


FIG. 14

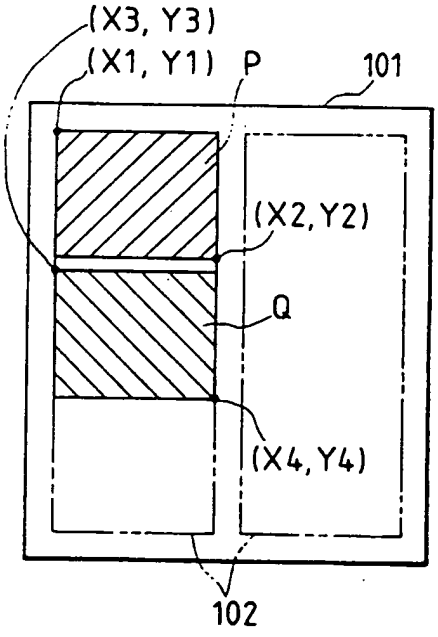


FIG. 15

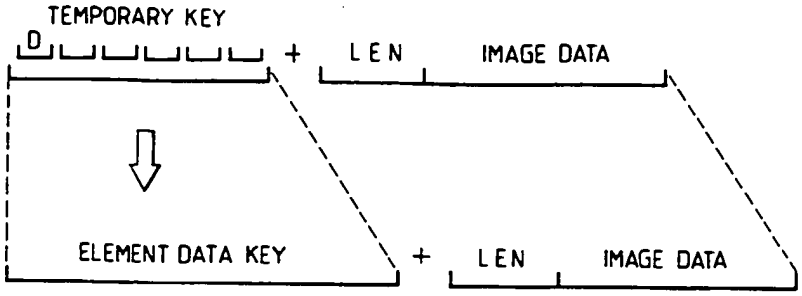


FIG. 16

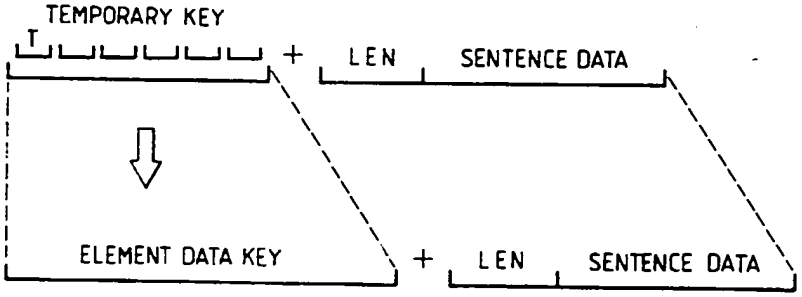


FIG. 23

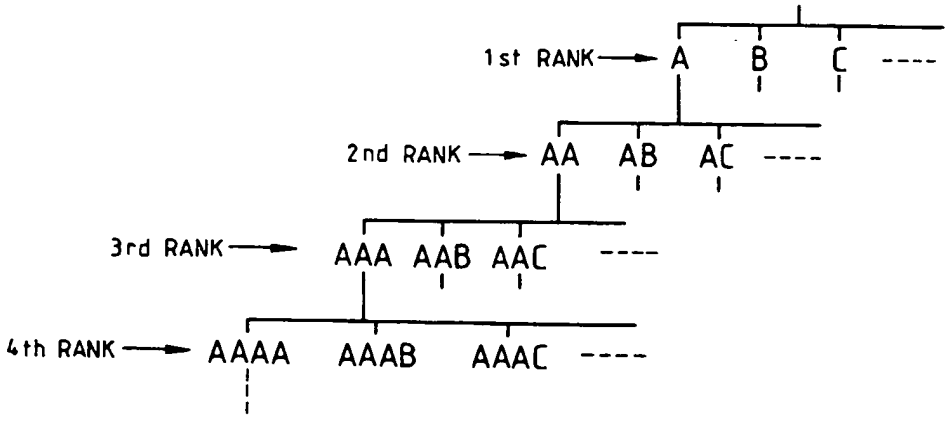


FIG. 17

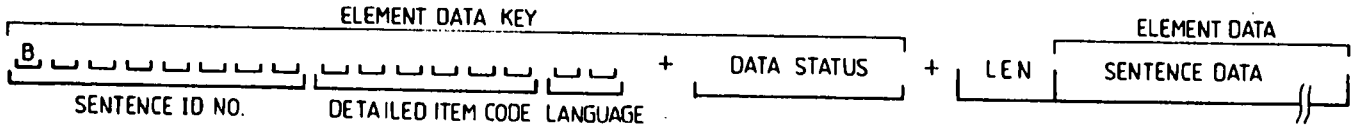


FIG. 18

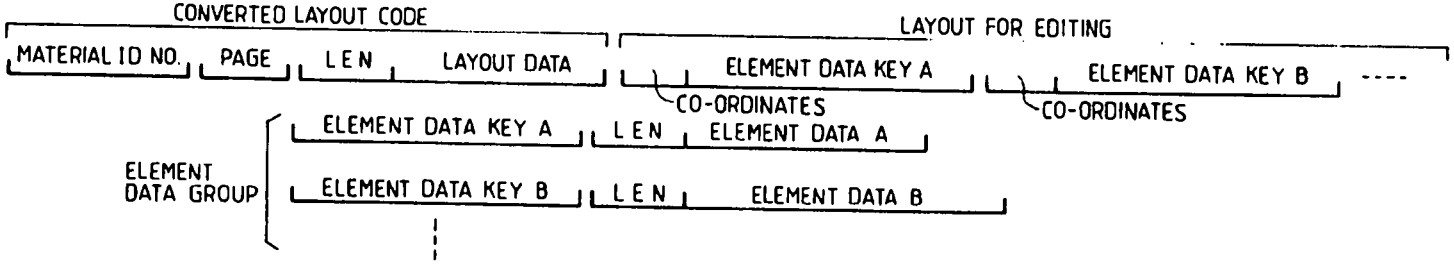


FIG. 19

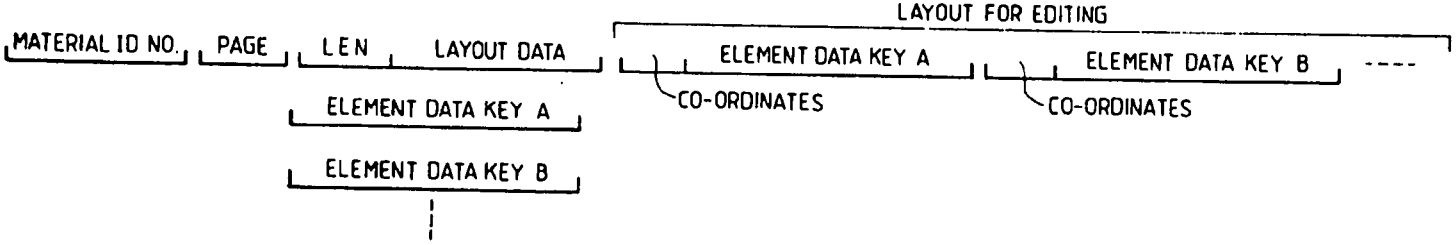


FIG. 20(1/6)

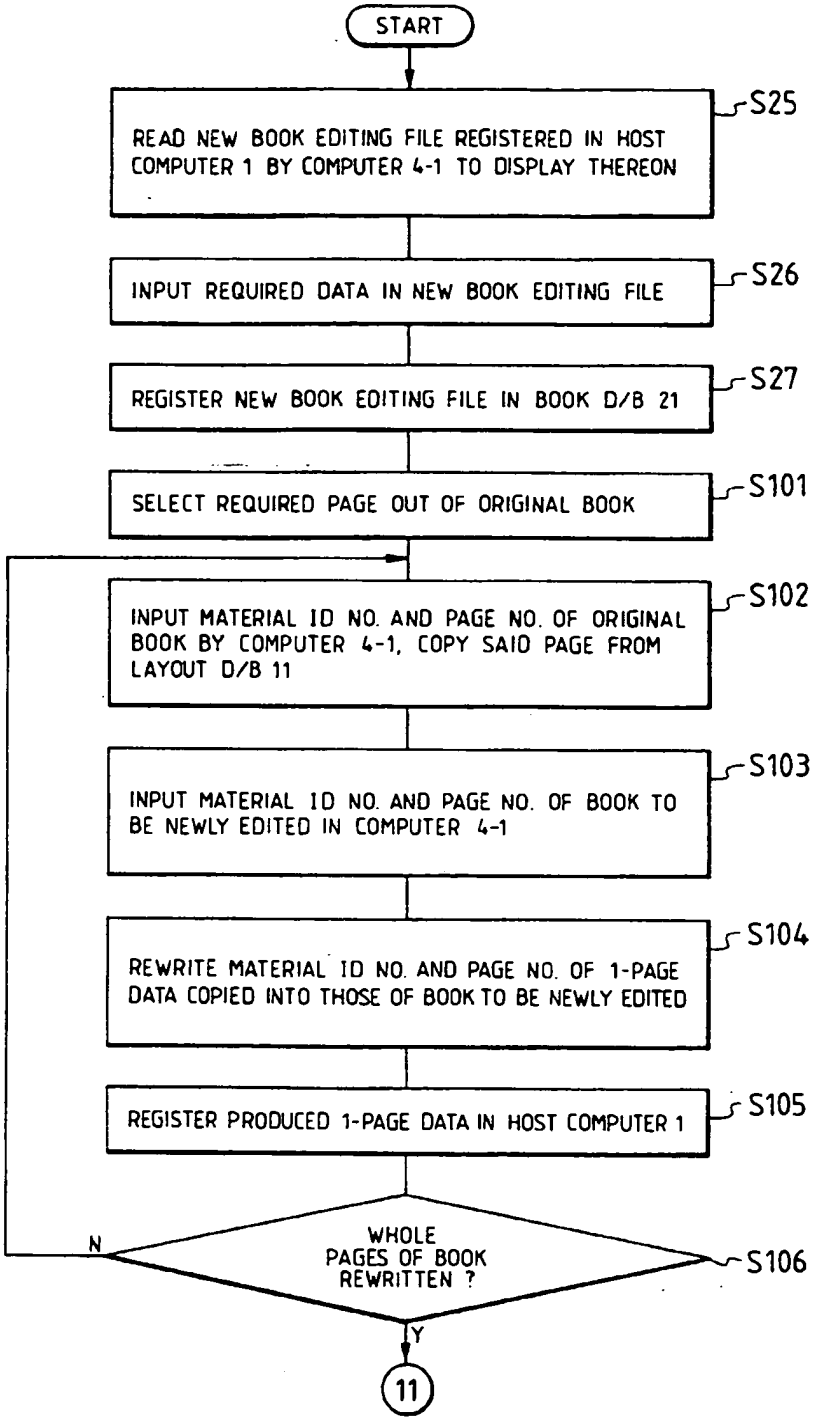


FIG. 20(2/6)

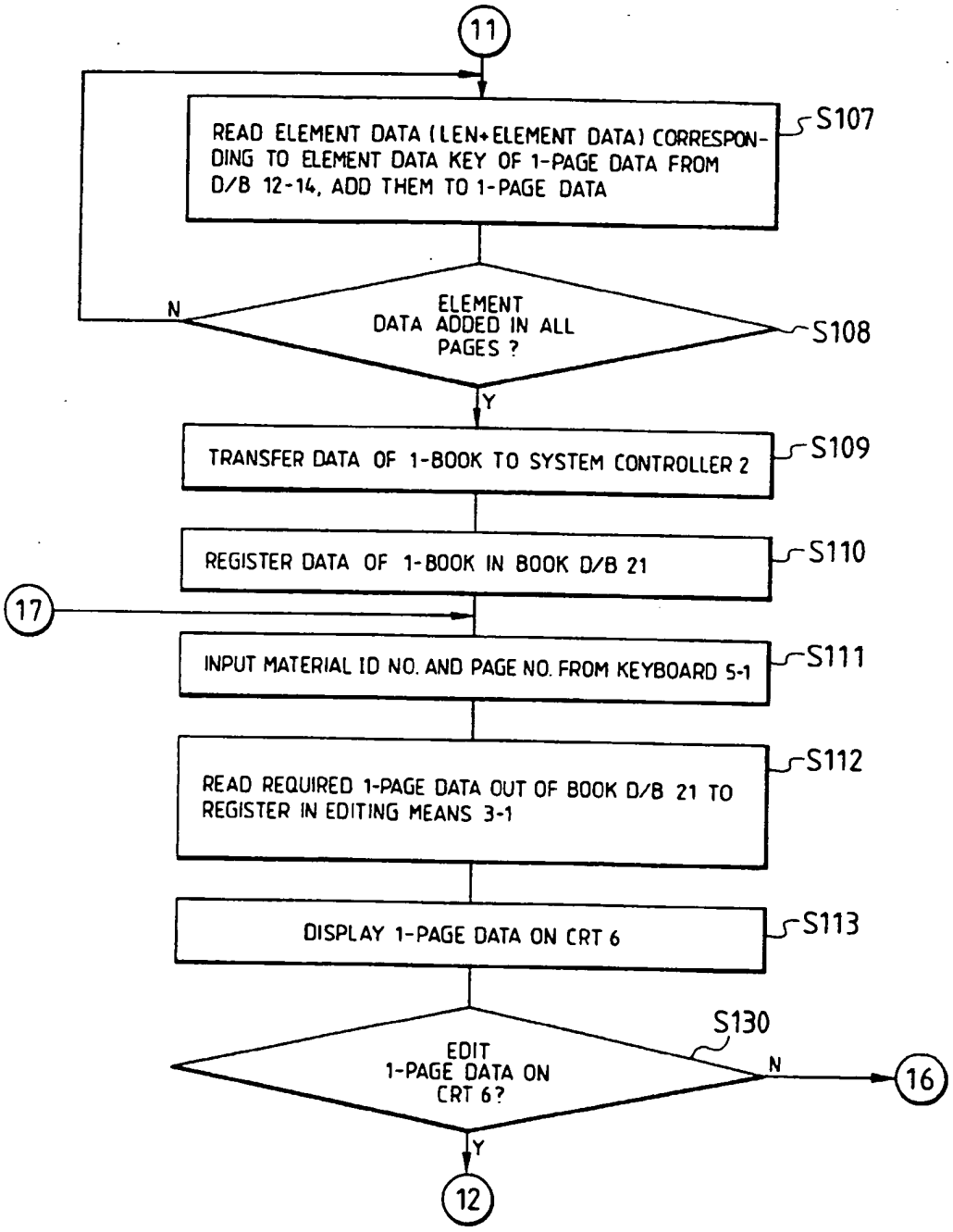


FIG. 20A

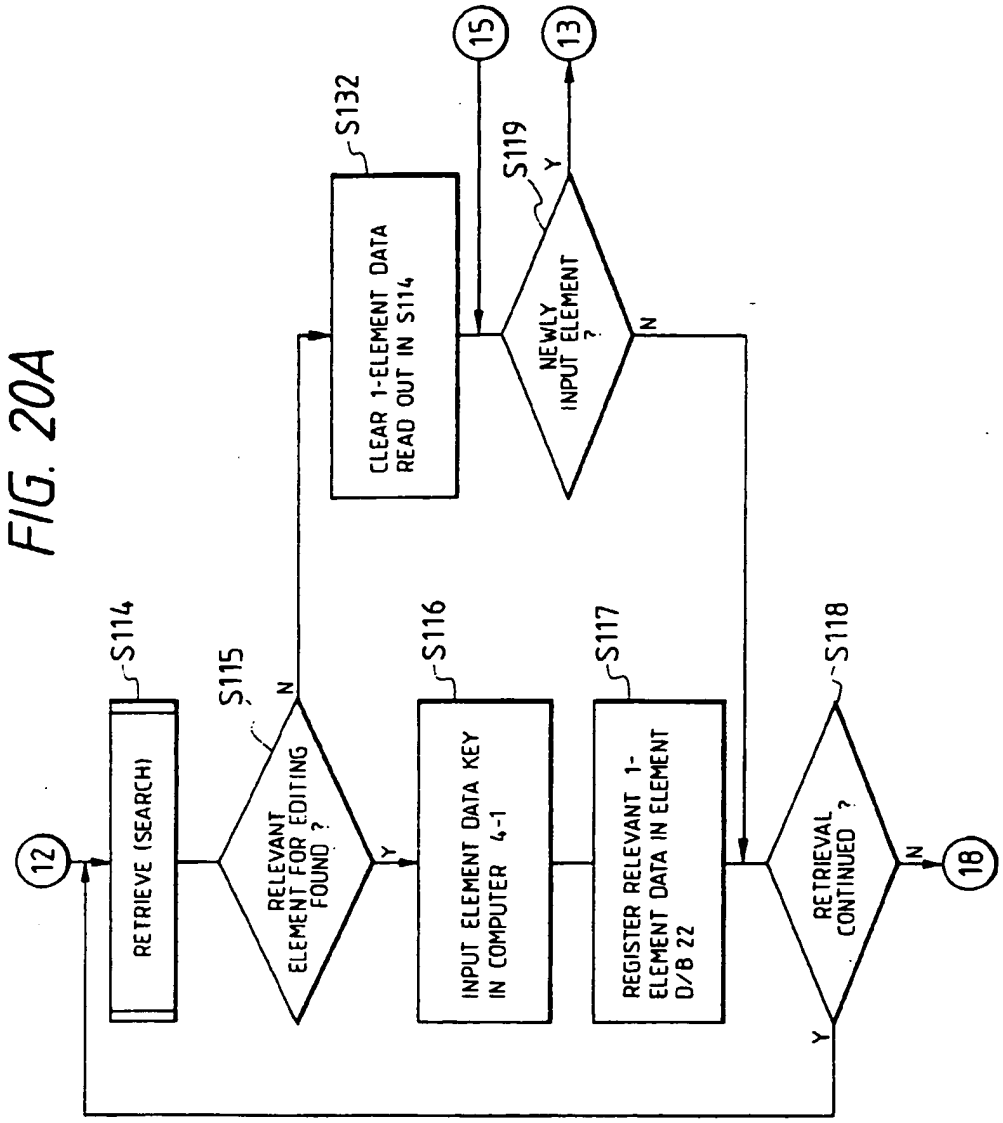


FIG. 20(3/6)

FIG. 20A	FIG. 20B

FIG. 20B

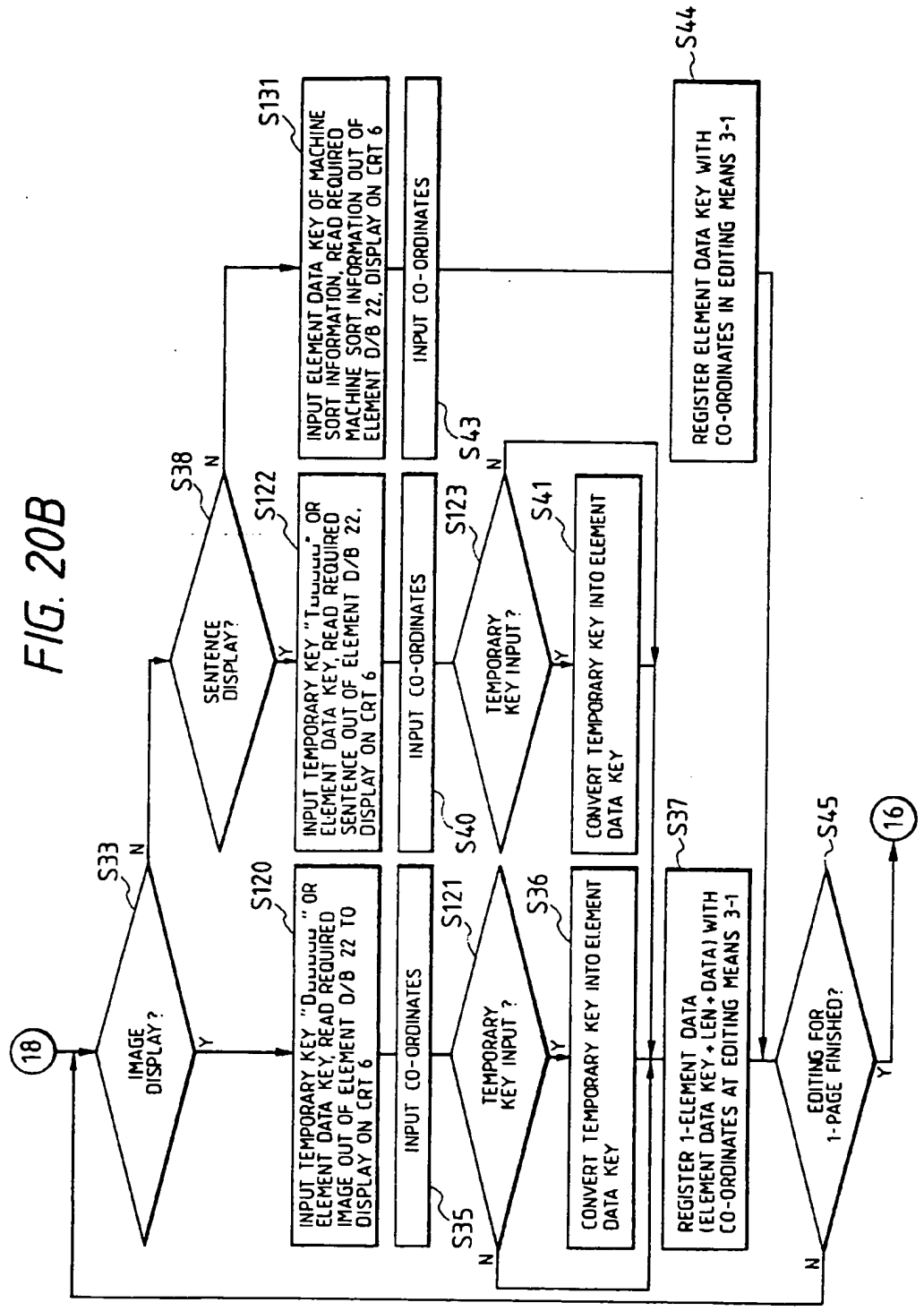


FIG. 20(4/6)

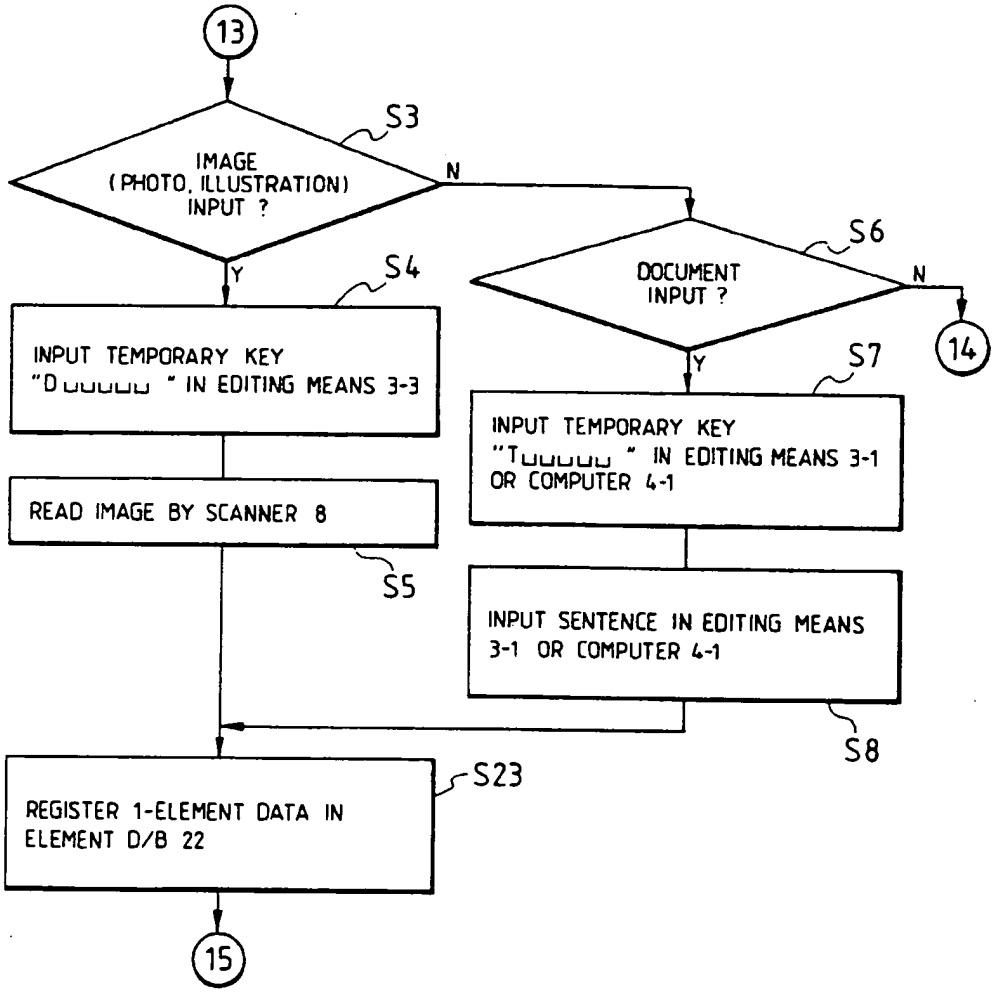


FIG. 20(5/6)

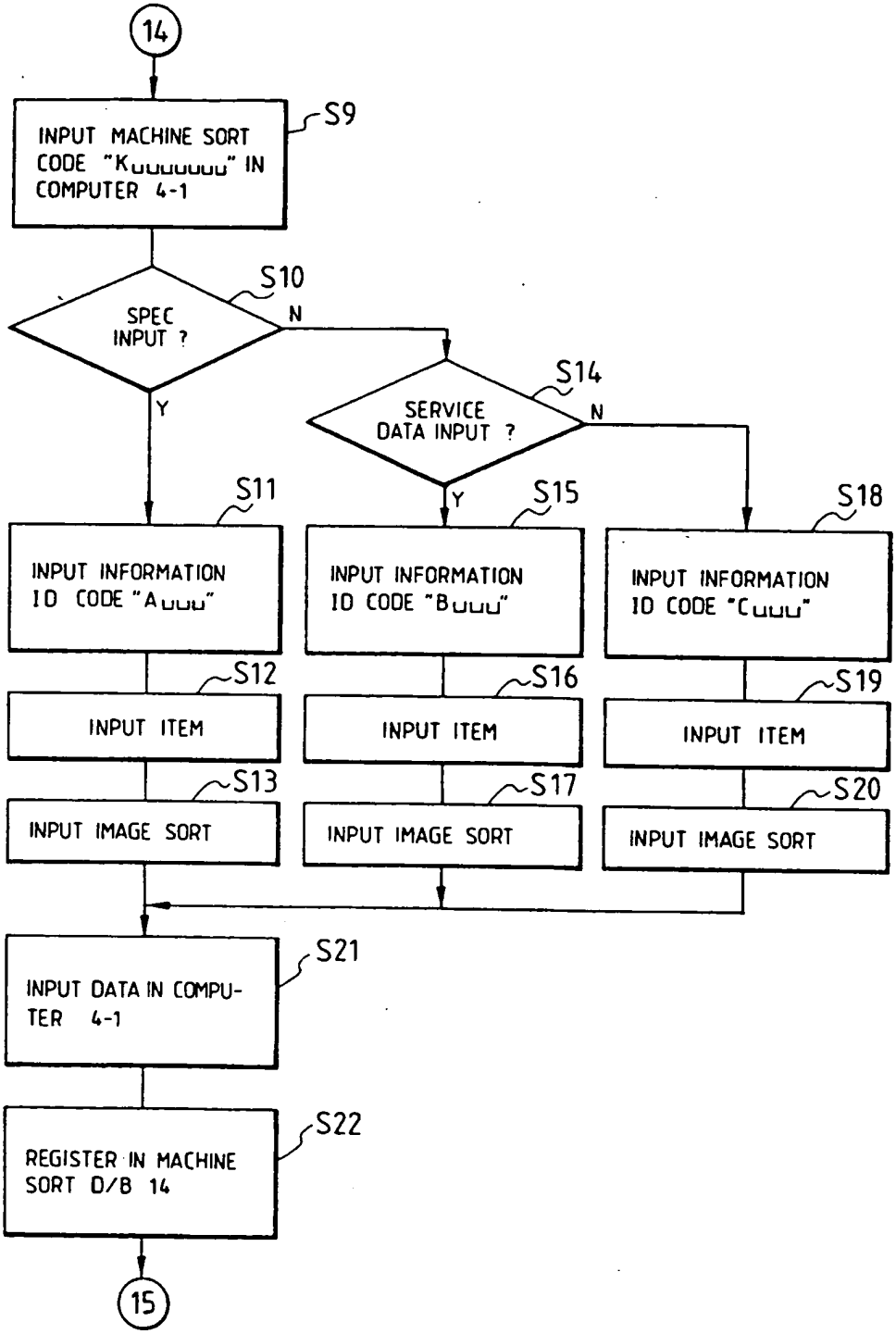


FIG. 20(6/6)

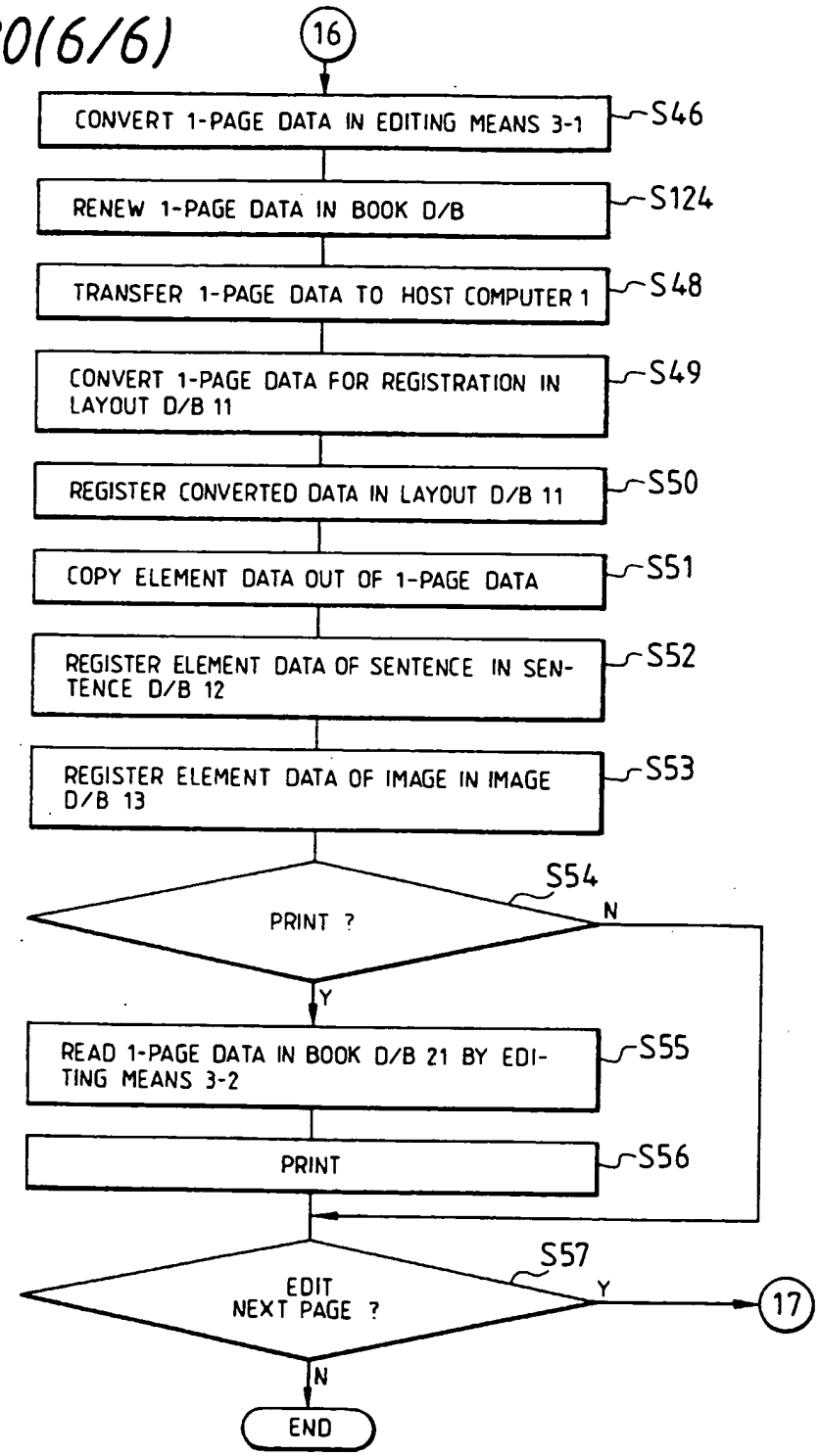
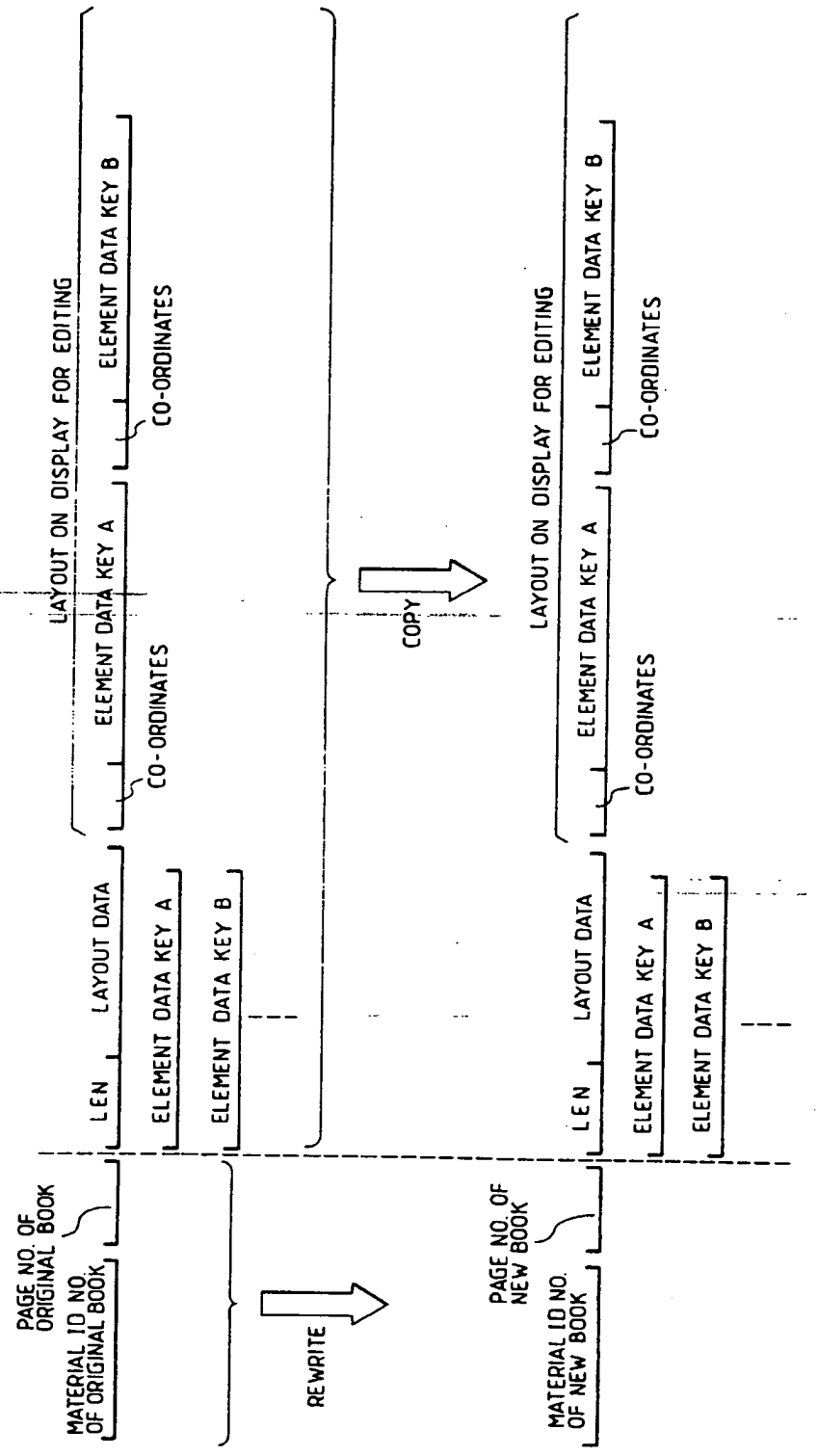


FIG. 21



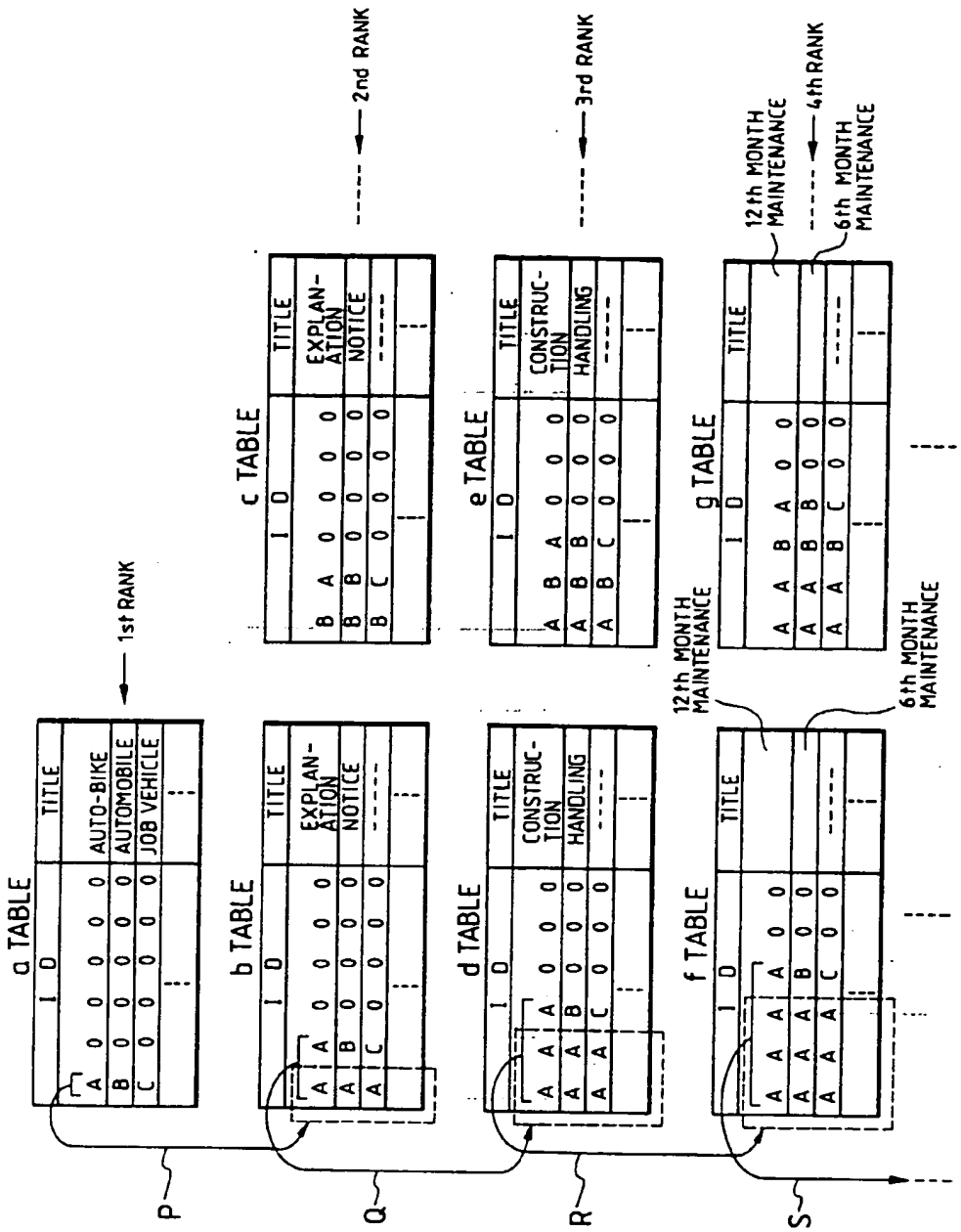


FIG. 22

FIG. 24

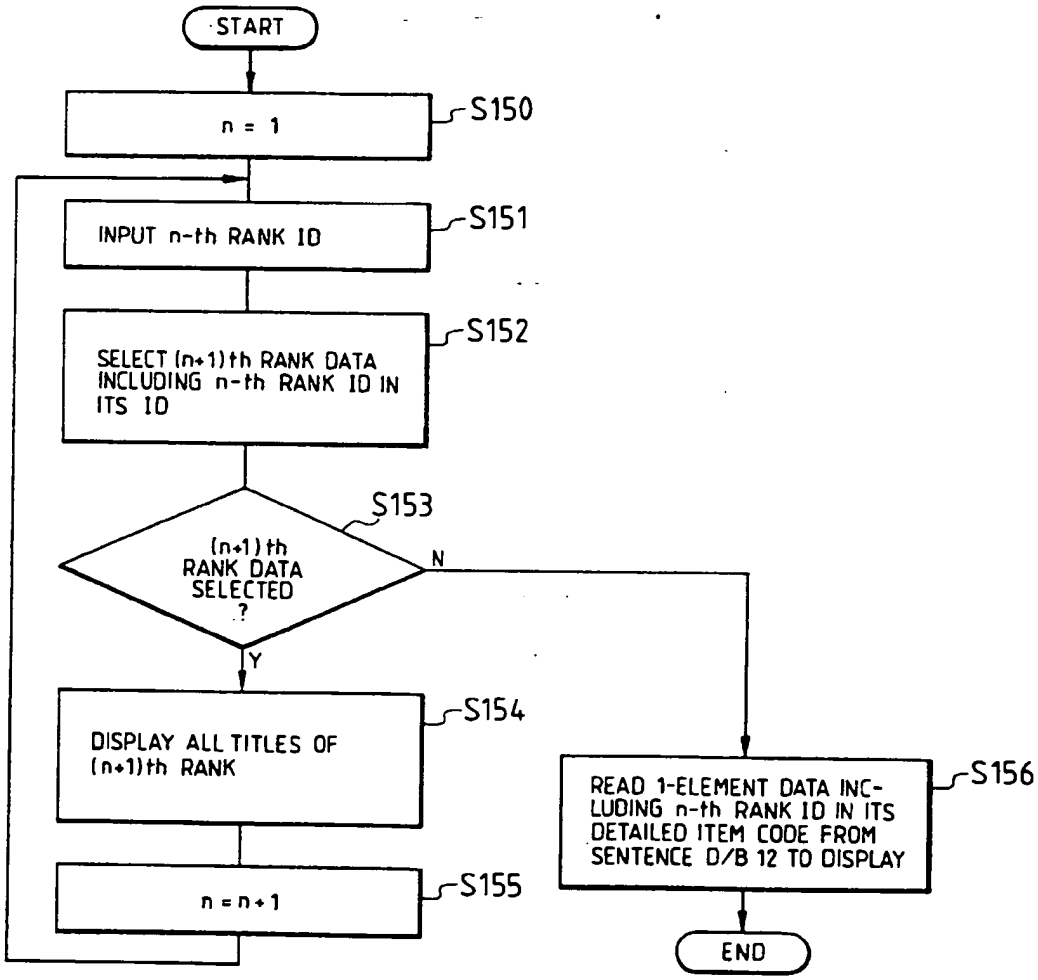


FIG. 25

CONVERSION CLASS	IMAGE CLASS	DISPLAY ON CRT
10	10	56.003-56.018mm
	20	56.003-56.018mm (2.2048-2.2054in)
	⋮	⋮
20	10	2.2048-2.2054in
	20	2.2048-2.2054in (56.003-56.018mm)
	⋮	⋮
40	10	56.003-56.018mm
	20	56.003-56.018mm (2.2048-2.2054in)
	⋮	⋮
⋮	⋮	⋮

FIG. 26

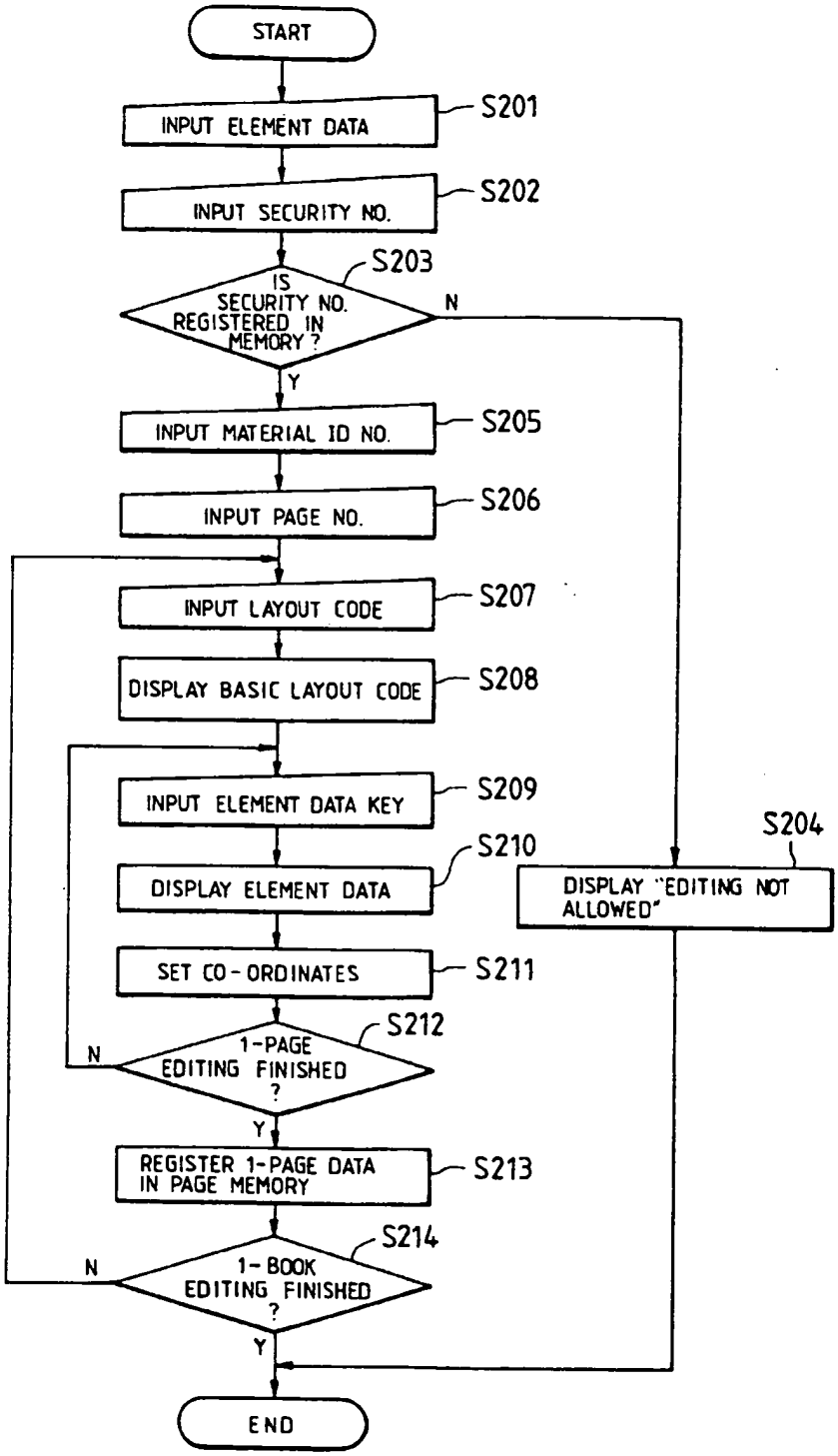


FIG. 27

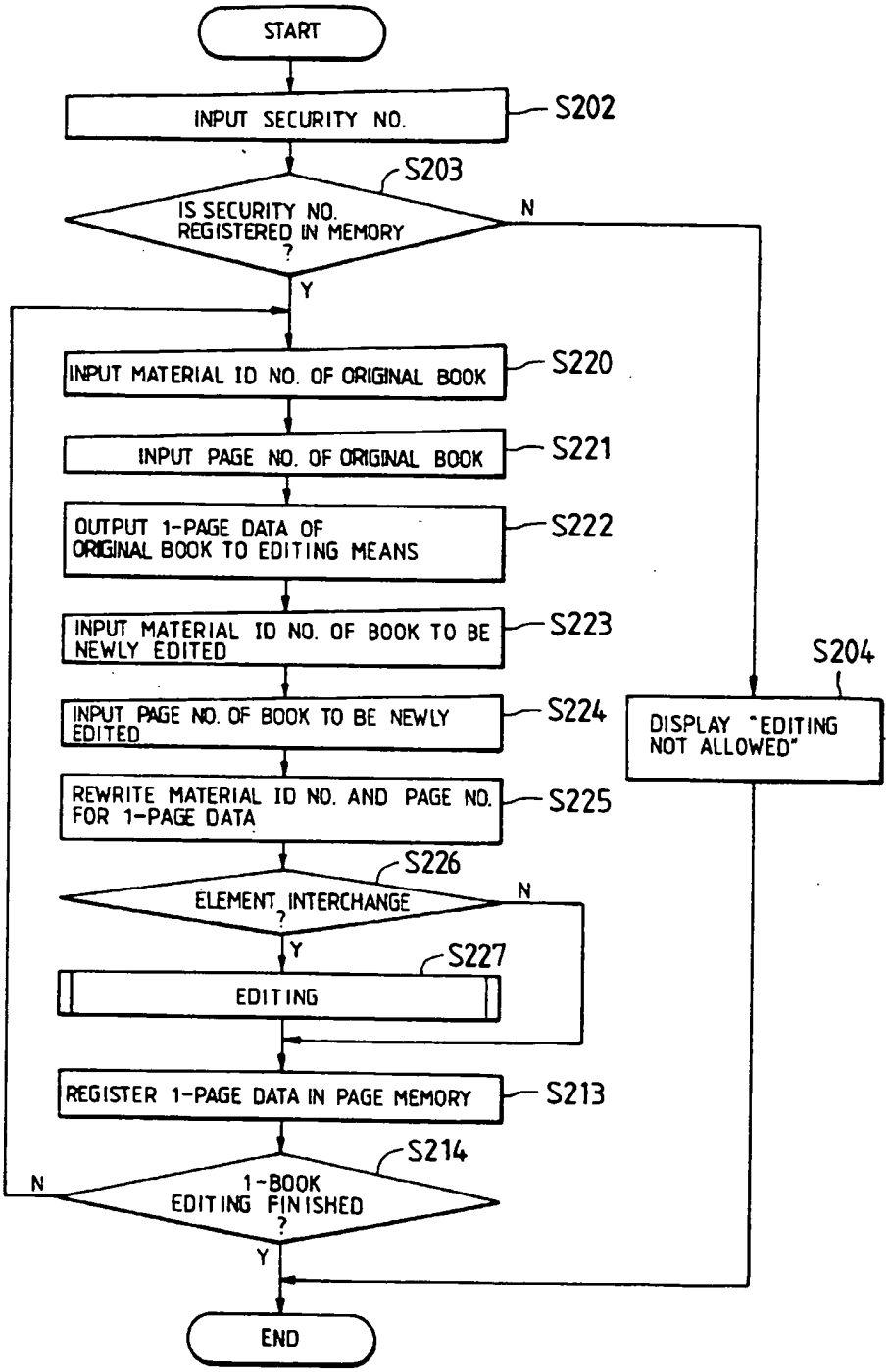


FIG. 28

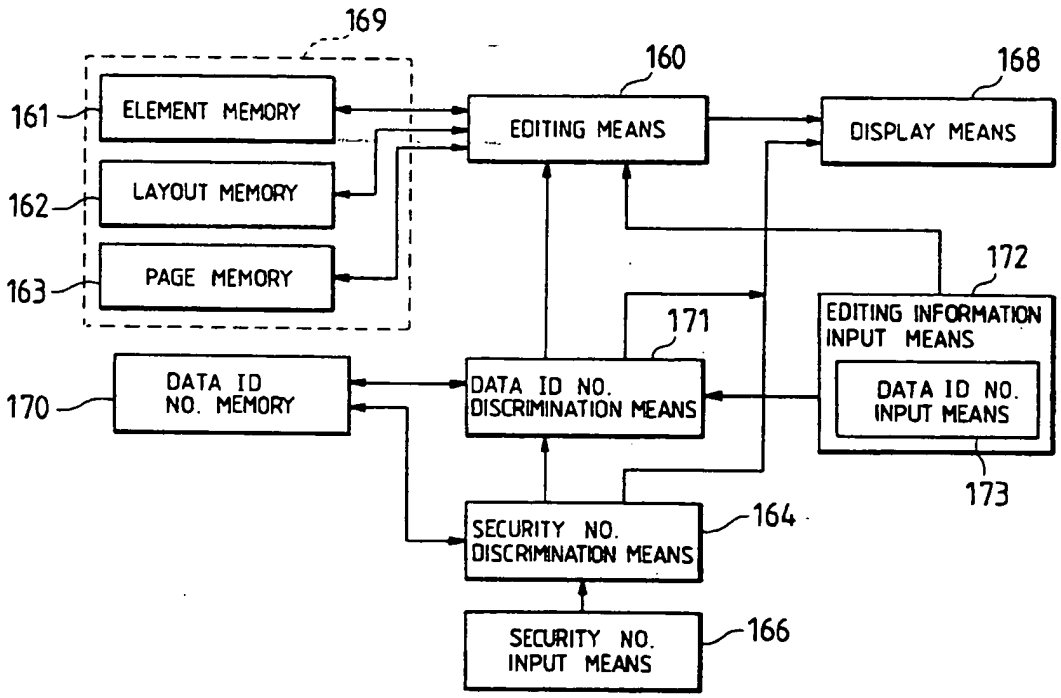


FIG. 29

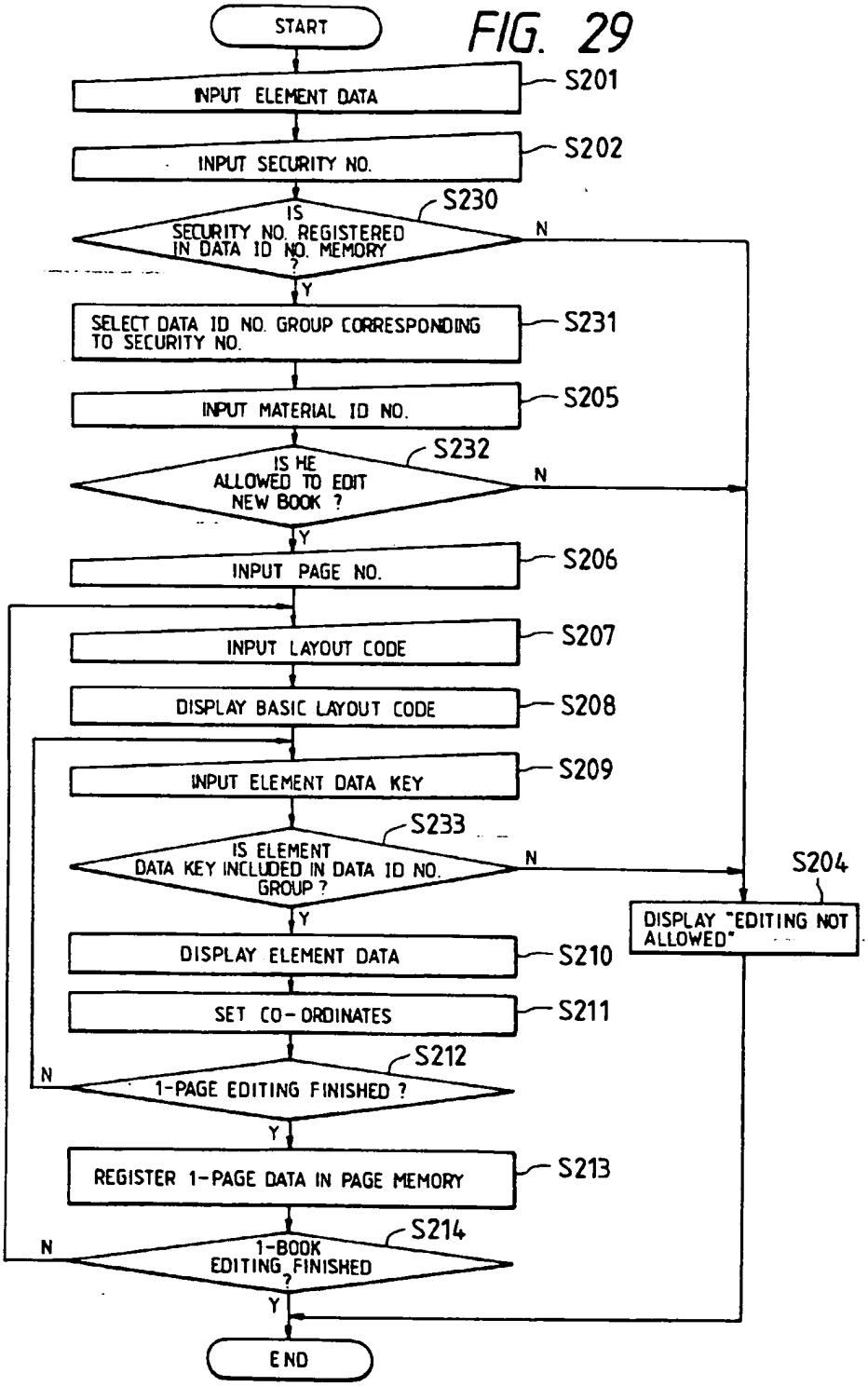


FIG. 30

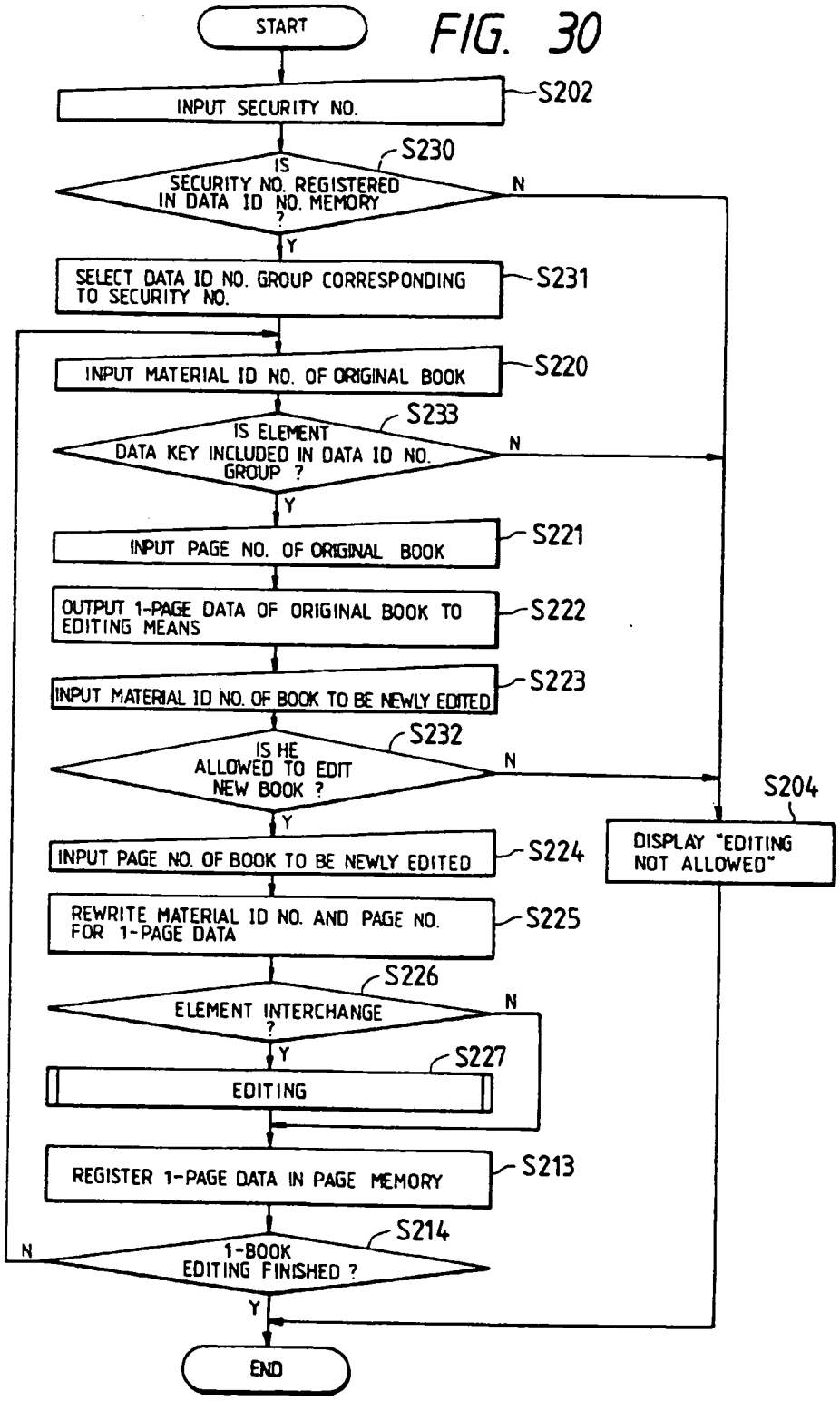


FIG. 31

TABLE a

SECURITY NO.	A1234
DATA ID NO.	B10
	B12
	B15
	B16
	NEW

TABLE b

SECURITY NO.	A1235
DATA ID NO.	B12
	B15
	B16

TABLE c

SECURITY NO.	A1236
DATA ID NO.	B10
	B14

FIG. 32

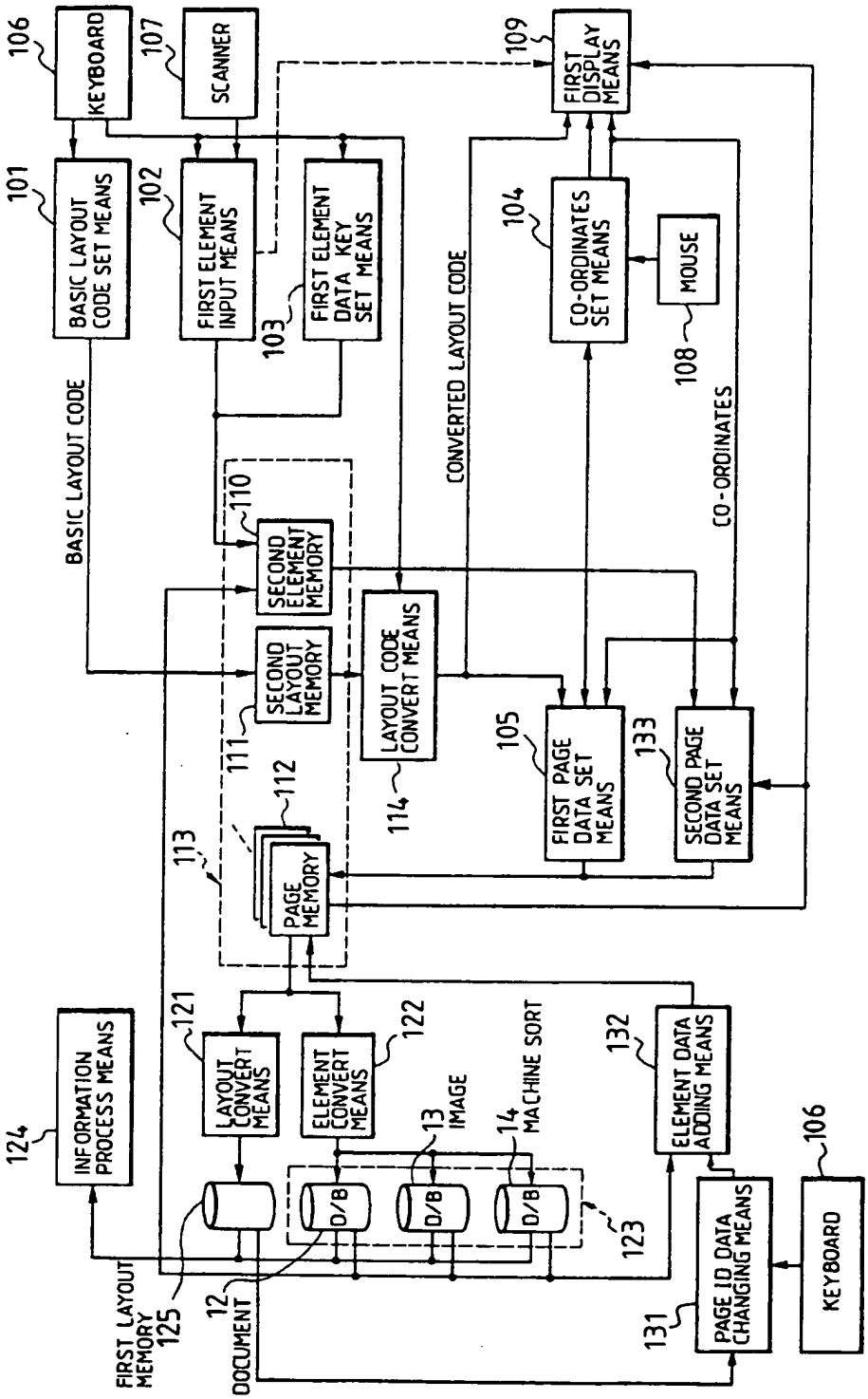


FIG. 33

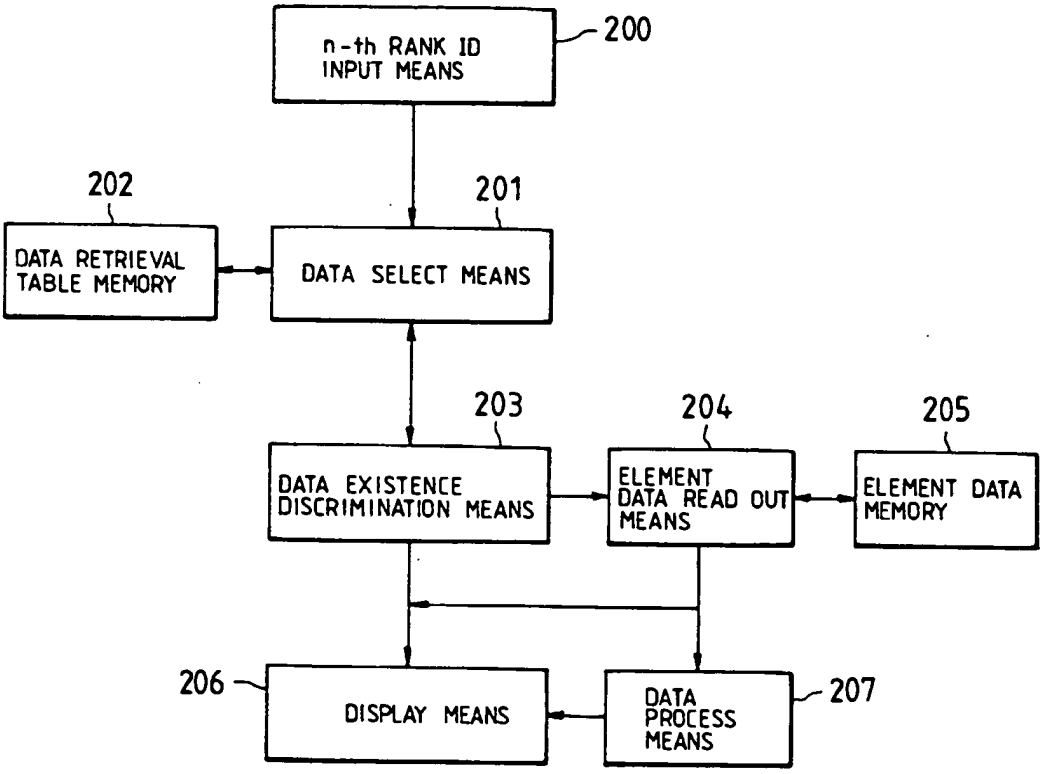


FIG. 34

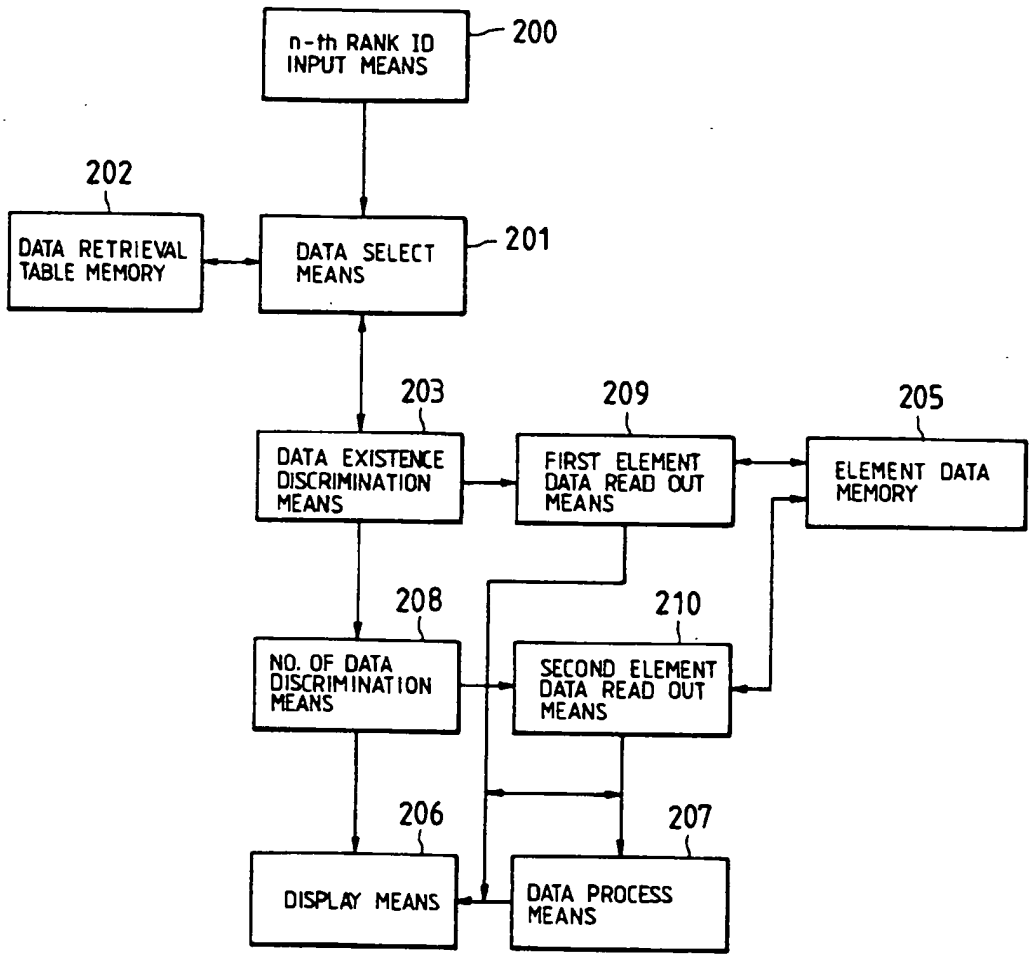
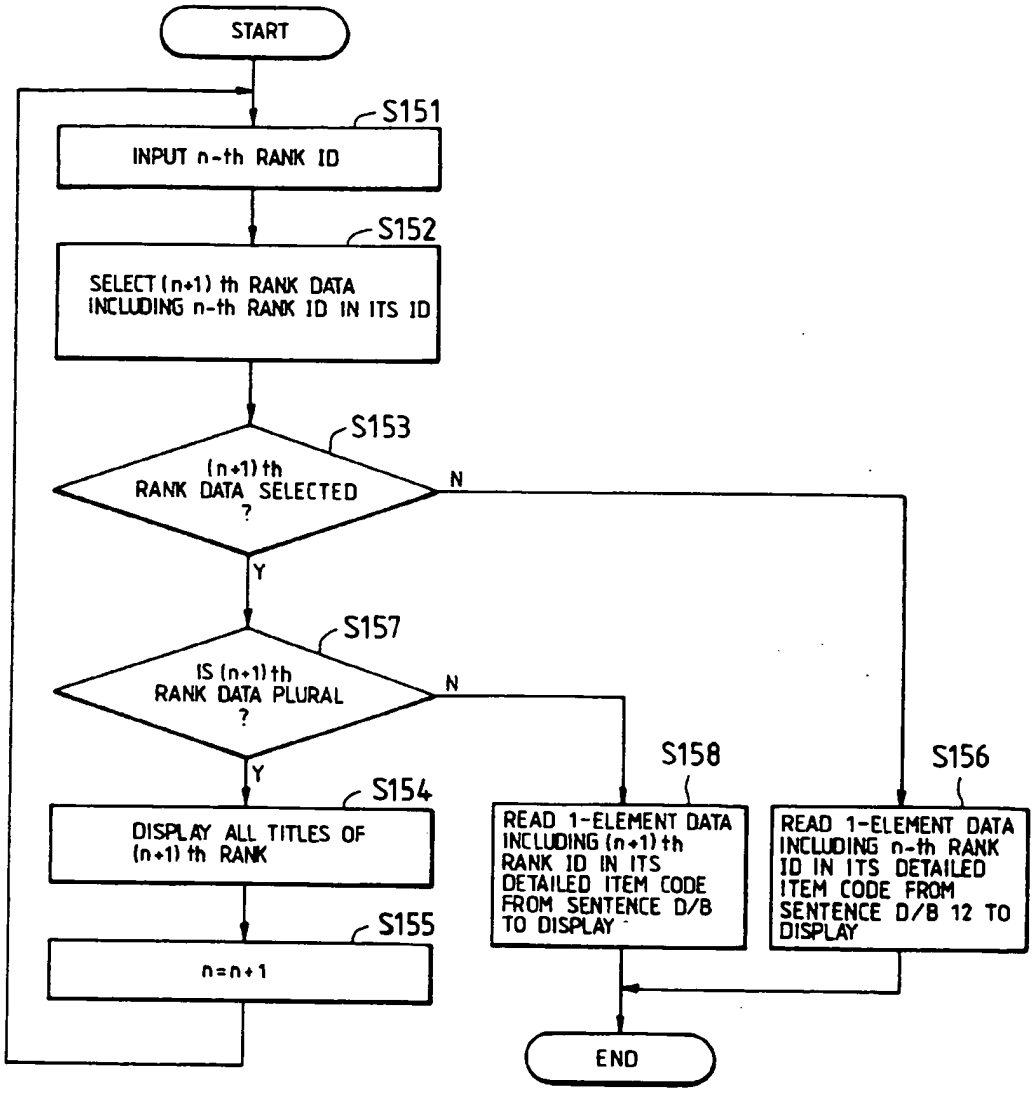


FIG. 35



Requested Patent: EP0651554A1

Title:

METHOD AND APPARATUS FOR THE ADDITION AND REMOVAL OF DIGITAL WATERMARKS IN A HIERARCHICAL IMAGE STORAGE AND RETRIEVAL SYSTEM. ;

Abstracted Patent: EP0651554 ;

Publication Date: 1995-05-03 ;

Inventor(s):

AXMAN MICHAEL STUART EASTMAN K (US); BARADAR ALI R EASTMAN KODAK CO (US); MELNYCHUCK PAUL W EASTMAN KODA (US); RABBANI MAJID EASTMAN KODAK CO (US) ;

Applicant(s): EASTMAN KODAK CO (US) ;

Application Number: EP19940420293 19941025 ;

Priority Number(s): US19930146371 19931029 ;

IPC Classification: H04N1/21; G06F1/00 ;

Equivalents: JP7212712 ;

ABSTRACT:

An image processing technique is described in the context of a hierarchical image storage and retrieval system. The method allows for the controlled addition and removal of digital watermarks from selected image components in the hierarchy. The method adds a digital watermark in a selected image resolution component and the means to remove it in an additional image component termed the watermark removal component. The method employs the encryption of the watermark removal component, and decryption with a special key, or password during authorized retrieval. This technique allows users of a distributed system the convenience of providing the entire image hierarchy on a single storage medium permitting images containing watermarks to be accessed without restriction for browsing and proofing, while the watermark removal requires knowledge and us of a controlled code.

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

(11) Publication number : **0 651 554 A1**

(12) EUROPEAN PATENT APPLICATION

(21) Application number : **94420293.6**

(51) Int. Cl.[6] : **H04N 1/21**, G06F 1/00

(22) Date of filing : **25.10.94**

(30) Priority : **29.10.93 US 146371**

(43) Date of publication of application :
03.05.95 Bulletin 95/18

(84) Designated Contracting States :
DE GB

(71) Applicant : **EASTMAN KODAK COMPANY**
343 State Street
Rochester, New York 14650-2201 (US)

(72) Inventor : **Rabbani, Majid, Eastman Kodak Company**
Patent Legal Staff,
343 State Street
Rochester, New York 14650-2201 (US)

Inventor : **Melnychuck, Paul W., Eastman Kodak Company**
Patent Legal Staff,
343 State Street
Rochester, New York 14650-2201 (US)
Inventor : **Axman, Michael Stuart, Eastman Kodak Company**
Patent Legal Staff,
343 State Street
Rochester, New York 14650-2201 (US)
Inventor : **Baradar, Ali R., Eastman Kodak Company**
Patent Legal Staff,
343 State Street
Rochester, New York 14650-2201 (US)

(74) Representative : **Boulard, Denis et al**
Kodak-Pathé
Département Brevets
CRT-Zone Industrielle
F-71102 Chalon-sur-Saône Cédex (FR)

(54) Method and apparatus for the addition and removal of digital watermarks in a hierarchical image storage and retrieval system.

(57) An image processing technique is described in the context of a hierarchical image storage and retrieval system. The method allows for the controlled addition and removal of digital watermarks from selected image components in the hierarchy. The method adds a digital watermark in a selected image resolution component and the means to remove it in an additional image component termed the watermark removal component. The method employs the encryption of the watermark removal component, and decryption with a special key, or password during authorized retrieval. This technique allows users of a distributed system the convenience of providing the entire image hierarchy on a single storage medium permitting images containing watermarks to be accessed without restriction for browsing and proofing, while the watermark removal requires knowledge and us of a controlled code.

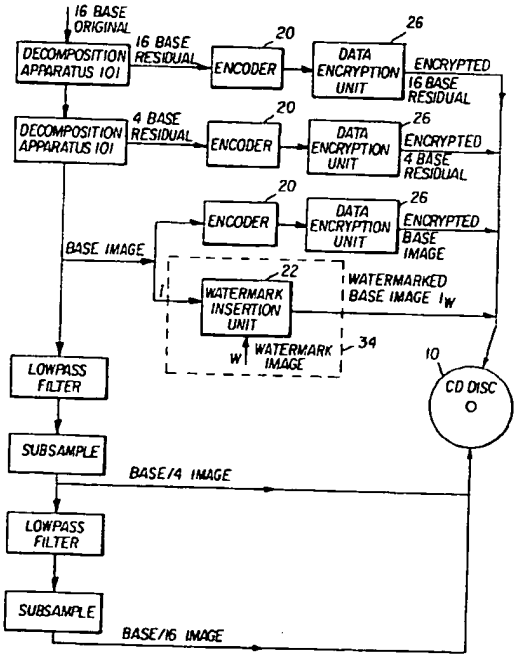


FIG. 2

EP 0 651 554 A1

Cross-reference to Related Application:

The present application is related to U.S. Patent Application Serial No. 08/026,726, entitled "Method and Apparatus for Controlling Access to Selected Image Components In An Image Storage and Retrieval System" filed March 5, 1993, by P. W. Melnychuck and assigned to Kodak, the assignee of the present application.

Technical field Of The Invention

The present invention is related to the field of digital image processing and more particularly to methods and associated apparatuses for adding and removing a digital watermark to and from a selected image resolution and the preventing of unauthorized use of associated higher resolution digital image components.

Background Of The Invention

A number of hierarchical techniques for image coding have been described in the open technical literature and in various patents. Of particular relevance to the present invention are the following publications:

P. J. Burt and E. H. Adelson, "The Laplacian Pyramid As A Compact Code," IEEE Trans. Comm., COM-31, 532-540 (1983).

J. Seberry and J. Pieprzyk, "CRYPTOGRAPHY: An introduction to Computer Security" Prentice Hall, 1988 and the following patents:

U.S. Pat No, 4,969,204 entitled "Hybrid Residual-Based Hierarchical Storage And Display Method For High Resolution Digital Images In A Multiuse Environment," by Paul W. Melnychuck and Paul W. Jones, 1990.

U.S. Pat No, 5,048,111 entitled "Hybrid Subband-Based Hierarchical Storage And Display Method For High Resolution Digital Images In A Multiuse Environment," by Paul W. Jones and Paul W. Melnychuck, 1991.

The publication by Burt, et al. teaches an encoding method for images termed the Laplacian pyramid, the Burt pyramid, or the residual pyramid. In this technique, the original image is lowpass filtered, and this lowpass image is subsampled to take advantage of its reduced bandwidth to provide an image of reduced dimension. This process of lowpass filtering and subsampling is repeated three times to generate a hierarchical structure, or pyramid of images of successively smaller dimensions. The total number of resolution levels are created depending on the application. Each lowpass image in this pyramid is then expanded to the dimensions of the next higher level by upsampling (inserting zeros) and filtering to form a prediction image for that level. This prediction image is subtracted from its corresponding lowpass image in a subtractor to generate difference, or residual, images. The residual images corresponding to the levels of the lowpass pyramid form another pyramid which is termed the Laplacian, Burt, or residual pyramid. This technique is motivated by the fact that the residual images have a reduced variance and entropy compared to the original or lowpass images and may be quantized and entropy encoded to provide efficient storage of the data. Reconstruction is performed by interpolating the decoded lowpass image at the bottom of the lowpass pyramid and adding in the corresponding decoded residual to generate the next level in the lowpass pyramid. This process is iterated until the original image size is reached. A progressive improvement in reconstructed image quality and resolution can thus be obtained by displaying the reconstructed lowpass filtered image at each level of the pyramid. Note that errors introduced in the encoding process are propagated from one level to the next higher level in the decoding process.

The patent to Melnychuck and Jones (U. S. Pat. No. 4,969,204) teaches a modification of the Burt pyramid scheme by extending the lowpass pyramid structure to include one or more lowpass filtered images of successively smaller dimensions beyond the set described by Burt, et al. The advancement in the method of Melnychuck and Jones is that the residual pyramid is not extended to include these corresponding extended smaller dimensions. Hence, the Melnychuck and Jones pyramid contains the Burt pyramid plus additional lowpass filtered images of smaller dimensions. In a hierarchical image storage and retrieval system, the additional lowpass filtered images of smaller dimension can be retrieved directly, without interpolation and addition of residual components. In the context of the present invention, the Melnychuck and Jones pyramid provides for low resolution images that can be used for browsing or proofing. The use of these additional low resolution images for browsing and proofing means that the customer may use a simple retrieval mechanism and need not possess a more complex and hence, more expensive retrieval device that would be used to decode the higher resolution components of the pyramid. Of course, higher resolution images requiring interpolation and residual addition may be used for browsing and proofing as well.

A hierarchical image processing method will be described for the addition and removal of digital watermarks in selected image components, and for the restriction of selected high resolution image components from unauthorized use. An image hierarchy is constructed in the context of a multi-resolution environment whereby the user has the option of selecting the type of display medium and the desired resolution of this display medium. In particular, two types of display media are considered: video monitors and color hard copies, although photographic, thermal imaging,

and other types are also of interest. In Fig. 1 a prior art technique for decomposing, storing, recomposing, and displaying, a digital image using a hierarchical process is shown. An original digital image is decomposed to provide image versions at various resolutions to allow for the display of an HDTV quality image on video, an NTSC quality image with PAL/SECAM compatibility on video, one or more sub-NTSC quality images on video for overviews and browsing, and a very high quality image on color hard copy. Intermediate to the decomposition and recomposition steps, generally are inserted an encoding step, to compress the data for storage which in turn requires a decoding step when the data is read from storage.

Summary Of The Invention

The present invention places a digital watermark in a selected image resolution component and the means to remove it in an additional image component termed a watermark removal component. Encryption of the watermark removal component is used to prevent use of the image for the generation of unauthorized high quality color hard copy. A watermark is a form of graphic overlay that may contain a copyright notice or information regarding the restricted use of the image. In a distributed image system it is common to deliver an image of compromised image quality for purposes of browsing or proofing. A compromised rendition of the image is commonly distributed to prevent full utility or fulfillment of the image without proper payment for the service that generated the image. The term browsing refers to the process of image selection from a plurality of images based on some user-defined criterion. Such is the case when a user may select an image from a catalog of images depicting a particular object. The term proofing refers to the process of image selection based on the degree of desirability of a given image from a plurality of images. Such is the case when a professional portrait photographer distributes a plurality of images to a customer for selection and approval. The terms watermark, browsing and proofing described herein are not limited to the examples described above.

Upon selection of the desired image by the customer, the professional delivers a high quality rendition of the image, most often in the form of a high quality color hard copy. At all times the professional possesses the sole means of generating the high quality hard copy. In a conventional photographic system the means would be the original negatives of the images; in a digital hierarchical system according to the present invention, the means are higher resolution residual components.

In a digital imaging system, and in particular one that includes a hierarchical form of digital storage and retrieval, the professional may use a suitable digital storage medium such as a CD for the distribution of proofs. In an unrestricted environment, the customer may choose a desired image resolution from the hierarchy for the purposes of browsing, proofing, or hard copy fulfillment. In those instances where it is desirable for the professional to deliver the digital storage medium containing the entire image hierarchy to the customer; it is also most economical to record the entire image hierarchy once onto the digital storage medium and avoid having to make a second copy containing only low resolution components for distribution. However, it is also desirable to restrict the use of selected high resolution components for the purpose of full image quality fulfillment until payment has been received. The professional may choose to provide low resolution image components for browsing or proofing, while maintaining restriction of the higher resolution components. Alternatively, he may be required to deliver a proof of high resolution. Such is the case when the image content contains information of small detail and the rendition of this detail is subject to approval via the proof. With traditional photographic prints, the professional may place a stamp, or watermark on a strategic location on the print, so as to render the print useless from a fulfillment point of view. Note with digital images that fulfillment may mean high quality video at NTSC/PAL/SECAM, HDTV, or hard copy. In the present invention, the professional places a digital rendition of the watermark on a selected image component. The removal of the watermark is done through an additional image component containing the reverse of the watermark. The customer, having possession of the digital storage medium CD would possess the means for generating his own high quality hard copy when authorized by the professional. Upon payment to the professional, the professional or his agent provides to the customer the information necessary to remove the watermark for full image quality fulfillment. In the present invention, that information would be an authorization code, key, or password that would be inputted to the image processing system accessing the storage medium, to unlock the restricted high resolution components. An advantage of this technique is that the customer may possess all information pertinent to generating high quality hard copy without the need to physically return to the professional for additional image components.

It may additionally be desirable to use some form of hierarchical image representation for the purpose of browsing or proofing in a distributed system because the hierarchy naturally provides a plurality of resolutions, and hence levels of image quality, from which to choose the proof image. No additional operation of compromising the image is necessary; the professional simply chooses at what resolution level(s) he wants to restrict access.

Systems that use a hierarchical structuring of the image data have not been employed in the past for

distribution purposes because of the lack of means to simultaneously provide low resolution components for browsing and proofing, while offering restricted access to the remaining hierarchical components for full quality image copy. Additionally, the means to generate and remove a digital watermark in a hierarchical image structure had not been previously considered.

The present invention permits the advantages of hierarchical image decomposition to create a series of residual components, direct retrieval of the additional low resolution images according to the Melnychuck and Jones pyramid, the addition and removal of a digital watermark in a selected image resolution component, and prior art encryption methods applied to the watermark removal component and the residuals, to provide for a system of browsing, proofing and restriction of the high resolution image components suitable in a distributed image system. It is assumed that the residual components and the watermark removal component are symbol encoded using the encoder box 20 in Figures 2 and 4 into a binary string of 1's and 0's either via fixed-length coding techniques (where a binary code word of a fixed-length is assigned to each symbol) or variable-length encoding techniques such as Huffman coding or arithmetic coding. The residual data may also be quantized prior to encoding, or it may be encoded in a lossless manner, i.e., without quantization. Data encryption box 26 is applied to the watermark removal component and if desired, also to the encoded quantized (or non-quantized) residual data. It is assumed that the encryption process is reversible. Hence, the decryption box 28 provides the exact data prior to data encryption.

In one embodiment of the invention a storage medium is called for having stored therein at least one low resolution digital image and at least one high resolution digital image, with said high resolution digital image encoded with a watermark that requires an authorization code for removal.

From the foregoing, it can be seen that it is a primary object of the present invention to provide a method and associated apparatus for storing and controllably retrieving digital images stored in a hierarchical format on a suitable digital storage distribution medium that allows the originator of the distribution medium to distribute the medium containing the entire image hierarchy and a controllably removable watermark for the purpose of retrieving low resolution images for browsing or proofing without compromising the originator's need to withhold the means for creating hard copies of the images without the watermark.

It is another object of the present invention to provide the means for controllably inserting and removing a watermark for a digital image.

It is another object of the present invention to provide the means for compromising a selected image component of a hierarchical formatted digital image by adding a digital watermark to the selected image component, and recording the selected image component containing the watermark as part of the image hierarchy on a digital storage distribution medium.

In association with a digital image, it is another object of the present invention to provide a means for creating a watermark removal component, and for controllably restricting access to the watermark removal component.

It is another object of the present invention to provide a means for affixing a watermark to a digital image and for controllably removing the watermark.

The above and other objects of the present invention will become more apparent when taken in conjunction with the following description and drawings wherein like characters indicate like parts and which drawings form a part of the present description.

Brief Description Of The Drawings

Fig. 1 is a block diagram illustrating the prior art Melnychuck and Jones hierarchical storage and display method.

Fig. 2 is a functional block diagram illustrating a hierarchical image decomposition technique incorporating a watermark insertion into an image component.

Fig. 3 is a functional block diagram illustrating a reconstruction technique for reconstructing the images decomposed by the system of Fig. 2.

Fig. 4 is a functional block diagram of another hierarchical image decomposition technique incorporating a watermark insertion into an image component.

Fig. 5 is a functional block diagram illustrating a reconstruction technique for reconstructing the images decomposed by the system of Fig. 4.

Detailed Description of the Invention

In the following description of the preferred embodiments, it will be assumed that the highest resolution of the image hierarchy is composed of 3072 x 2048 pixels and that this resolution is adequate to produce photographic quality originals on an appropriate digital output device. It is also assumed that a moderately high resolution level of the hierarchy composed of 1536 x 1024 pixels is adequate to generate a high quality HDTV display, or a small-sized photographic quality print on an appropriate digital output device. It is also assumed that the lowest resolution levels of 192 x 128 pixels, 384 x 256 pixels, and 768 x 512 pixels are generated and stored onto a digital storage medium such as a CD. These resolution levels are provided to give the reader an insight as to the operation of one or more embodiments of the invention

with the understanding that other resolutions or arrangements may be chosen to suit specific needs without detracting from the teachings of the present invention.

Referring now to Fig. 2, a hierarchical residual decomposition technique, for decomposing a 16BASE original image to form a 16BASE residual, a 4BASE residual, a BASE, a BASE/4, and a BASE/16 image, incorporating the teachings found substantially in Fig. 7 of the patent to Melnychuck and Jones (U. S. Pat. No. 4,969,204), in combination with the present invention is shown. The BASE image is processed in box 34 to incorporate a watermark and to provide a watermarked BASE image.

An example of a watermark insertion box 34 is given by the watermark insertion unit 22 whereby a watermark image W is combined with the input image I to create a watermarked image I_W. In this example, it is assumed that the input image I and the watermark image W are of the same size and the same bit-depth. For example, if the input image I is an 8-bit image representing the luminance component of a color image, the watermark image W would also be an 8-bit image. Similarly, the watermarked image I_W would have the same size and each pixel value would be represented with 8 bits. An example of a watermark insertion unit 22 is one where the input image I and the watermark image W are combined according to the following equation to create the watermarked image I_W

$$I_W(i,j) = I(i,j) + \alpha\, W(i,j)$$

Where (i,j) denotes the two-dimensional location of the pixels in the image and the operation is performed for all the pixels in the input image. The watermark image W is prepared by the originator of the storage medium and may contain the logo of the originator or any other pattern that the originator may wish to use as a watermark. The parameter α, which can be either positive or negative, controls the watermark contrast and is also selected by the originator and can vary from one image to another. Larger magnitudes of α would, in general, create a higher contrast watermark. Also, to guarantee that the watermarked image I_W has the same bit-depth as the input image I, the watermarked image I_W is clipped to the same range as the input image. For example, for an 8-bit image with pixel values in the range of 0 to 255, for every pixel location (i,j), the value of $I_W(i,j)$ is clipped to 255 if the result of the above equation exceeds 255 and is set to zero if that result is less than zero. It should be noted that this example illustrates only one method of implementing the watermark insertion box 34 and the originator of the storage medium may incorporate any other method to generate a watermark that creates the desired effect of inhibiting the use of the image.

The BASE/16, BASE/4, and watermarked BASE images are stored on the digital storage medium 10 in direct (unencrypted) form. The BASE image, which in this case serves as the watermark removal record, is encrypted in the data encryption unit 26. The data encryption unit 26 consists of either a private-key data encryption algorithm (also referred to as symmetric data encryption algorithm) or a public-key data encryption algorithm (also referred to as asymmetric data encryption algorithm) both of which have been explained in the prior art and in the reference book by Seberry and Pieprzyk cited before. Examples of private-key encryption algorithms that can be used in the data encryption unit 26 are either block ciphers such as the Data Encryption Standard (DES) which uses a 56-bit key and operates on blocks of data of length 64 bits at a time, or a stream cipher algorithm such as RC-4, a commercially available encryption software that uses a 40-bit key component. The encrypted BASE image is also stored on the storage medium 10. The 4BASE and 16BASE residual components are also stored on the digital storage medium 10 either in direct (unencrypted) form or in encrypted form depending on the level of security desired by the application. In the case that the encryption of any or all of the residual data are needed, either the same key used in encrypting the BASE image is used or a separate key is used. The use of multiple encryption keys provides the originator of the storage medium with more flexibility in controlling the access to the various resolutions of the image hierarchy.

For browsing or proofing, a procedure illustrated by Fig. 3 is employed. A user retrieves the BASE/16, BASE/4, or watermarked BASE image directly without decryption from the digital storage medium 10. Upon authorization, the user inputs a decryption key(s) to the data decryption unit 28 to allow the decryption of the original BASE image (and the residuals) to be performed. An example of a data decryption unit 28 is a software implementation of a decryption algorithm corresponding to the reverse operation of the encryption algorithm employed in the data encryption unit 26. One example of a set of encryption/decryption algorithms is the Data Encryption Standard (DES) which has been explained in full detail in the reference book by Seberry *et al* mentioned before. Note that the decryption key(s) must be provided by the originator of the storage medium. Upon the decryption of the BASE image and the residual components, these components can be used to arrive at full image quality fulfillment.

In a second embodiment, illustrated in Fig. 4, the 16BASE image is decomposed by decomposition apparatus 101 into a residual pyramid consisting of the 16BASE, 4BASE, and BASE. The BASE image is further decomposed to create the BASE/4 and BASE/16 images, through low pass filtering and subsampling. BASE/4 and BASE/16 are not part of the residual pyramid and hence they are available directly for display on a monitor.

A watermark, as described in the previous em-

bodiment in Fig. 2, is inserted in the BASE image in box 34 to arrive at a watermarked BASE image. This watermarked BASE image is then interpolated to the size of the 4BASE image using linear interpolation as indicated by the interpolator box 24. A difference is formed in subtractor 32 between the original 4BASE image and the interpolated watermarked BASE image to form a modified 4BASE residual that serves as the watermark removal record. The difference in this embodiment versus the first embodiment is that the watermark removal record is the modified 4BASE residual instead of the BASE image. This modified 4BASE residual is encrypted using the data encryption unit 26 as described before and is then stored on the storage media 10 along with the BASE/16, BASE/4, and watermarked BASE image in direct (unencrypted) form. Finally, the 16BASE residual data is stored on the digital storage medium either in direct or encrypted form depending on the application.

For browsing or proofing, the system of Fig. 5 is employed. The user retrieves the BASE/16, BASE/4, or watermarked BASE image directly without decryption from the digital storage medium 10. Upon authorization, the user inputs the decryption key to the data decryption unit 28 to allow the decryption to be performed to generate the modified 4BASE residual. The watermarked BASE image is interpolated using linear interpolation and is added to the decrypted modified 4BASE residual in the reconstruction apparatus 210 to recover the original 4BASE image. If the residuals have not been quantized, the 4BASE image can be exactly recovered. In the case where the residuals have been quantized, some discrepancy between the original 4BASE image and the 4BASE image recovered according to the above scheme would exist. The degree of this discrepancy would depend on the coarseness of the quantizer employed in the quantization of the residual components. Note that the decryption key must be provided by the originator of the storage medium.

It is to be understood that in some instances it may be desirable to place a watermark upon the low resolution images to control their access.

While there has been shown what are considered to be the preferred embodiments of the invention, it will be manifest that many changes and modifications may be made therein without departing from the essential spirit of the invention. It is intended, therefore, in the annexed claims, to cover all such changes and modifications as may fall within the scope of the invention.

Parts List:

10 Digital storage medium (CD-Disc)

20 Encoder
22 Watermark insertion unit
24 Interpolator
26 Data encryption unit
28 Date decryption unit

30 Decoder
32 Subtractor
34 Watermark insertion box

101 Decomposition apparatus

201 Reconstruction apparatus

Claims

1. A storage medium having stored therein at least one low resolution digital image and at least one high resolution digital image, with said high resolution digital image encoded with a watermark that requires an authorization code for removal.

2. The storage medium according to claim 1 and further having stored thereon at least one additional high resolution digital image that is not encoded with a watermark and is accessed with the authorization code in place of the high resolution digital image encoded with the watermark.

3. A storage medium having stored therein at least one low resolution digital image and at least one high resolution digital image in the form of a BASE image, residual image components and a watermark component, with said low resolution digital image, said BASE image or said high resolution image formed by the combination of the BASE image with said residual image components and a watermark component being accessible without an authorization code.

4. The storage medium of claim 3 in combination with an authorization code to remove the watermark component from an accessed high resolution image.

5. A system for controlling the uncompromised use of a high resolution digital image stored on a storage medium as BASE and residual components, comprising:

means for encrypting the residual components stored on said storage medium using a watermark code;

means for accessing the BASE and encrypted residual components;

means for combining the accessed BASE and residual components to reconstruct the high resolution digital image with the watermark code; and

means for authorizing the removal of the watermark code.

6. A system for controlling the uncompromised use of a high resolution digital image comprising:

means for forming a hierarchy of lower resolution digital images from the high resolution digital image;

means for forming residual images that are a function of differences between adjacent images in the hierarchy of lower resolution digital images;

means for encrypting at least one of the formed residual images with a watermark code;

storage means for storing the formed hierarchy of lower resolution images and the at least one encrypted residual image;

means for reconstructing high resolution images by accessing and combining a lower resolution image with a residual image;

means for displaying of the at least one encrypted residual image with the watermark; and

means for controllably removing the watermark code to permit an uncompromised use of the high resolution digital image.

7. A recording medium having stored thereon a plurality of digital images with each of the digital images being comprised of a low resolution digital image component and at least one residual digital image component which is combinable with the low resolution digital image component to form a higher resolution digital image incorporating a watermark which is removable with an authorization code.

8. A method for controlling the use of a digital image stored on a storage medium in a hierarchical form comprised of a BASE image and at least one residual image component, comprising the steps of:

a) associating a watermark with said at least one residual image component;

b) permitting access to the BASE image for low resolution viewing of the digital image;

c) combining the BASE image with the at least one residual image component and an associated watermark to form the digital image for viewing, printing and/or storing; and

d) controllably providing a watermark removal code to remove the watermark from the formed digital image of step c.

9) A storage medium having stored thereon at least one digital image encoded with a watermark that requires an authorization code for removal.

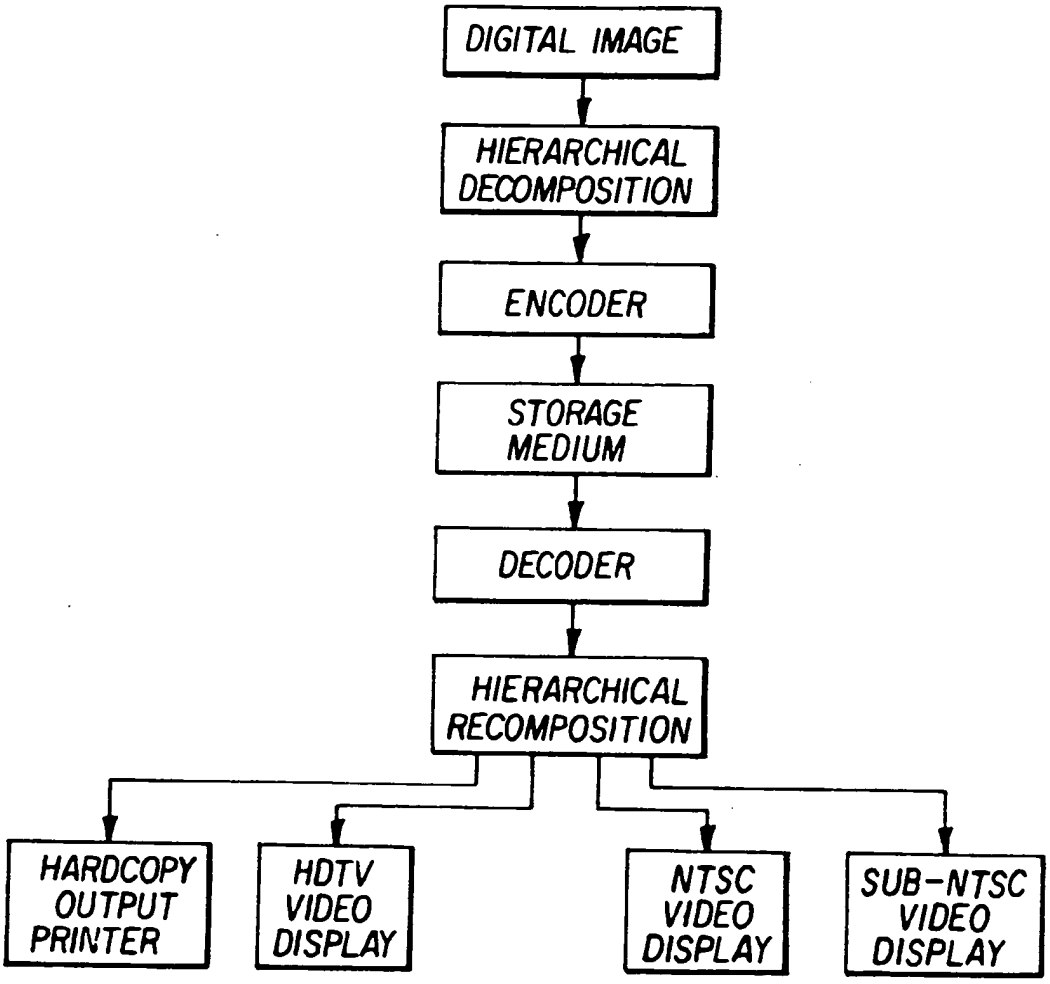


FIG. 1
(prior art)

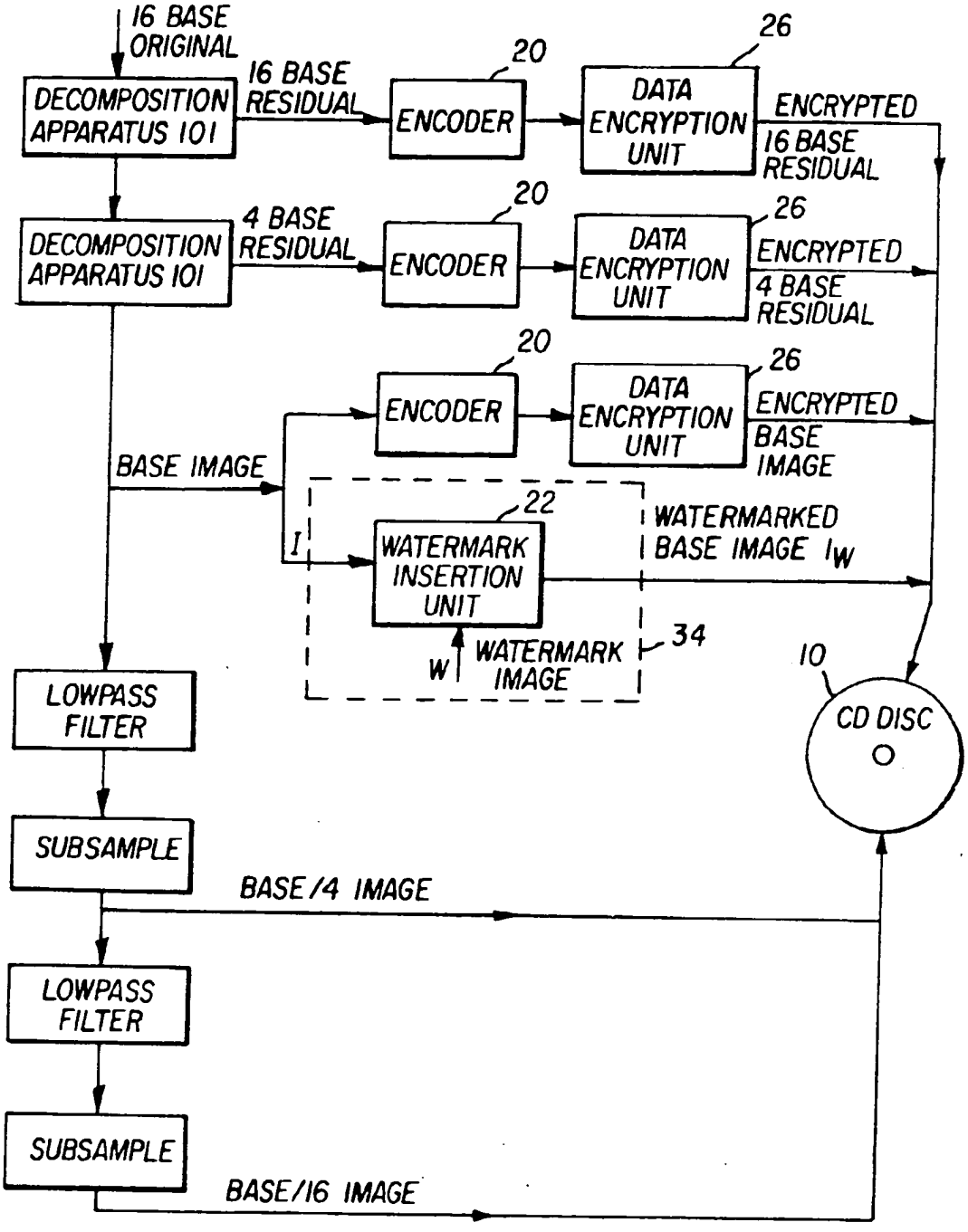


FIG. 2

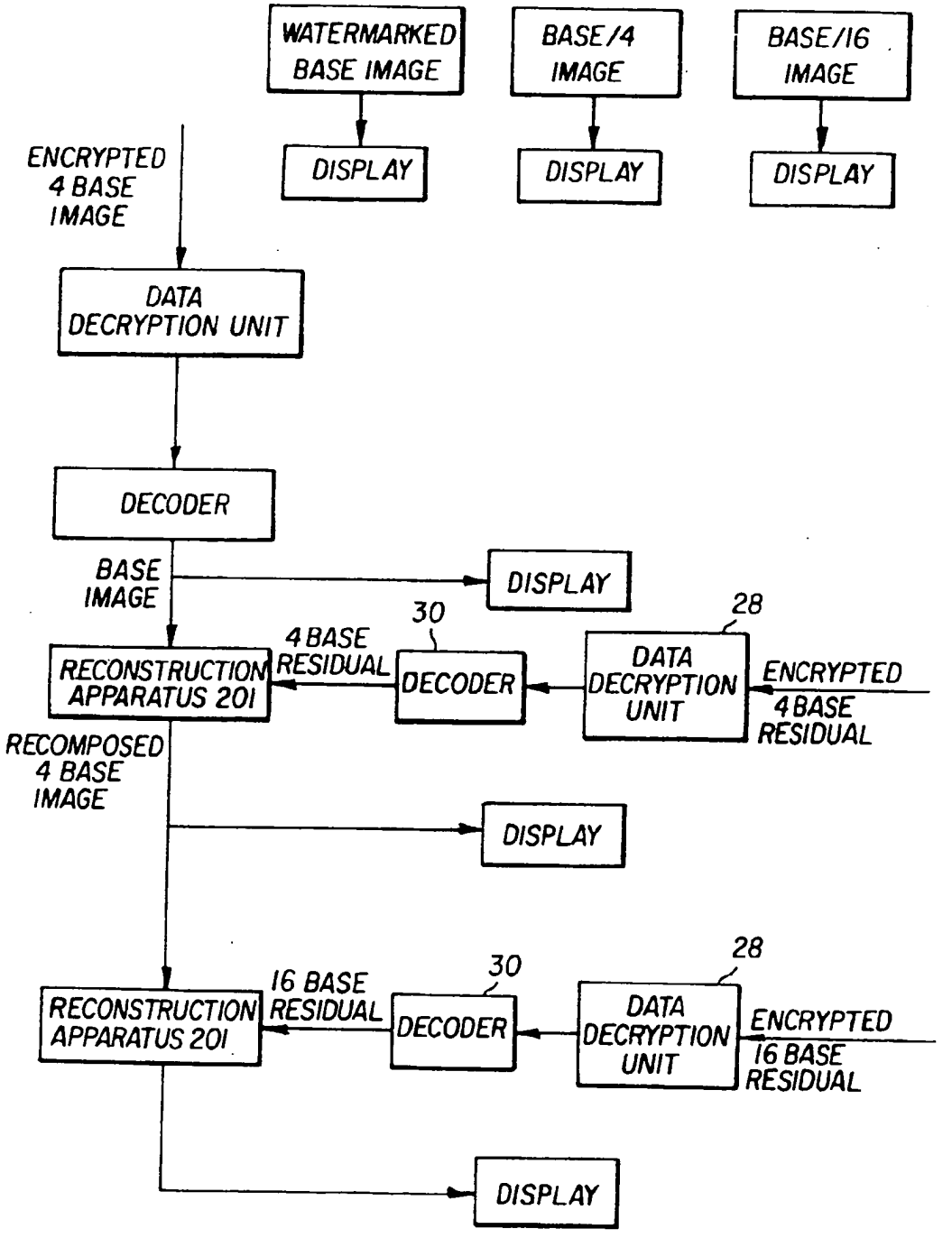


FIG. 3

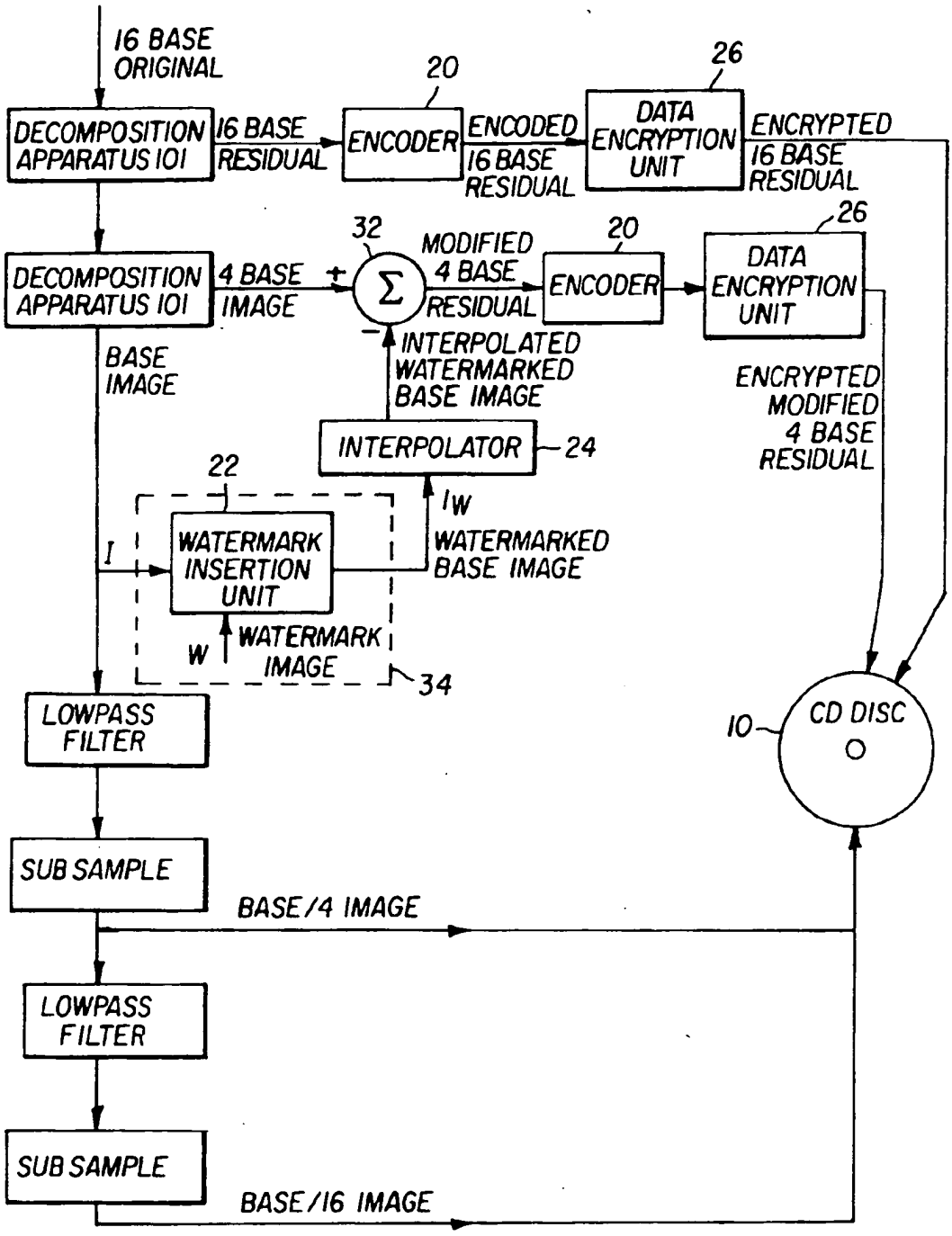


FIG. 4

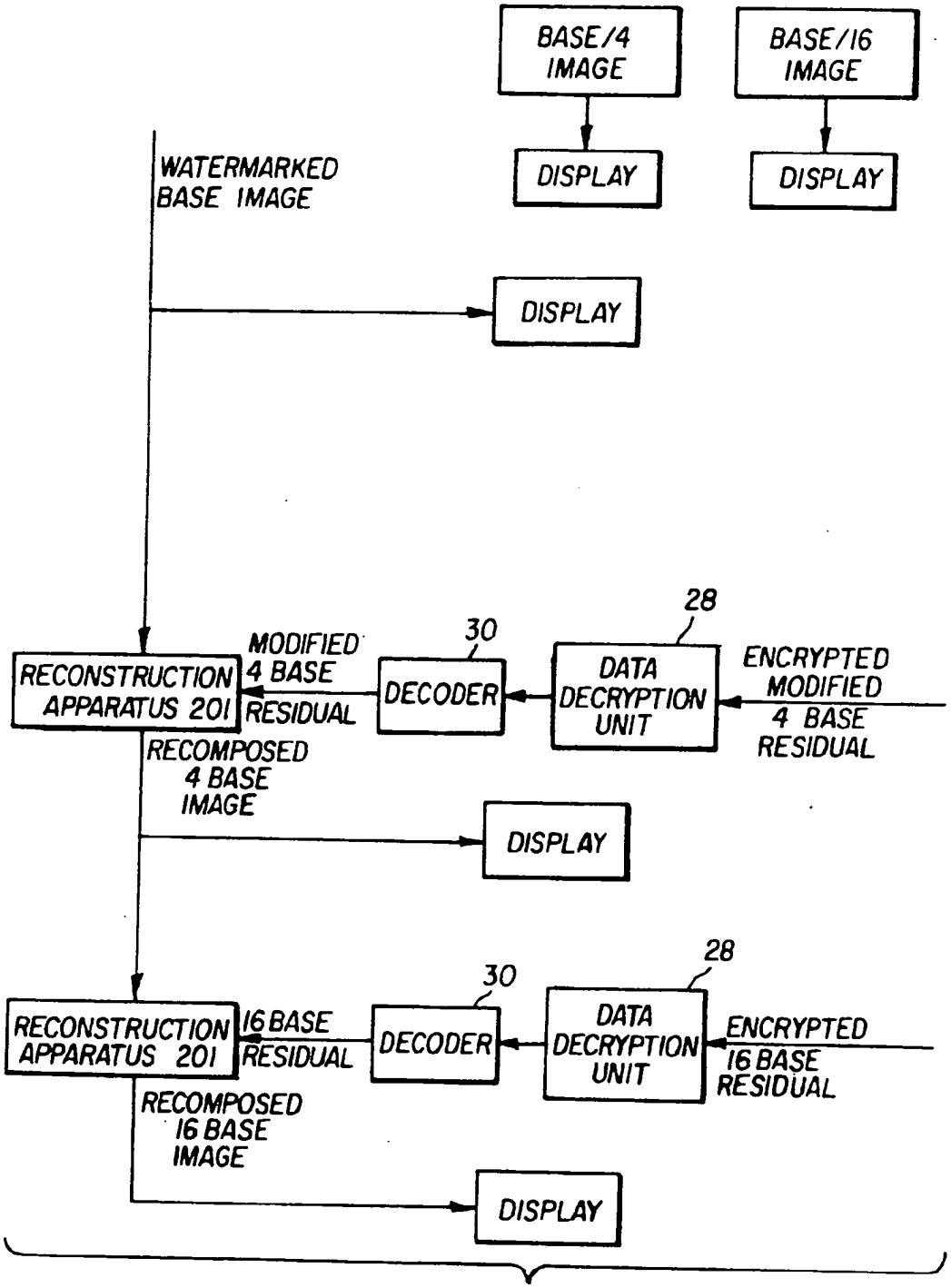


FIG. 5

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 94 42 0293

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)
D,Y	US-A-4 696 204 (MELNYCHUCK ET AL) * the whole document *	1-9	H04N1/21 G06F1/00
Y	ELECTRONICS AND COMMUNICATIONS IN JAPAN, vol.73, no.5, May 1990, NEW YORK, US; pages 22 - 33 N.KOMATSU ET AL 'A Proposal on Digital Watermark in Document Image Communication and Its Application to Realizing a Digital Signature' * figures 1-5 * * page 22, left column, line 1 - page 27, left column, line 23 *	1-9	
P,Y	EP-A-0 614 308 (EASTMAN KODAK) * abstract; figure 2 * * column 4, line 52 - column 5, line 4 *	2	

TECHNICAL FIELDS SEARCHED (Int.Cl.6)
H04N G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	30 January 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)

Requested Patent: EP0668695A2

Title: METHOD FOR RESTRICTING DATA REPRODUCTION. ;

Abstracted Patent: EP0668695 ;

Publication Date: 1995-08-23 ;

Inventor(s): SUGAHRA TAKAYUKI (JP) ;

Applicant(s): VICTOR COMPANY OF JAPAN (JP) ;

Application Number: EP19950300895 19950214 ;

Priority Number(s): JP19940047762 19940222 ;

IPC Classification: H04N5/913; G11B20/00 ;

Equivalents:

CN1066892B, CN1116803, DE69516326D, DE69516326T, JP2853727B2, JP7235131, KR201232 ;

ABSTRACT:

In a reproduction apparatus, for reproducing an original signal conveyed as main data by a data medium such as a recording disk or broadcasting system, with medium protection data which are specific to the data medium being conveyed together with the main data, the apparatus includes a section (12) for generating apparatus protection data which are specific to the reproduction apparatus, a section (13) for combining the apparatus protection data with the medium protection data to define a protection level, and a section (14) for applying the protection level to restrict reproduction of of the original signal, with stepwise variations in restriction occurring in accordance with changes in protection level. The medium protection data may include information for specifying restricted reproduction of portions of the original signal, such as by producing degraded resolution within specified regions of specified frames of a video signal.

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

(11) Publication number : **0 668 695 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number : **95300895.0**

(22) Date of filing : **14.02.95**

(51) Int. Cl.[6] : **H04N 5/913**, G11B 20/00

(30) Priority : **22.02.94 JP 47762/94**

(43) Date of publication of application :
23.08.95 Bulletin 95/34

(84) Designated Contracting States :
DE FR GB

(71) Applicant : **VICTOR COMPANY OF JAPAN, LIMITED**
3-12, Moriya-cho
Kanagawa-ku Yokohama (JP)

(72) Inventor : **Sugahra, Takayuki**
14-7-101, Nakahama,
Isogo-ku
Yokohama (JP)

(74) Representative : **Dempster, Benjamin John Naftel et al**
Withers & Rogers
4 Dyer's Buildings,
Holborn
London EC1N 2JT (GB)

(54) Method for restricting data reproduction.

(57) In a reproduction apparatus, for reproducing an original signal conveyed as main data by a data medium such as a recording disk or broadcasting system, with medium protection data which are specific to the data medium being conveyed together with the main data, the apparatus includes a section (12) for generating apparatus protection data which are specific to the reproduction apparatus, a section (13) for combining the apparatus protection data with the medium protection data to define a protection level, and a section (14) for applying the protection level to restrict reproduction of of the original signal, with stepwise variations in restriction occurring in accordance with changes in protection level. The medium protection data may include information for specifying restricted reproduction of portions of the original signal, such as by producing degraded resolution within specified regions of specified frames of a video signal.

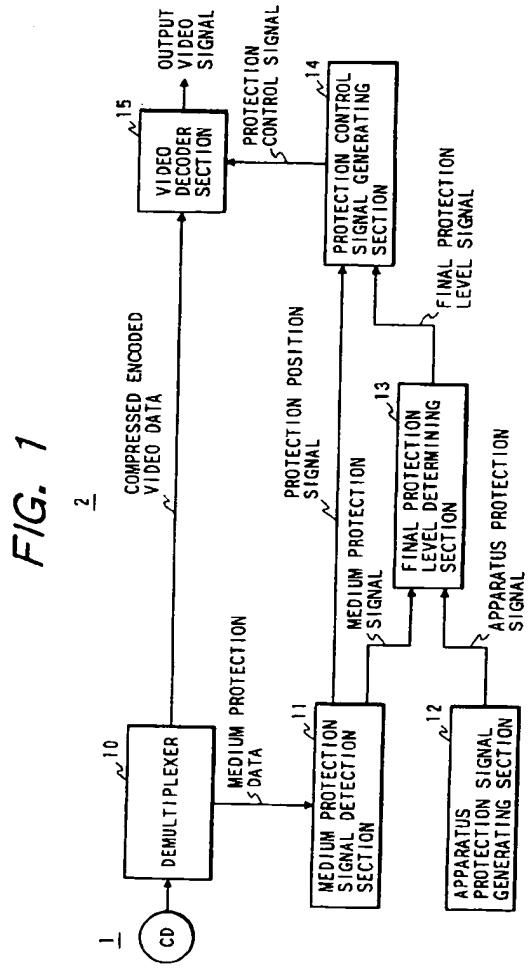


FIG. 1

EP 0 668 695 A2

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a data reproduction protection method, and a data reproduction apparatus for implementing such a protection method, whereby reproduction of a signal represented by digital data such as a recorded digital video signal can be selectively restricted.

Description of the Prior Art

In the following, the term "data medium" is to be understood in a very general sense, as applying for example to broadcasting systems which transmit data such as video and/or audio data, in which case the received data may be the object of reproduction protection, and as applying also to any type of recording medium such as recording disks or tapes, etc., in which case playback data derived from the recording medium may be the object of reproduction protection. The reproduction protection may serve to selectively restrict viewing, hearing or copying of the data.

In the prior art, various types of reproduction protection method have been applied in fields such as CATV (cable television) and satellite television broadcasting. One method is to execute scrambling processing of transmitted video and audio data, and to insert a copyright code into the data, for thereby dividing the data into portions which can be freely reproduced and portions for which a fee must be paid in order to reproduce the data. When a program for which payment of a fee is necessary is received by a receiving apparatus, the program can be unscrambled and reproduced only if specified payment conditions are satisified.

In the case of recorded media, one method of reproduction protection which is applicable to the DAT (digital audio tape recorder) recording system is the SCMS (serial copy management system). With that method, the playback DAT signal from a DAT playback apparatus has a main ID (identification) number which includes a copy inhibit code, whereby a single [copy enable - copy inhibit] sequence is ensured, so that a user can only make a single copy of a pre-recorded digital audio tape.

However with such prior art methods of reproduction protection there are only two control possibilities, i.e. reproduction is made either possible or impossible. It has not been possible hitherto to provide a gradually varying degree of restriction of reproduction of a signal conveyed by a data medium. Thus, such a reproduction protection method can only be used for a single purpose, e.g. for management of payment fees, or for copyright protection. Moreover with such a prior art reproduction protection method, since the data which are to be protected exist only in a transmitting medium or recording medium prior to being reproduced, it has not been possible to provide a varying degree of limitation of reproduction capability in accordance with some condition of the reproduction apparatus. Thus in some cases, the degree of protection may be excessively severe, or excessively lax, so that it is difficult to achieve an effective degree of protection. For example, certain types of scenes recorded on a video tape may be permitted to be viewed in a certain country, such as the U.S.A., but may not be permissible in other countries. It would be thus advantageous to ensure that when that video tape is played on a reproduction apparatus which is sold to the public in such other countries, reproduction protection is automatically applied such that the aforementioned scenes will not be reproduced, or will not be clearly reproduced. However in the prior art, such a feature has not been possible.

SUMMARY OF THE INVENTION

It is an objective of the present invention to overcome the problems of the prior art set out above, by providing a reproduction protection method and apparatus whereby information specifying a degree of restriction of reproduction of an original signal is conveyed (e.g. by a recording medium or signal transmission medium) together with data expressing the original signal, whereby information specifying a degree of restriction of reproduction of the original signal are generated by a reproduction apparatus which operates on the conveyed data, and whereby information specifying a degree of restriction which is actually applied to reproduction of the original signal is derived based on a combination of the restriction information conveyed by the data medium and the restriction information generated by the reproduction apparatus.

More specifically, the invention provides a reproduction protection method comprising:

attaching medium protection data to main data which are conveyed by a data medium, said main data representing an original signal;

supplying the main data and medium protection data, via the data medium, to a reproduction apparatus;

generating apparatus protection data by the reproduction apparatus;

determining a protection level by combining the medium protection data and the apparatus protection data; and

controlling the reproduction apparatus to utilize the main data to reproduce the original signal in accordance with the protection level.

It is a further objective of the invention to overcome the above problems by providing a reproduction apparatus providing reproduction protection, for operating on main data representing an original signal and medium protection data expressing a medium protec-

tion level, said main data and medium protection being conveyed by a data medium, the apparatus comprising:

means for detecting said medium protection data to obtain a medium protection signal expressing said medium protection level;

means for generating an apparatus protection signal expressing an apparatus protection level which has been assigned to said reproduction apparatus;

means responsive to said medium protection signal and apparatus protection signal for determining a final protection level in accordance with a combination of said medium protection level and apparatus protection level;

means for executing reproduction of said original signal by utilizing said main data, including means for selectively restricting said reproduction in accordance with said final protection level.

With such a method and apparatus for reproduction protection, the protection level can determined in accordance with the medium protection data, and hence can be determined in accordance with the wishes of the manufacturer of the data medium, or of the copyright owner of the main data. In addition, the protection level which is actually applied (i.e. the final protection level) is also determined in accordance with the apparatus protection data, which can be specified by the manufacturer or the seller of the reproduction apparatus. As a result, when the main data are to be reproduced (for example, during playback of a recording disk or tape), a graduated degree of limitation of reproduction is implemented, with that degree of limitation being determined by the final protection level, i.e. being determined in accordance with a combination of the requirements of the data medium manufacturer or the copyright owner of the main data and the requirements of the manufacturer or seller of the reproduction apparatus. In that way, considerable flexibility can be ensured in selectively restricting reproduction of signals which are conveyed for example by recording disks or tapes or by broadcasting systems.

In particular, the invention enables such reproduction restriction to be applied to specific frames or sequences of frames of a video signal, or to specific regions within each of a sequence of frames.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a general system block diagram of a first embodiment of the invention, which is a CD player providing reproduction protection in accordance with the present invention, for use in describing the basic principles of the invention;

Fig. 2 is a matrix diagram showing an example of how final protection levels are determined in a reproduction apparatus according to the present invention;

Fig. 3 shows specific examples of how medium protection levels can be assigned;

Fig. 4 shows specific examples of how apparatus protection levels can be assigned;

Fig. 5 is a general system block diagram of a second embodiment, which is a specific configuration for the apparatus of Fig. 1, wherein protection control is applied to a data decompression section;

Fig. 6 shows examples of relationships between final protection level values and video picture visibility grades, for three different methods of protection control of a video signal;

Fig. 7 is a general system block diagram of a third embodiment, which is a specific configuration for the apparatus of Fig. 1, wherein protection control is applied to a video reproduction control section;

Fig. 8 is a conceptual diagram for describing how time-axis protection and spatial-domain protection control can be applied to a video signal with the present invention;

Fig. 9 is a diagram for describing how video, audio and protection data can be conveyed in a data stream in accordance with the MPEG1 standards;

Fig. 10 is a block diagram of an example of the internal configuration of a video reproduction control section in the embodiment of Fig. 7;

Fig. 11 is a general system block diagram of a fourth embodiment, which is a specific configuration for the apparatus of Fig. 1, wherein protection control is applied to a video reproduction control section and also to an audio reproduction control section;

Fig. 12 shows examples of relationships between final protection level values and audio signal audibility grades, for three different methods of protection control;

Fig. 13 is a block diagram of an example of the internal configuration of an audio reproduction control section in the embodiment of Fig. 11;

Fig. 14 is a matrix diagram for illustrating how variable ranges can be defined for final protection level values, by a fifth embodiment of the invention;

Fig. 15 is a general system block diagram of the fifth embodiment, wherein a final protection level can be modified by operation of a switch provided on the reproduction apparatus; and

Fig. 16 shows relationships between medium protection level values and settings of a modification switch in the embodiment of Fig. 15, for each of respective values of the apparatus protection level.

DESCRIPTION OF PREFERRED EMBODIMENTS

Embodiments of the invention will now be described, in which the data medium is assumed to be a recording medium, specifically a CD (compact disk), having video and audio signal data recorded thereon. In the following, such data representing original signals which are to be reproduced from the data medium will be referred to in general as the main data, to distinguish these from protection data, which are also conveyed by the data medium as described hereinafter. It will further be assumed that the original video and audio signals have been encoded by high-efficiency compression encoding using the MPEG1 algorithm, prior to recording. The MPEG1 algorithm is described for example in "International Standards for Multimedia Encoding", edited by Yasuda, published by the Maruzen company in Japan. Fig. 1 is a diagram for describing the general features of the reproduction protection method and reproduction protection apparatus.

Protection Information Provided on Recording Medium

The concept of medium protection data will first be described, referring to Fig. 1, in which data recorded on a CD (compact disk) 1 are read out from the disk to obtain an input signal for a demultiplexer 10 of a CD player 2, which is shown in block diagram form and which is configured to provide reproduction protection control in accordance with the present invention. The CD 1 has video and audio signals recorded thereon as digital data (referred to in the following as the main data), using MPEG1 compression encoding. Protection data, which are predetermined in accordance with the contents of the main data and will be referred to as the medium protection data, are also recorded on the CD 1. The medium protection data consist of information to be used in selectively restricting reproduction of the main data, as described hereinafter. The medium protection data can for example be recorded within the main code and sub-code header regions of the CD 1, or in the user region of the compressed data. The medium protection data expresses a protection level, referred to as the medium protection level, which can for example take values which successively increase in five steps, from 1 to 5, as shown in Fig. 2 and described hereinafter. In that case, the medium protection data can express the medium protection level by 3 bits. If separate medium protection levels are provided for the recorded video and audio data, then these can be expressed by two sets of 3 bits. The higher the number of the medium protection level, the greater is the degree of protection (i.e. the greater becomes the degree of restriction of reproduction of the recorded video or audio signal). The limitation of reproduction can for example by effected, in the case of video data, by operating on the playback data obtained from the CD1 such as to produce an output video signal, from the apparatus of Fig. 1 which will result in a mosaic pattern being produced within all or a specific region of a resultant displayed picture, as described hereinafter.

The medium protection level which is expressed by the medium protection data is predetermined in accordance with the wishes of the manufacturer or seller of the data medium (CD 1), or of the owner of the copyright for the main data. Fig. 3 shows two examples of medium protection data, each expressing 5 values of medium protection level. In the first example, limitation of reproduction is based upon the film classification system (i.e. movie ratings system) which is used in the U.S.A. In the case of a film which is rated "free", no restriction on reproduction is imposed by the medium protection data. If the movie is rated "PG" (i.e. parental guidance), then a moderate degree of restriction (protection level 1) is applied, and so on with increasing degrees of restriction for the "R" and "X" ratings. In the second example, limitation of reproduction is based on the rights of the copyright owner, providing successively increasing degrees of restriction of reproduction as the protection level increases from 1 to 5.

The medium protection data is a combination of data for expressing at least one medium protection level as described above, and protection position information which specifies the position (within the encoded main data) at which the protection is to be applied. It is an essential feature of the present invention that the medium protection data can assign the medium protection level in units of frames of the video signal. That is to say, limitation of reproduction of individual frames can be controlled. In addition, limitation of reproduction of one or more specific regions within a specific frame (or sequence of specific frames) can also be predetermined by the medium protection data. In that case, for example, a region formed of a number of fixed-size blocks of pixels can be converted to a blank region, or filled with a mosaic pattern, in the final display picture that is obtained from the output video signal from the apparatus of Fig. 1. That is to say, the protection position information can be used to specify not only a specific frame, but also to specify one or more specific regions within a frame.

Alternatively, the protection position information can specify an identical medium protection level for the entirety of each of a succession of video signal frames, or specify an identical medium protection level for one or more specific regions within each of a succession of frames.

The term "protection position information" is used here, since it may not be necessary to record explicit data constituting the protection position information on the CD1. Instead, that information can be inherent-

ly constituted by the positions at which respective medium protection data are located within the stream of compressed encoded main data that have been recorded on the CD 1. For example, the apparatus may be configured such that if the encoded data for a video signal frame are immediately preceded by a portion of medium protection data, then that indicates that the medium protection data portion is to be applied to that frame. If the medium protection data portion is to be applied to one or more regions within that frame, rather than the entire frame, then the medium protection data portion which immediately precedes the compressed encoded frame in the recorded data can include at least two values for each of these regions, for specifying the respective positions of the regions. In that case, explicit protection position information must be recorded as data on the CD 1, as part of the medium protection data.

Protection Information Generated by Reproduction Apparatus

With the method and apparatus of the present invention, a reproduction apparatus can be configured to produce predetermined protection data which are specific to that reproduction apparatus. For example, the CD 1 in Fig. 1 is played by a reproduction apparatus 2 which is provided with a presettable memory device which will be assumed to be a ROM (read-only memory) which generates apparatus protection data expressing a protection level referred to in the following as an apparatus protection level. The apparatus protection level is specified beforehand by the manufacturer or the seller of the reproduction apparatus. In the same way as for the medium protection data described above, the apparatus protection level can take a plurality of values, corresponding to respectively different degrees of limitation of reproduction of main data which are obtained from a data medium. It will be assumed that the number of apparatus protection level values is 4, i.e. from 1 to 4, so that the apparatus protection data can be consist of two bits. The higher the apparatus protection level number, the greater becomes the degree of protection, i.e. the greater becomes the degree of reproduction limitation.

The contents of the apparatus protection data ROM cannot be rewritten by the user. In the embodiment of Fig. 1, each time that power to the apparatus is switched on, the apparatus protection data are read out from the ROM, and thereafter reproduction is executed in accordance with a combination of the apparatus protection level specified by the apparatus protection data and the medium protection level which is specified by the medium protection data.

Fig. 4 shows three examples of how the apparatus protection level can be assigned. In the first example, the apparatus protection level is preset in accordance with the country in which the reproduction apparatus is to be used. If the reproduction apparatus is to be used in the U.S.A. for example, then it is possible that the apparatus protection level could be set to a low value such as 1. In the case of a reproduction apparatus which is to be used in other parts of the world, such as Europe, Japan and Taiwan, which have varying degrees of restrictions on video software, the apparatus protection level could be set to higher values, as illustrated. In that way, video scenes which are not permissible in one country can be automatically eliminated (partially or completely), e.g. by insertion of mosaic pattern regions in the resultant display picture, by using the apparatus protection level and medium protection level in combination as described hereinafter.

With the second example in Fig. 4, the apparatus protection level is predetermined in accordance with the type of person who is expected to use the reproduction apparatus. If the reproduction apparatus is to be used only by adults, for example, then the apparatus protection level can be set at a low value such as 1. If the reproduction apparatus will be used by children, the apparatus protection level can be set to a high value such as 4. In that way, video scenes or audio content which are considered unsuitable for children can be partially or completely restricted from being reproduced.

With the third example in Fig. 4, the protection level that is set by the apparatus protection data is predetermined in accordance with the applications for which the reproduction apparatus will be used. For example if the reproduction apparatus is to be sold to the public, then the apparatus protection level can be set to a value such as 1 or 2, whereas if the reproduction apparatus is intended to be used for demonstration purposes in a shop, then the apparatus protection level can be set to a different value, i.e. 3 or 4, and the apparatus of Fig. 1 controlled such that only certain scenes which should be of interest to possible customers of the shop will be displayed. In that way, the apparatus protection level can be set in accordance with the application objectives of the reproduction apparatus.

The overall features of reproduction protection will now be described, referring to Fig. 1. Firstly, data recorded on the CD 1 are read out, as an input signal to the demultiplexer 10. The recorded data on the CD 1 consist of the compressed encoded main data (i.e. compressed encoded video and audio data) which are multiplexed with the medium protection data. The demultiplexer 10 separates the compressed encoded main data from the medium protection data, and supplies the medium protection data to a medium protection signal detection section 11 while supplying the compressed encoded main data to a video decoder section 15. As described hereinabove, the medium protection data may inherently specify protection

position information, or may include explicit protection position information. The medium protection signal detection section 11 serves to detect the protection position information, and generates a corresponding protection position signal, which indicates those portions of the main data to which the medium protection data applies (e.g. specific video signal frames, and/or block regions within specific frames). The protection position signal is supplied to a protection control signal generating section 14. The medium protection signal detection section 11 is further responsive to the medium protection data for generating a corresponding medium protection signal, which expresses the medium protection level and is supplied to a final protection level determining section 13.

The final protection level determining section 13 basically consists of a matrix ROM in this embodiment, i.e. a ROM which stores a pattern of relationships between respective combinations of medium protection levels and apparatus protection levels and resultant final protection levels. The operation of the matrix ROM will be described referring to the matrix diagram of Fig. 2, which shows an example of how the contents of that ROM are read out in response to combinations of medium protection level and apparatus protection level values. There are five possible values (designated as A to E respectively) for the final protection level, successively increasing in degree of reproduction limitation in the sequence A, B, C, D, E. Fig. 2 shows an example of various values of the final protection level which are determined by respective combinations of values of the medium protection level and apparatus protection level, i.e. the values in the range A to E which are located at respective intersections between rows and columns of the matrix in Fig. 2. Thus for example if the medium protection level is 4 and the apparatus protection level is 2, then the final protection level will be B.

It is necessary to clearly distinguish between the ROM of the apparatus protection signal generating section 12 and the matrix ROM of the final protection level determining section 13. The contents of the ROM of the apparatus protection signal generating section 12 can be set in accordance with the requirements for a particular reproduction apparatus, whereas the contents of the matrix ROM of the final protection level determining section 13 will in general be common to a large number of reproduction apparatus units.

The final protection level determining section 13 generates an output signal, referred to as the final protection level signal, which expresses the final protection level that has been determined, and supplies that signal to the protection control signal generating section 14. In response to that signal, and the protection position signal, the protection control signal generating section 14 generates a signal referred to as the protection control signal, which is supplied to control a section which will be referred to as the video decoder section 15. For simplicity of description, only video signal reproduction will be considered at this stage, and the video decoder section 15 should be understood as a section which converts the compressed encoded video data (main data) from the demultiplexer 10 to a standard (analog) video signal. As the stream of compressed encoded main data flows into the video decoder section 15 from the demultiplexer 10, the protection control signal controls the video decoder section 15 such as to apply reproduction protection in accordance with the final protection level values, at the respective positions within that data flow which are specified by the protection position information. The video decoder section 15 thereby produces an output video signal which will result in video pictures in which reproduction of the main (video) data is limited in accordance with the final protection level values. As will be understood from the above, the final protection level values may change from frame to frame of the video signal, in accordance with changes in the medium protection level.

A first example of limitation of reproduction in accordance with the final protection level will be described, which is implemented by controlling the expansion and decoding of the main data by the video decoder section 15. The example will be described referring to the embodiment of Fig. 5, in which the reproduction apparatus is again a CD playback apparatus, designated by numeral 3. Only the operation with regard to the video data of the main data from the CD 1 will be described. In Fig. 5, a specific configuration for the video decoder section 15 of Fig. 1 is shown, made up of a variable-length decoding section 15a, a dequantizer section 15b, an inverse transform section 15c and a video reproduction control section 15d, with the compressed encoded video data being supplied from the demultiplexer 10 to the variable-length decoding section 15a and with the final output (analog) video signal being produced from the video reproduction control section 15d. In this embodiment, the video data have been recorded on the CD 1 after being subjected to compression by discrete cosine transform processing, and reproduction limitation is controlled by controlling the accuracy of inverse DCT (discrete cosine transform) processing which is effected by the inverse transform section 15c. The protection control signal which is produced from a protection control signal generating section 14a and supplied to control the inverse transform section 15c is derived based on the medium protection data and apparatus protection data, in combination, as described above for section 14 in Fig. 1, i.e. the protection control signal applies control in accordance with the final protection level.

The demultiplexed compressed encoded video data read from the CD 1 are subjected to variable-length decoding in the variable-length decoding sec-

tion 15a, and then to dequantization in the dequantizer section 15b. The resultant compressed data are then subjected to inverse DCT processing in the inverse transform section 15c, with the transform processing being selectively modified in accordance with the protection control signal. The resultant decompressed data are then processed in the video reproduction control section 15d to obtain the final output video signal. The effect of reproduction limitation controlled by the protection control signal acting on the inverse transform section 15c is to selectively produce a degree of blurring or formation of a mosaic pattern within the pictures which are displayed using the final output video signal. Such a degree of blurring will be referred to as the visibility grade. An example of the relationship between the visibility grade and the final protection level values A to E (which are determined by the protection control signal generating section 14a as described hereinabove, and are expressed by the protection control signal) is shown in the leftmost column of Fig. 6. If the protection level is A, i.e. minimum limitation of reproduction of the video data, then the protection control signal is set to a state whereby it does not affect the operation of the inverse transform section 15c. If the protection level is B, then the protection control signal controls the inverse transform section 15c such as to operate on a block size of 8 x 8 picture elements (i.e. the same block size which was utilized in the original DCT processing), using the DC component value for each block, but using only two of the AC transform coefficients, with the values of all of the other transform coefficients being forcibly set to zero. This will result in a substantial lowering of resolution of a display picture that is produced based on the output video signal. If the protection level is C, then the protection control signal controls the inverse transform section 15c such as to operate on a block size of 8 x 8 picture elements of a video signal frame, using only the DC component for each block, (i.e. the DC component is the only transform coefficient used). In this case, since all of the picture elements within an 8 x 8 picture element block will have identical video signal values, this will result in a mosaic pattern being formed in the finally obtained picture. If the protection level is D, then the protection control signal controls the inverse transform section 15c such as to operate on a block size of 16 x 16 picture elements, using only the DC component values. This will again result in a mosaic pattern being formed, in which the blocks of the pattern are of larger size then for the case of protection level C, i.e. a mosaic pattern of macroblocks is formed, thereby further degrading the degree of visibility of the resultant picture. If the protection level is E, then the video data obtained from the inverse transform section 15c are replaced by different video data (produced from a source not shown in the drawing), which are produced from the video reproduction control section 15d as the final video signal, and which will produce a predetermined picture or pictures. Such a predetermined picture might for example display a warning message concerning copyright protection.

With the MPEG1 algorithm, a block size of 8 x 8 picture elements is used in the DCT processing. The transformed block is expressed by a DC component (i.e. DC coefficient) and a plurality of coefficients (the AC coefficients) which represent signal level values at respectively different successively increasing frequencies. Thus if for example the inverse transform section 15c is controlled such that only the DC coefficient and the two lowest-frequency AC coefficients are used, in the inverse DCT operation for each block, then a specific reduction in resolution of all (or a specific part) of the resultant display picture can be achieved in a very simple manner.

Similarly if only the DC component for a block is used in the inverse DCT processing, with all of the AC coefficients set to zero, then all of the signal level values for the picture elements of a block will be set to an identical value, in the resultant video signal obtained from the inverse DCT operation. Hence, a mosaic pattern can very easily be formed in the resultant picture that is obtained using the final output video signal. Moreover if, for each of respective 16 x 16 element macroblocks (i.e. each consisting of four 8 x 8 picture element blocks), only the DC coefficient for a specific one of the 8 x 8 element blocks is used (for example, the DC coefficient for the upper leftmost one of the 8 x 8 element blocks) in the inverse DCT processing for all of the four 8 x 8 element blocks constituting the 16 x 16 element macroblock, with all of the AC coefficients set to zero, then a mosaic pattern will be formed which is substantially coarser than the mosaic pattern which is formed by using the 8 x 8 element blocks.

It can thus be understood that with the above embodiment of the invention, applied to video data which have been subjected to high-efficiency compression encoding using a data transform operation, stepwise changes in a degree of restriction of reproduction of the video data can easily accomplished by effecting stepwise changes in a degree of resolution of a finally obtained picture, or in a portion of that picture, and that such stepwise changes in resolution can be easily controlled in accordance with the final protection level which has been established based on the medium protection data and apparatus protection data. In particular, when such control is applied to the inverse DCT processing, it is possible to easily effect stepwise changes in the visibility grade, i.e. in the picture resolution, through use of unit blocks of picture element values which are basic to the transform processing. Such a type of control of the visibility grade, operating within each frame of the video signal, can be considered as applying protection in a (2-dimensional) spatial domain.

With a video data encoding method such as the DCT method, the output digital signal that is produced from the inverse DCT circuit consists of sequential sets of data, each consisting of successive picture element values for the respective picture elements of a unit block (e.g. a block of 8 x 8 picture elements) of a video signal frame. In order to convert such a digital signal into a normal digital video signal, it is necessary to first temporarily store the data produced from the inverse DCT circuit in a video memory (e.g. a frame memory), then to read out the video data in the correct sequence (i.e. as successive picture element values in successive picture scanning line intervals). That operation is the basic function of the video reproduction control section 15d in Fig. 5. Fig. 7 shows another embodiment of the invention, in which control for reproduction protection is applied to the video data which have been produced from the inverse transform section 15c, i.e. in which control by the protection control signal in accordance with the final protection level is applied to a video reproduction control section which is configured such as to respond appropriately to the protection control signal, and is designated as 15d'. Apart from this feature, the configuration and operation of this embodiment is identical to that of Fig. 5 described above. The video reproduction control section 15d' includes a frame memory, into which output data from the inverse transform section 15c are temporarily written, and then read out in the appropriate sequence as described above, to obtain the final output video signal.

In this embodiment, the degree of reproduction limitation is controlled by "thinning out" frames of the video data that are used to form the final output video signal, with the degree of "thinning out" being determined by the final protection level. That is illustrated by the central column in Fig. 6, in which such a type of control is referred to as time domain protection. In the example of Fig. 6, when the final protection level is A, then all of the frames of video data which are successively written into the frame memory of the video reproduction control section 15d' are used to form the final output video signal. If the final protection level is B, then the protection control signal from the protection control signal generating section 14b controls the video reproduction control section 15d' such that only one out of every 15 frames of video data supplied from the inverse transform section 15c is used to form the final output video signal. Specifically, one out of every fifteen frames of video data from the inverse transform section 15c is held stored in the frame memory of the video reproduction control section 15d' for fifteen successive frame periods, and is repetitively read out during that time, to form the final output video signal. Thus a type sample-and-hold operation is performed using the frame memory in the video reproduction control section 15d', whereby the finally obtained picture will change once in every 0.5 seconds. If the final protection level is C, then the video reproduction control section 15d' is controlled such that the contents of the frame memory are updated only once in every 60 frame periods, i.e. the finally obtained picture will change only once in every 2 seconds. If the final protection level is D, then only the video data of certain specific frames (or one specific frame) are written into the frame memory of the video reproduction control section 15d' and read out to obtain the final output video signal. In that way, for example, only a portion of the video data (e.g. a portion which is not subject to copyright protection) will be displayed. If the protection level is E, then the video data for a predetermined picture are written into the frame memory of the video reproduction control section 15d' and repetitively read out, to display only that predetermined picture, which can be for example a warning message concerning copyright protection.

Alternatively, control of the degree of reproduction limitation can be performed by arranging that the protection control signal from the protection control signal generating section 14b acts on the video reproduction control section 15d' such as to vary (in accordance with the final protection level) the number of gradations provided by each video data sample, i.e. to vary the number of amplitude levels that can be expressed by each sample. That can be performed by setting one or more low-order bits of each data sample to a fixed value, e.g. 0. For example if the LSB is always set to 0, then the number of possible gradations is reduced by half, and a corresponding lowering of resolution of the finally obtained display picture is achieved. The bits in each digital data sample which are not fixed in that way will be referred to in the following as the effective bits of the sample. Such gradation control based on the numbers of effective data bits is illustrated by the right-side column in Fig. 6. In that example, if the final protection level is A, then the protection control signal from the protection control signal generating section 14b has no effect on the operation of the video reproduction control section 15d', so that each video data sample used to form the output video signal has the standard number of effective bits, i.e. 8 bits. If the final protection level is B, then the protection control signal controls the video reproduction control section 15d' such that the number of effective bits/sample of the output video signal is reduced to 4 (i.e. by rounding-off the low-order 4 bits to zero). If the final protection level is C, then the protection control signal controls the video reproduction control section 15d' such that the number of effective bits/sample is 2 (i.e. all except the two high-order bits are set to zero), so that the picture resolution is further degraded. Similarly, if the final protection level is D, then the video reproduction control section 15d' is controlled such that the number of effective bits is reduced to 1. If the protection level is E, then the video data for a predetermined picture are written

into the frame memory of the video reproduction control section 15d' and repetitively read out as the final output video signal, to display only that predetermined picture.

It would be equally possible, as indicated by the broken-line connection from the protection control signal generating section 14b to the dequantizer section 15b in Fig. 7, to arrange that the protection control signal from the protection control signal generating section 14b acts on the dequantizer section 15b such as to vary (in accordance with the final protection level) the number of effective bits of each output datum from the dequantizer section 15b. That will provide a similar effect to that described above for the case in which control is effected through the video reproduction control section 15d'.

Another method which may be used to control the degree of reproduction limitation is to apply the protection control signal from the protection control signal generating section 14b such as to control the variable-length decoding section 15a. In that case, the protection control signal is arranged to act on the variable-length decoding section 15a such that, as the protection level is increased from A to E, data having a long code length are set to zero, i.e. are ignored. This will result in a lowering of resolution in the final picture that is obtained from the output video signal.

It should be noted that reproduction limitation control can be executed by a combination of control acting along the time axis and control acting in a spatial domain (i.e. within individual frames). That point is illustrated conceptually in Fig. 8, in which successive vertical lines 20 represent sequential frames of the video signal that is recorded on the CD 1. (For simplicity of description, it will be assumed that the final protection level is identical to the medium protection level). Together with each video signal frame data portion on the CD 1, a medium protection data portion is recorded, which may include position information specifying a region within the frame within which display resolution is to be lowered, to a degree that is in accordance with the final protection level. In this example there are two possible basic display conditions for each frame, i.e. non-display or display (with one or more degraded resolution regions possibly being formed). To achieve this, the medium protection data assigned to each frame includes a 1-bit flag, whose 1 or 0 logic state designates either display or non-display for the frame. If that flag bit indicates that none of the frame is to be displayed, the condition is indicated by a "x" symbol in Fig. 8, while if the flag bit indicates that the frame is to be completely or partially displayed, that condition is indicated by a "o" symbol in Fig. 8. In the example of Fig. 8, a "o" condition is specified for each of the ten consecutive frames designated as F_A, indicating that each frame is to be displayed. In addition, the medium protection data of each of these frames includes position information for a degraded resolution region. The resultant display picture is designated by numeral 21, containing a degraded resolution region 22, which is rectangular and is shown as a hatched-line region. The degraded resolution region 22 is formed by a plurality of 16 x 16 element macroblocks, and for each of the frames F_A, the corresponding position information in the medium protection data specifies two addresses of macroblocks (designated as 22a and 22b, located at the upper left-side and lower right-side corners of the mosaic region 22) within the frame, to thereby specify the position and size of a rectangular region which is the degraded resolution region 22.

Similarly, a set of four successive frames F_B is each to be displayed, but with a degraded resolution region formed in the final display picture, as indicated by numeral 24. In this case the degraded resolution region is formed of two adjoining rectangular regions, so that it is necessary for the position information in the medium protection data to specify the positions of two pairs of macroblocks within the frame, i.e. the pair of addresses of macroblocks 24a, 24b and the pair of addresses of macroblocks 24c, 24d in Fig. 8.

It can thus be understood that in this case, reproduction protection is applied by a combination of control with respect to the time axis, and control with respect to (2-dimensional) space within each frame. It can be further understood that the invention enables extremely precise control of reproduction limitation, which is determined in accordance with the final protection level.

With data transmission in accordance with the MPEG1 system (i.e. based on the ISO-11172-3 standards), data are transmitted as successive packs of data, which are time-division multiplexed, as illustrated by the data flow 40 shown in Fig. 9. Each pack is made up of a leading portion such as the portion 41, which contains information including a pack start code and a stream identifier which distinguishes the data conveyed by that pack from that of other packs (e.g. to distinguish between video, audio or other data), and a main data portion such as portion 42. In this example each main portion consists of either compressed encoded video data such as portion 42, encoded audio data such as portion 43, or protection data such as portion 44. In this example it will assumed that protection of the form shown in Fig. 8 is applied to the video data, so an individual protection data portion may be assigned to each of a plurality of video signal frames, i.e. video data frames can be conveyed by respective packs, each preceded by a protection data pack. In that case, each protection data pack in the example of Fig. 9 consists of a 3-bit portion which specifies the medium protection level, a 1-bit frame flag specifying whether or not the frame is to be displayed (as described above for Fig. 8), a portion which specifies the number of macroblock start/end address pairs (to be utilized when at least

one degraded resolution region is to be formed within the frame, e.g. as for each of the frames F_A and F_B in Fig. 8), followed by the pairs of macroblock start/end addresses (e.g. the pair of addresses of macroblocks 24a, 24b, then the pair of addresses of macroblocks 24c, 24d, for each of the frames F_B in the example of Fig. 8).

Fig. 10 shows a specific internal configuration for the video reproduction control section 15d' of the embodiment of Fig. 7. In Fig. 10, the output data from the inverse transform section 15c are supplied via a line 50 to a gradation control section 51, which is controlled by one of two protection control signals that are generated from the protection control signal generating section 14b, and resultant output data from the gradation control section 51 are transferred through a switch 52, which is controlled by the other one of the protection control signals. Data transferred through the switch 52 are written into a frame memory 53, and are subsequently read out from the frame memory 53 in the appropriate sequence to constitute successive frames of the original video signal. The digital video signal thereby produced from the frame memory 53 is supplied to a digital/analog converter 54, to obtain an analog video signal as the final output signal.

So long as the switch 52 is held closed, the contents of the frame memory 53 will be completely updated once in each frame period of the video signal, so that data of a new frame will be sequentially read out from the frame memory 53. However if the switch 52 is held closed during an integral number of frame intervals, then the most recently stored contents of the frame memory 53 will be repetitively read out during each of these frame intervals, i.e. the last frame will be continuously outputted. It will thus be apparent that this circuit can implement the time-axis protection operation described above on successive frames, if the switch 52 is controlled in accordance with the status of the frame bit that is contained in the medium protection data.

The gradation control section 51 operates on each digital video signal sample (in general, each 8-bit datum) that is supplied from the inverse transform section 15c, to set the low-order bits of each sample in accordance with the final protection level. For example referring to the right-side column in Fig. 6, if the final protection level is B, then the gradation control section 51 sets all of the four low-order bits of each sample to a predetermined value, e.g. 0. If the protection level is D, then all of the seven low-order bits of each sample are set to 0.

If reproduction restriction is to be applied within a video signal frame, then the protection control signal generating section 14b responds to the protection position signal such as to apply the above-mentioned protection control signal to the gradation control section 51 during one or more specific time intervals within the corresponding frame interval, with each of these specific time intervals being determined based on one of the macroblock start/end address pairs which are shown in Fig. 9, described above. In that way, reproduction limitation can be applied within specific regions of a frame, as illustrated in Fig. 8, i.e. the spatial-domain protection operation described above can be applied.

It will be understood that in practical terms, each "start address" will define a time-axis position, within a frame period, of the data sample corresponding to an uppermost left-side pixel of a rectangular region within the frame, while the "end address" similarly defines the position of a data sample corresponding to a lowermost right-side pixel of that region. Such time relationships can be readily established by well-known techniques for operating on a digital video signal, so that detailed description is omitted.

In the above embodiments, only reproduction protection of video data has been described. Fig. 11 shows another embodiment of the invention, in which reproduction protection of both audio and video data is applied. Only the points of difference between this embodiment and previous embodiments will be described. A CD reproduction apparatus 5 of this embodiment differs from that of Fig. 7 by including circuits for decoding and dequantizing an encoded digital audio signal that has been recorded on the CD 1, by an audio decoding section made up of a variable-length decoding section 16a, a dequantizer section 16b, a sub-band combining section 16c and an audio reproduction control section 16d. The compressed encoded audio data are separated from the video and protection data contained in the input data stream, by the demultiplexer 10, and are supplied as input data to the variable-length decoding section 16a, with an output audio signal being produced from the audio reproduction control section 16d. This embodiment further differs from that of Fig. 7 in that the protection control signal generating section 14c of this embodiment produces not only a first protection control signal which acts on either the video reproduction control section 15d' or dequantizer section 15b to apply video signal reproduction protection by varying the number of bits per datum, as described hereinabove for the protection control signal of the embodiment of Fig. 7, but also a second protection control signal which acts on the audio reproduction control section 16d or dequantizer section 16b to apply audio signal reproduction protection, as described in the following. For simplicity of description, it will be assumed that the second protection control signal is produced in accordance with the final protection level that is derived for reproduction protection of the the video data, as described hereinabove. However in general, separate medium protection levels and separate apparatus protection levels would be specified for the video and audio data, i.e. to obtain separate final protection levels for video and audio data.

Fig. 12 is a table illustrating three possible methods of applying audio signal reproduction protection with the embodiment of Fig. 11. With each of the three examples shown in Fig. 12, five different audibility grades can be selected for the output audio signal produced from the audio reproduction control section 16d, in accordance with the final protection level, to effect audio signal reproduction protection. Firstly, the method illustrated by the leftmost column in Fig. 12 will be described. In this case, audio signal reproduction protection is applied by selectively restricting the bandwidth of the output audio signal produced from the audio reproduction control section 16d in accordance with the final protection level. If MPEG1 audio signal compression is used, then assuming a sampling frequency of 48 KHz, a bandwidth of 24 KHz is available for the output audio signal. In this example, if the final protection level is A, then no bandwidth restriction is applied, i.e. the audio bandwidth is 24 KHz. If the final protection level is B, then the audio signal bandwidth is restricted to 18 KHz, if the protection level is C the bandwidth is restricted to 12 KHz, if the protection level is D the bandwidth is restricted to 6 KHz, and if the protection level is E then no audio output signal is produced.

Since MPEG1 audio compression utilizes sub-band encoding with 32 bands, such bandwidth restrictions can be effected by causing the second protection control signal to act on the dequantizer section 16b such as to set the inverse quantization values corresponding to certain high-frequency bands to zero. Thus, audio reproduction protection by bandwidth control can be easily implemented.

A second method of audio signal reproduction protection will be described referring to the central column in Fig. 12. In this case, time-axis protection is applied, by "thinning-out" of audio signal sample values that are supplied from the sub-band combining section 16c and used in the audio reproduction control section 16d to obtain the output audio signal. In this example, if the final protection level is A, then all of the audio sample values are used in deriving the output audio signal. If the final protection level is B, then one in every two samples is held for two consecutive sample periods, by a sample-and-hold circuit, i.e. only half of the total samples are used in deriving the output audio signal. If the protection level is C, then only one in every three samples is used in deriving the output audio signal, i.e. one in every three successive samples is held for three consecutive sample periods. If the protection level is D, then only samples which occur during a specified interval are used in producing the output audio signal. For example, this operation could be performed when only a specified part of the recorded audio signal is to be allowed (by the copyright owner) to be reproduced. If the final protection level is E, then no audio output signal is produced.

A third method of audio signal reproduction protection will be described referring to the rightmost column in Fig. 12. In this case, protection control is executed by effecting control of the number of gradation levels provided by the audio data samples, in a similar manner to that described hereinabove for the video data. Generally, a digital audio signal has 16 bits/sample. When the final protection level is A, then all of these 16 bits are utilized, i.e. there is no limitation of audibility. If the final protection level is B, then the number of effective bits/sample is reduced to 12 (i.e. the low-order 4 bits of each 16-bit sample are fixed at 0), causing a lowering of quality of the reproduced audio signal. If the protection level is C, the number of effective bits is further reduced to 8, if the final protection level is D then the number of effective bits/sample is reduced to 4, and if the final protection level is E, then no audio output signal is produced.

It can thus be understood that the invention enables precise limitation of reproduction of a recorded audio signal together with limitation of reproduction of the recorded video signal, in accordance with a combination of medium protection data and apparatus protection data.

Fig. 13 shows an example of the internal configuration of the audio reproduction control section 16d of this embodiment. For the purpose of description, it is assumed that each of the above-mentioned three methods of protection control of the audio reproduction control section 16d is utilized, although in practice only one of these could be utilized. The circuit consists of a digital low-pass filter 61 which receives the output data samples from the sub-band combining section 16c via an input line 60, a sample-and-hold circuit 62 which can be controlled to hold and output each data sample for a specific interval, a gradation control section 63 which effects the aforementioned control of low-order bits of each audio data sample, to thereby control the gradation levels which can be expressed by each sample, a memory 64 for temporarily holding successive data samples, and a digital-to-analog converter 65 for converting the digital audio data to an analog audio signal. The gradation control section 63 is controlled by a protection control signal to provide varying degrees of gradation in accordance with the final protection level that has been determined for the audio data, by setting varying numbers of low-order bits of each 16-bit audio data sample to a fixed value as described above.

As shown, the protection control signal can also be applied to control the sample-and-hold circuit 62, to effect the above-described method of reproduction protection control utilizing sample-and-hold processing of the audio data samples. Similarly, the protection control signal can be applied to control the LPF 61, to achieve reproduction protection control by varying the bandwidth of the audio signal. It will be clear that a simpler circuit configuration can be util-

ized that that shown in Fig. 13, if only one of the above three methods of protection control is applied.

In each of the embodiments described above, there is a fixed relationship pattern between combinations of the protection levels which can be expressed by the medium protection data and the protection levels which can be expressed by the apparatus protection data, and the respective final protection levels which are thereby obtained, i.e. the relationship pattern which is stored in the matrix ROM of the final protection level determining section 13, an example of which is shown in Fig. 2. However in some cases there may be a requirement for enabling such a relationship pattern to be selected from a number of different relationship patterns, which have varying degrees of protection severity. Specifically, it may be advantageous to provide the reproduction apparatus with a switch which can be operated by the person who is in charge of the reproduction apparatus, such that the switch can be used to select from a plurality of different relationship patterns, so that the degree of reproduction protection can be flexibly determined by that person. In that way, the person in charge of the reproduction apparatus can ensure that the reproduction protection will be appropriate for the viewing audience. For example, an adult can set the switch such as to ensure that unsuitable scenes cannot be viewed by any children who may use the reproduction apparatus. In that case, assuming for example that the relationship pattern which is the least severe is that shown in Fig. 2, the entire range of possible relationship patterns is illustrated in the table of Fig. 14. Here, for each combination of protection levels obtained from the medium protection data and apparatus protection data, there is a corresponding range of one or more possible values of final protection level, with that range extending from a least severe value to the most severe value (i.e. protection level E). One method of implementing such a capability would be to provide a plurality of matrix ROMs (or to define a plurality of separate matrix regions in a ROM), for storing the respectively different relationship patterns. However an embodiment of the invention will now be described whereby such a capability can be easily implemented by a simple modification of any of the embodiments of the invention that have been previously described.

The embodiment, which is a modification of the embodiment of Fig. 5, is shown in Fig. 15. The embodiment differs from that of Fig. 5 in being provided with a severity modification circuit 30 and a severity modification setting switch 31. The severity modification circuit 30 is connected between the final protection level determining section 13 and the protection control signal generating section 14a, and functions to selectively modify each protection level value which is read out from the matrix ROM of the final protection level determining section 13 (as described hereinabove for the embodiment of Fig. 5), and to supply a resultant modified final protection level to the protection control signal generating section 14a. The severity modification setting switch 31 can be adjusted by the person who is in charge of the CD player, to select one of five possible switch conditions which will be designated as P_A to P_E, respectively. The severity modification setting switch 31 is coupled to control the severity modification circuit 30 such that the severity modification circuit 30 executes protection level modification in accordance with the specific position at which the switch is set, as described in the following.

The relationships between the five possible values of the medium protection level (determined by the medium protection signal detection section 11 as described hereinabove) and the five positions P_A to P_E of the severity modification setting switch 31, are shown for each of the four possible values of the apparatus protection level (determined by the apparatus protection signal generating section 12), in diagrams (A) to (D) in Fig. 16. Referring first to diagram (A), if the switch position is P_A, then the relationship between the medium protection level values, the apparatus protection level values and the final protection level values is left unchanged from those of the corresponding column in Fig. 2, i.e. this setting of the severity modification setting switch 31 provides the least severe degree of reproduction protection. If the switch position is set to P_B, then the least severe value which can be taken by the modified final protection level becomes level B. That is to say, if a protection level A is established by the final protection level determining section 13, that is changed by the severity modification circuit 30 to a modified final protection level B. If the switch position is P_C, then the least severe value of the modified final protection level is changed to C. If the switch position is P_D, then the least severe value of the modified final protection level is changed to D, and if the switch position is P_E, then the modified final protection level is fixed as E.

The same is true for each of the apparatus protection level values 2, 3 and 4, as illustrated in diagrams (B), (C) and (D) in Fig. 16, which correspond to the second, third and fourth columns in Fig. 2 respectively.

It will be apparent that the severity modification circuit 30 can be easily configured using a logic circuit, which implements a simple algorithm in accordance with the setting of the severity modification setting switch 31, i.e. the algorithm would begin:

[If the severity modification switch 31 is set at P_A, transfer the protection level value established by the final protection level determining section 13 directly to the protection control signal generating section 14a, as the final protection level.

If the severity modification switch 31 is set at P_B, and if the protection level value established by the final protection level determining section 13 is level A,

change that to level B and transfer to the protection control signal generating section 14a as the (modified) final protection level. Otherwise, transfer the protection level value produced from the final protection level determining section 13 unchanged, as the final protection level.

If the severity modification switch 31 is set at P_C, and if the protection level established by the final protection level determining section 13 is level A or level B, change to level C, and transfer to the protection control signal generating section 14a as the final protection level. Otherwise, transfer the protection level established by the final protection level determining section 13 unchanged, as the final protection level .....], and so on.

Thus with this embodiment, if the severity modification setting switch 31 is set to its least severe position (P_A), then the relationship pattern between combinations of the medium protection level and reproduction apparatus protection level values will be as shown in Fig. 2. If the severity modification setting switch 31 is set to the most severe position (P_E), then the relationship pattern will be such that the final protection level will always be the highest level, i.e. level E. As the severity modification setting switch 31 is successively changed from positions P_A to P_E, the least severe degree of reproduction restriction (within the range of possible degrees of restriction which can be set by the final protection level) is increased to a more severe degree, by successive steps.

For example if the apparatus protection level is 2 and the medium protection level is 4, then the final protection level will be B. However by altering the setting of the severity modification setting switch 31, the user can change the final protection level to a higher value, in the range B to E. Hence with this embodiment, although protection level values can be specified by the manufacturer or copyright owner of the recording medium, and by the manufacturer or seller of the reproduction apparatus, the final degree of protection can be determined by the person who is in charge of the reproduction apparatus. Such a feature is highly useful.

As can be understood from the above description of embodiments, the invention enables a final protection level to be established, for controlling reproduction of recorded or transmitted video or audio signals, with that final protection level being determined based on a combination of protection levels which are respectively separately established by the manufacturer or copyright owner of recorded signals (or broadcaster of transmitted signals, or copyright owner of transmitted signal) and by the manufacturer or seller of the reproduction apparatus. The final protection level can be applied such as to achieve extremely precise protection of reproduction, whereby for example specific frames of a video signal, and/or specific regions within a frame, can be protected by restricting reproduction, with the degree of restriction being variable in a stepwise manner. The invention can at the same time provide corresponding protection of an audio signal which is being reproduced in conjunction with a video signal.

Although the invention has been described in the above with reference to a CD player apparatus, it will be understood that the invention is not limited in any way to such an apparatus, and is in general applicable to reproduction protection in any type of apparatus which reproduces a recorded or transmitted video and/or audio signal.

Claims

1. A reproduction protection method comprising:

attaching medium protection data, which are specific to a data medium, to main data which represent an original signal, and conveying said medium protection data and main data by said data medium;

supplying said main data and medium protection data via said data medium to a reproduction apparatus;

generating apparatus protection data which are specific to said reproduction apparatus, within said reproduction apparatus;

determining a protection level by combining the medium protection data and the apparatus protection data; and

controlling said reproduction apparatus to utilize said main data to reproduce said original signal in accordance with said protection level.

2. A reproduction apparatus providing reproduction protection, for operating on main data which are conveyed by a data medium and represent an original signal and on medium protection data which are specific to said data medium and are conveyed by said data medium, the apparatus comprising:

means for generating apparatus protection data which are specific to said reproduction apparatus;

means for defining a protection level based on said medium protection data and apparatus protection data in combination; and

means for executing reproduction of said original signal by utilizing said main data, including means for selectively restricting said reproduction in accordance with said protection level.

3. A reproduction apparatus as claimed in claim 2, wherein the apparatus further includes:

means (10,11) for detecting said medium protection data to obtain a medium protection signal expressing said medium protection level;

said generating means comprises means (12) for generating an apparatus protection signal expressing an apparatus protection level which has been assigned to said reproduction apparatus;

said defining means comprises means (13) responsive to said medium protection signal and apparatus protection signal for determining a final protection level in accordance with a combination of said medium protection level and apparatus protection level, and generating a final protection level signal expressing said final protection level.

4. A reproduction apparatus providing reproduction protection, for operating on main data which are conveyed by a data medium and represent an original signal and on medium protection data which are specific to said data medium and are conveyed by said data medium, the apparatus comprising:

means (10,11) for detecting said medium protection data to obtain a medium protection signal expressing said medium protection level;

means (12) for generating an apparatus protection signal expressing an apparatus protection level which has been assigned to said reproduction apparatus;

means (13) responsive to said medium protection signal and apparatus protection signal for determining a final protection level in accordance with a combination of said medium protection level and apparatus protection level, and generating a final protection level signal expressing said final protection level;

means (15) for utilizing said main data to execute reproduction of said original signal, including means (14) responsive to said final protection level signal for selectively restricting said reproduction in accordance with said final protection level.

5. A reproduction apparatus according to claim 2, 3 or 4, wherein said main data are conveyed by said data medium after having been compressed and encoded, wherein said reproduction apparatus includes means (15a, 15b, 15c) for decoding and means for decompressing said main data, and wherein said means for selectively restricting reproduction of the original signal comprise means for controlling at least one of said decoding means and said decompressing means in accordance with said final protection level.

6. A reproduction apparatus according to claim 5, wherein said main data comprise video data which are conveyed by said data medium after having been compressed by applying a transform operation upon each of fixed-size blocks of pixel values within each frame of a video signal, wherein said decompressing means includes means (15c) for applying an inverse transform operation, and wherein said means for selectively restricting reproduction of the original signal serves to selectively eliminate specific ones of a plurality of transform coefficients utilized in executing said inverse transform operation.

7. A reproduction apparatus according to any of claims 2 to 6, wherein said main data are conveyed by said data medium after having been compressed, quantized and encoded, wherein said reproduction apparatus includes means (15b) for dequantizing said main data, and wherein said means for selectively restricting reproduction of the original signal comprise means for controlling said dequantizing means in accordance with said final protection level.

8. A reproduction apparatus according to any of claims 2 to 7, wherein said means for selectively restricting reproduction of the original signal comprise means (52, 62) for periodically omitting selected portions of said main data from use by said reproduction control means in reproducing said original signal.

9. A reproduction apparatus according to claim 8, wherein each of said selected portions comprises at least one digital data sample.

10. A reproduction apparatus according to claim 8 or claim 9, wherein said main data express a digital video signal, and wherein each of said selected portions comprises at least one video signal frame.

11. A reproduction apparatus according to any of claims 2 to 10, wherein said means for selectively restricting reproduction of the original signal comprises gradation control means (51, 63) for selectively varying a number of signal amplitude gradations which can be expressed by each data sample of said main data.

12. A reproduction apparatus according to any af claims 2 to 11, wherein said means for selectively restricting reproduction of the original signal comprises controllable digital low-pass filter means (61) for filtering said main data, to thereby controllably vary a bandwidth of a reproduced original signal obtained from said apparatus.

13. A reproduction apparatus according to any of claims 2 to 12, wherein said medium protection data specify a medium protection level and pro-

tection position information, and wherein said protection position information specifies a portion of said main data which is to subjected to reproduction restriction in accordance with said final protection level.

14. A reproduction apparatus according to claim 13, wherein said main data comprise a digital video signal, and wherein said protection position information specifies at least one region within each of a plurality of frames of said video signal.

15. A reproduction apparatus according to any of claims 2 to 14, further comprising protection severity modification means, operable for modifying said final protection level which is determined by said combination of medium protection data and apparatus protection data, to obtain a modified final protection level.

16. A reproduction apparatus according to claim 15, wherein said severity modification means comprise a manually operable switch (31) provided on said reproduction apparatus, and severity modification circuit means (30) coupled to said switch and coupled to receive a signal representing said final protection level, for modifying said final protection level in accordance with a setting position of said switch and thereby producing an output signal expressing a modified final protection level.

17. A reproduction apparatus according to claim 16, wherein said final protection level has a value which defines one of a plurality of successively increasing degrees of restriction of reproduction of said original signal, extending between a minimum and a maximum degree of restriction, and wherein said severity modification means is responsive to a change in said setting position of the switch for changing only said minimum degree of restriction.

18. A data medium for transferring main data representing an original signal to a reproduction apparatus, to reproduce said original signal by said reproduction apparatus, wherein said data medium further transfers medium protection data which are specific to said data medium and wherein said reproduction apparatus generates apparatus protection data which are specific to said reproduction apparatus, said medium protection data and apparatus protection data in combination specifying a degree of restriction on said reproduction of the original signal by said reproduction apparatus.

FIG. 1

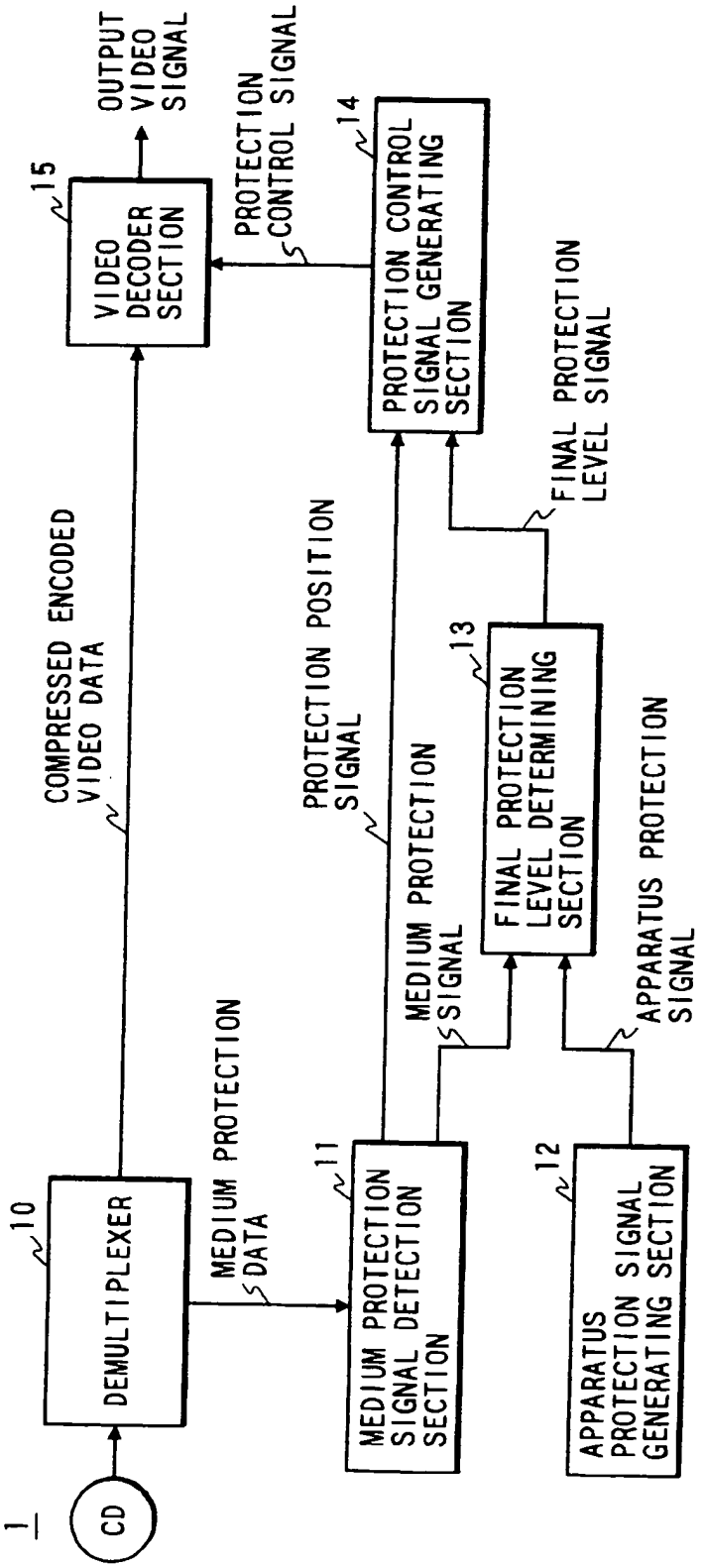


FIG. 2

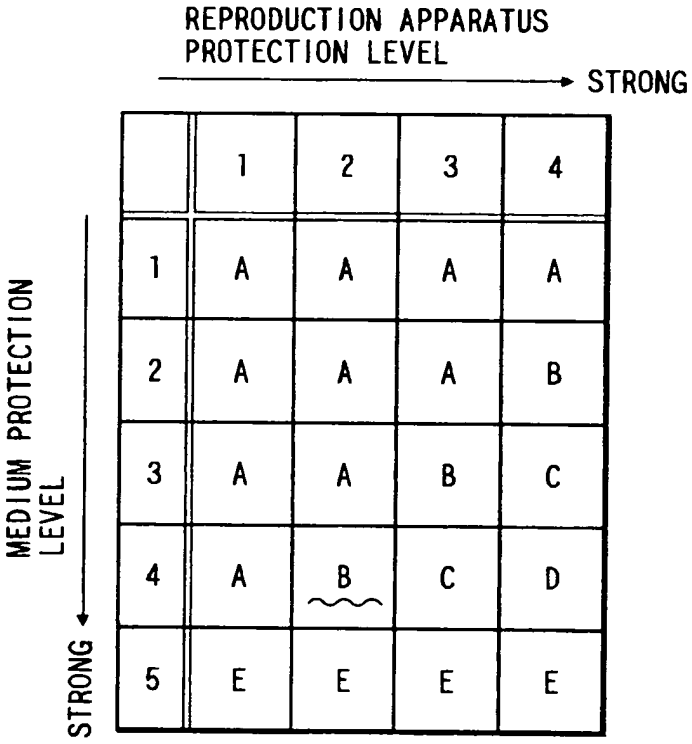


FIG. 3

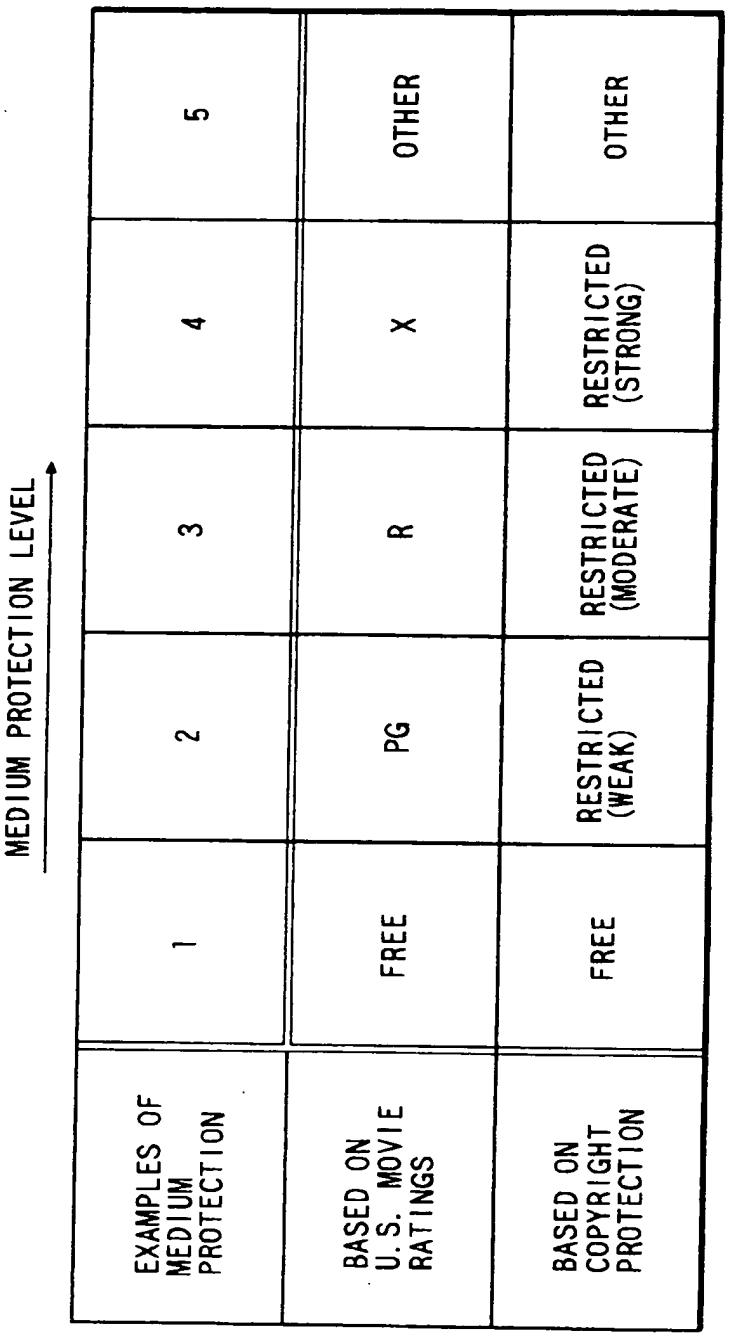


FIG. 4

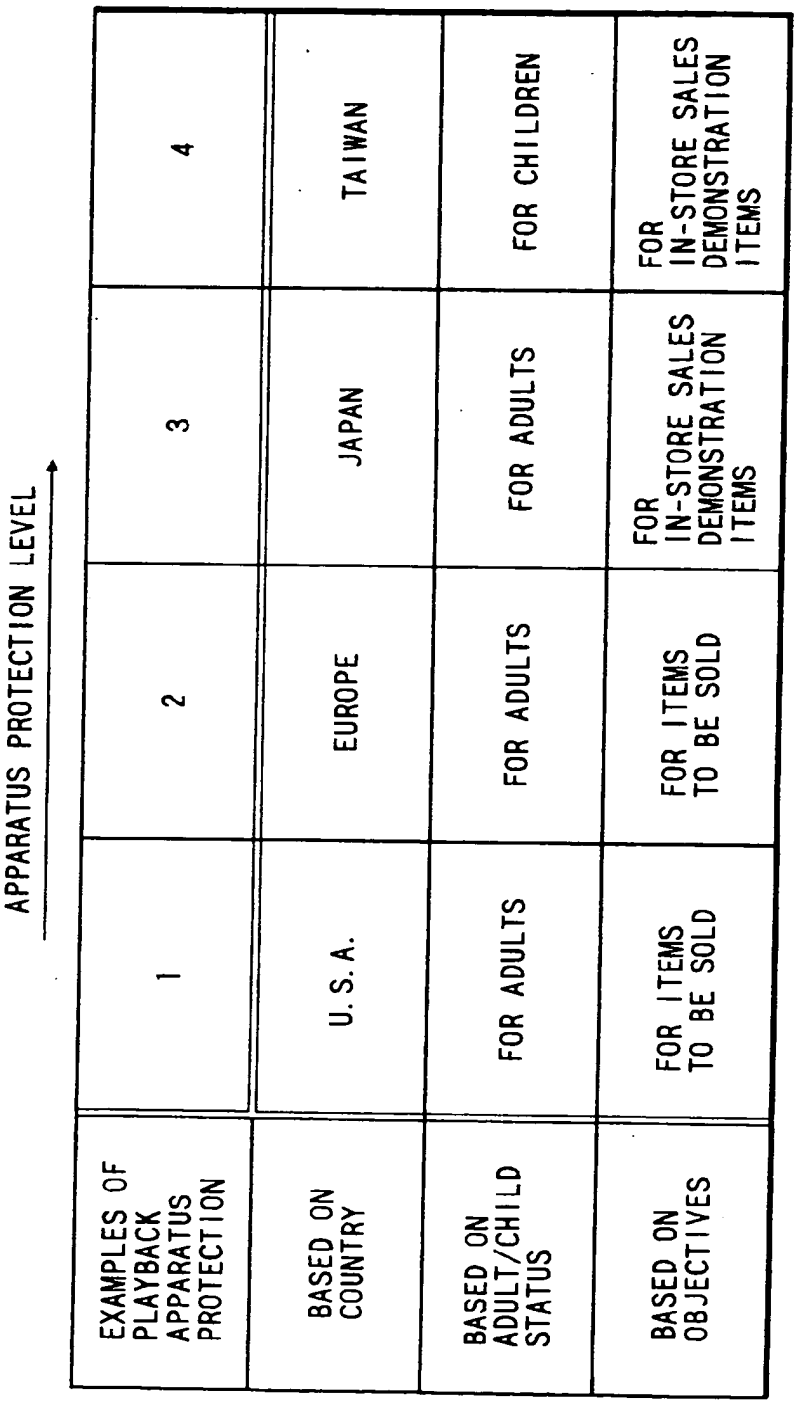


FIG. 5

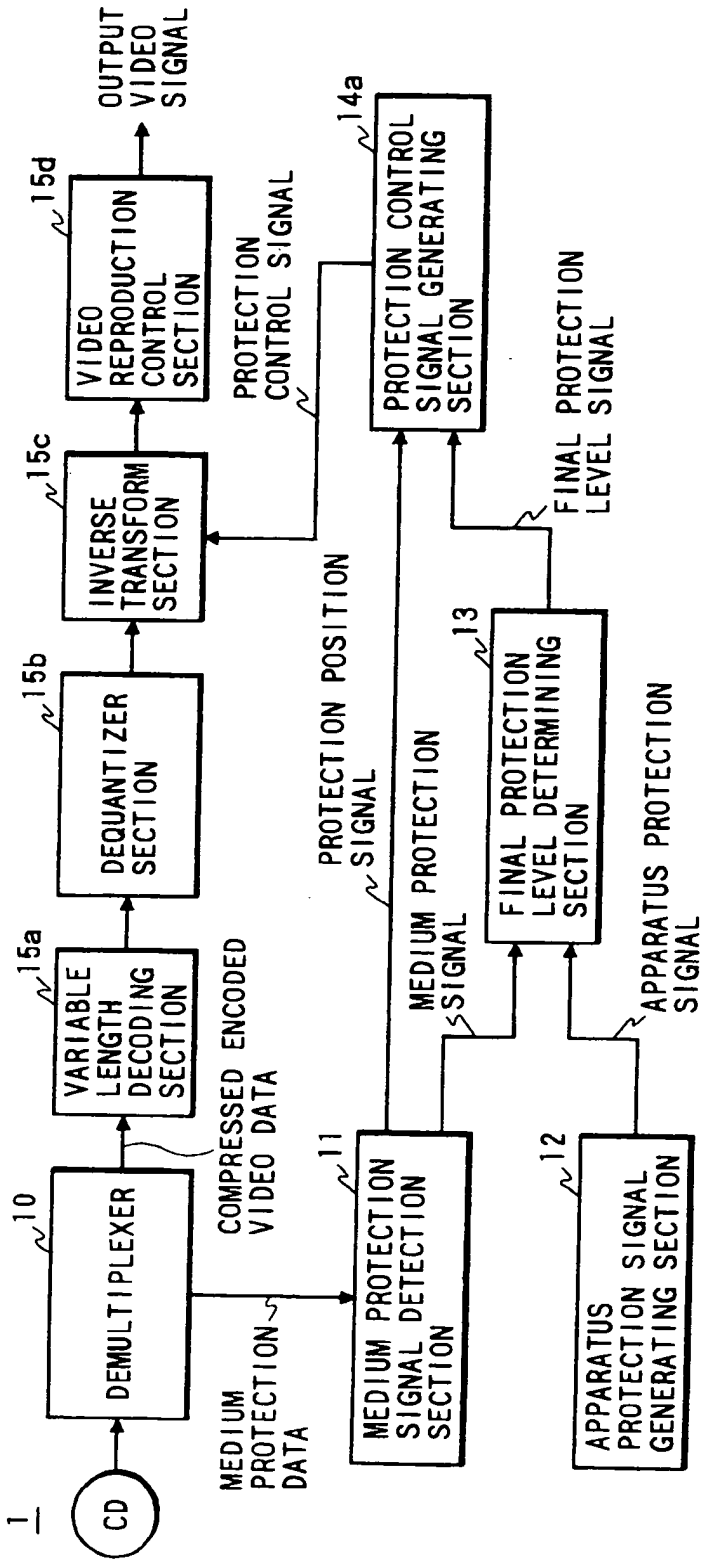


FIG. 6

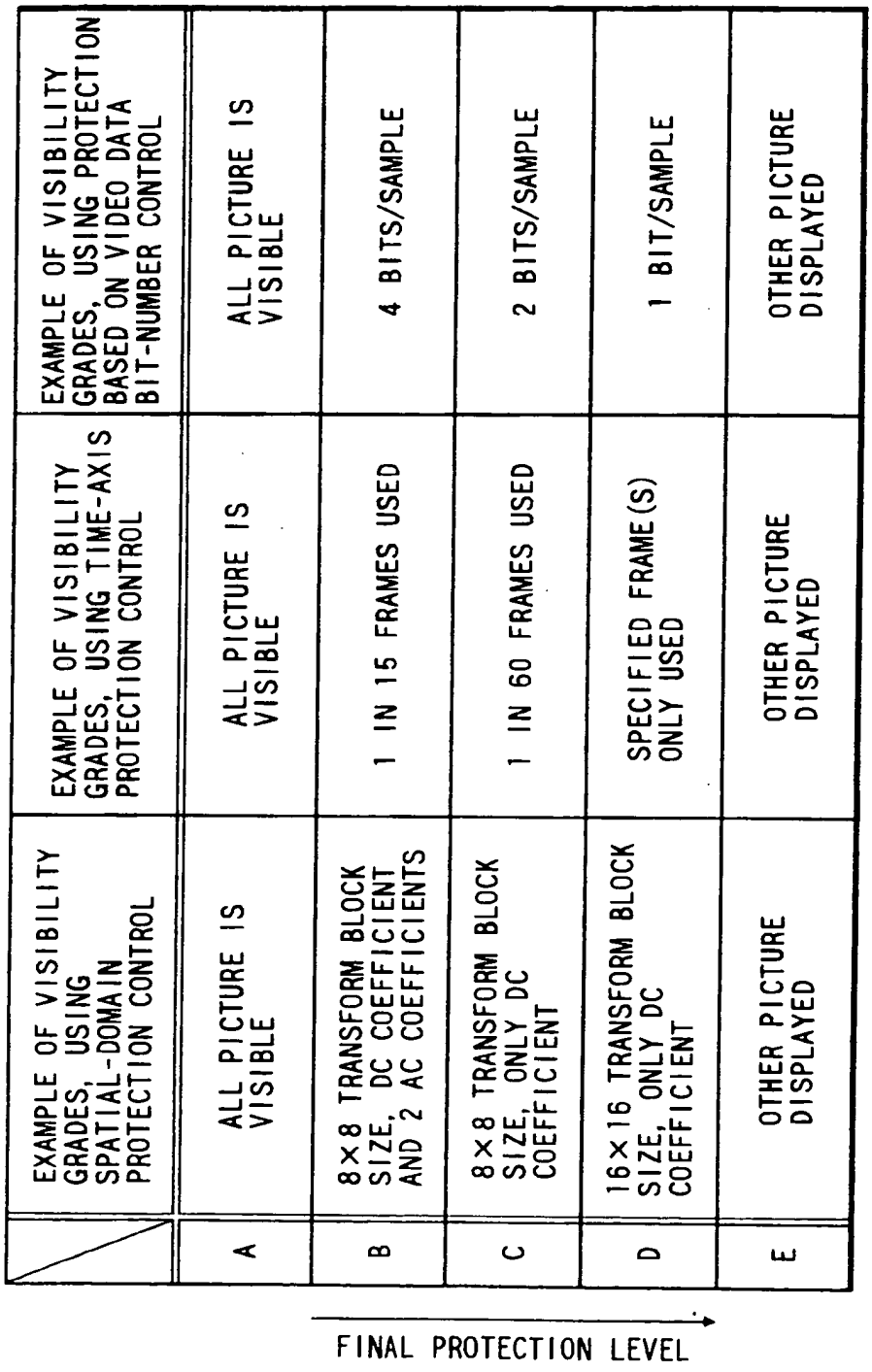


FIG. 7

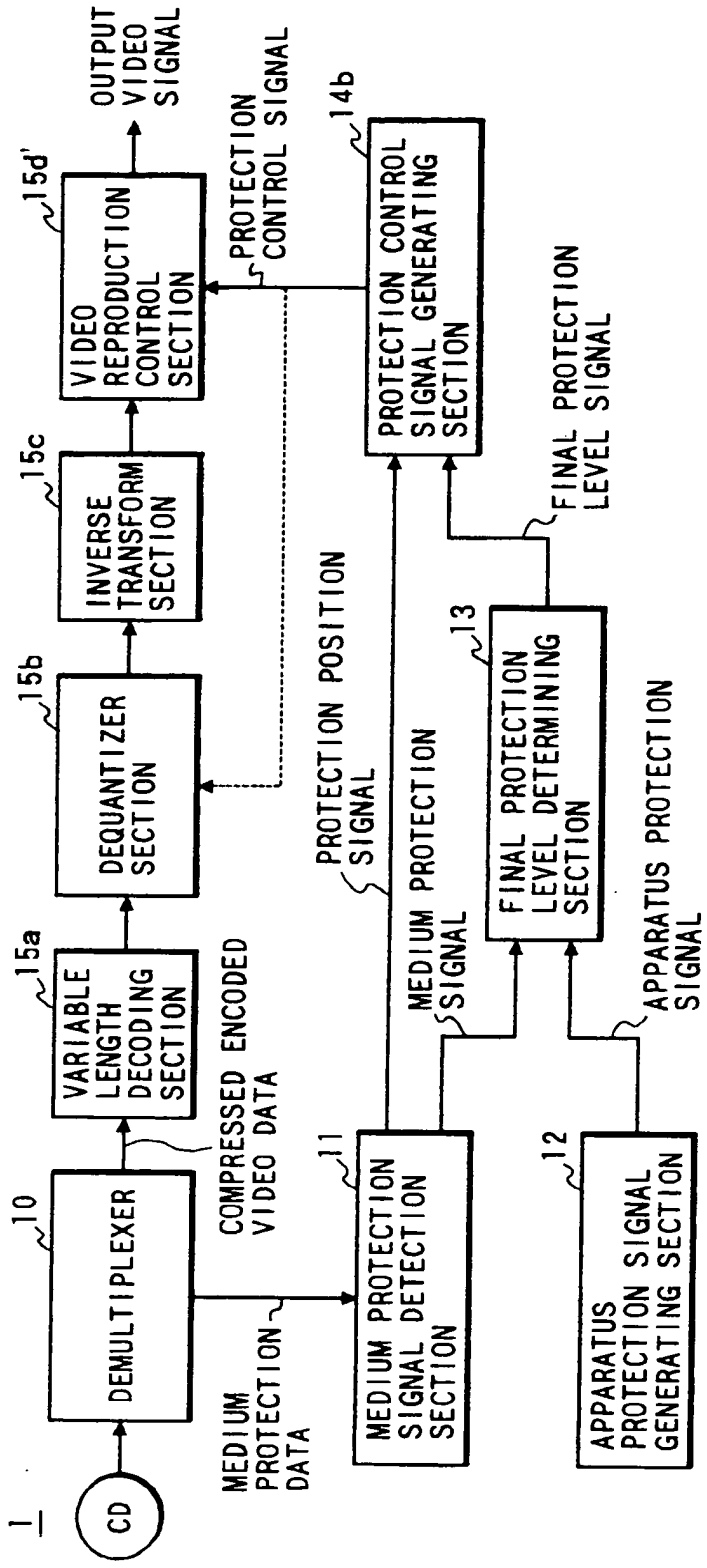


FIG. 8

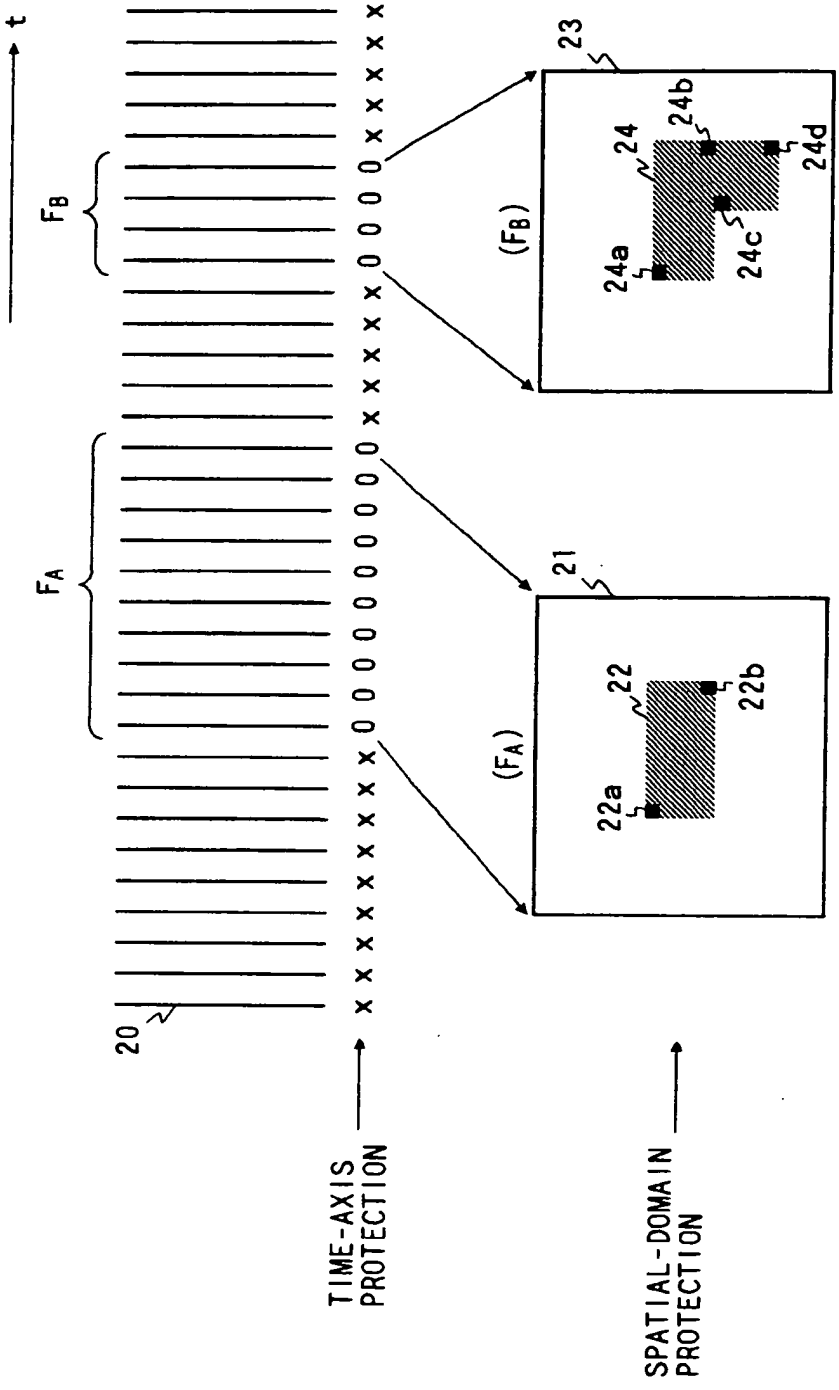


FIG. 9

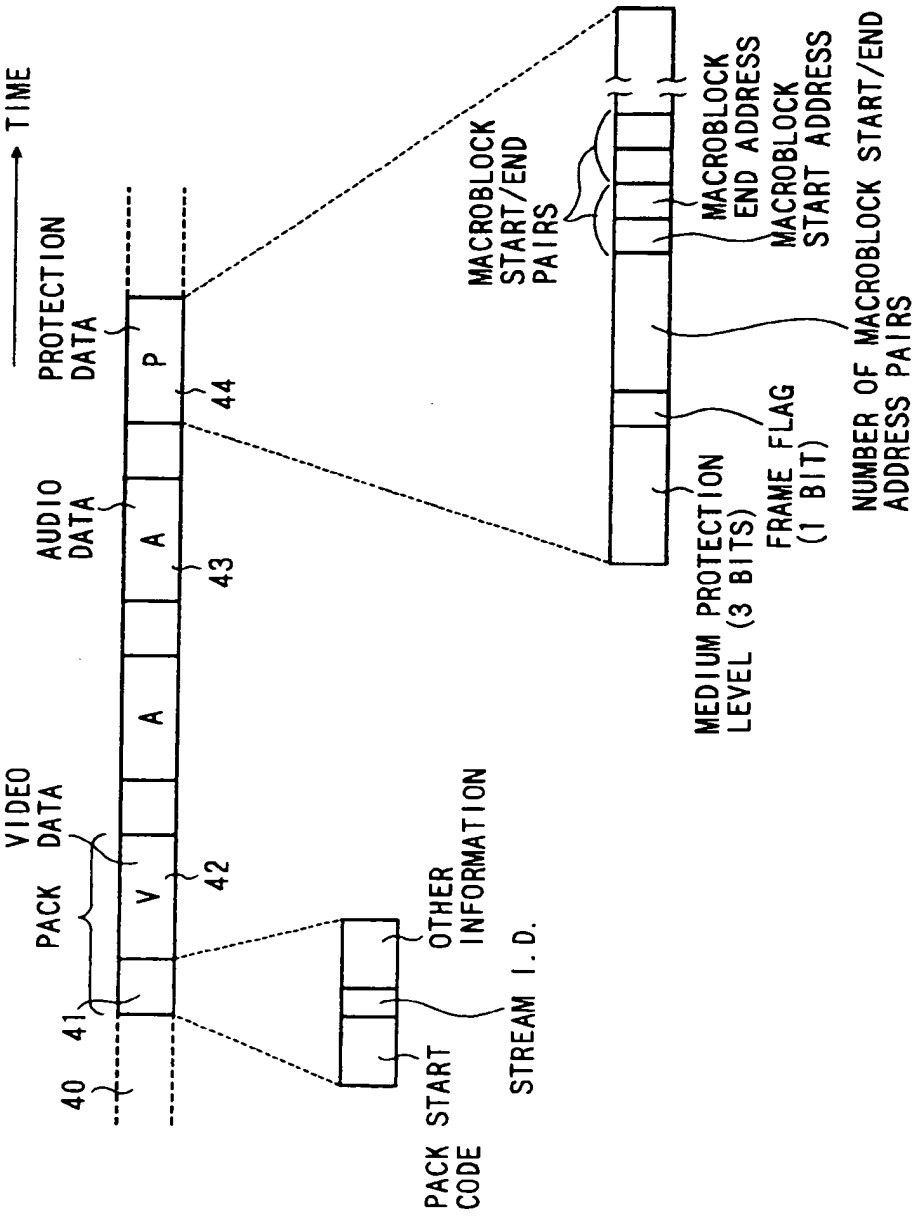


FIG. 10

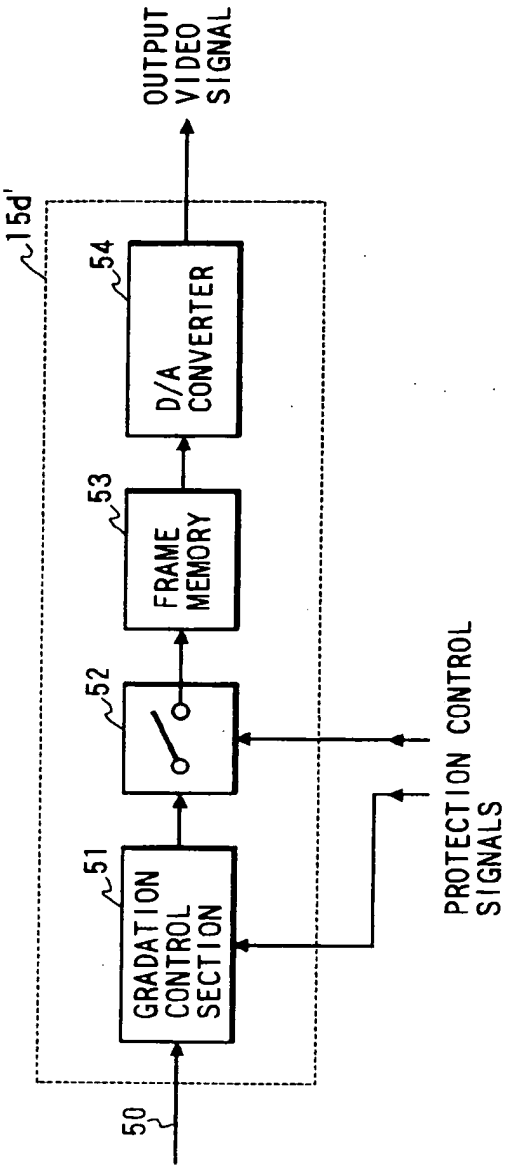


FIG. 11

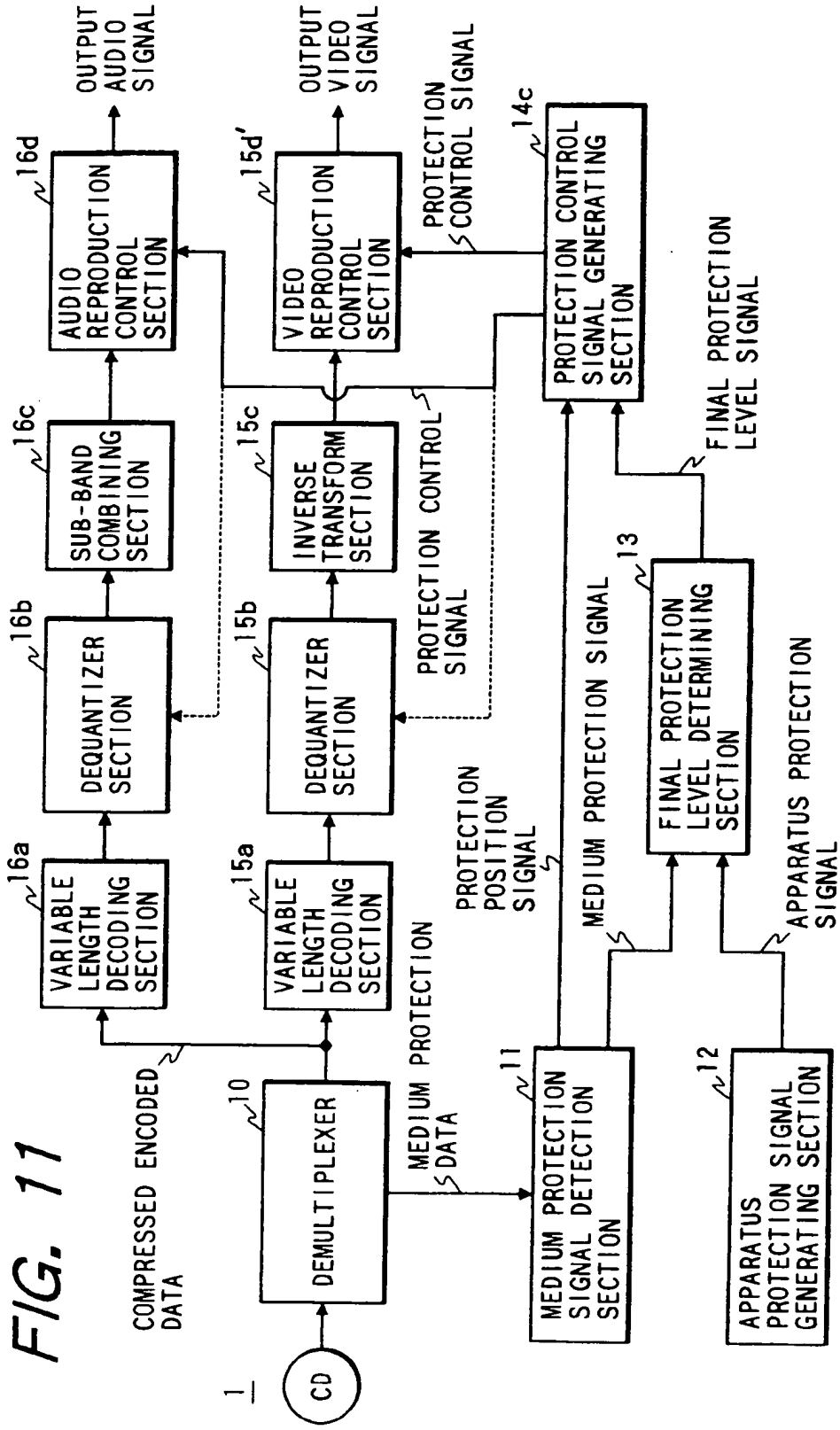


FIG. 12

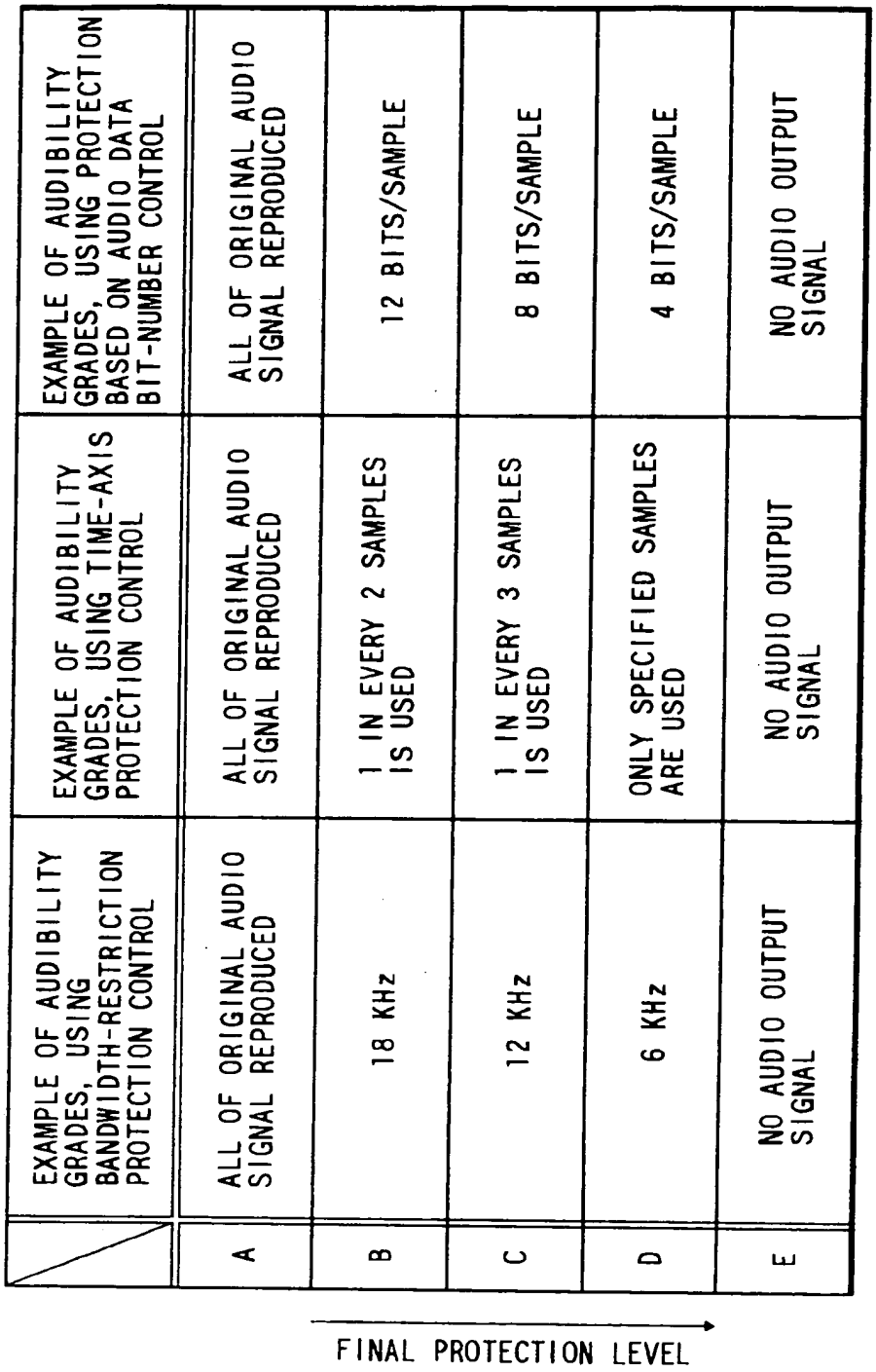


FIG. 13

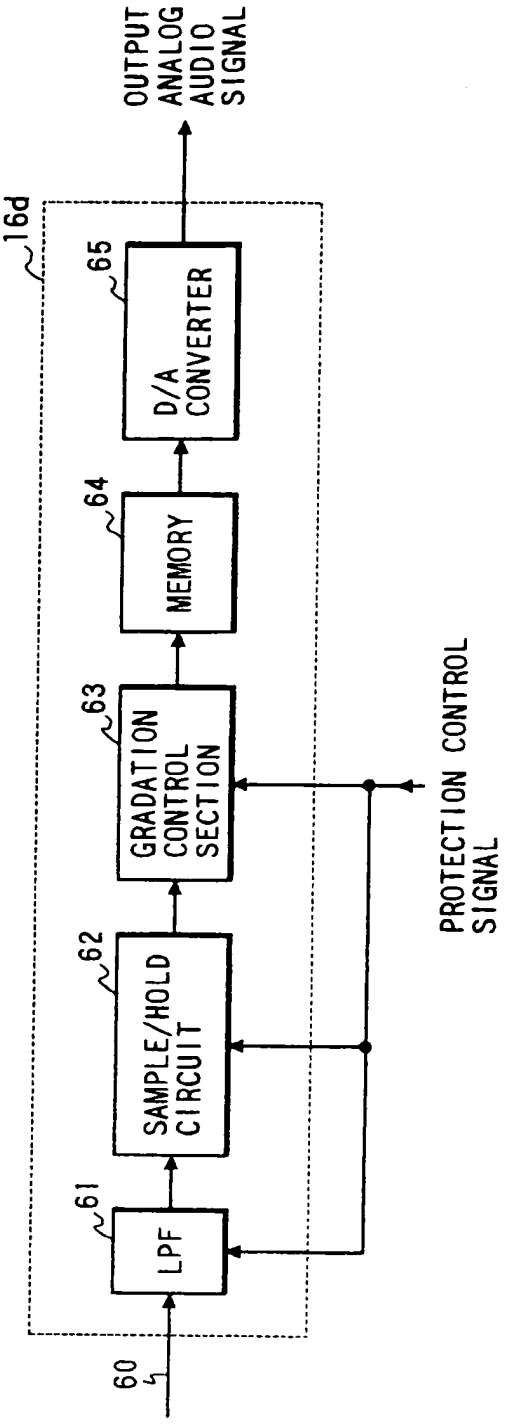
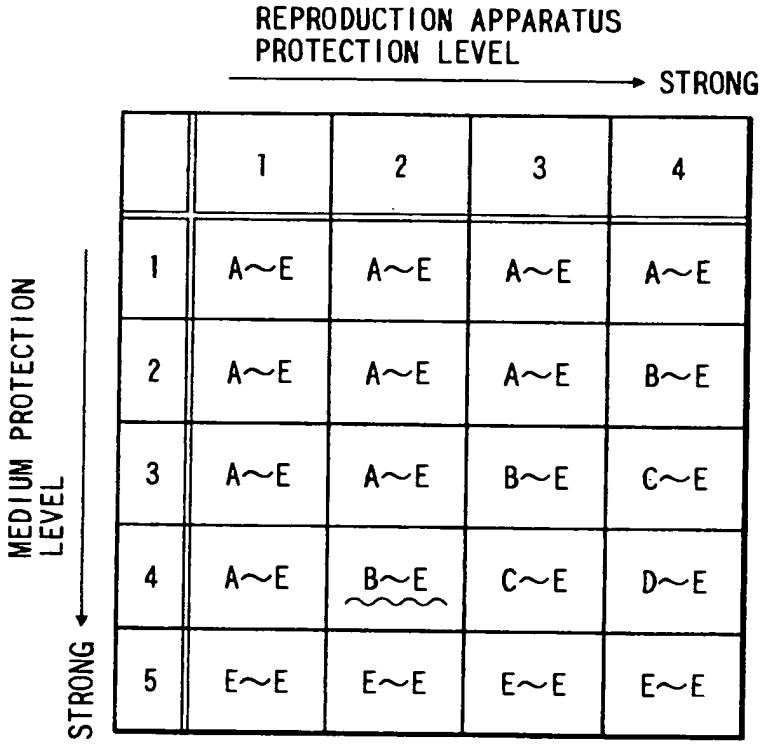


FIG. 14



	1	2	3	4
1	A~E	A~E	A~E	A~E
2	A~E	A~E	A~E	B~E
3	A~E	A~E	B~E	C~E
4	A~E	B~E	C~E	D~E
5	E~E	E~E	E~E	E~E

FIG. 15

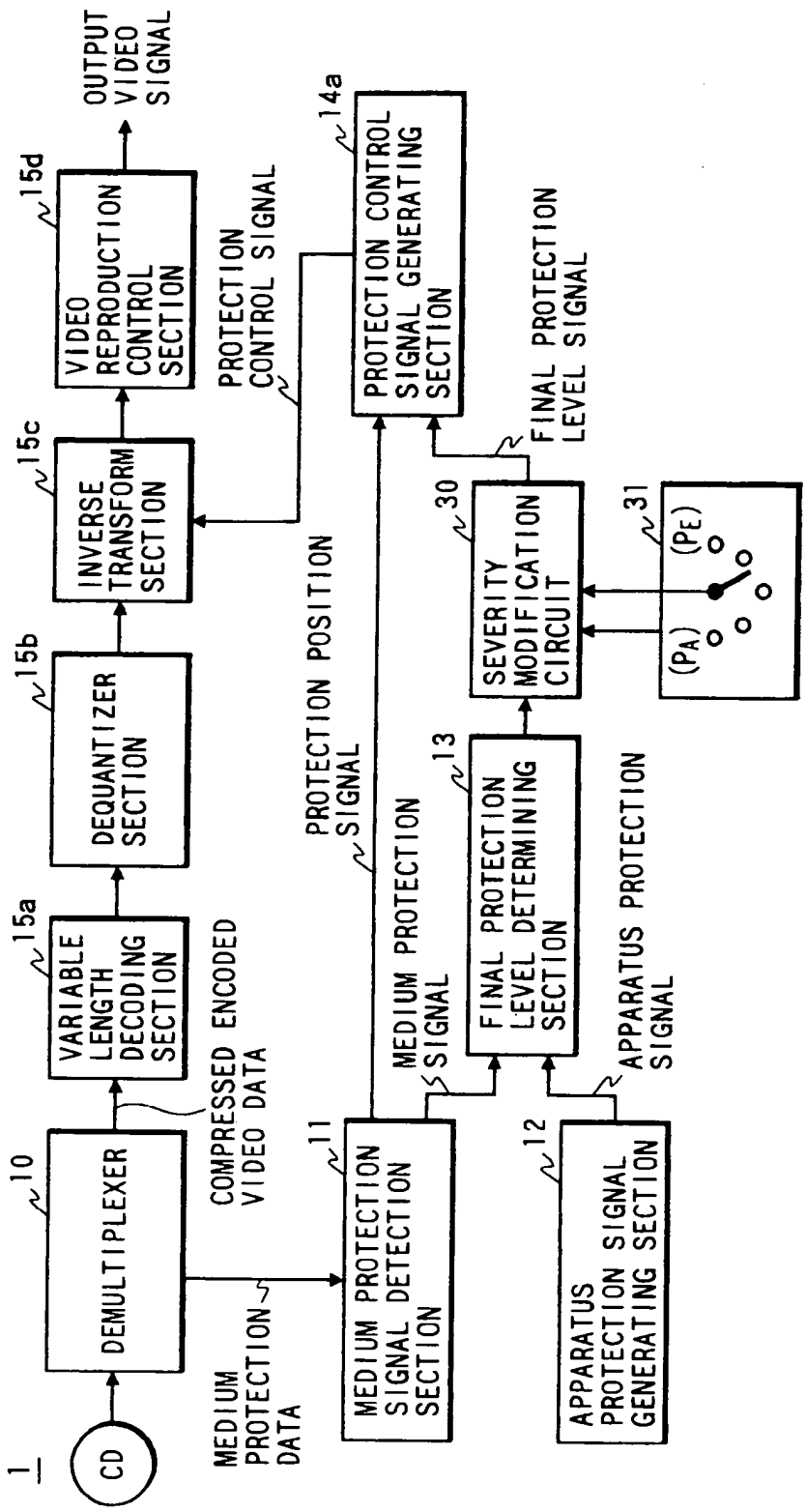
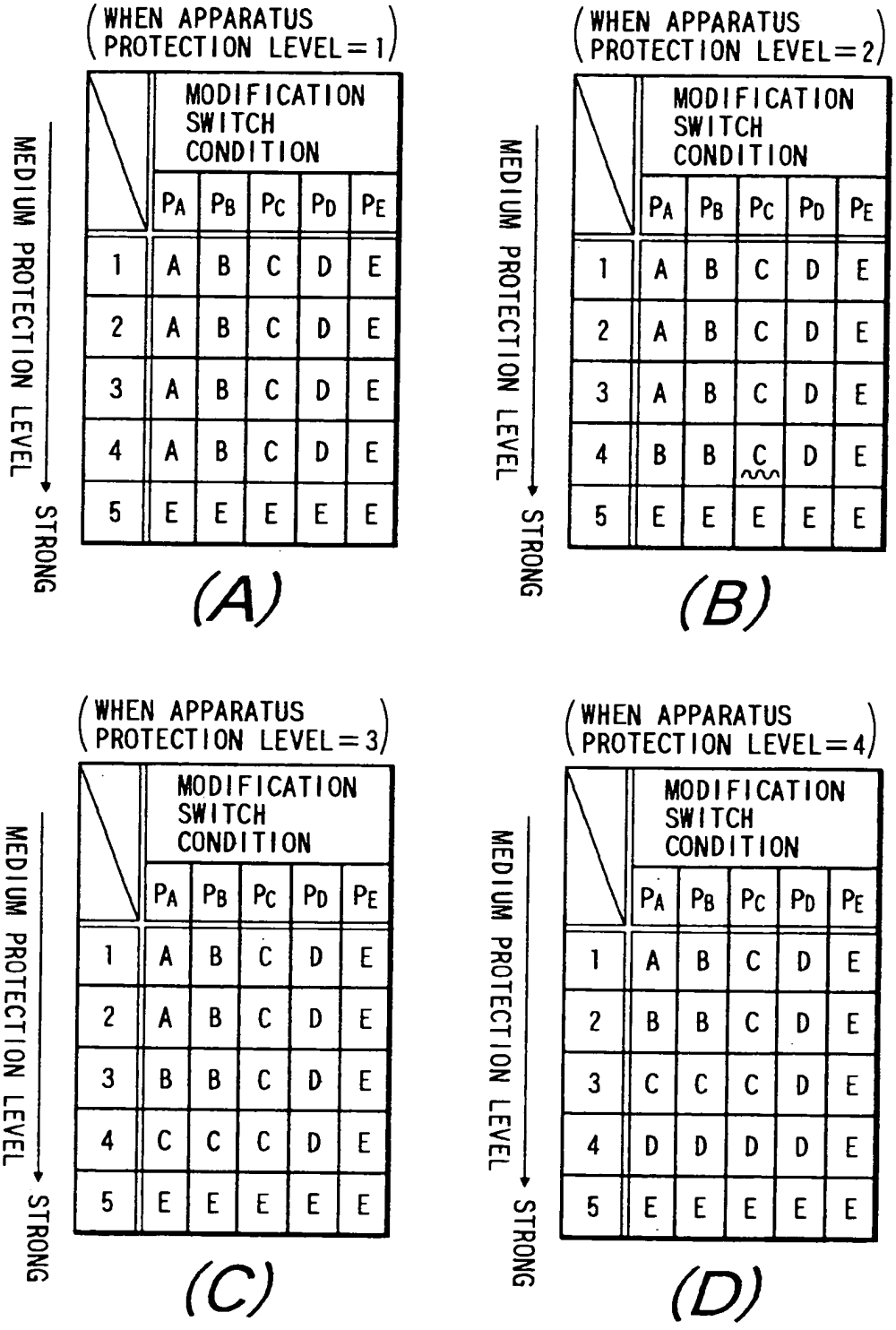


FIG. 16



Requested Patent: EP0725376A2

Title:

CHARGING METHOD AND CHARGING SYSTEM IN INTERACTIVE ON-LINE SERVICE ;

Abstracted Patent: EP0725376 ;

Publication Date: 1996-08-07 ;

Inventor(s):

NAKANO HIROAKI (JP); NIIJIMA MAKOTO (JP); SONODA YUMIE (JP); KUMAGAI YOSHIAKI (JP); NAGAHARA JUNICHI (JP); NASHIDA TATSUSHI (JP) ;

Applicant(s): SONY CORP (JP) ;

Application Number: EP19960300509 19960125 ;

Priority Number(s): JP19950017885 19950206 ;

IPC Classification: G07F19/00; G07F7/00 ;

Equivalents: JP8214281, US5845260 ;

ABSTRACT:

A charging method in interactive on-line service where a server and terminals of users are mutually connected by means of a transmission medium, and a predetermined service is provided from the server to the terminals via the transmission medium, and the fee for such service is collected from each user individually. The charging method comprises the steps of opening, in the server, an imaginary account for a child of the relevant user with a limited maximum amount, and withdrawing from the imaginary account the fee for the service provided to the user's child. The server can provide a predetermined service to the relevant user's child within a range of the limited maximum amount preset in the imaginary account. And when withdrawing the fee from the imaginary account, the server can restrict the service providable to the terminal. Thus, the parent enables his child to receive a desired on-line service, such as on-line shopping or video-on-demand, on the basis of the child's own judgment by setting an upper limit of a service utilisable by the child and still limiting the services providable for the child, hence realising promoted utilisation of the service by children.

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

(11) EP 0 725 376 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
07.08.1996 Bulletin 1996/32

(51) Int Cl.[6]: **G07F 19/00**, G07F 7/00

(21) Application number: **96300509.5**

(22) Date of filing: **25.01.1996**

(84) Designated Contracting States:
DE FR GB NL

(30) Priority: **06.02.1995 JP 17885/95**

(71) Applicant: **SONY CORPORATION**
Tokyo 141 (JP)

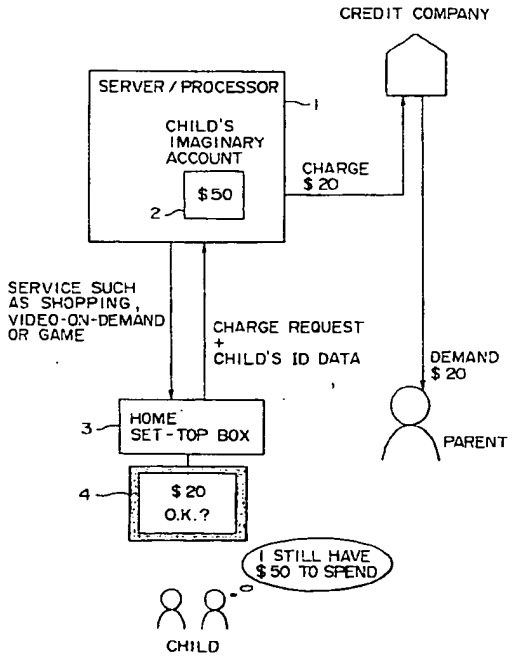
(72) Inventors:
- **Nagahara, Junichi, c/o Sony Corp.**
  Shinagawa-ku, Tokyo (JP)
- **Nashida, Tatsushi, c/o Sony Corp.**
  Shinagawa-ku, Tokyo (JP)
- **Nakano, Hiroaki, c/o Sony Corp.**
  Shinagawa-ku, Tokyo (JP)
- **Nijima, Makoto, c/o Sony Corp.**
  Shinagawa-ku, Tokyo (JP)
- **Sonoda, Yumie, c/o Sony Corp.**
  Shinagawa-ku, Tokyo (JP)
- **Kumagai, Yoshiaki, c/o Sony Corp.**
  Shinagawa-ku, Tokyo (JP)

(74) Representative: **Nicholls, Michael John**
J.A. KEMP & CO.
14, South Square
Gray's Inn
London WC1R 5LX (GB)

(54) Charging method and charging system in interactive on-line service

(57) A charging method in interactive on-line service where a server and terminals of users are mutually connected by means of a transmission medium, and a predetermined service is provided from the server to the terminals via the transmission medium, and the fee for such service is collected from each user individually. The charging method comprises the steps of opening, in the server, an imaginary account for a child of the relevant user with a limited maximum amount, and withdrawing from the imaginary account the fee for the service provided to the user's child. The server can provide a predetermined service to the relevant user's child within a range of the limited maximum amount preset in the imaginary account. And when withdrawing the fee from the imaginary account, the server can restrict the service providable to the terminal. Thus, the parent enables his child to receive a desired on-line service, such as on-line shopping or video-on-demand, on the basis of the child's own judgment by setting an upper limit of a service utilisable by the child and still limiting the services providable for the child, hence realising promoted utilisation of the service by children.

FIG. 1



EP 0 725 376 A2

Printed by Jouve, 75001 PARIS (FR)

Description

The present invention relates to a charging method and a charging system adapted for use in interactive on-line service or mail-order sale such as on-line shopping or video-on-demand service.

In a conventional charging system where a central server and home television receivers are mutually connected by means of a cable and the server distributes video or data to the television receivers and charges each user individually for the service offered, it is generally customary that the charging process is executed by utilizing the user's credit card. In this case, the user acknowledges the charge by inputting the number of his credit card every time a service is received. Then the service provider company receives the fee from a credit company, and the credit company mails a bill with a detailed account to the actual user to thereby demand payment of the fee.

Thus, in the conventional charging system, any person other than the owner of a credit card is unable to receive the provision of a service. Since none of children under age has a credit card in general, there exists a problem that a child is incapable alone of receiving a service.

In an attempt to solve the above problem, there may be contrived an idea of giving general credit cards to children as well. However, this idea is considered unpractical under the present circumstances where a credit card is usable similarly to cash. Meanwhile in a situation where any child is permitted to use his parent's credit card, it may probably occur that some unnecessary charge is demanded of the parent. Consequently, whenever the child wants to receive a certain service, the parent is obliged to input the number of his credit card each time to approve provision of the service, and therefore it follows that the child is not allowed to receive the desired service unless the parent is always with the child, hence raising an issue that the child's independency is impeded.

On the other hand, as viewed from the service provider company, there exists a problem of probable failure in promoting utilization of the service by children.

It is an object of the present invention to enable a child to receive a desired on-line service on the basis of his own judgment by setting an upper limit of a service utilizable by children and still limiting the services providable for children, hence realizing promoted utilization of the service by children.

According to a first aspect of the present invention, there is provided a charging method in interactive on-line service where a server and terminals of users are mutually connected by means of a transmission medium, and a predetermined service is provided from the server to the terminals via the transmission medium, and the fee for such service is collected from each user individually. This charging method comprises the steps of opening, in the server, an imaginary account for a child of the relevant user with a limited maximum amount, and withdrawing from the imaginary account the fee for the service provided to the user's child.

The server can provide a predetermined service to the relevant user's child within a range of the limited maximum amount preset in the imaginary account.

And when withdrawing the fee from the imaginary account, the server can restrict the service providable to the terminal.

According to a second aspect of the present invention, there is provided a charging system in interactive on-line service where a server and terminals of users are mutually connected by means of a transmission medium, and a predetermined service is provided from the server to the terminals via the transmission medium, and the fee for such service is collected from each user individually. This charging system comprises a setting means for setting an imaginary account of each user's child with a limited maximum amount, and a charging means for withdrawing from the imaginary account the fee for the service provided to the relevant user's child.

The server can provide a predetermined service to the relevant user's child within a range of the limited maximum amount preset in the imaginary account.

And when withdrawing the fee from the imaginary account, the server can restrict the service providable to the terminal.

In the charging method or the charging system of the present invention where an imaginary account of each user's child is opened in the server, the fee for the service provided to the relevant user's child is withdrawn from the imaginary account. Therefore the charging mode is changeable in accordance with the serviced user.

In one modified charging method or system of the present invention, a child is permitted to receive provision of a predetermined service within the limited maximum amount preset in the imaginary account, so that it is possible to prevent any excessive utilization of the service by the child.

And in another modified charging method or system of the present invention, the service to be provided is restricted when the fee is withdrawn from the imaginary account. Consequently, it becomes possible to restrict provision of any service that is not suited for participation of children.

The invention will be further described by way of non-limitative example, with reference to the accompanying drawings, in which:-

Fig. 1 shows a constitution of an exemplary embodiment representing the charging system of the present invention;

Fig. 2 is a flow chart showing a flow of equivalents in the constitution of Fig. 1;

Fig. 3 is a block diagram showing a detailed consti-

tution of the charging system in Fig. 1;

Fig. 4 is a flow chart showing an example of a charging process executed when a child is serviced with a game program;

Fig. 5 is a flow chart showing an example of a process executed on a closing day of a month in a service provider company; and

Fig. 6 is a flow chart showing an example of a process executed when a parent sets a balance in his child's imaginary account.

Fig. 1 shows a constitution of an exemplary embodiment which represents the charging system of the present invention, and Fig. 2 shows a flow of equivalents in the constitution thereof. First at step S1, a parent sets an imaginary account 2, which is to be used exclusively for charging his child, in a server/processor 1 (setting means, charging means) installed in a network service company. As will be described later with reference to Fig. 2, the server/processor 1 consists of a server (data base) 1a and a management computer 1b. Next at step S2, an adequate amount is preset in the charging imaginary account 2. Suppose now that an amount of $50 is preset in the imaginary account 2.

Subsequently at step S3, the child operates a display device 4 such as a television receiver (hereinafter referred to as TV) and transmits a charge request for a desired service to be received, such as on-line shopping, video-on-demand or game, to the server and processor 1 via a set-top box 3. And simultaneously the child's identification data is also transmitted with the charge request. As a result, the fee for the desired service relative to the charge request is withdrawn from the child's imaginary account 2 preset in the server/processor 1.

In case the fee for the desired service relative to the child's charge request is $20 for example, an amount of $20 is withdrawn from the current balance $50 in the imaginary account 2, whereby the balance is reduced to $30. And at the end of each month, the service provider company submits a bill of the total amount ($20 in this example) during the month to a credit company as a credit for the parent.

Thereafter at step S4, the credit company sends a detailed statement and a bill to the parent at the end of the month to thereby claim payment of the total fee ($20 in this case) in that month. The present situation regarding utilization of the service can be checked by inquiring at the server/processor 1 without the necessity of waiting for arrival of the detailed statement from the credit company.

Fig. 3 is a block diagram showing the charging system of Fig. 1 in further detail. The set-top box 3 installed in each contractor home transmits or receives data to or from a service provider company via a cable or the like and supplies the received predetermined data to the display device 4 such as a TV, or transmits to the service provider company a signal which corresponds to a predetermined command obtained by manipulation of an undermentioned remote controller 6.

The display device 4 is capable of displaying characters, patterns or pictures corresponding to predetermined data supplied from the set-top box 3. The remote controller 6 outputs a predetermined command to the set-top box 3 for selecting a desired service such as a game program to be transmitted from the service provider company, or for approving the charge.

In the server (data base) 1a of the service provider company, an imaginary account 2 of the contracted user's child is opened, and there are recorded data of the user inclusive of his credit card number and so forth specified at the time of contract. The management computer 1b transmits or receives predetermined data to or from the set-top box 3 installed in each contractor home for updating the balance of the imaginary account 2 opened in the server 1a, or for reading out some other data therefrom or writing predetermined data therein. Further the computer 1b is capable of transmitting or receiving other data such as charge data to or from the credit company via a telephone line, network or cable.

In a server (data base) 5a of the credit company, various data inclusive of users names, addresses and so forth are recorded. In response to an inquiry or a charge request based on a predetermined credit card number from the service provider company, a management computer 5b retrieves from the server 5a the user's name and address corresponding to the relevant credit card number, and sends a bill to the address. In case the credit company and the contractor home are mutually connected via a telephone line, network or cable, it is possible to execute an on-line demand for payment.

For example, when a desired service such as a movie is selected by manipulating the remote controller 6 in a contractor home and a purchase of the relevant service is commanded, this command is supplied from the set-top box 3 via the cable to the service provider company. At this time, the set-top box 3 supplies also an inherent user ID simultaneously to the service provider company. Then the service provider company recognizes a purchase demand of the desired service by a predetermined user who is identified by the user ID.

In the service provider company, the credit card number stored correspondingly to the user ID, which is supplied from the contractor home, correspondingly in advance at the time of contract is retrieved from the server 1a, and an inquiry and a charge request are executed to the credit company via a network or the like. And a desired service demanded for purchase, i.e., a movie in this example, is transmitted via a cable to the contractor home.

In the credit company, the fee for the service is demanded to the user in response to the charge request from the service provider company, while the fee for the

service is paid to the service provider company. As described above, a demand for payment of the fee to the user is executed by mailing a bill or by on-line transmission of the necessary data in case the credit company and the contractor home are mutually connected via a telephone line or the like.

In the case of an access by a child, it is regarded as a partial charge to his parent and is so processed in the credit company. More specifically, the imaginary account 2 of the child opened in the service provider company is set actually in his parent's account, so that the fee withdrawn from the child's imaginary account 2 is withdrawn actually from his parent's account.

For the purpose of enabling the server/processor 1 to recognize an access by a child, it is possible to devise the system in such a manner that a child needs to input a predetermined code number when making a charge request, or that a remote controller is prepared to be used exclusively by the child and, when a desired service is selected or a charge request is made by manipulating such a remote controller, it is regarded as an access by the child, and then predetermined identification data of the child is transmitted from the set-top box 3 to the server/processor 1. Consequently, the server/processor 1 is rendered capable of recognizing whether the accessing user is a child or not.

Fig. 4 is a flow chart for explaining the detailed operation of the charging system performed when a child is serviced with a game. First at step S11, a child selects a desired game program by manipulating the TV 4 or the remote controller 6, and approves a charge for the fee. Then the identification number of the selected game program and a charge request thereof are transmitted from the set-top box 3 via a predetermined cable to the server/processor 1.

Subsequently the operation proceeds to step S12, where the server/processor 1 recognizes a charge to the service requested by the child, i.e., a game program in this case, and makes a decision as to whether the fee for the requested service is less than or equal to the balance in the child's imaginary account 2, that is, whether the balance in the imaginary account 2 is sufficient or not. And if the result of such a decision signifies that the balance in the child's imaginary account 2 is insufficient for the fee of the game program, the operation proceeds to step S16 where a message indicating insufficiency of the balance is displayed on the screen of the TV 4. Then the operation returns to step S11, and the steps subsequent thereto are repeated.

Meanwhile, if the result of the above decision signifies that the balance in the child's account 2 is sufficient for the fee of the game program, the operation proceeds to step S13, where a check message of confirmation is further displayed on the screen of the TV 4 for example.

Thereafter the operation proceeds to step S14, where the server/processor 1 makes a decision as to whether the child has pushed the check button or not. More specifically, when the child has pushed the check button provided for confirmation on the TV 4 or the remote controller 6 while watching the check message displayed on the screen of the TV 4 at step S13, then a corresponding signal is transmitted from the set-top box 3 via a cable or the like to the server/processor 1, so that the above decision is made in accordance with this signal.

If the result of the decision at step S14 signifies that the child has not pushed the check button, the operation returns to step S11, and the steps subsequent thereto are repeated. Meanwhile, if the result of the above decision signifies that the child has pushed the check button, the operation proceeds to step S15 where the fee for the service is withdrawn from the imaginary account 2, and the entire process is completed.

In the fee charging process to the child in this embodiment, it is not necessary for the child to input any credit card number differently from a usual case where a parent receives a service. Accordingly, there is no method of ascertaining if the user is really a regular contractor or not. However, the amount is not much since the account is an imaginary one for charging a child and, considering an advance approval of the parent who is an actual owner of the credit card, a simpler means has been selected for convenient utilization of any service by the child. If this selection raises a problem, a modification can be achieved with facility to further enhance the safety by necessitating the child to input a code number.

Fig. 5 is a flow chart for explaining the detailed operation performed in the service provider company. First at step S21, the situation of the service utilization and the balance are referred at the end of each month by the server/processor 1. Next at step S22, a decision is made by the server/processor 1 as to whether any service has been utilized or not.

If the result of the above decision is negative to signify no utilization of any service, the operation proceeds to step S25, where the entire balance in the imaginary account 2 is carried forward to the next month. Then the operation returns to step S21, and the steps subsequent thereto are repeated. Meanwhile, if the result of the decision at step S22 is affirmative to signify utilization of some service, the operation proceeds to step S23.

At step S23, a detailed statement of the service and the fee therefor are sent by the parent's credit card number to the credit company.

Then the operation proceeds to step S24, where the current balance in the imaginary account 2 is carried forward to the next month by the server/processor 1. After that, the operation returns to step S21, and the steps subsequent thereto are repeated.

In this case, only the amount used is billed to the card company, and the current balance is carried forward to the next month. It is usual that the balance tends to decrease as the child is provided with a service and pays the fee for the same. However, in an exemplary future case where the child receives a prize by giving a

correct answer in a quiz program or the like or by winning in an on-line game, the balance may increase due to transfer of the prize money to the imaginary account 2. Since it is impossible in this system to cash the balance in the imaginary account 2, the prize money is consumed also in a form of on-line shopping or the like, so that the service provider company can acquire a merit that the cash never flows out therefrom.

Fig. 6 is a flow chart for explaining the operation performed when a parent sets a balance in a child's imaginary account 2. Initially at step S31, the parent opens the child's imaginary account 2 in the server/processor 1 in the service provider company.

Then the operation proceeds to step S32, where the parent inputs an initial amount to be preset in the imaginary account 2 and further inputs his credit card number. Subsequently the operation proceeds to step S33, where the server/processor 1 inquires of the corresponding card company to make a collation as to whether the inputted credit card number is correct or not. And if the result of such a collation signifies that the inputted credit card number is not correct, the operation returns to step S31 and the steps subsequent thereto are repeated. Meanwhile, if the result of the above collation signifies that the inputted credit card number is correct, the operation proceeds to step S34.

Then at step S34, the balance of the amount inputted at step S32 is set in the imaginary account 2.

The processing flow for opening a new imaginary account 2 is similar to that for practically opening a bank account. This idea is derived from a prepaid card such as a telephone card, and the right of spending a fixed amount of money, which is spendable freely by a child, is purchased by his parent in advance using the parent's card number. The service provider company first inquires of the card company to ascertain if the card number is correct or not, then opens an imaginary account 2 and presets a predetermined amount therein.

Thus, in the service for providing information to each home via a cable, telephone line or network as described above, it becomes possible for a parent to easily manage his child's utilization of the information provision type service by opening a child-charging imaginary account in the server/processor 1 in the service provider company.

In an example where a parent opens his child's imaginary account 2 in the server/processor 1 and, in the same manner as to give a cash allowance to the child per month, presets in the account 2 the amount equivalent to the child's monthly spending money, the parent is rendered capable of managing the child's expense properly so that the child may not spend much amount carelessly for the service, hence realizing prevention of any trouble that results from purchase by the child in mail order or the like, e.g., any trouble relative to purchase of unnecessary commodities or payment therefor.

Meanwhile, as viewed from the service provider company, participation of children to the service can be promoted to consequently increase expectable charging amounts, since the parent opens the above-described imaginary account 2 and presets a predetermined amount therein, whereby the child is permitted to receive a desired service on the basis of his individual judgment even in the absence of the parent in front of the set-top box 3.

Further, any service can be provided in a state where the independency of each child is respected, so that it becomes possible for the child at home to have quasi-experience of a business transaction extremely similar to a real one.

It is a matter of course that the parent can grasp the contents of the detailed account statement on the basis of the bill sent from the credit company for settlement, and is informed successively of the details by receiving from the service provider company the report of the imaginary account 2 in the server/processor 1, without the necessity of waiting for delivery of the detailed account statement.

And due to the construction where the balance of the imaginary account 2 is displayed on the display device 4 such as a television receiver, the child is enabled to be conscious of paying an equivalent for the provided service, and this brings about an advantage of raising the independency of the child.

Opening the child's imaginary account 2 is quite similar to the case of giving a family credit card to the child and limiting the amount of its credit. However, although a general credit card is usable as cash in nearly entire business transactions, this electronic imaginary account 2 is totally different therefrom in the point that it is available merely in the world of interactive on-line service such as on-line shopping or video-on-demand.

Therefore, in contrast with the case of a credit card where the situation of service utilization is transmitted with a detailed account statement sent per month, it is possible in the imaginary account to successively monitor the situation with a merit of on-line service. For example, a parent manipulates the TV 4 or the remote controller 6 to transmit a command, which is used for monitoring the situation of service utilization in the imaginary account 2, to the server/processor 1 via the set-top box 3. Then, in response to the command from the TV 4 or the remote controller 6, the server/processor 1 inquires into the situation of service utilization in the imaginary account 2 and transmits the result via the set-top box 3 to the TV 4. Consequently, the current situation of service utilization in the imaginary account 2 is displayed on the screen of the TV 4. Since the child's imaginary account is thus opened, an application program for enabling the parent to successively monitor the child's service utilization can be constructed with facility.

In case a child is permitted to carry an ordinary credit card with him, it is impossible for his parent to limit the place where the child may use the card. According to the present invention, however, the parent is capable of

preventing his child's access to any undesirable service by previously restricting some programs which are selectable by the use of the child's imaginary account 2. Consequently, provision of any service or information unsuitable for participation of children can be restricted.

Although in the above embodiment the server/processor 1 and the set-top box 3 are connected mutually via a cable, any other transmission medium such as a telephone line, an optical fiber cable or a network cable may also be employed for the connection.

Further, a monthly withdrawal from the imaginary account in the above embodiment is changeable to be weekly, daily or per service utilization.

Thus, according to the charging method or the charging system of the present invention in interactive on-line service, an imaginary account for a child of the relevant user is opened, and the fee for the service provided to the user's child is withdrawn from the imaginary account. Therefore the charging mode is changeable depending on the person to be provided with the service, whereby the child's utilization of the service is rendered allowable under supervision of the parent. Consequently it becomes possible to promote participation of children to the service to eventually enhance the added value of the service.

In one modification of the charging method or system, the child is permitted to receive provision of desired service within a range of the limited maximum amount preset in the imaginary account, so that any excessive service utilization by the child is rendered preventable to consequently eliminate any excessive charge.

And in another modification, providable service programs are limited at withdrawal of the fee from the imaginary account, so that it becomes possible to restrict provision of any service that is not suited for participation of children, and therefore the content of the service can be changed in accordance with the serviced person.

The invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. The present embodiment is therefore to be considered in all respects as illustrative and not restrictive, the scope of the invention being indicated by the appended claims rather than by the foregoing description and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced therein.

Claims

1. A charging method in interactive on-line service where a server and terminals of users are mutually connected by means of a transmission medium, and a predetermined service is provided from said server to the terminals via the transmission medium, and the fee for such service is collected from each user individually, said charging method comprising the steps of:

   opening, in said server, an imaginary account for a child of the relevant user with a limited maximum amount; and

   withdrawing from said imaginary account the fee for the service provided to the user's child.

2. A charging system in interactive on-line service where a server and terminals of users are mutually connected by means of a transmission medium, and a predetermined service is provided from said server to the terminals via the transmission medium, and the fee for such service is collected from each user individually, said charging system comprising:

   a setting means for setting an imaginary account of the relevant user's child with a limited maximum amount; and

   a charging means for withdrawing from said imaginary account the fee for the service provided to the relevant user's child.

3. A charging method according to claim 1, or system according to claim 2, wherein said server provides the predetermined service to the relevant user's child within a range of the limited maximum amount preset in said imaginary account.

4. A charging method according to claim 1, or system according to claim 2, or any claim dependent therefrom, wherein, when withdrawing the fee from said imaginary account, said server restricts the service providable to the terminal.

5. The charging method according to claim 1, or system according to claim 2, or any claim dependent therefrom, wherein, in response to an input code number preset for the relevant user's child, said server identifies an access by the child and then withdraws from said imaginary account the fee for the service provided to the relevant user's child.

6. A charging method according to claim 1, or system according to claim 2, or any claim dependent therefrom, wherein, in response to input identification data indicative of the child and added in the terminal by a manipulation of the child's exclusive remote control means, said server identifies an access by the child and then withdraws from said imaginary account the fee for the service provided to the relevant user's child.

7. The charging method according to claim 1, or system according to claim 2, or any claim dependent therefrom, wherein, in response to an input request generated by a manipulation in the terminal for con-

firming the situation of service utilisation in said imaginary account, said server transmits to said terminal the data representing the situation of service utilisation in said imaginary account.

FIG. 1

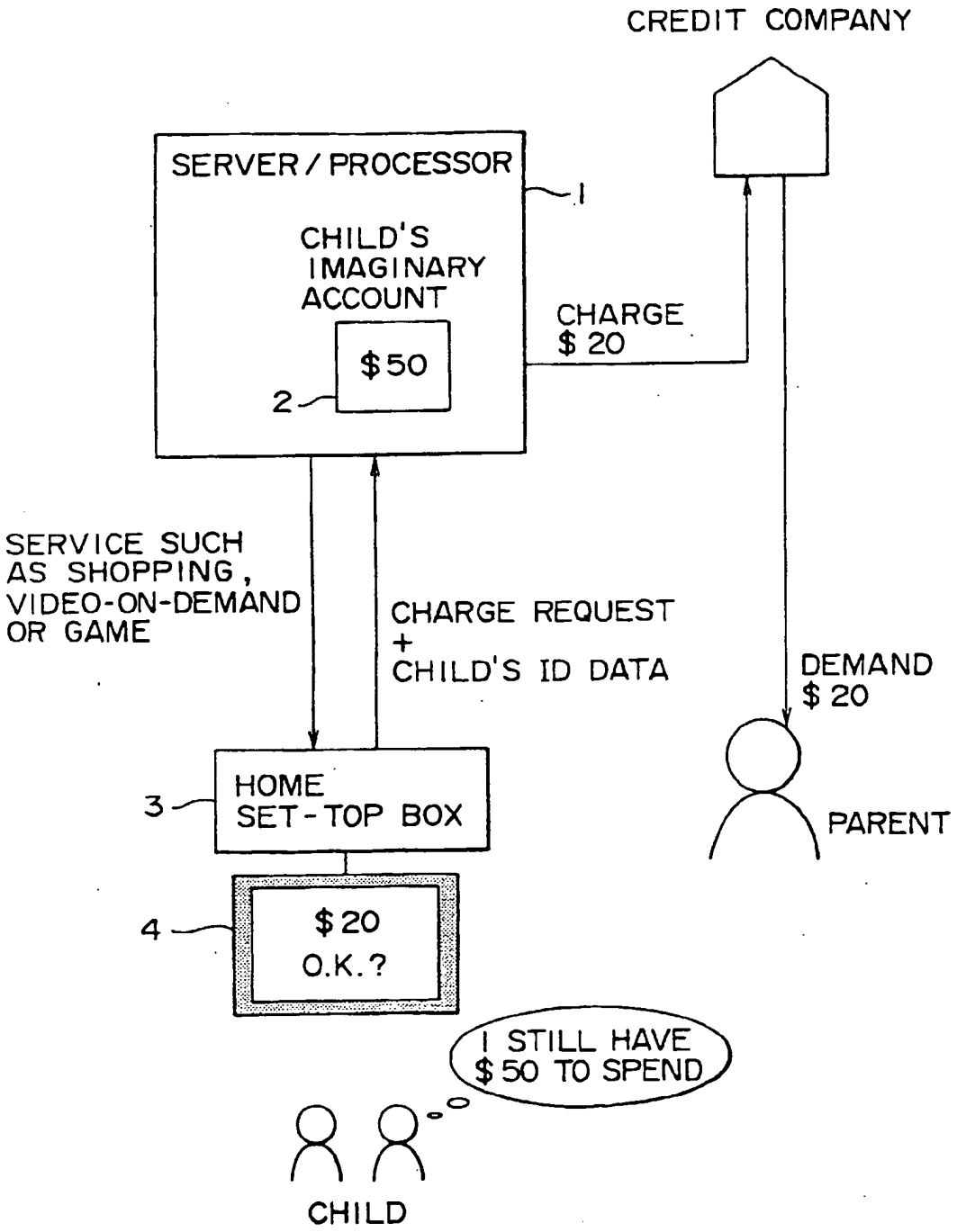


FIG. 2

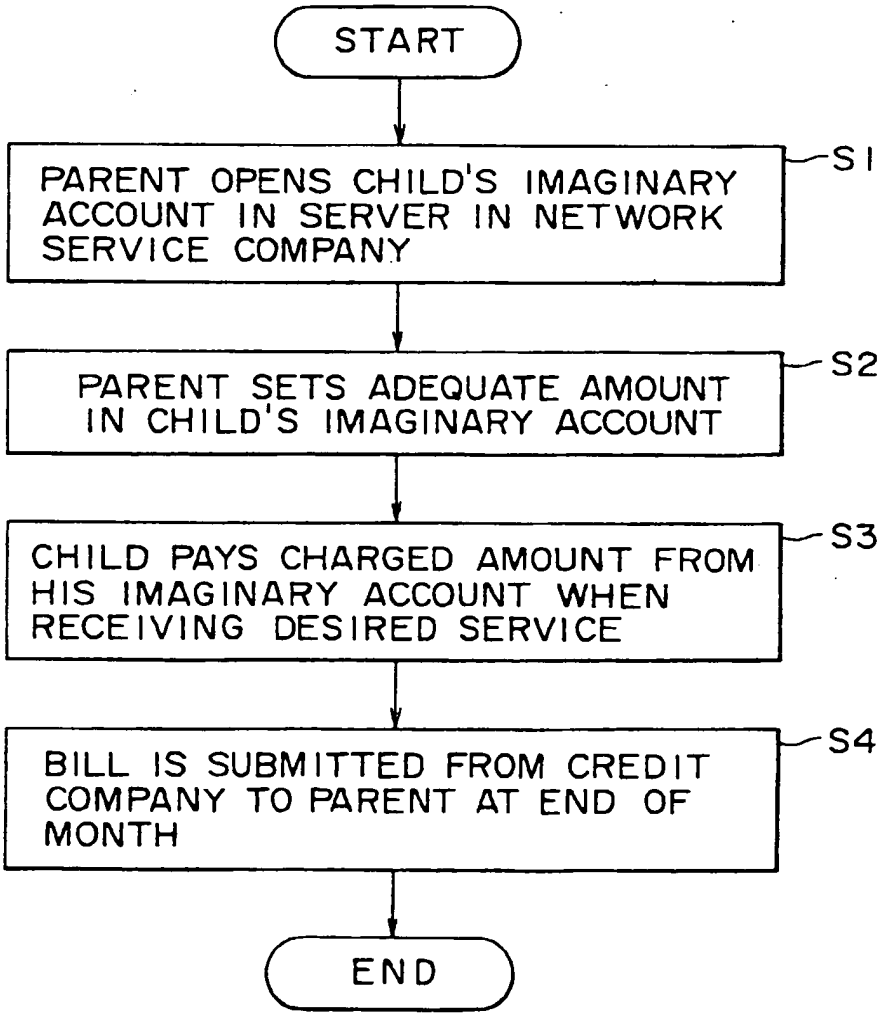


FIG. 3

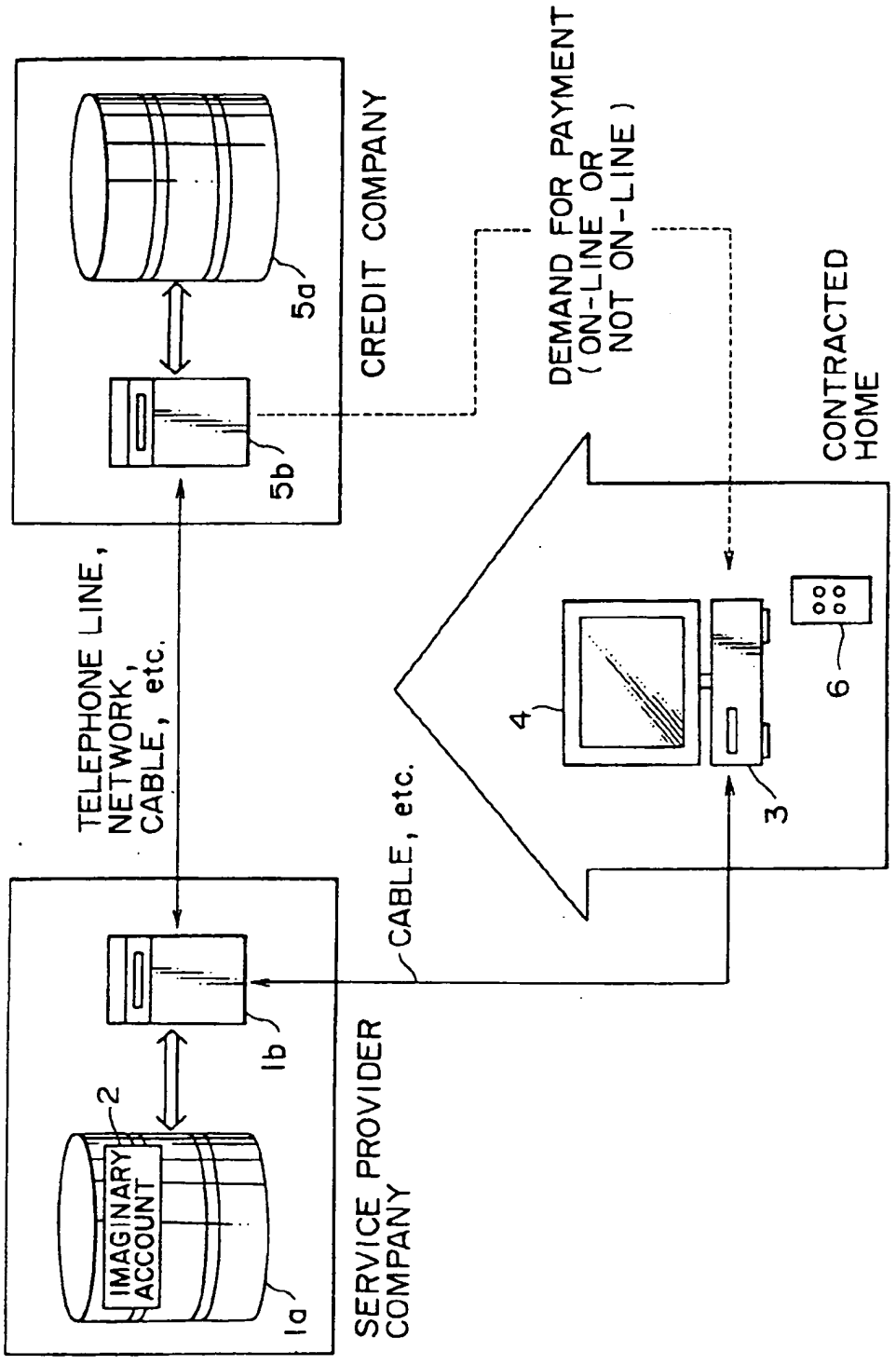


FIG. 4

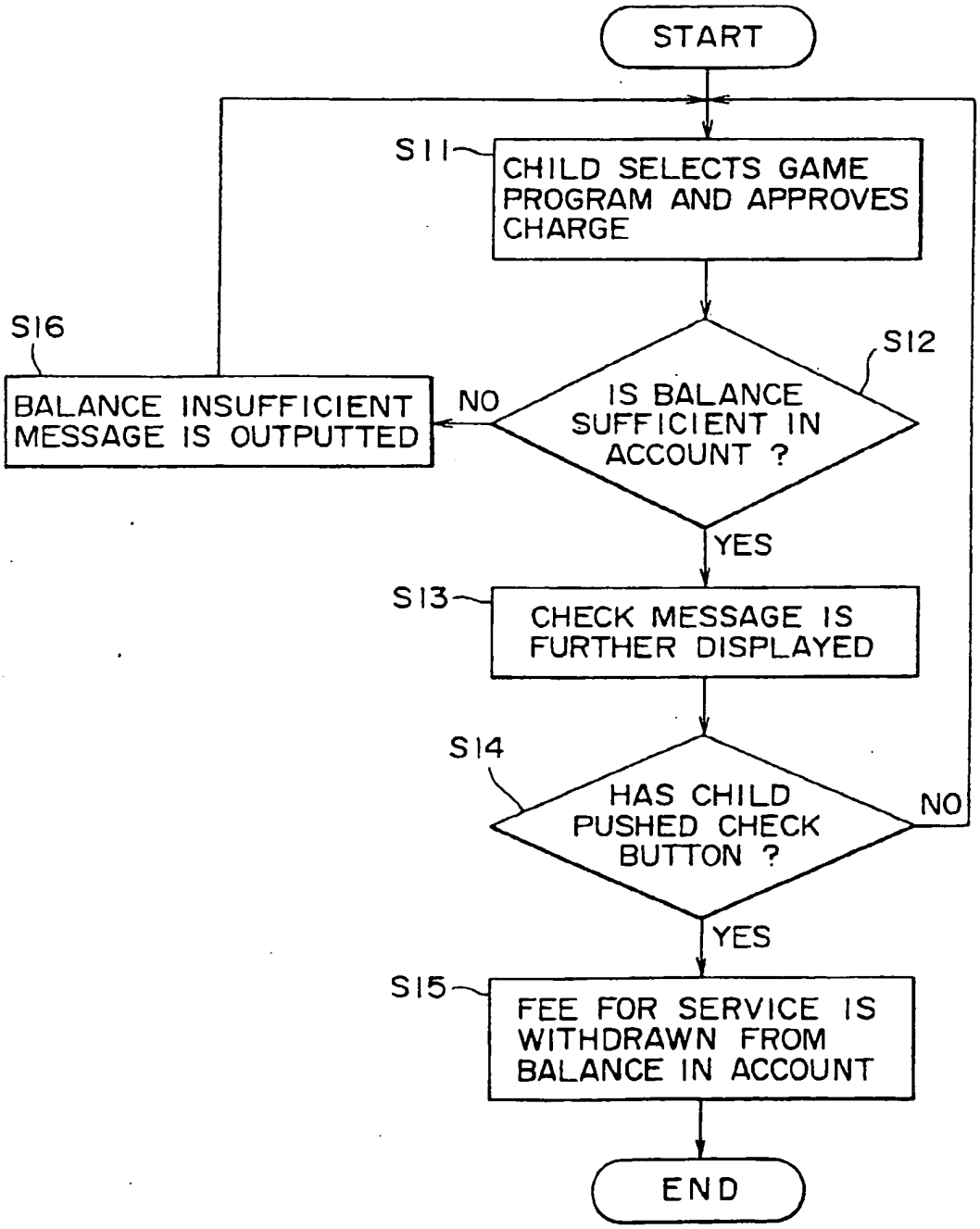


FIG. 5

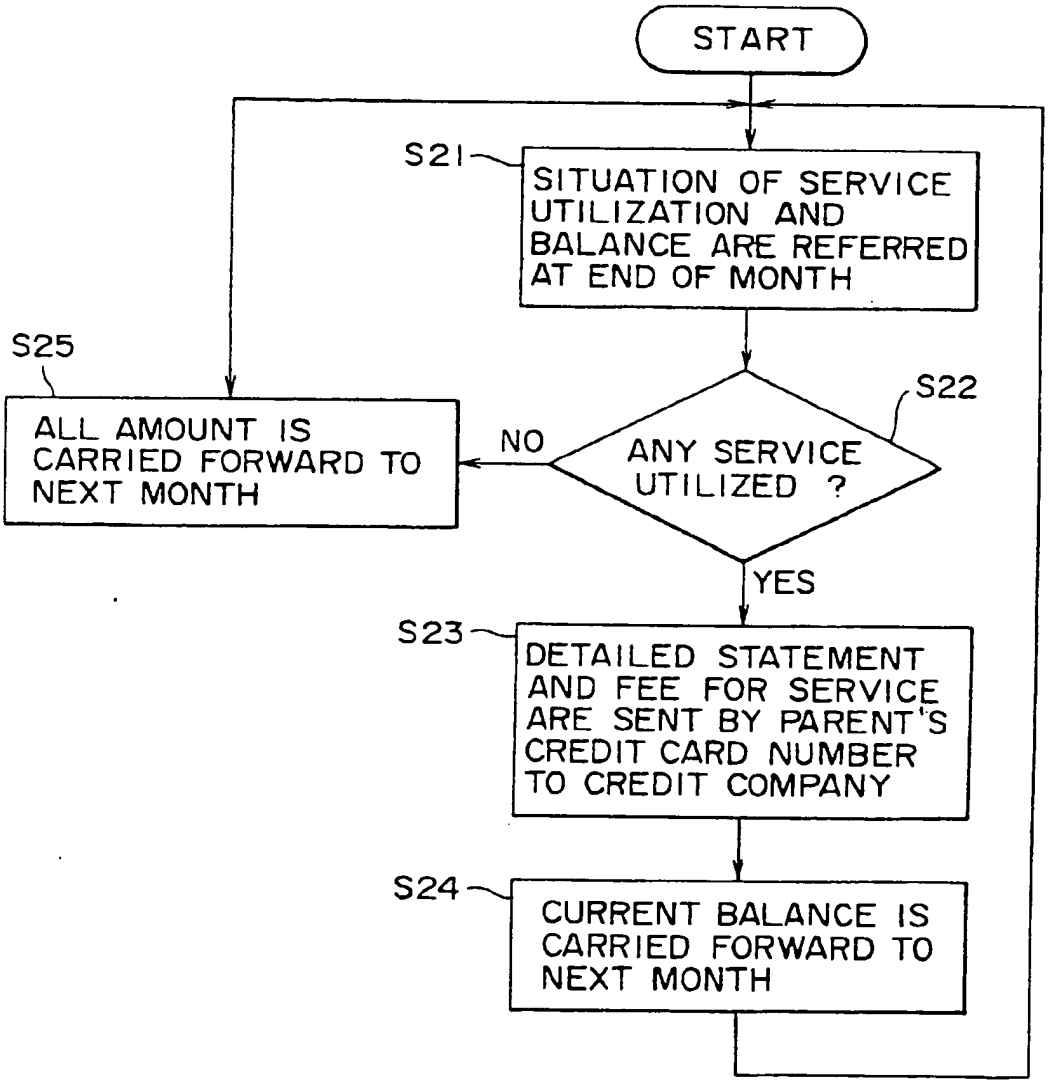
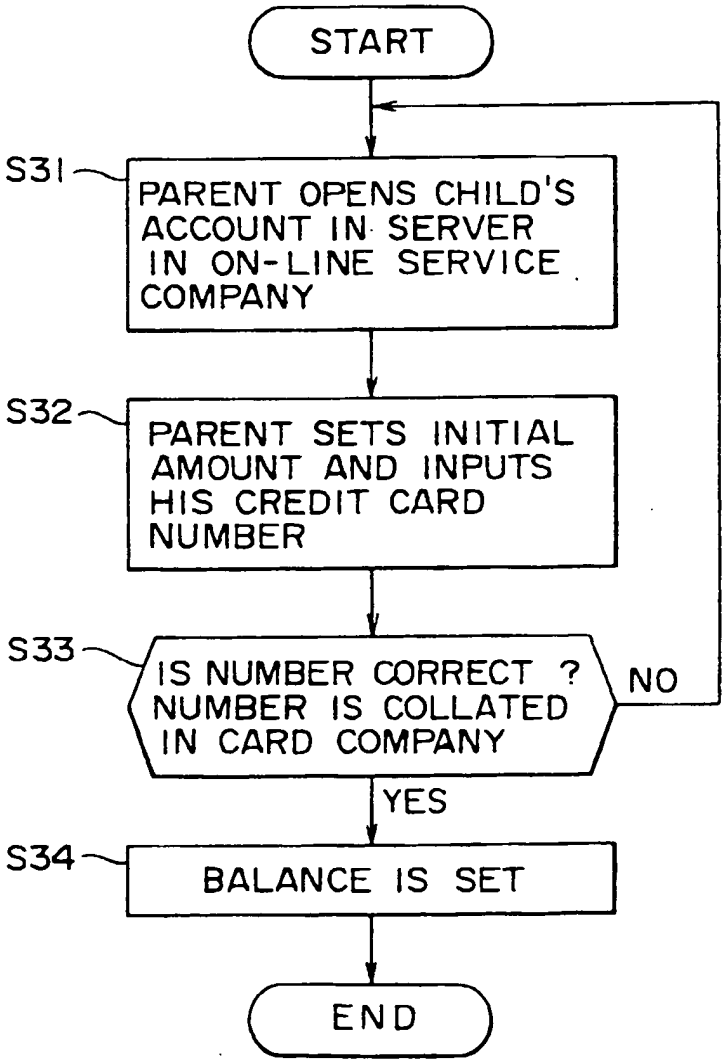


FIG. 6



Requested Patent: GB2136175A

Title: FILE ACCESS SECURITY METHOD AND MEANS ;

Abstracted Patent: US4588991 ;

Publication Date: 1986-05-13 ;

Inventor(s): ATALLA MARTIN M (US) ;

Applicant(s): ATALLA CORP (US) ;

Application Number: US19830472609 19830307 ;

Priority Number(s): US19830472609 19830307 ;

IPC Classification: ;

Equivalents:

DE3407642, FR2542471, JP1434850C, JP59169000, JP62042304B, ZA8401400 ;

ABSTRACT:

An improved file access security technique and associated apparatus accesses data which is stored in encrypted form under one encryption key and re-stores the data re-encrypted under another encryption key, and produces a record of each access and data re-encryption both as the control source of encryption keys for access and re-entry of encrypted data and as a secured audit record of users that had access to each file.

(12) UK Patent Application (19) GB (11) 2 136 175 A

(43) Application published 12 Sep 1984

(21) Application No 8405950

(22) Date of filing 7 Mar 1984

(30) Priority data

(31) 472609 (32) 7 Mar 1983 (33) US

(51) INT CL³
H03K 13/24 G06F 15/00

(52) Domestic classification
G4A AP

(56) Documents cited
None

(58) Field of search
G4A

(71) Applicant
Atalla Corporation, (USA—California),
2363 Bering Drive, Sane Jose, California 95131,
United States of America

(72) Inventor
Martin M. Atalla

(74) Agent and/or address for service
Peter A. Oliver, 8 Coombe Close, Frimley, Surrey,
GU16 5DZ

(54) File Access Security Method and Means

(57) An improved file access security technique and associated apparatus 9 accesses data which is stored at 17 in encrypted form under one encryption key and re-stores the data re-encrypted under another encryption key, and produces at 19 a record of each access and data re-encryption both as the control source of encryption keys for access and re-entry of encrypted data and as a secured audit record of users that had access to each file.

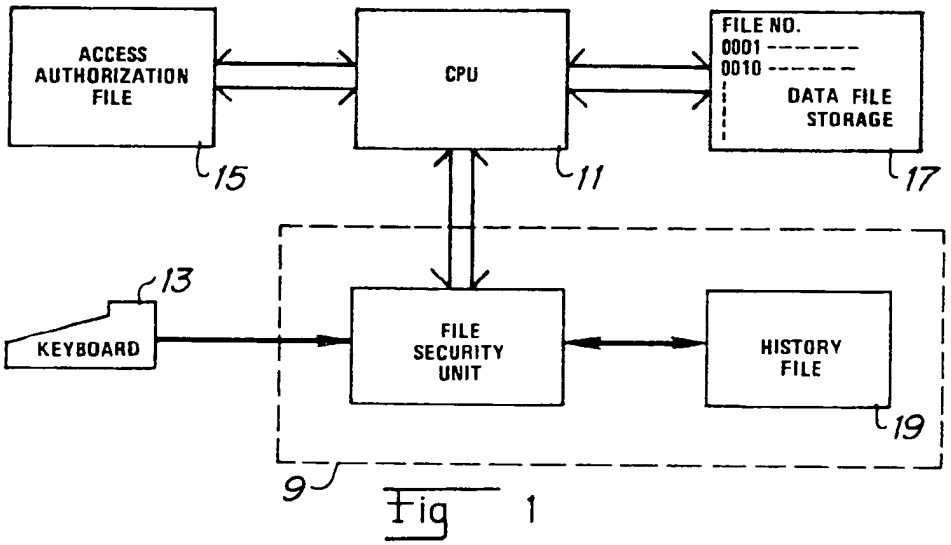


Fig 1

GB 2 136 175 A

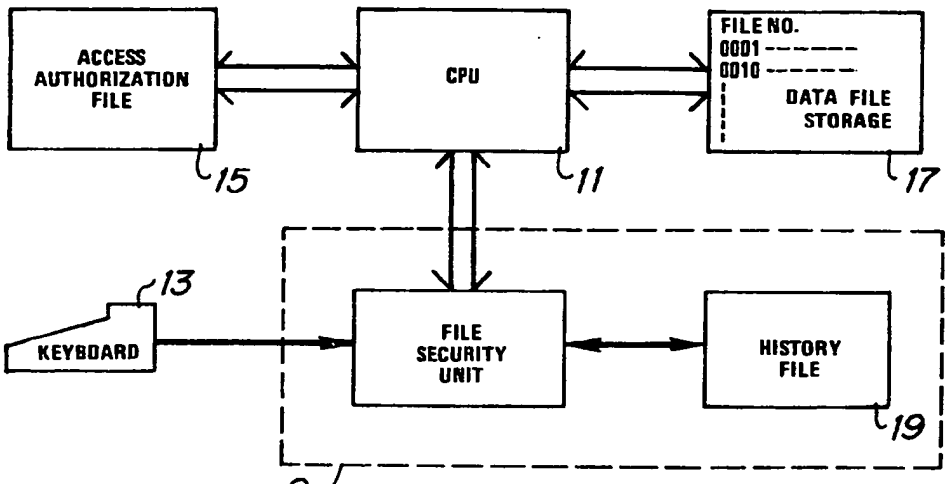

ACCESS AUTHORIZATION FILE
15
CPU
11
FILE NO.
0001 -------
0010 -------
DATA FILE STORAGE
17
13
KEYBOARD
FILE SECURITY UNIT
HISTORY FILE
19
9

Fig 1

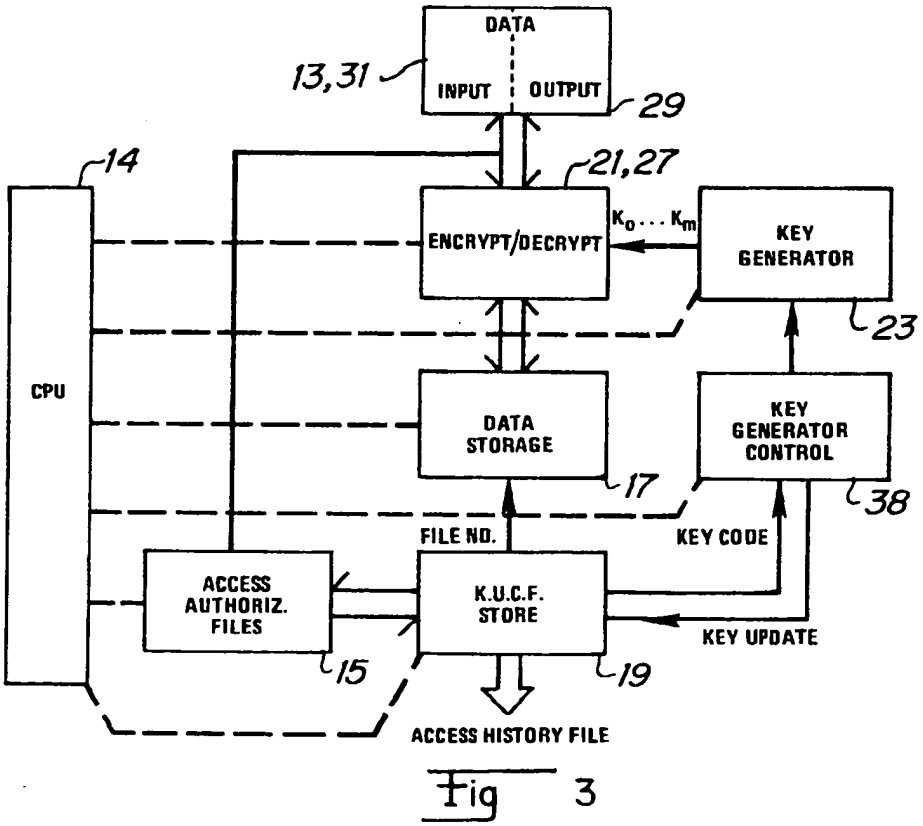

DATA
13,31
INPUT
OUTPUT
29
14
21,27
ENCRYPT/DECRYPT
$K_0 \ldots K_m$
KEY GENERATOR
23
CPU
DATA STORAGE
17
KEY GENERATOR CONTROL
38
FILE NO.
KEY CODE
ACCESS AUTHORIZ. FILES
15
K.U.C.F. STORE
19
KEY UPDATE
ACCESS HISTORY FILE

Fig 3

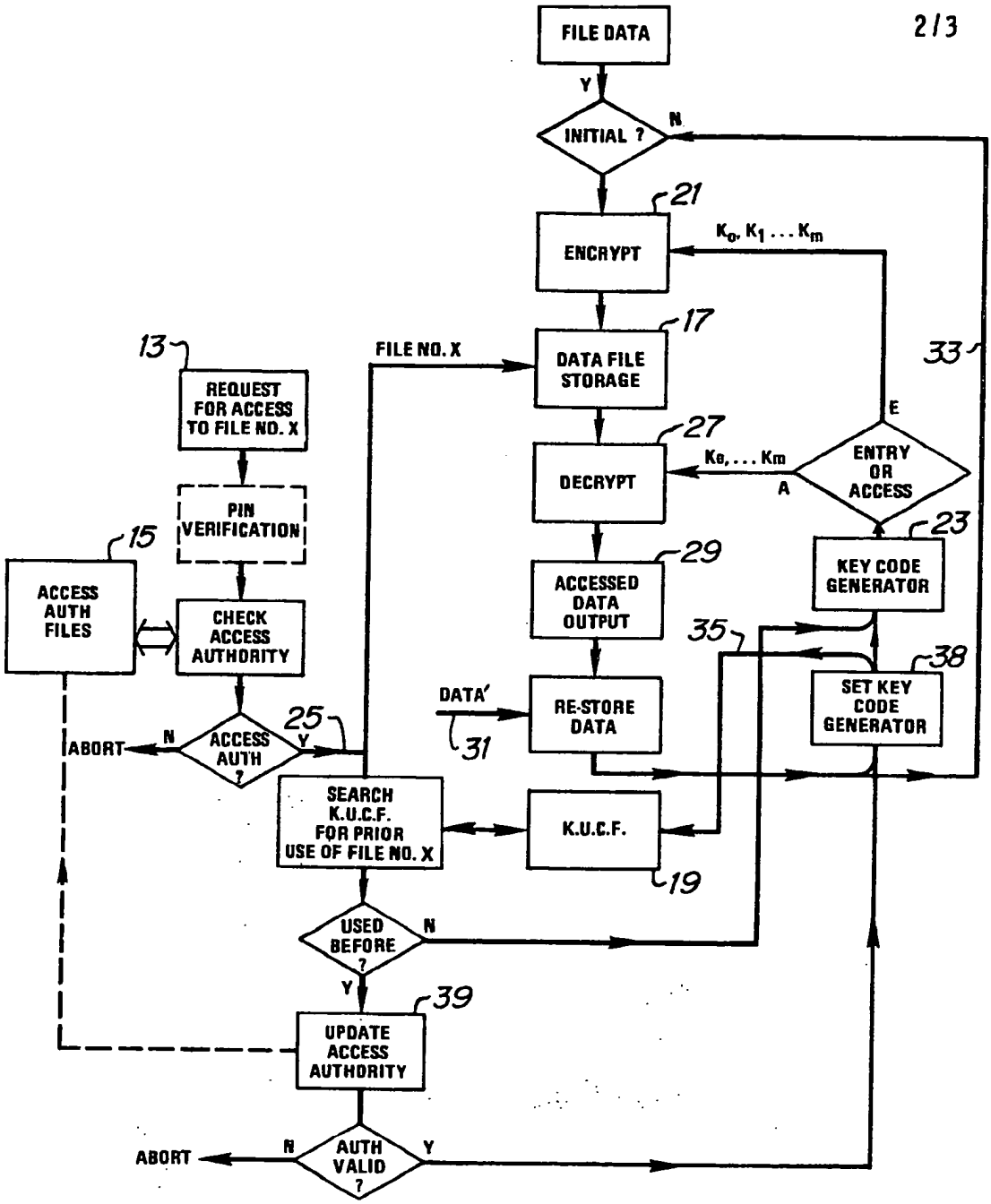

FILE DATA
Y
INITIAL ?
N
21
ENCRYPT
$K_0, K_1 \ldots K_m$
17
DATA FILE STORAGE
FILE NO. X
33
E
27
DECRYPT
$K_0, \ldots K_m$
A
ENTRY OR ACCESS
23
KEY CODE GENERATOR
13
REQUEST FOR ACCESS TO FILE NO. X
PIN VERIFICATION
15
ACCESS AUTH FILES
CHECK ACCESS AUTHORITY
29
ACCESSED DATA OUTPUT
35
38
SET KEY CODE GENERATOR
DATA'
31
RE-STORE DATA
ABORT
N
ACCESS AUTH ?
Y
25
SEARCH K.U.C.F. FOR PRIOR USE OF FILE NO. X
K.U.C.F.
19
USED BEFORE ?
N
Y
39
UPDATE ACCESS AUTHORITY
ABORT
N
AUTH VALID ?
Y

Fig 2

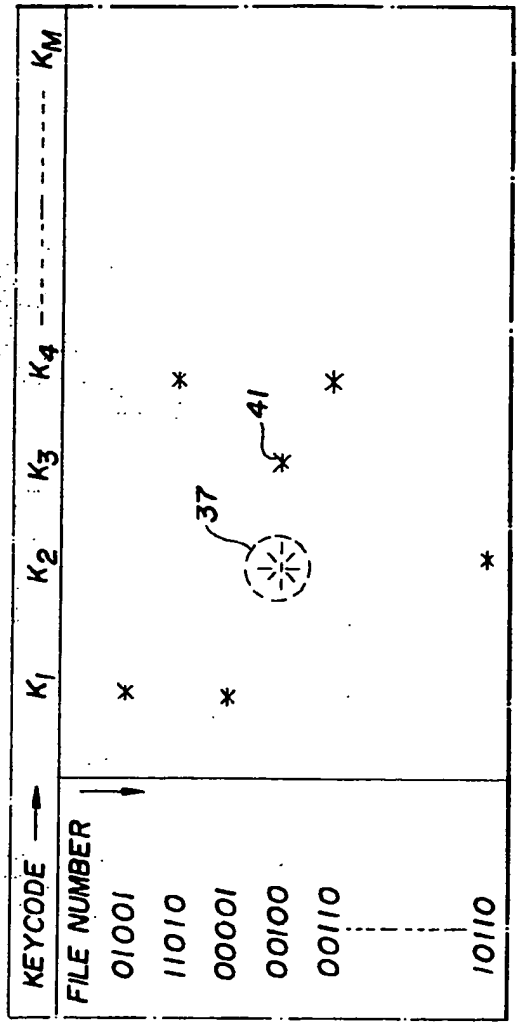

KEYCODE
FILE NUMBER
K1
K2
K3
K4
KM
01001
11010
00001
00100
00110
10110
37
41

FIG. 4

SPECIFICATION

File Access Security Method and Apparatus

This invention is concerned with a method of and apparatus for securing data files in storage.

Many known computer-controlled operations on secured data files require verification of the identity of an individual seeking to access a file before the data (usually in encrypted form) can be accessed (see, for example, U.S. Patents 3,938,091, 3,587,051, 3,611,293 and 4,198,619). In addition, many known record-securing schemes including those associated with credit cards, require verification of both the authority of the using individual and the authenticity of the data in the record, to protect against unauthorized users and against counterfeit or duplicate records. Schemes of this type are disclosed in U.S. Patents 4,304,990, 4,328,414 and 4,357,429.

One disadvantage associated with computer-controlled security schemes of these types is that there is typically no indication left on file of who gained access to a secured record.

The present invention provides a method of securing data files in storage against unauthorized access, the method comprising the steps of encrypting file data as a selected logical combination thereof with an initial one of a plurality of encryption key codes to produce file data in encrypted form for storage at selected file address locations, establishing a record of accesses to each selected file address location and the one of the plurality of encryption key codes with which the file data at the address location is encrypted, processing a request for access to file data at a selected file address location by determining from the record the number of prior accesses thereof and the encyrption key code associated therewith, decrypting file data at the selected file address location using said associated encryption key code, re-encrypting file data for said selected file address location using a new one of said plurality of encryption key codes in said selected logical combination, storing the newly re-encrypted file data at the accessed file address location, and modifying the record to indicate an additional access to the selected file address location and the new encryption key code associated therewith.

In performing a method as set forth in the last preceding paragraph, it is preferred that in carrying out the step of decrypting, file data at a selected filed address location is decrypted using said initial encryption key code in response to determination from the record that said selected file address location was not previously accessed.

A method as set forth in either one of the last two immediately preceding paragraphs may further comprise the additional steps of establishing a file of user access authorizations, and prior to accessing a selected file address location, determining the authorization status of a user to gain access to the selected file address location.

A method as set forth in the last preceding paragraph may further comprise the additional step of selectively altering the access authorization of a user to gain subsequent access to the selected file address location in response to re-encryption of the file data for storage at the selected filed address location.

A method as set forth in any one of the last four immediately preceding paragraphs may further comprise the steps of reinitializing all the file data by decrypting the file data at each selected file address location using the encryption key code thereof determined from the record, and re-encrypting the file data at each such file address location using a new initial one of a plurality of key codes.

In performing a method as set forth in the last preceding paragraph, it is preferred that in carrying out the reinitialization step, the file data at any file address location which is not indicated in the record to have been accessed previously is decrypted using the initial encryption key code.

The present invention also provides apparatus for securing data filed in storage against unauthorized access, comprising storage means for storing file data in encrypted form at selectable file address locations, encryption means for supplying encrypted file data to a selected file address location as the logical encoding combination of file data and an encryption key signal applied thereto, generator means for applying selected encryption key signals to the encryption means, record means for producing indication of selected file address locations and key code signals associated with encryption of file data stored therein, circuit means responsive to indentification of a selected file address location for determining from said record means the encryption key signal associated therewith for setting the generator means to supply the associated encryption key signal, decryption means disposed to receive encryption key signals from the generator means and encrypted file data from the storage means and operable in accordance with said logical encoding combination to decrypt the file data at said selected file address location, and means operable upon the decrypted file data for altering the generator means to supply a new encryption key signal for re-storing the file data at the selected file address location newly encrypted with a new encryption key signal, said means altering the record means to produce an indication of the new encryption key signal associated with file data in the selected file address location.

In apparatus as set forth in the last preceding paragraph, it is preferred that said circuit means is responsive to the indication in said record means that a selected file address location was not previously accessed for setting said generator means to supply the initial encryption key signal to the decryption means.

Apparatus as set forth in either one of the last two immediately preceding paragraphs may further comprise access record means for storing data representative of the authorization of users to selectively access file data in said storage means, and means disposed to receive identification data from a user, and coupled to said circuit means for inhibiting the generator means from supplying an encryption key signal to said decryption means for an unauthorized, identified user.

Apparatus as set forth in the last preceding paragraph may further comprise means responsive to re-storing of file data at the selected file address location newly encrypted with a new encryption key signal for altering the identified user's authorization in said access record means to access said selected file address location.

Apparatus as set forth in the last preceding paragraph but three may further comprise initializing means coupled to said generator means, said encryption means and decryption means and to said record means for setting the generator means to selectively decrypt file data in each file address location using the encryption key signals from said generator means established from the record means for each such file address location, and for re-encrypting the decrypted file data for each file address location using a new initial encryption key signal for re-storage at the respective file address location.

In apparatus as set forth in last preceding paragraph, it is preferred that said initializing means responds to indication from said record means of no previous access to a selected file address locations for decrypting file data therein in using an initial encryption key signal and for re-encrypting the decrypted file data using a new initial encryption key signal to re-store the newly encrypted file data at the respective file address location.

The present invention further provides a file access record produced by a process comprising the steps of storing at selected file address locations file data that is encrypted as the logical combination of file data and selected ones of a plurality of encryption key signals; decrypting file data at a selected file address location using the encryption key signal associated therewith in accordance with said logical combination, re-encrypting the decrypted file data as a logical combination thereof and a new encryption key signal for re-storing at the corresponding file address location, and producing said file access record as the compilation at least of the number of times each selected file address location was decrypted and information indicative of the encryption key signals with which the file data at each selected file address location was reencrypted and re-stored therein.

In accordance with the preferred embodiment of the present invention, a dynamic record of encryption control keys used to gain access initially and at all subsequent occasions to secured encrypted files is generated both as an active element of the accessing scheme and as a secured, historic record for audit purposes of all accesses to encrypted files. In addition, substitutions of outdated files are prevented once a file is accessed, even merely for display without alteration, so that a file once accessed, and therefore with its security compromised, can be resecured against duplication, substitition, and re-use. Schemes of this type are particularly useful in banking and funds-transfer operations where proper access initially to an account file, for example, to effect a withdrawal of funds, must thereafter be carefully controlled to avoid such disastrous practices as multiple replication of the same operation coupled with substitution of the original balance back into the file. Further, the historic record of accesses to files produced by the present invention constitutes an audit record in encrypted form of such accesses.

There now follows a detailed description which is to be read with reference to the accompanying drawings of a method and apparatus according to the present invention; it is to be clearly understood that this method and apparatus have been selected for description to illustrate the invention by way of example and not by way of limitation.

In the accompanying drawings:

Figure 1 is a pictorial block diagram showing one application of the apparatus of the present invention;

Figure 2 is a flow chart illustrating the operation of the apparatus of Figure 1;

Figure 3 is a block diagram of the illustrated embodiment of the present invention; and

Figure 4 is a chart illustrating the formation and operation of the key usage control file according to the present invention.

Referring now to Figure 1, there is shown a pictorial block diagram of the present invention illustrating the addition of an access-securing module 9 to a typical computer system comprising a central-processing unit 11, keyboard controller 13, and memory means 15, 17 for storing files. The memory means 15, 17 may use any conventional form of storage technology such as semiconductor memory, magnetic memory in core, crystal, disc, drum, or tape form, and any combinations thereof, to provide means 17 for storing the data to which access is to be controlled, and to provide means 15 for storing access authorization information about individuals and entities that may access the stored data means 17. The keyboard controller 13 provides manual-entry access to the computer system in conventional manner and is representative of other computer-accessing schemes such as by another computer system, and the like.

In accordance with the present invention, such a typical computer system is modified to include the access-securing module 9 which operates with the computer system to progressively re-encrypt the data in storage in memory means 17 each time a file is accessed, and optionally to

update the access authorization information in storage in the memory means 15 in response to authorizations granted, and to generate historic files in encrypted form of the encryption keys used to decrypt and re-encrypt each file accessed from the memory means 17. In addition, the module 9 operates in a controlled reinitialization mode to restore all files in the memory means 17 to a new, standard encryption key after numerous accesses of files in the memory means 17 have been authorized. The number of accesses before requiring reinitialization is determined by the memory capacity in the module 9.

Referring now to Figures 2 and 3 in addition to Figure 1, there are shown a flow chart and a block diagram, respectively, illustrating the operation of the system of Figure 1 under control of a central processing unit 11. In operation, a person or entity, R, requesting access to a particular file may enter personal identification numbers, information about the particular file, and the like, via the keyboard controller 13. Optionally, a personal-identity verification routine may be performed in conventional manner (as disclosed, for example, in U.S. Patent 3,938,091 or 4,198,619) and the access-authorization files in the memory means 15 may be searched for authorization to access the requested file. All such files in memory means 17 are initially encrypted with an initial key code, K_0, in a conventional manner (for example, using the *D*ata *E*ncryption *S*tandard module available from the *N*ational *B*ureau of *S*tandards) by encrypting the file data in an encryption module 21 with key code, K_0, from a key code generator 23.

With authorization established 25, the particular file #X may be accessed, but decrypting the file #X requires the correct key code. For this purpose, a key-usage control file 19, later described herein in detail, is searched to determine if the file #X was previously accessed. The conditions of prior access, namely, that it was, or it was not previously accessed, are possible. If it was not, then file #X will not appear in the key-usage control file, an indication that it appears in storage provided by the memory means 17 encrypted with the initial key code, K_0. The key code generator 23 is capable of generating a sequence of different key codes K_0, $K, K_2, K_3 \ldots K_n$ and is set to supply key code K_0 to a decryption module 27 (which, of course, may be the same type of DES module, or may be the same module, as the encryption module 21). The requested file #X may therefore be decrypted in conventional manner using the key code K_0 to provide accessed data 29 in clear text. The data is then returned to storage, either without or with new data modifications 31 that reflect a data-oriented transaction such as sale, deposit, withdrawal, or the like, and is re-stored in encrypted form using a new key code K_1. This is accomplished by resetting 38 the key code generator 23 to supply the key code K_1 to the encryption module 21 and returning the data 33 with or without modifications for encryption in the module 21 with the key code K_1. In addition, the key-usage control file 19 is updated to reflect that the file #X was accessed and now resides in storage newly-encrypted with the new key code K_1 in the sequence. Further, the access-authorization in the memory means files 15 may be updated optionally to inhibit further access to file #X by user R, for example, to inhibit R's further access until a "new date", or until accessed by another user, or the like. Subsequent access to file #X by user R, if continuously authorized, or by any other user must be via decryption with the key code K_1.

If file #X was previously accessed, then the key-usage control file 19 will contain the entry of file #X having been previously accessed and returned to storage encrypted with a new key code $K_1, K_2 \ldots K_n$, depending upon the number of previous accesses to file #X. Thus, with reference to the chart of Figure 4 which illustrates the typical entries in the key-usage control file 19, if file #X is file #00100, then the previous accesses to this file resulted in its being re-stored encrypted with key code K_2 (at entry 37). The search of the key-usage control file 19 thus indicates that file #00100 was previously accessed twice and now requires decryption with key code K_2. If authorization of the requesting user is still valid 39, then the key code generator 23 is set to supply the key code K_2 to the decryption module 27 in order to furnish the data in this file in clear text 29. Re-storing the data from this file in modified or unmodified form is accomplished by resetting 38 the key code generator 23 to supply the key code K_3 (entry 41 in Figure 4) to the encryption module 21 for encryption therein of the returned data with the new key code K_3. All retrievals of data in storage in the memory means 17 may be by destructive read of information in the addressed file so that data for restoring therein may be written in in the newly-encrypted form. After numerous accesses to files in storage in the memory means 17, the key-usage control file 19 will typically include entries as illustrated in Figure 4. Such a file optionally may also include codes to identify the particular users who gained access to each file. The file 19 thus provides an audit record of the accesses to the files in the memory means 17. In addition, the key-usage control file 19 is in encrypted form since it neither reveals the data in storage in the memory means 17 nor the actual key codes $K_1 \ldots K_n$ (only generated by the generator 23) required to decrypt the data in storage. Further, the key codes $K_0 \ldots K_n$ which serve as file-protect codes can be generated internally in conventional manner, for example, by a random-number generator 23 and therefore need not be known to anyone.

After numerous accesses to the data in storage 17 which approaches the limit of the sequence of key codes for any particular file, or on a periodic basis, the entire collection of files in storage 17 may be re-encrypted with a new initial key code K_0' of a sequence of new key codes $K_0', K_1' \ldots K_n'$

using the apparatus illustrated in Figure 3 under control of the central processing unit 14. However, since the files in storage 17 are encrypted with different key codes, the key-usage control file 19 must be consulted to determine which key code to use to decrypt the data in each file for re-encryption with a new initial key code K_0'. After completion of this reinitialization mode of operation, the key-usage control file 19 for the sequence of key codes $K \ldots K_n$ may be retired to serve as an historic record of access to the data in storage 17 without compromising the security of the system or of the data in storage 17 under new encryption codes.

CLAIMS

1. A method of securing data files in storage against unauthorized access, the method comprising the steps of:

encrypting file data as a selected logical combination thereof with an initial one of a plurality of encryption key codes to produce file data in encrypted form for storage at selected file address locations;

establishing a record of accesses to each selected file address location and the one of the plurality of encryption key codes with which the file data at the address location is encrypted;

processing a request for access to file data at a selected file address location by determining from the record the number of prior accesses thereof and the encryption key code associated therewith;

decrypting file data at the selected file address location using said associated encryption key code;

re-encrypting file data for said selected file address location using a new one of said plurality of encryption key codes in said selected logical combination;

storing the newly re-encrypted file data at the accessed file address location; and

modifying the record to indicate an additional access to the selected file address location and the new encryption key code associated therewith.

2. A method according to claim 1 wherein, in carrying out the step of decrypting, file data at a selected file address location is decrypted using said initial encryption key code in response to determination from the record that said selected file address location was not previously accessed.

3. A method according to either one of claims 1 and 2 comprising the additional steps of establishing a file of user access authorizations; and prior to accessing a selected file address location

determining the authorization status of a user to gain access to the selected file address location.

4. A method according to claim 3 comprising the additional step of selectively altering the access authorization of a user to gain subsequent access to the selected file address location in response to re-encryption of the file data for storage at the selected file address location.

5. A method according to any one of the preceding claims and further comprising the steps of:

reinitializing all the file data by decrypting the file data at each selected file address location using the encryption key code therefor determined from the record; and

re-encrypting the file data at each such file address location using a new initial one of a plurality of key codes.

6. A method according to claim 5, wherein, in carrying out the reinitialization step the file data at any file address location which is not indicated in the record to have been accessed previously is decrypted using the initial encryption key code.

7. A method of securing data files in storage against unauthorized access substantially as hereinbefore described with reference to the accompanying drawings.

8. Apparatus for securing data files in storage against unauthorized access, comprising:

storage means for storing file data in encrypted form at selectable file address locations;

encryption means for supplying encrypted file data to a selected file address location as the logical encoding combination of file data and an encryption key signal applied thereto;

generator means for applying selected encryption key signals to the encryption means;

record means for producing indication of selected file address locations and key code signals associated with encryption of file data stored therein;

circuit means responsive to identification of a selected file address location for determining from said record means the encryption key signal associated therewith for setting the generator means to supply the associated encryption key signal;

decryption means disposed to receive encryption key signals from the generator means and encrypted file data from the storage means and operable in accordance with said logical encoding combination to decrypt the file data at said selected file address location; and

means operable upon the decrypted file data for altering the generator means to supply a new encryption key signal for re-storing the file data at the selected file address location newly encrypted with a new encryption key signal, said means altering the record means to produce an indication of the new encryption key signal associated with file data in the selected file address location.

9. Apparatus according to claim 8 wherein said circuit means is responsive to the indication in said record means that a selected file address location was not previously accessed for setting said genrator means to supply the initial encryption key signal to the decryption means.

10. Apparatus according to either one of claims 7, 8 and 9 and further comprising:

access record means for storing data representative of the authorization of users to selectively access file data in said storage means; and

means disposed to receive identification data from a user, and coupled to said circuit means for inhibiting the generator means from supplying an encryption key signal to said decryption means for an unauthorized, identified user.

11. Apparatus according to claim 10 comprising means responsive to re-storing of file data at the selected file address location newly encrypted with a new encryption key signal for altering the identified user's authorization in said access record means to access said selected file address location.

12. Apparatus according to claim 8 comprising initializing means coupled to said generator means, said encryption means and decryption means and to said record means for setting the generator means to selectively decrypt file data in each file address location using the encryption key signals from said generator means established from the record means for each such file address location, and for re-encrypting the decrypted file data for each file address location using a new initial encryption key signal for restorage at the respective file address location.

13. Apparatus according to claim 12 wherein said initializing means responds to indication from said record means of no previous access to a selected file address location for decrypting file data therein in using an initial encryption key signal and for re-encrypting the decrypted file data using a new initial encryption key signal to re-store the newly encrypted file data at the respective file address location.

14. Apparatus for securing data files in storage against unauthorized access substantially as hereinbefore described with reference to the accompanying drawings.

15. A file access record produced by a process comprising the steps of:

storing at selected file address locations file data that is encrypted as the logical combination of file data and selected ones of a plurality of encryption key signals;

decrypting file data at a selected file address location using the encryption key signal associated therewith in accordance with said logical combination;

re-encrypting the decrypted file data as a logical combination thereof and a new encryption key signal for restoring at the corresponding file address location; and

producing said file access record as the compilation at least of the number of times each selected file address location was decrypted and information indicative of the encryption key signals with which the file data at each selected file address location was re-encrypted and re-stored therein.

Printed in the United Kingdom for Her Majesty's Stationery Office, Demand No. 8818935, 9/1984. Contractor's Code No. 6378.
Published by the Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.

(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) 415884 (32) 02.10.1989 (33) US

(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

(71) Applicant
Sun Microsystems Inc

(Incorporated in the USA - Delaware)

2550 Garcia Avenue, Mountain View, California 94043, United States of America

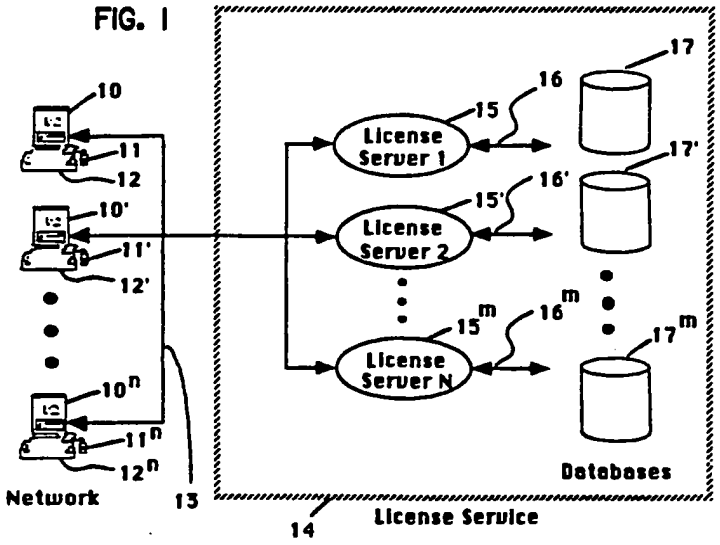
(72) Inventor
John Richard Corbin

(74) Agent and/or Address for Service
Potts Kerr and Co
15 Hamilton Square, Birkenhead, Merseyside, L41 6BR, United Kingdom

(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

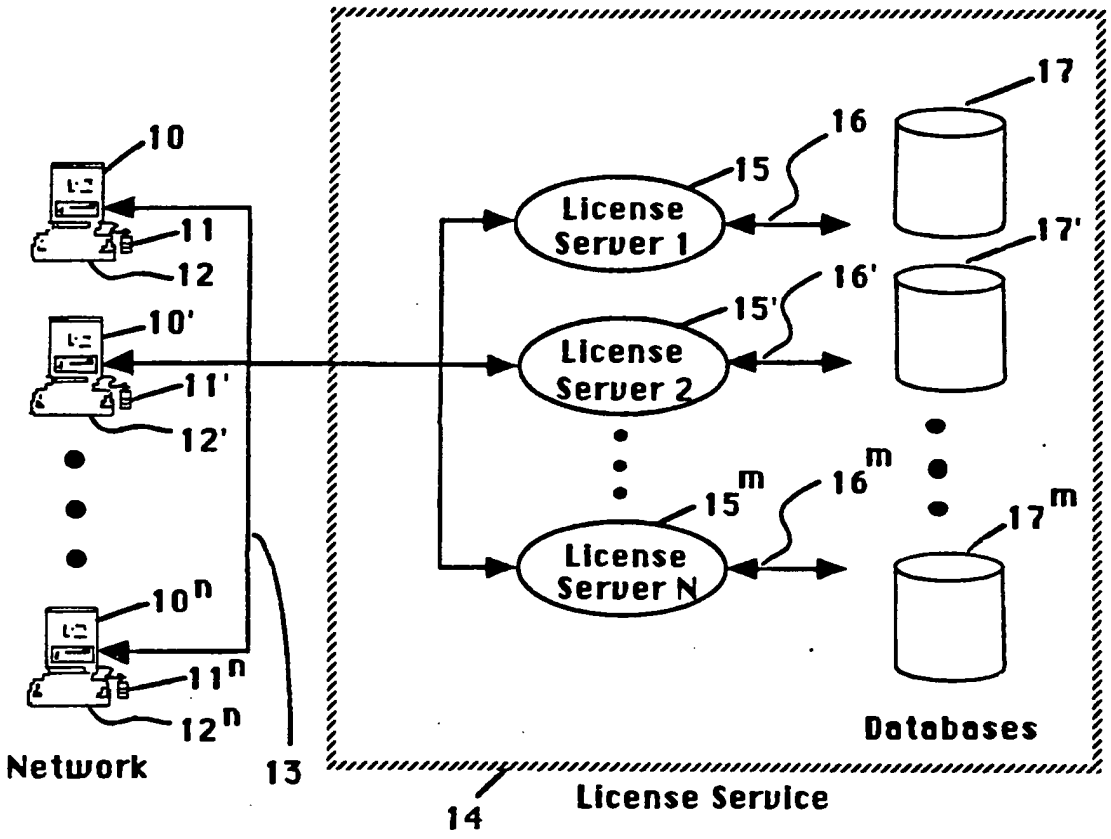

10
11
12
10'
11'
12'
10^n
11^n
12^n
Network
13
License Server 1
License Server 2
License Server N
15
16
17
15'
16'
17'
15^m
16^m
17^m
Databases
14
License Service

FIG. 1

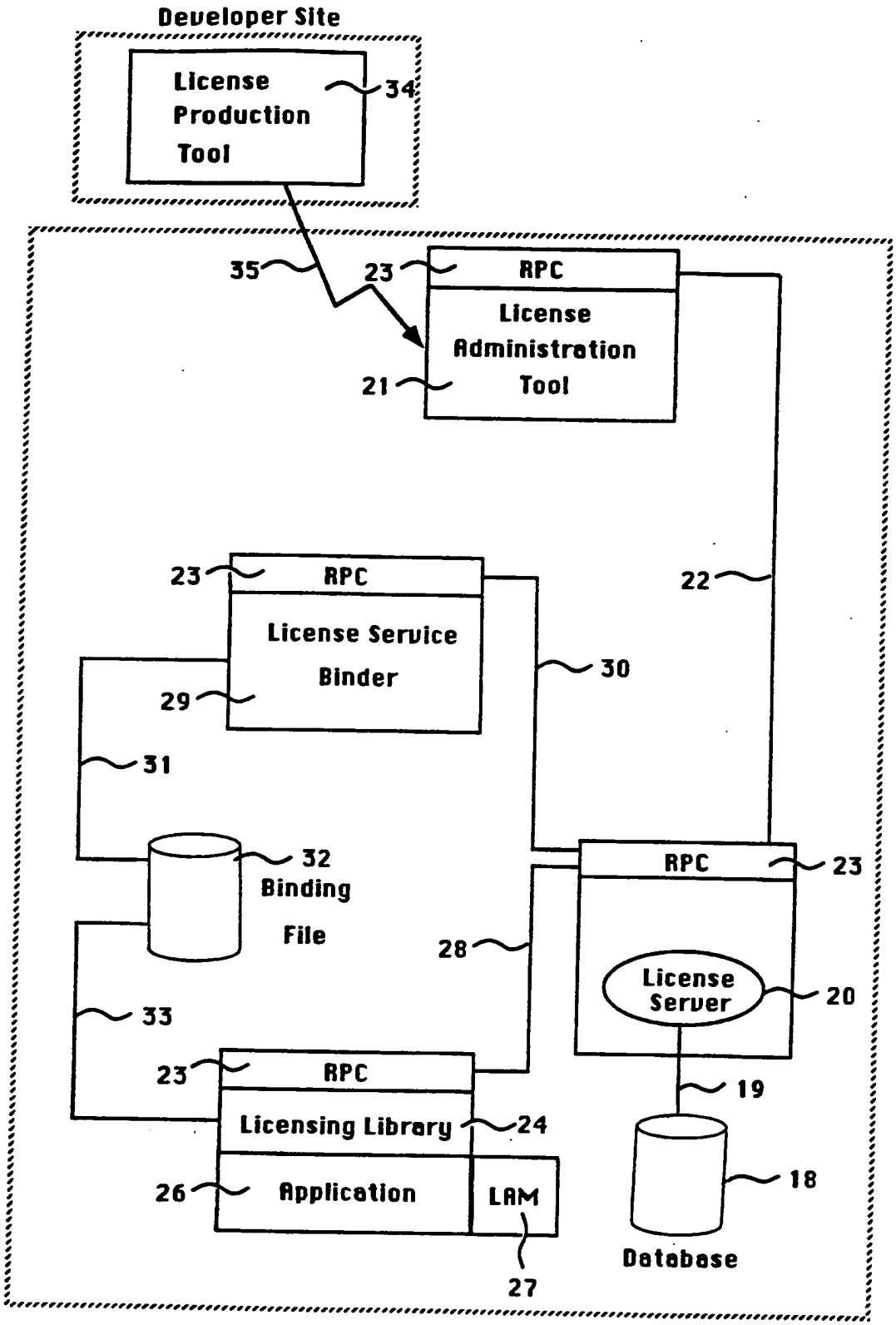

Developer Site
License Production Tool
34
35
23
RPC
License Administration Tool
21
22
23
RPC
License Service Binder
29
30
31
32
Binding File
RPC
23
28
License Server
20
33
23
RPC
19
Licensing Library
24
26
Application
LAM
27
18
Database
Customer Site

FIG. 2

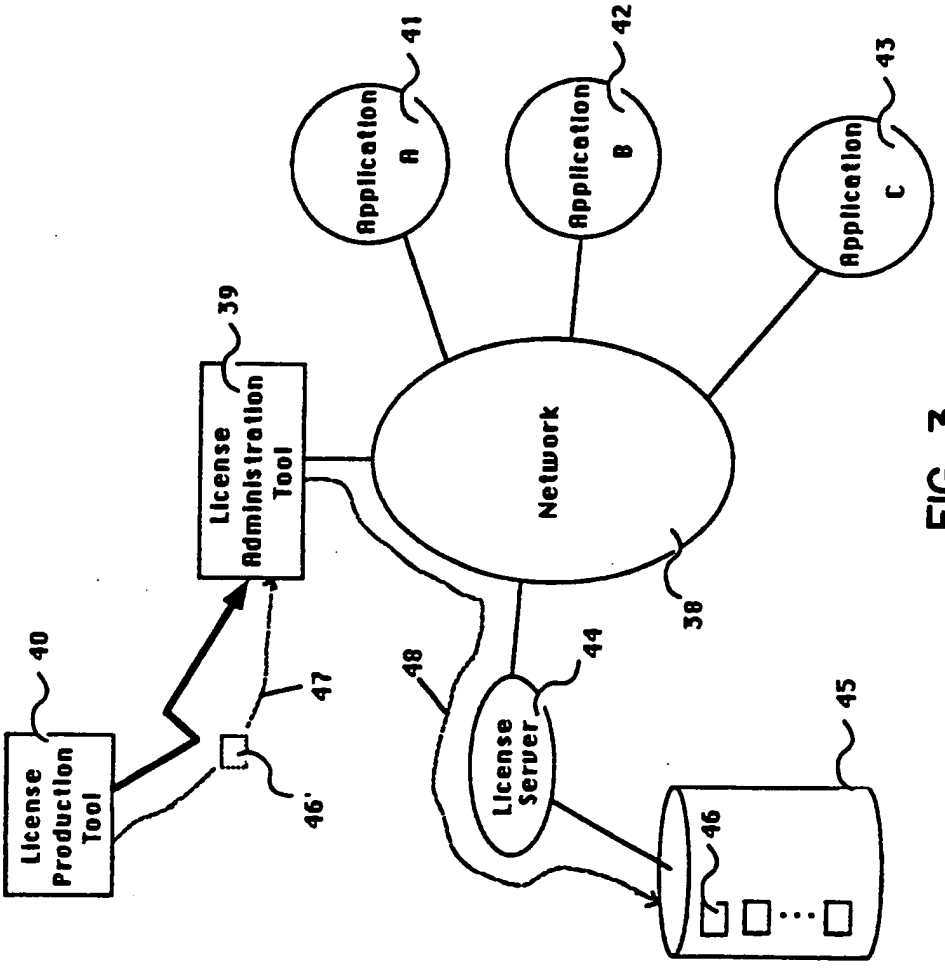

Application A
41
Application B
42
Application C
43
39
License Administration Tool
Network
38
40
License Production Tool
47
46'
48
License Server
44
45
46

FIG. 3

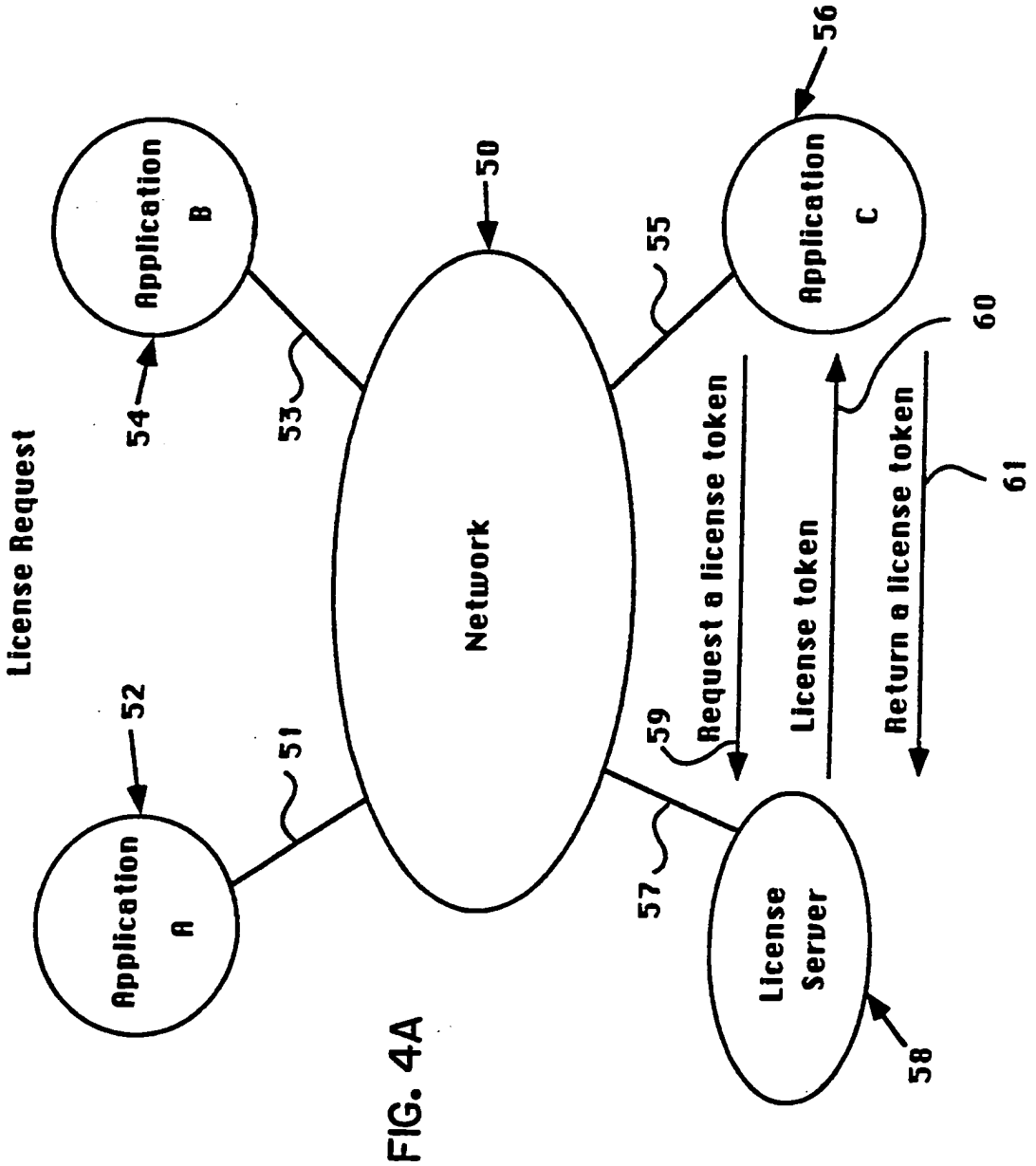

License Request
Application A
52
51
Network
50
Application B
54
53
Application C
56
55
License Server
58
57
59
Request a license token
License token
Return a license token
60
61

FIG. 4A

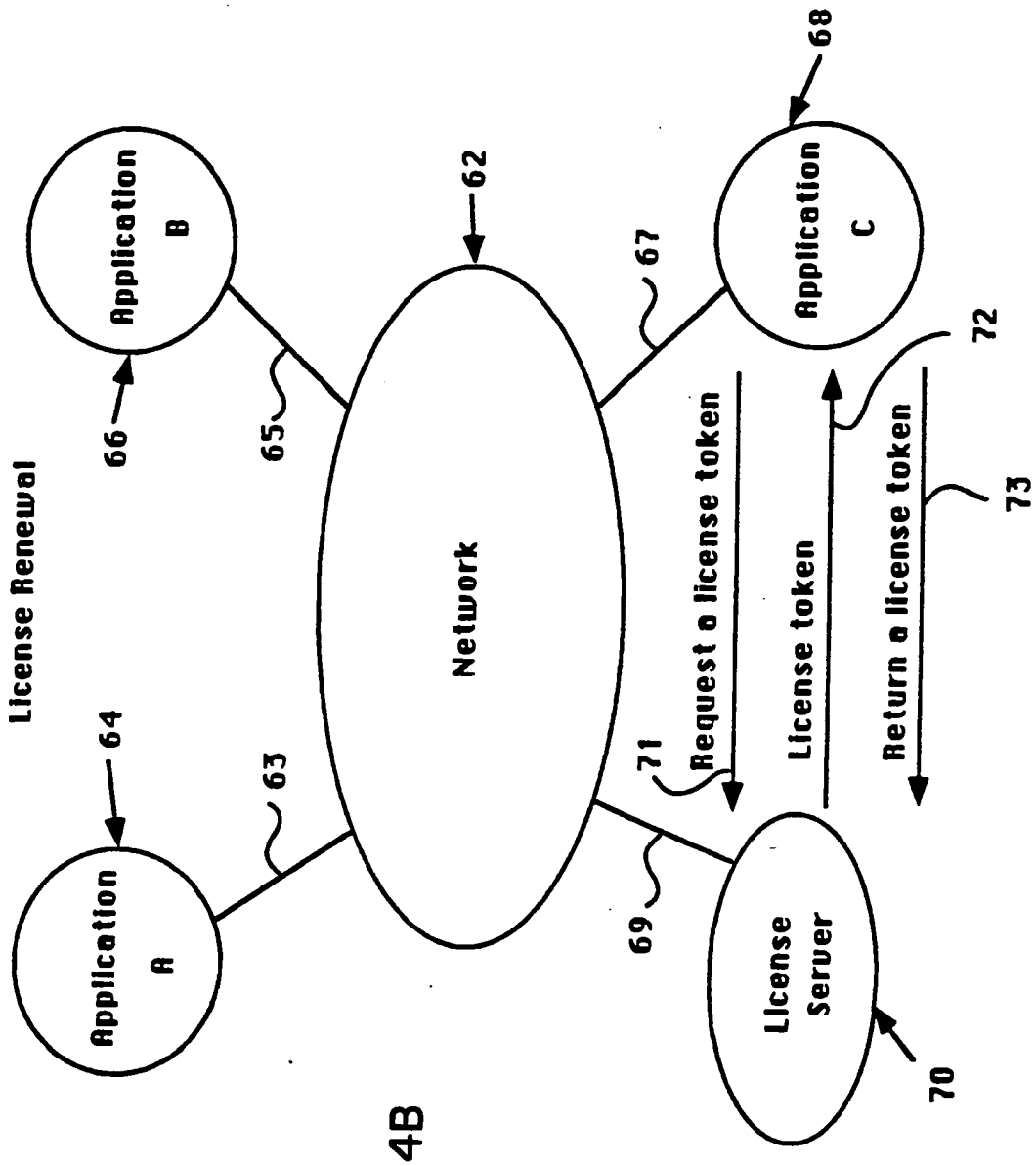

License Renewal
Application A
Application B
Application C
Network
License Server
Request a license token
License token
Return a license token
62
63
64
65
66
67
68
69
70
71
72
73

FIG. 4B

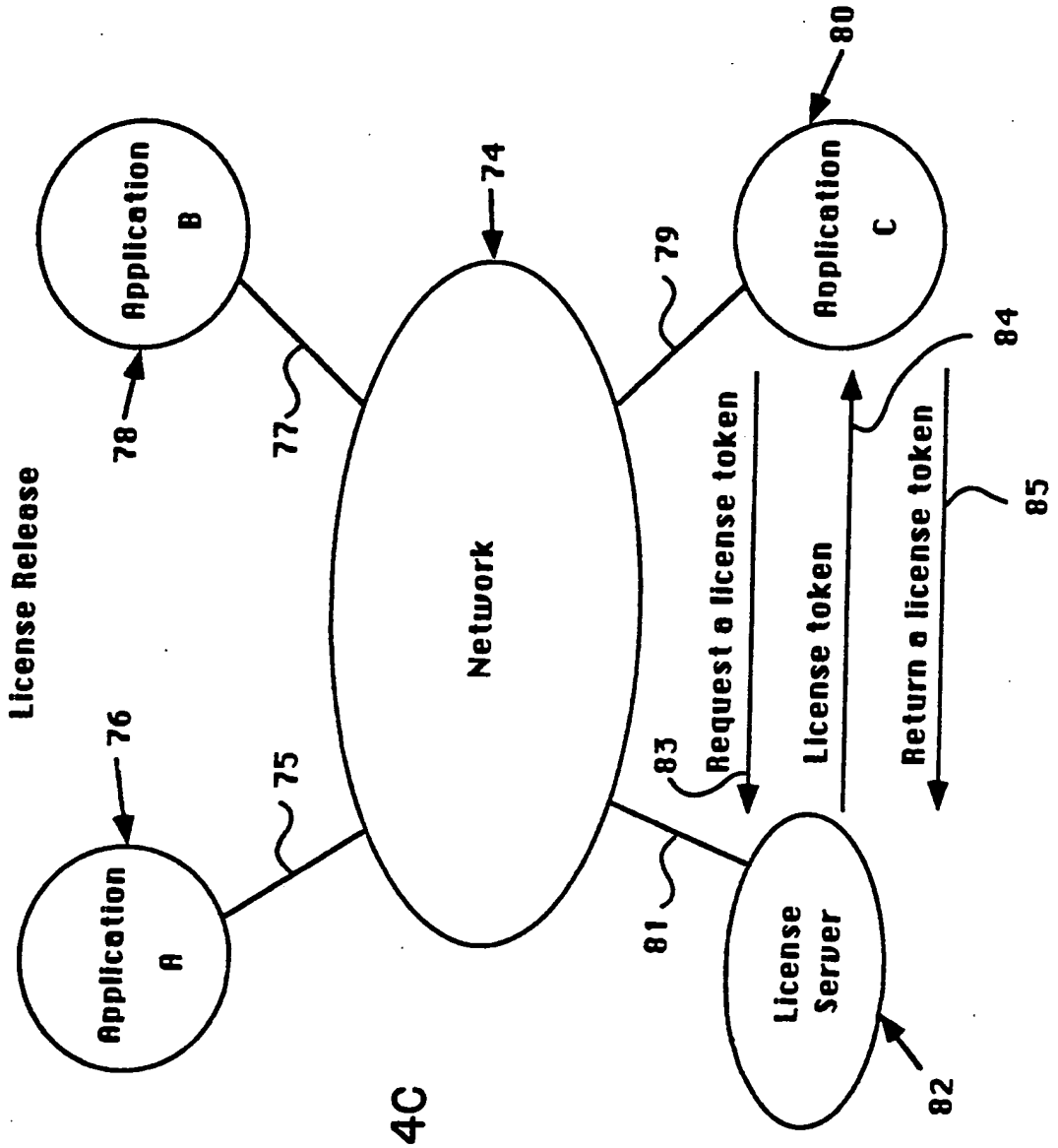

Application A
Application B
Application C
Network
License Server
License Release
Request a license token
License token
Return a license token
74
75
76
77
78
79
80
81
82
83
84
85

FIG. 4C

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 licensé 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) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 August 2001 (30.08.2001)

PCT

(10) International Publication Number
WO 01/63528 A1

(51) International Patent Classification[7]: G06F 17/60

(21) International Application Number: PCT/US01/05706

(22) International Filing Date: 22 February 2001 (22.02.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/511,537 23 February 2000 (23.02.2000) US

(71) Applicant *(for all designated States except US)*: **IPDN CORPORATION** [US/US]; 104 E. Main Street, DuQuoin, IL 62832 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **SAIGH, Michael, M.** [US/US]; 535 East Main Street, DuQuoin, IL 62832 (US). **BARRETTE, Pierre, Philip** [US/US]; 662 Lake Shore Drive, Murphysboro, IL 62966 (US).

(74) Agents: **BEULICK, John, S.** et al.; Armstrong Teasdale LLP, Suite 2600, One Metropolitan Square, St. Louis, MO 63102 (US).

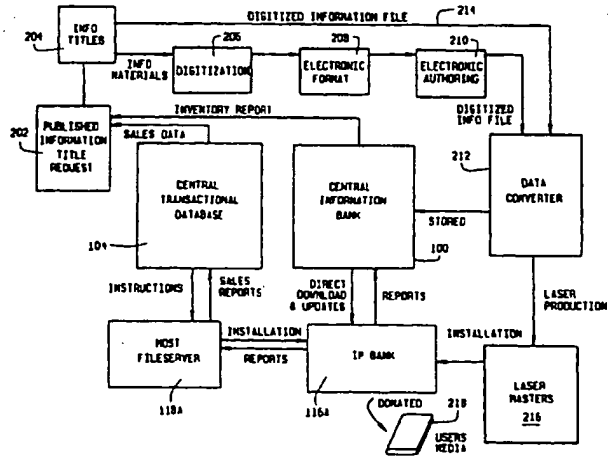
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, 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, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

[Continued on next page]

(54) Title: METHODS AND DEVICES FOR STORING, DISTRIBUTING, AND ACCESSING INTELLECTUAL PROPERTY IN DIGITAL FORM



(57) Abstract: In one embodiment, the present invention is an apparatus for facilitating obtaining text of an IP, the apparatus including a storage device (116A) having stored therein text of a plurality of IP, wherein the text includes electronically stored representations of printed text works, movies, films, video presentations, television programming, music, audio works, audio presentations, radio programs, graphic material, art works, plays, operas, novels, writings, photographs, pictures, images (including electronic images of all types including virtual images), advertising copy, or software, or portions and combinations thereof; a processor (118A) connected to the storage device (116A), the storage device (116A) further having stored therein a program for controlling the processor to: receive an IP selection request; receive a user identification associated with the IP selection request; and output encrypted text of the selected IP if the user identification and IP selection are valid utilizing a determined level of IP encryption.

WO 01/63528 A1

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.

METHODS AND DEVICES FOR STORING, DISTRIBUTING, AND ACCESSING INTELLECTUAL PROPERTY IN DIGITAL FORM

CROSS REFERENCE TO RELATED APPLICATIONS

This application is a continuation-in-part of U.S. patent application serial no. 09/175,559, filed October 20, 1998, which is a continuation-in-part of U.S. patent application serial no. 09/049,321, filed March 27, 1998, which is a continuation of U.S. patent application serial no. 08/687,292, filed July 25, 1996 (now U.S. Patent No. 5,734,823, issued March 31, 1998), which is a continuation of U.S. patent application serial no. 08/367,056, filed December 30, 1994 (now abandoned), which is a continuation-in-part of U.S. patent application serial no. 08/296,120, filed August 25, 1994 (now abandoned), which is a continuation of U.S. patent application serial no. 07/787,536, filed November 4, 1991 (now abandoned), all of which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

The present invention relates to methods and apparatus for the secure consolidated electronic storage and distribution of Intellectual Property in digital form and more particularly to methods and apparatus for electronically storing and transmitting information in consolidated, digital form to and from centralized storage facilities and users.

As used herein, "Intellectual Property" ("IP") refers to that, which in non-electronic form, would be referred to as books, text, pictures, photographs, videos, movies, films, audio, music, software, computer games, video games, and other types of expressions of ideas or concepts that can be stored and distributed in digital form. IP also includes all material that is capable of being protected by copyright, as well as other materials. In some embodiments, IP's include

compilations and/or portions of one or more of the above types of works as well as the works themselves.

Current networks for dissemination of IP lack a uniform distribution system. There are, for example, several different encryption technologies providing digital security (i.e., prevention of piracy or illegal copying of IP) for different media such as books, video, software, and multi-media. Similarly, there are different network distribution systems competing with each other to provide IP with little or no compatibility between the systems as seen from the standpoint of users, publishers (i.e., content providers), or the IP itself. Each entity has its own web-site, data center, and encryption algorithm. Uniformity is lacking because many different publishers, studios, and intellectual property owners have their own information retrieval service. Present methods and systems available for electronically distributing IP are not believed to provide a sufficient level of security against unauthorized use and copying to be acceptable vehicles for the commercial distribution of IP. This fragmentation produces inefficiencies in the distribution of intellectual property. There is currently no operator of an overall distribution network capable of providing publishers assurance of IP protection and easy-to-use, efficient, inexpensive, and universal dissemination of IP. There are currently no standards available that establish a criteria for the development and operation of the infrastructure, networking and systems for the commercial electronic delivery of IP with a level of security that would be desirable to most commercial providers of IP.

Traditionally, each type of IP has had its own unique way of being provided to the consumer. Intellectual Property (hereinafter referred to as IP) have been printed on paper and in books, for example, and music has been "burned" onto CDs. Retrieval methods for obtaining IP via digital readers, MP3 music hardware players and other hardware are currently specialized for the particular forms of IP for which the reader, player or other hardware is adapted. This form of specialization further fragments the IP market into specialized content data centers and reduces efficiency in distribution while failing to provide a level of security against

redistribution and copying that would be desirable to most commercial providers of IP. In addition, a proliferation of web-sites and market access locations make it more difficult and confusing for a consumer to access the available choices in a meaningful manner. Furthermore, a lack of uniformity in methods for identifying particular digitally-formatted IP makes it more difficult to track the flow of use of IP, because multiple tracking methods must be employed to accommodate multiple methods of identification. However, information provided in each case can be reduced, ultimately, to binary information. It would therefore be desirable to provide a distribution network designed for distributing various different IP types and a distribution method that can securely deliver each of these various IP types to the consumer, while safeguarding the transmitted IP types from illegal reproduction.

Present methods for distributing IP rely upon straight-forward transfer of data without inherent security built into the transferred data files. These methods provided adequate protection before the proliferation of public networks and the Internet. However, if one person were to "crack" a security code, any or all of the IP distributed in a file would be available for unencrypted transfer for all to have and use without regard for licensing and ownership. It would therefore be desirable to provide methods and devices for securing transferred data files that avoid making the IP available for unencrypted transfer after a code is cracked. Further, it would be beneficial to provide a distribution system and mechanism that regularly and automatically updates and/or changes encryption and decryption algorithms, keys, and/or formulae applicable to each IP electronically distributed to recapture security features for any IP for which previous encryption and decryption algorithms, keys, and/or formulae have been broken or compromised, to thereby inhibit cracking or compromising of IP encryption and decryption mechanisms.

Furthermore, the present proliferation of web sites and the Internet increases the difficulty of the user in finding and accessing specific IP's. There is currently a lack of uniformity in methods of identifying particular IP in digital form. This lack of uniformity makes it more difficult to track the flow and use of IP because

multiple tracking methods are required to accommodate multiple methods of identification. Lack of uniformity in electronic storage and distribution of IP also make distribution and delivery of IP more expensive and less efficient. Moreover, users of IP are inconvenienced because they must acquire multiple devices or software programs to read many different IP data formats.

It would therefore be desirable to provide methods and apparatus for consolidating all forms of digitally distributable IP.

It would further be desirable to provide methods and apparatus that provide a consistent interface between IP organizations and content owners on the one hand, and consumers of IP on the other.

BRIEF SUMMARY OF THE INVENTION

In one embodiment, the present invention is an apparatus for facilitating obtaining text of an IP, the apparatus including a storage device having stored therein text of a plurality of IP, wherein the text includes electronically stored representations of printed text works, movies, films, video presentations, television programming, music, audio works, audio presentations, radio programs, graphic material, art works, plays, operas, novels, writings, photographs, pictures, images, advertising copy, computer games, video games, or software, or portions and combinations thereof; a processor connected to the storage device, the storage device further having stored therein a program for controlling the processor to: receive an IP selection request; receive a user identification associated with the IP selection request; and output encrypted text of the selected IP if the user identification and IP selection are valid utilizing a determined level of IP encryption.

As is explained below, the terms "text" and "IP" as used herein are to be interpreted broadly. For example, a "IP" that is a musical recording includes "text" that is audio, or more specifically, music. Other types of "IP" may include more than one type of "text."

This embodiment of the present invention consolidates various forms of digitally distributable IP and provides a consistent interface between IP organizations and content owners on the one hand, and consumers of IP on the other.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates one embodiment of the present information distribution system architecture;

Fig. 2 illustrates information flow in the system architecture shown in Fig. 1;

Fig. 3 is a block diagram illustration of a point of purchase delivery system;

Fig. 4 is a more detailed block diagram illustration of the host fileserver shown in Fig. 3;

Fig. 5 is a perspective view of an IP Bank embodiment;

Fig. 5A is an alternative IP Bank embodiment;

Fig. 5B is yet another alternative IP Bank embodiment;

Fig. 6 is a block diagram illustration of the IP Bank circuitry;

Fig. 7 is a block diagram illustration of an end user's storage medium;

Fig. 8 illustrates the information flow for the point of purchase configuration;

Fig. 9 is a block diagram illustration of certain elements of a point of rental delivery system;

Fig. 10 is a block diagram illustration of certain elements of an IP Bank subsystem;

Fig. 11 is a block diagram illustration of certain elements of a promotional delivery system; and

Fig. 12 is a flow chart illustrating the encryption process implemented in accordance with the present invention; and

Fig. 13 is another IP Bank embodiment.

Fig. 14 is a representation of a network showing, in one embodiment of the present invention, how IP from content suppliers is provided through a central data center to a user device.

Fig. 15 is a more detailed block diagram of the user device shown in Fig. 14.

Fig. 16 is a block diagram of the user device shown in Fig. 15 to which an external authentication device is attached.

DETAILED DESCRIPTION OF THE INVENTION

As used herein, the term "network" refers to any electronic interconnection between two or more electronic devices over which data is transferred, including, but not limited to, the Internet, an intranet, a land line or traditional telephone network, a cellular or wireless mobile network, a wireless microwave network, television or radio wave transmissions, a cable network, a wireless connection (for example, infrared or microwave connections), satellite, a localized land network system, induction connection using electric lines, a wireless network using lasers as the transmitting medium, any combination of any of the preceding or any other system for the transmission of data between two or more units. A "secure network" is a network employing security measures against unauthorized access to data being transmitted via the network or data stored within a memory storage area of a device connected to the secure network.

The term "IP Bank," as used herein, refers to an interface between a network and a user. Such interface may be physically located in proximity to a central information storage facility, at a location remote from a central information storage facility (for example, in a physical housing such as a kiosk located at a retail establishment), or in proximity to and directly connected to a user's computer. In one embodiment of the invention, the IP Bank is wholly or partially a virtual device generated from an interaction a software program stored on a user's computer and software located at another IP Bank or a central information storage facility. In embodiments involving a virtual device, some of the hardware necessary for operation of an IP Bank is located proximate to or within a user's computer, or proximate to or within the IP Bank or the central information storage facility, depending upon a location at which user contact is made. For example, in one embodiment, a "IP Bank" comprises memory storage, a processing unit, a keyboard, slots for credit cards, slots for storage media on which downloaded digital data such as electronic versions of printed IP texts are stored, and a printer (for bills and receipts). A virtual "IP Bank" comprises, for example, only a memory and a processor (located at, e.g., a retail establishment, a data storage center, or at a security encryption compression module. Keyboards, slots for credit cards and storage media, etc. are supplied in this embodiment by the user's device and the user's software. Although the term "IP Bank" and "text" may suggest book-type or written material based upon a reader's prior notions of the subject matter to which these terms refer, the terms are not intended to be so limited in this description. Other types of material, such as movies, music, voice, video, graphic images, natural language and other text, audio, computer games, video games computer software material, and any other type of intellectual property capable of being converted to and stored in digital form is intended to be included when the terms "IP" and "text" are used, unless otherwise specified or made clearly obvious from context. (For example, "natural language text" does not refer to video or music.) An IP Bank is a self-service, user interactive information vending device, which, in one embodiment, is a separate, stand-alone device. In another embodiment, an IP bank includes hardware and software located at a central data

storage facility and hardware and software located within or proximate to a user's computer, with both being in electronic communication via a suitable network, thereby providing operation coordinated in a manner such that the components function in the same manner as a stand-alone IP bank. In yet another embodiment, an IP bank is a device located proximate to a user's computer or a computer board located within the user's computer and which is electronically connected to the user's computer via a suitable network connection, and which is also connectable to a central information storage facility using a suitable network. In another embodiment of the present invention, for example, each IP bank contains a high capacity, local memory storage having a customized portfolio of the most demanded information products for a particular site at which the IP Bank is located. In another embodiment, each IP Bank comprises a processor, a monitor, a network connection, and one or more slots configured for insertion of a portable memory storage medium and/or a connection to a user device, for example, a portable digital assistant (PDA). Other information is transferred via a network to an IP Bank for supplemental, secondary, and less demanded purposes. A processing unit within the IP Bank and coupled to the IP Bank local memory and storage controls downloading and dynamic encryption of information.

The term "IP," as used herein, includes all different types of intellectual property that is capable of being electronically stored in digital form. The term "IP" includes traditional printed text works, movies, films, video presentations, television programming, music, audio works or presentations, radio programming, graphic material, art works, plays, operas, novels, writings, photographs, pictures, images, advertising copy, computer games, video games, computer software, any portion of combination of the above, and/or other types of intellectual property.

The term "IP file," as used herein, refers to a digital version of an IP. The description is not dependent upon the format of the stored digitized material, and is equally applicable whether the digitized material is stored using HTML, XML, Adobe PDF (portable document format), SHOCKWAVE® format (Macromedia, Inc.,

San Francisco, CA), or any other format. Where the context so indicates, the term "IP selection request" refers to a request made by a user to obtain a copy of a given IP file. The term "selected IP," as used herein, refers to a particular IP file for which a user has made a request for a copy. The term "text," as used herein, when used in connection with a selected IP or IP file, refers to the content of the digitally stored IP file under consideration. Because the term "IP" is used herein includes not only traditional printed text works, but also other types of intellectual property capable of being electronically stored in digital form, the term "text" as used herein is intended to be interpreted with equal generality so as to include the content of the "IP," whether it be traditional printed text or the content of another type of IP to which the general term "IP" is applied herein. The general applicability of various of the embodiments of the invention disclosed herein will be appreciated if it is understood, for example, that a "IP" that is a television program includes "text" that is television programming. As another example, a "IP" that is a musical recording includes "text" that is audio, or more specifically, music. Extending the analogy further, it is easily understood that one "reads" the "text" of the musical recording "IP" by playing the music via a system that converts the digitally encoded audio into sound. Some types of "IP's" may include more than one type of "text," for example, digital representations of printed text, music, and pictures. All uses herein of the terms "IP" and "text" are intended to be generalized in a similar manner, unless it is explicitly stated that such use is intended to refer only to printed text or natural language text, or the context is such that the term "text" is obviously limited thereto.

The term "user's computer" or "user's device," as used herein, refers to any electronic device performing some or all of the functions traditionally associated with a typical desktop computer, including, but not limited to, a traditional desktop computer (such as an IBM PC, a MacIntosh® (Apple Computer, Inc., Cupertino, CA) or a clone of either), a laptop computer, PDA device, wireless connecting device, Internet connecting device (e.g., WEBTV™ (WebTV Networks, Inc., Palo Alto, CA) or others), digital telephone, video gaming device, ebook, or another electronic device capable of being electronically connected with a network.

The terms "user's computer" and "user's device" do not necessarily refer to different types of units.

As used herein, the term "nonvolatile memory" refers generally to a type of memory that does not depend upon power being continuously applied to retain information. For example, static RAM, segregated hard drive, floppy disk, CD-RW, magnetic optical disk, flash memory, electronically erasable programmable read only memory (EEPROM), and WORM (write once, read many time) memories are types of nonvolatile memory. ROM (read only memory) and field programmable gate arrays are also types of nonvolatile memory that are sometimes mask-programmed. A circuit designer would be able to choose a suitable type of nonvolatile memory for use in a circuit, giving due consideration to voltage, current, data access rate, data transfer rate, physical size of the device desired, and memory density requirements, and to whether the application requires the nonvolatile memory to be field programmable, erasable, and/or reprogrammable.

Also as used herein, the term "encryption algorithm" unless otherwise stated, includes methods for encryption, keys used for encryption, and formulae used for encryption. The term "decryption algorithm," unless otherwise stated, includes methods for decryption, keys used for decryption, and formulae used for decryption. However, keys and/or formulae are sometimes explicitly mentioned along with the algorithms in the explanation to add emphasis.

In one embodiment, a user may select portions or entire contents of one or more IPs. The selected information is then combined and downloaded to the user's storage device, for example, a "cartridge." A "cartridge," "memory module," or "storage medium," as used herein, refers to any device capable of electronic storage of digitized data and which is capable of providing access thereto in a controlled manner. The terms "cartridge," "memory module," and "storage medium" are interchangeable, unless otherwise indicated by context. The terms encompass, for example, devices such as appropriately configured compact disks (CDs), DVDs

(including DVD-R, DVD-RAM and DVD-RW), flash memory cards, and removable storage devices (including floppy disks, memory chips, ZIP™ disks, and other units), where "appropriately configured" means, for example, having instructions or data recorded thereon to control access by an access device to data recorded thereon. Also included are protected segregated portions of a data storage portion of a hard disk, and any other device for the storage of data electronically in digital form and providing controlled access thereto. In one embodiment, a cartridge includes a unique identification number and a predetermined amount of memory for storing the selected information. In another embodiment, a cartridge also includes specialized software (cartridge controlling software, or "CCS") stored in nonvolatile memory, either on a separate memory chip of the cartridge, or in a segregated protected space within a general memory storage space of the cartridge. The CCS is electronically linked to the cartridge on which it is stored, and controls and regulates reading and use of encrypted and tagged data information files stored within the general memory of the cartridge. As long as any portion of encrypted data stored on the cartridge is being accessed, CCS remains functioning and regulates and controls functions that the user is permitted to perform with regard to encrypted data in question. For example, in most cases, when an encrypted data file is accessed for reading, CCS operates to restrict the ability of the user to copy the data in a decrypted form or to print data onto hard copy. CCS also restricts the user from copying any portion of the decrypted data to any data file other than a temporary file in RAM that is automatically erased when CCS stops operating as an encrypted file that is only readable using CCS operating on authorized equipment. When the user ceases accessing encrypted data files, and CCS ceases operation, just prior to closing down, CCS totally erases all record of any decrypted data and all temporary files in which such data may have been stored. Once CCS has ceased operation, the user is no longer able to access any of the temporary files generated during previous operation, as all record of such files ever existing and the content contained therein will have been permanently excised from the user's system. Thus, in connection with related application software stored within the user's device, CCS utilizes one or more of the dynamic encryption and decryption

features, a unique serial or registration number, and various data registration headers to regulate and govern any use the user makes of encrypted files and the data stored within. Without CCS operating, a user is unable to gain access to the encrypted data. In one embodiment of the invention, CCS contains monitoring features that prevent operation of CSS should any attempt be made to alter its operation. In one embodiment, when CCS becomes inoperable for any reason, the user of the device has to bring the affected cartridge to an authorized agent for repair or replacement.

In one exemplary embodiment of CCS, CCS finds the information necessary to decrypt an encrypted text. While the user device in which CCS resides is operating, CCS restricts what the user device is capable of doing. For example, it restricts the user device so that it cannot write information to an external storage device while an internal storage device having an encrypted text is being used. When a cartridge is disengaged from the user device by removal, CCS shuts downs the user device, for example, either by shutting down its program or by operating an electronic switch to remove power from the user device. CCS also removes all temporary files related to the program from the user device. In addition, CCS reads a secure real-time clock in the user device to determine whether the user device is presently authorized to access data on a cartridge. CCS also registers when tampering occurs, such as a change in the real-time clock data caused by an obvious backdating attempt.

One embodiment of the invention enables a user to obtain updates to any data information file he or she has acquired, by utilizing a dial-up network to dial into an IP Bank or a central data storage facility, or by using the Internet to access an IP Bank or a central data storage facility through an appropriate link via a web site, or the use of a wireless network (e.g., digital satellite, cellular, wireless mobile, microwave, infrared, etc.) to gain access to an IP Bank or the central data storage facility over any network or connection. In one embodiment, a Secure Universal Resource Locator (SURL) is used, including both a secure phone number as well as an Internet-based URL. The same restrictions apply to obtaining an update as to acquire the data information file being updated.

The following sections provide a brief overview of one embodiment and a detailed description of its architecture. Following the detailed architecture description is a detailed description of point of sale delivery embodiment configurations. A detailed description of the various levels of encryption which may be used in the present system is then provided.

A. Brief Overview

In accordance with one embodiment of the present invention, information is distributed from a central information bank to a user's personalized storage medium. Information to be so distributed by the present system is received from outside sources either electronically, over various communication networks (e.g., telephone lines, cable systems, cellular systems, wireless mobile systems or other similar commercial communication networks) or from various storage mediums (e.g., magnetic or electronic disks, cartridges, or tape reels or compact disks, laser disks, tape cassettes, etc.), or in hard copy format. If information is received in a hard copy format, it is initially converted to a standard digital format (e.g., ASCII text, DOS text or other similar standard commercially available text format) by scanning or direct transcription. If information is received in a videotape, NTSC or PAL format video, then the information is digitally encoded in a ".avi" file, a Quicktime file, or other format, including the application of appropriate MPEG-X compression as needed. A content provider, i.e., an outside source, providing information can specify instructions relating to authorized use, access, and cost of the provided information that can be permanently linked to a master data file of the information when it is electronically stored or accessed. In one embodiment, instructions are electronically generated when information is received electronically. These instructions include a description of allowed uses (e.g., copy generation, printing limits and/or authorization, rental options, if any, purchase options, etc.), desired level of security against unauthorized copying or use (from level zero [virtually no security] to a maximum level [most robust security available]) and cost to a user, optionally as a function of user-selected access right. Whenever a request is made to obtain a copy of

the information file (i.e., IP File), the instructions are used in connection with the choices made by the user, within the allowed options, to generate user-specific electronic instructions for limiting use of the requested copy of the IP File. Then the information is digitized, formatted, compressed and initially encrypted to form an electronic master copy which is stored in the central information bank. The master copy is duplicated electronically and dispatched electronically through a communication network, such as a telephone or satellite network, to a point-of-sale delivery system. IP Banks form a part of such a delivery system, and the electronic copies are retained in the IP Banks for downloading into a user's personalized storage medium. Initially, a user selects the information to be downloaded and a tracking entry is made into a transactional database to record the transfer. Prior to and during downloading of the copy on the user's storage medium, the information is dynamically encrypted utilizing a varying level of encryption which is dependent upon a variety of variables, for example, an economic value of the information. A "dynamic" encryption process is utilized so that only the electronic reader associated with the user card used to access the information from the IP Bank and download the information to the user storage cartridge can be utilized to display the information in an understandable text format. In one embodiment, the "dynamic" encryption process creates, encrypts, and transmits data files, in real time, to one or more different user devices or storage media located in one or more different physical locations using a network. For example, in one embodiment, an instructor's lecture notes are electronically encrypted and transmitted in a secure manner to devices and/or storage media of students attending a lecture in real time. The notes are stored on these devices and/or storage media for current or later use.

As explained in greater detail below, in one embodiment, the dynamic encryption process uses a unique serial number (or other identification, for example, an alphanumeric or binary identification) associated with the particular user's storage medium to which the encrypted data is to be downloaded, the unique registration number given the user on registration, and the unique file number associated with the file to be downloaded, to secretly generate the encryption and decryption algorithms

and formulae to uniquely encrypt the copy of the data file as it is downloaded to the user's storage medium and to allow the user's system to decrypt the data for later reading. A copy of the algorithms and formulae are secretly stored at the central data storage facility in a portion of the memory storage having limited access. A copy of the algorithms and formula necessary to decrypt each of the acquired encrypted data files is stored in read-only form on the user's storage medium. Storage space for this read-only copy is designed so that any attempt to access the data stored thereon other than using an unaltered CSS program renders the data stored thereon unreadable. In another embodiment, instead of utilizing a unique registration number permanently tied to the storage medium being used for developing the unique encryption and decryption algorithms and formulae, a unique electronically-generated electronic number is assigned to the storage medium at the time of initial registration with the system, and that assigned number is used in developing the applicable encryption and decryption formulae and algorithms. In this embodiment, the unique electronically-generated number is a number that is randomly generated by the system and checked for uniqueness. In another embodiment, the unique electronically-generated number is generated by a system program that searches features of the user data storage medium to find a unique representation of the device, from which a unique electronic registration number is developed. In other embodiments, other methods, including combinations of these methods, are used to generate unique numbers.

In another embodiment, software programming periodically changes the applicable encryption and decryption algorithms and formulae associated with a particular encrypted IP file to provide enhanced security. In this embodiment, the changes in the applicable encryption and decryption algorithms and formulae occur when a user logs on and connects with the system through an IP Bank connected to the central information storage facility. The changes occur after expiration of a predetermined period of time from a date on which the particular encrypted IP file is initially created, or after expiration of a predetermined period of time from a date on which the applicable encryption and decryption algorithms and formulae are last changed, whichever last occurs. The periodic changing of applicable encryption and

decryption algorithms and formulae makes it more difficult for the encryption to be broken and allows recapturing of IP files into the encryption system if previous encryption has been broker. The frequency of such encryption changes is dependent upon a time value of the IP involved and a length of time that a user of the IP file in question has been authorized to access the particular encrypted IP file.

In one embodiment, security is enhanced through the use of an external authorization device affixed or linked to the reader device and a related identifier. Examples of suitable, currently-available authentication devices include parallel port software locks, iButtons™ (Dallas Semiconductor Corp., Dallas, TX), and "Smart Cards" plugged in via a pc-card.

In another embodiment of the invention, when one or more IP Banks are operated in conjunction with one or more retail establishments, each retail establishment is given a unique identification number that is used to tag each user device, user storage medium and/or encrypted IP file acquired at the retail establishment or through an IP Bank linked to the retail establishment. Using the tag information, particular user devices, storage media and/or IP files, a predetermined portion of the system revenue generated in regard to such tagged user devices, storage media and/or IP files is allocated to and shared with the retail establishment in question. In another embodiment, the tagging is further delineated to allow revenue sharing with manufacturers and distributors of user devices and/or storage media.

IP digital data transmission can occur between two or more than two mobile wireless or cellular devices. The devices can be multipurpose, for example, they may have a combination of PDA, cell phone, visual display, and/or audio capabilities. The devices transmit IP in a secure manner whether the IP is digital audio, video, text, software, or multimedia. The mobile device can also asynchronously obtain, through a wireless access portal (WAP), music, video, text, and other IP through various databases. For example, if a user wishes to access his or her favorite songs, the mobile wireless browser can be used to browse the music

database, access a music title by artist, CD title, or by some other fashion and customize the desired transfer of music IP. An actual order and payment then takes place after the user requests whether they wish to rent (and thus, autoerase) or own a copy of a requested song. A similar mechanism is used, in one embodiment, for ordering natural language texts (e.g., an "ebook"), software games, video and movies. Asynchronous transfer refers to the downloading of IP from one or more than one database using the wireless mobile device. The actual transfer of data takes place and is stored in a customer's SEC (security encryption compression) device. A user could utilize the IP at his or her discretion as opposed to real-time usage with a data stream transfer.

In one embodiment, multiple databases and portals with transparent links can be used and accessed with the mobile wireless device. It is immaterial to the practice of this embodiment of the present invention whether access is through mobile wireless or cable, fiber optics, telephone access to an Internet portal or database. The same end-to-end security as described in this invention for IP is consistent with the invention. The same dynamic encryption and mechanical and/or technical means exists throughout the data transfer distribution cycle. After the transfer occurs, the user (i.e., customer) SEC unit contains data and rules linked to data governed by the "smart" storage device. The smart storage device determines the time of autoerasure, watermarking, level of security from the highest strata to a more open system and other rules directly linked to that data or piece of the IP data. The rules can apply as minutely as a note, word or bit, etc.

Synchronous data transfer refers to a continuous stream of IP data to the mobile cellular device. Here, the same rules for the IP exist in real time and the wireless connection must be maintained throughout its transfer. A combination of synchronous and asynchronous transfer could exist within the database and multipurpose consumer device. A single purpose device, for example, one providing mobile wireless access to an internet music data portal without other functional abilities is also within the scope of the present invention.

In one embodiment, the wireless device can also be linked to other devices synchronously or asynchronously. For example, data transferred on the mobile wireless device after downloading a music IP could be transferred to a car or home stereo system, when music (audio) is played. Various transfer mechanisms can be used for the transfer of music or other IP. One embodiment of the present invention includes the transfer of IP through radio frequency to a car stereo or a home stereo, and the transfer of movies and television programs through radio frequency transfers to television sets and/or computers. In the case of a mobile wireless device such as a car stereo, the mobile wireless device is equipped with radio circuitry that can interface with the car stereo's AM or FM frequency. Transfers to this circuitry can be either synchronous or asynchronous. In a similar manner, the mobile wireless device could access other forms of IP, such as video, movies, software, text, and multimedia, etc., and transfer the IP in an encrypted manner, digitally secured as stipulated by a content owner's rules regarding the IP.

Digital content security can be produced and distributed in real time "on the fly." Professors, musicians, authors, instructors, film producers and any individual or organization or corporation can secure "their" content from redistribution when producing the content in a live setting or to be released on the web or other method of business-to-business or business-to-consumer digital distribution. The scope and quantity of content can be a word, note, picture frame based on DOI (Digital Object Identifier) or a complete or part of an entire work of art or content. Course packs present a classic example of bits and pieces of information gathering. The scope could also mean "on the fly content."

In one embodiment, rules formulated by content providers are matched with technological capabilities. The rules applied toward a certain content will constantly adjust in accordance to market demands, functionality and use of content as well as with technological advancement. The rules relative to a DOI, text, audio, visual, or content form will be tied to the content in a secured manner.

Digital content rules will be assigned with digital security to each and all content in any form (text, music, video, software, multimedia.) Such rules will be established by content owners and providers and will not be alterable or capable of manipulation by digital content users.

Functionality is defined as how content is allowed to be used. For example, the content (i.e., "text") can be autoerased, allowed to be printed or transferred to another user (or not), watermarked, and a host of rules applied governing activity, use, and manipulation of the content. The content can include natural language text, music, video, software, or multimedia. The rules regarding digital content and security also includes the tracking, accounting and verification of transactions.

Consolidation of intellectual property for content providers can exist in one master database or several databases or subsystems. Transparent links to outside databases to reserve intellectual property are used in one embodiment of the present invention to provide a customer with greater depth while giving content providers a sense of database control and management. Security, uniformity and integration are maintained throughout the system's network distribution chain. Customers may not know the location from which content is ultimately accessed. This seamless transfer greatly enhances content-driven access and continuity.

Since intellectual property content is fragmented into thousands of providers, content continuity and access is important to help mediate content sourcing and availability. A smart browsing and retrieval system that will, in effect, assist a consumer in accessing information by title, subject, author, artist, studio, publisher, etc., is built into the platform to be standardized. Embodiments of the present invention develop systems integration and an active digital distribution database that apply security access rules if needed, and then redistribute the information to business users, household users, and/or organizational users.

Customization, business users, household consumers, etc., can, if content providers allow via rules applied to the content, customize their content. This customization of content can extend to any form of digital intellectual property and polarity. In one embodiment, customers or content providers and integrators can add music to text material, animation, software, pictures, video, or multimedia. Cumulative printing or release play - music, video, etc. - is determined by content owners percentage of total content. A certain percentage of a song could be taken out of security, for example.

Advertising and promotions can be used to supplement supplier's income from IP and/or reduce the cost to customers. Content owners will decide, as will customers, to what extent advertisement will be embedded or linked to content, and thus, how much income will be supplemented and/or cost reduced.

Because of content security and use restrictions, digital content returns are made possible.

In one embodiment, autoerasure is provided based on a secured, unalterable (by a user or customer) calendar clock. The clock can be reset when the customer retrieves information via wireless, internet, or other means. Autoerasure rules include the elimination of title (i.e., file) access when time is expired. A customer could then reactivate the affected title access for an additional period of time. If a customer does not choose to renew a rental IP, the content of the IP is totally removed from the system.

B. System Architecture

Figure 1 illustrates one embodiment of the present information distribution system. The system is shown, for illustration purposes only, as being implemented across the world. Referring to Figures 1 and 2, one of the facilities provided is one or more "central data centers" or "central information banks" 100. Each central data center 100 acts in conjunction with any others present in the

network to store and control delivery of IP. Additional data centers 100 each comprising one or more computers acting in conjunction with each other are provided. A central information bank 100 is a central "library," or storage location, for information. Peripheral information banks 102A-F, coupled to central information bank 100, are libraries, or storage locations, for community oriented information. For example, the information stored in central information bank 100 accessed most often from the San Francisco bay area peripheral information bank 102A may not be accessed often from the peripheral information bank for Rome, Italy 102E. In any event, central information bank 100 is coupled to each peripheral information bank 102A-F to enable sharing of information. As explained in more detail hereinafter with respect to peripheral information bank 102F, each peripheral information bank 102A-F is coupled to one or more point-of-sale sites.

A central transactional database 104 coupled to the central information bank 100 and the peripheral information banks 102A-F, serves a central record keeping function for central information bank 100 and peripheral information banks 102A-F. Central information bank 100 and central transactional database 104 preferably, are commercially available main frame computers, such as an IBM main frame computer. The particular main frame model selected depends on the amount of information to be centrally stored in the network, the extent of record keeping functions to be performed, and the speed at which transfer and processing of information is to occur. Importantly, the present invention is not limited to any one particular computer to serve as the central information bank and/or the central transactional database.

As shown in Figure 1 is an exploded view of the various couplings between central information bank 100 and transactional database 104, peripheral information bank 102F and various point-of-sale delivery sites, particularly, point of purchase sites 108A-C, point of rental sites 110A-D, promotional sites 112A-D, and IP Bank subsystem sites 114A-C. Each point of purchase site 108 includes a point of purchase transactional database, represented by a box, and a user interface,

represented by a circle. As explained above, the user interface is sometimes referred to herein as the "IP Bank." Specifically, point of purchase site 108A contains IP Bank 116A and transactional database 118A, site 108B contains IP Bank 116B and transactional database 118B, and site 108C contains IP Bank 116C and transactional database 118C. Since the central information bank 100 and peripheral information bank 102F, and specifically peripheral information bank memory storage unit 106A, also could serve as IP Banks, such units are illustrated as circles. Further details regarding IP Banks and transactional databases are provided below in Section C.

As illustrated in Figure 1, each point-of-sale delivery system, such as systems 112A, 108A-B, 110A, and 114A-B, may be networked directly to peripheral information bank 102F, or the point-of-sale delivery system, such as systems 108C, 110B-D, 112B, 112D and 114B-C, may be networked to the point of purchase site 108B, which is networked to the peripheral information bank 102F. Point-of-sale delivery system configurations are explained in more detail below in Section C. At the level illustrated in Figure 1, however, it is important to understand that the delivery systems may be integrated into various combinations, such as a promotional point of rental system as shown by 110B and 112B, or a promotional point of purchase system as shown by 108B and 112C or a combination of a promotional, point of purchase, and point of rental systems as shown by 108C, 110D and 112D.

Communication network links between the central information bank 100 central transactional database 104, peripheral information banks 102A-F, and point of sale sites can be made utilizing one or a combination of many commercials available networks such as telephone, satellite or cable networks or any other medium suitable for transmitting information in digitized format. Many well-known protocols could be used in connection with the present system. For example, if the Internet is used as the "backbone" network, the well-known TCP/IP protocol could be used.

Figure 2 illustrates the flow of information in accordance with the embodiment of the system architecture illustrated in Figure 1. For ease of illustration

only, peripheral memory storage unit 106A is consolidated into central information bank 100, and peripheral transactional database 106B is consolidated into central transactional database 104. It should be understood, of course, that communication links between the peripheral information bank 102F and central information bank 100 and central transactional database 104 are provided.

As illustrated by the inputs provided to block 202, a publisher will receive inventory reports from the central information bank 100 and sales data from central transactional database 104. Based on this and other information the publisher can determine whether to place additional information on the network. For ease of reference such information is sometimes referred to herein as "information titles" as shown in block 204. If the information is not present in an electronic format, then the information is digitized 206, disposed in an electronic format 208 and then undergoes electronic authoring 210. The digitized information is then transmitted to a data converter 212 for converting the digitized information into a uniform format. For example, if the central information bank 104 and central transactional database 104 are DOS- or UNIX-based systems, the data converter will convert the information into a DOS or UNIX format, as appropriate. If the information titles are in a digitized format, the information titles are transmitted directly to data converter 212 for direct conversion into the uniform format as illustrated by line 214.

Once the data is in a uniform, digitized format, it undergoes an initial encryption and compression to both reduce the amount of storage space required to store the data and to make the data ready for being transmitted with less risk of unauthorized use while being transmitted through a communications network. The compression is accomplished through the use of one of the commercially available compression protocols. The initial encryption is performed using one of the standard available encryption protocols as discussed below in Section D.

Once in uniform, encrypted and digitized form, the information titles are stored in central information bank 100. An electronic index listing all titles

available and accessible by author, title, subject or ISBN codes and/or ISSN codes is prepared. As new information titles are added, the electronic index is updated to include the new titles. The information titles may then be downloaded to IP Bank 116A. The information titles and corresponding electronic index information may, in addition to or rather than being stored in central information bank 100, be disposed on laser disk masters as illustrated at block 216. Laser disk masters 216 can then be installed directly into IP Bank 116A.

In one embodiment, as desired information data titles are being downloaded from IP Bank 116A to a user's storage medium 218, each file is being dynamically encrypted and encoded with authorization and user identification data to restrict access and user of the information being transferred. The encryption process uses algorithms and formulae generated using a combination of the unique serial number of the storage medium to which the encrypted file is to be downloaded, the unique registration number given the user requesting the downloading at the time of registration and the unique hidden number given the file, a copy of which is to be downloaded. In this manner, the data being downloaded is tied to the user requesting the downloading and the storage medium on which the encrypted data is to be stored. The unique serial number of the storage medium may be a number that is permanently associated with and stored upon the storage medium at the time of manufacture or initial formatting or may be a number that is electronically generated and assigned when the storage medium is initially registered with the system. In both cases the number will be electronically readable and stored on a nonvolatile portion of the memory storage space associated with the user storage medium. In one embodiment, a current time and date are included with the data titles downloaded from IP Bank 116A so that a check is made of a clock associated with user storage medium 218 or an associated user device to ensure that no backdating of the clock in end user's device has taken place.

Prior to downloading desired information titles, the user may access an electronic index which contains all the information titles available for downloading

from IP Bank 116A. Through the electronic index, the user obtains the listing for available information titles by author's name, by specific title of the work, by ISBN code or by subject matter. Once compiled, a listing of the available information titles included in the index category selected and the other necessary information to allow the user to purchase or rent any information title contained in the index category listed is displayed on the video screen. Using the video listing, the user selects any title listed thereon and obtains a printout of the relevant information through the printer slot 342. Upon proper access by a user, the information titles may then be downloaded from IP Bank 116A onto a user's storage medium 218.

After downloading of information and corresponding electronic index information form central information bank 100 or installation of laser masters 216 to IP Bank 116A, inventory reports are generated by IP Bank 116A and transmitted to central information bank 100. These inventory reports reflect the information titles presently stored in IP Bank 116A. These reports are then are sent to publisher 202. Also, a download completion report is sent from IP Bank 116A to transactional database 118A, sometimes referred to herein as a host fileserver, which in turn generates a status sales report. The sales report is transmitted to central transactional database 104. Transactional database 104 sends the necessary action instruction back to host fileserver 118A and a transaction report to publisher 202 for uses such as accounting and auditing.

The cartridge also contains programming (CCS) that regulates access and use of encrypted data files, decrypted an encrypted file when selected for opening, controls the operation of the user's computer when a decrypted version of an encrypted file is being accessed, and removes all trace of any decrypted version of an encrypted file from the user's hard drive or RAM when being closed. (RAM may include or consist of DRAM, SDRAM, and/or VRAM).

In one embodiment, a content provider can specify instructions relating to authorized use, access and cost of the provided information that can be permanently

linked to a master data file of the information when it is electronically stored or accessed. Instructions are electronically generated when information is received electronically. These instructions include a description of allowed uses (e.g., copy generation, printing limits and/or authorization, rental options, if any, purchase options, etc.), desired level of security against unauthorized copying or use (from level zero [virtually no security] to a maximum level [most robust security available]) and cost to a user, optionally as a function of user-selected access right. Whenever a request is made to obtain a copy of the information file (i.e., IP File), the instructions are used in connection with the choices made by the user, within the allowed options, to generate user-specific electronic instructions for limiting use of the requested copy of the IP File. For example, content providers who wish to allow a user to be able to produce a hard printed copy of a portion of an information file for study purposes, when the information files are being downloaded into the master file, can select that special authorization codes be included with the master data file. Once established, the instruction codes so generated accompany the information file to the storage medium, or cartridge, of the user and regulate the use of the information file by the user. The cartridge retains information relating to such printing and restricts further printing once the limits have been reached. The user determines, within the defined limits, or authorized purposes, the portion of the text (natural language text or other IP) to be produced as a hard copy by using the high lighting features of the reader programming to make a selection. In one embodiment, if no special codes are selected, default codes apply that embody the most restrictive instruction selections. In another embodiment, the content provider can allow the user to select the applicable use and access instructions from a variety of authorized choices with the selection being made at the time of acquisition. The cost of acquisition to be paid by a user will vary based upon the user choices regarding desired use. In one embodiment, rules can be updated or adjusted by a content provider.

In another embodiment of the present invention, when one or more IP Banks are operated in conjunction with one or more retail establishments, each retail establishment is given a unique identification number that is used to tag each user

device, user storage medium and/or encrypted IP file acquired at the retail establishment or through an IP Bank linked to the retail establishment. Using the tagged information, particular user devices, storage media and/or IP files, a predetermined portion of the system revenue generated in regard to such tagged user devices, storage media and/or IP files is allocated to and shared with the retail establishment in question. In another embodiment, the tagging is further delineated to allow revenue sharing with manufacturers and distributors of user devices and/or storage media.

C. Point-of-Sale Delivery System Configurations

The point-of-sale delivery systems, as previously discussed, are classified by function. The functions include one or more of the following: (1) point of purchase delivery system, (2) point of rental delivery system, (3) IP bank subsystem, and (4) promotional delivery system. The configurations for each of these functions are separately discussed in detail below.

Point of Purchase Delivery System

A point of purchase system is illustrated in block form in Figure 3. The point of purchase system is described herein for illustration purposes only as a system from which IP can be purchased. As pointed out above, however, the system is not limited to a particular type of IP and other media capable of being expressed in electronic form such as computer software, music and video could be purchased utilizing the present system. In addition, although the system described here is a point-of-purchase delivery system, other embodiments employ either synchronous or asynchronous IP network protection, or both, making possible real-time IP transmission.

The point of purchase system illustrated in Figure 3 includes an IP Bank 302 coupled to host fileserver 304. Server 304 is coupled to a customer service terminal 306 (of course, there could be more than one terminal) and a cashier's station

308A which is further interconnected to other cashier stations 308B-D. Server 304 also is coupled to an institution network 310, which in turn connects to institution terminals 312A-E. Service terminal 306, cashier stations 304A-D and institution network 310 are connected to server 304 via a computer communication link such as a commercially available computer networking system such as CompuServe, a public network such as the Internet, an intranet, or a virtual private network (VPN). IP Bank 302 and server 304 are connected to central information bank 100 and central transactional database 104 as explained above with reference to Figs. 1 and 2.

Cashier stations 308A-D are in serial, linear networking connections which allows the addition and removal of a number of cashier stations at any time. This configuration accommodates extra cashier stations required during rush seasons or rush hours and the desire to remove cashier stations for better utilization of space after the rush seasons. Customer service terminal 306 has local processing capability that provides customer services such as personal identification initiation, personal identification number changes, processing of complimentary IP, IP refunds, customer information entries and updates. The customer services terminal 306 can also provide the retail outlet with internal administration and the management functions, such as the IP inventory cards management, the IP list management, IP requests, IP reports, financial reports, and e-mail and Bulletin Board management.

Referring now to Figure 4, point of purchase fileserver 304 is shown in more detail. Particularly, server 304 includes one or more central processing units (CPUs) 316, a primary power supply 318, an uninterrupted power supply 320 to assure continuous operation during power failure, and a high density storage 322 that holds all the programs and the databases required for server 304 operation.

Server 304 has four (4) interfaces, i.e., a network interface 324, a maintenance interface 326, a customer service station interface 328 and a cashier station interface 330. CPU 316 transmits instructions to IP Bank 302, creates

transaction databases and reports, and processes orders from cashier stations 308A-D and customer service terminals 306A-D.

From network interface 324, server 304 communicates with central transaction database 104 for electronic filing of transaction reports and communicates with IP Bank 302 to give IP Bank 302 downloading instruction orders and to receive the status reports and the inventory reports from IP Bank 302. Server 304 also is coupled, through network interface 324 to an IP Bank subsystem to receive subsystem reports in order to give instructions and orders whenever necessary, as hereinafter discussed. External network systems such as institutional or corporate network systems with local merchants terminals, community bulletin board services and others can also be coupled to the network interface 324. The network interface 324 also allows two-way connecting with inter-bank networks such as Cirrus, Plus or other similar data transfer network. Coupling to merchants' terminals, promotional system provides local merchants and the local business direct access to update their promotions and coupons. Maintenance interface 326 enables remote or on-site diagnosis and repair of server 304.

Customer service station interface 328 provides for communication between server 304 and customer service terminals 306A-D to handle customer service transactions. Customer service terminals 306A-D are illustrated as being coupled through a data switch 332 to a printer 334. Cashier station interface 330 provides that cashier stations 308A-D can communicate with server 304.

Figure 5 illustrates one embodiment of IP Bank 302. IP Bank 302 includes a high resolution color graphic display 336 which is a touch screen device used to display, for example, instructions, messages, and status reports to the user, indexing information and to receive the user's touch screen input selections. IP Bank 302 also has a keypad 338 that is for the user to input a personal identification as well as other inputs. A magnetic code or other generally accepted card reader 341, shown as an insertion slot, is provided for customers' transactions with a bank card, credit

card or some other form of debit card. A bar code reader 340, shown as an insertion slot, is provided to allow users to insert cards containing ISBN and/or ISSN codes for desired information titles for reading by the IP Bank. ISBN and/or ISSN codes may also be manually inserted, for example, by typing the relevant numbered keys on the keypad 338 or with a joystick type device (not shown). A printer slot 342 also is provided to enable the user to access the output of IP Bank 302, a printer (not shown in Fig. 5), as hereinafter described, to retrieve receipts and transactions reports and ISBN access vouchers. IP Bank 302 also includes a base member 344 with a cut-out portion 346 to enable a user to stand comfortably at keypad 338. Other configurations are possible for users having special physical needs. Importantly, IP Bank 302 also includes a cartridge slot 348 for the user to input a reading cartridge, as explained in detail hereinafter, to obtain a copy of the information selected for downloading.

In another embodiment and as shown in Figure 5A, the physical embodiment of IP Bank 302 may be altered. More specifically and in one embodiment, IP Bank 302 is positioned on a desk 339 using a personal computer 349, for example those available from IBM Corporation or Dell. The user operates IP Bank 302 as described above with reference to Figure 5 except, the user may for example, sit in a chair (not shown). Such a configuration may be used, for example, in a corporate, dormitory, library, or other similar environment where the user may be accessing information for longer periods of time or in a professional type environment. In other embodiments, readers 340 and 341and cartridge slot 348 each may be located within computer 349 or in separate devices which are electrically coupled to computer 349.

In yet another embodiment of the invention, and as is shown in Figure 5B, a different physical embodiment of the IP Bank 302 is provided. More specifically, IP Bank 302 is placed in proximity to central information data storage facility 100 and is in electronic communication with central information data storage facility 100 via any suitable means (e.g., cable, wireless, etc.). In this embodiment, IP Bank 302 is contained within a housing 500 for combined central storage 100 and IP

Bank 302. User access to IP Bank 302 is via dial-up modem 502, 512 over the telephone, via the Internet through a web site 504, 506 maintained for that purpose, via satellite linking, via wireless (such as narrowband wireless, broadband wireless, infrared, microwave, radio wave or other similar connection) or via any other network capable of electronically transmitting data in digital form, for example, a cable modem, ADSL-X or broadband wireless network. Once IP Bank 302 is accessed by the user using specialized software contain in a WORM (write once, read many times) memory or other nonvolatile memory on the IP Bank and on the user's storage medium 218, a virtual IP Bank is generated for viewing and access using the user's computer. The virtual IP Bank then performs all of the same functions as IP Bank 302 in another embodiment, except that memory of central information storage facility 100 is used as storage for the information titles and the transaction information, the appropriate hardware connected to the user's computer operates as an access port to IP Bank 302, and specialized software programming stored on the user's storage medium in connection with specialized software programming stored on the WORM (write once, read many times) memory of the IP Bank operates as the software programming to drive the IP Bank.

In yet another embodiment of the invention, IP Bank 302 is a special board configuration located within the user's computer or a separate device electronically linked to the user's computer by a cable, a phone line, a wireless connection (infrared, microwave or otherwise) or any other suitable means. In this embodiment, IP Bank 302 is configured in such a manner that any attempt to read the data contained thereon other than as authorized renders the data totally unreadable. IP Bank 302, in this embodiment, is connected to the central information storage facility 100 or another IP Bank 302 using a secure network. In one embodiment, IP Bank 302 also functions as a controller of the user's computer when being used to access encrypted files or the secure network. In this embodiment, IP Bank 302 also contains operating software that is stored in WORM (write once, read many times) memory within IP Bank 302.

Figure 6 is a block diagram description of IP Bank 302 circuitry. Particularly, IP Bank 302 includes a central processing unit (CPU) 350, which is coupled to display 336, keypad 338, magnetic strip reader 341 and bar code reader 340. Although CPU 350 is illustrated as one unit, it is contemplated that CPU 350 could be a parallel processor or distributed processor arrangement. Selection of CPU 350 type depends on the amount of information to be processed, the desired speed of processing and costs. CPU 350 also is coupled to an automatic teller machine (ATM) module 352 to allow transactions with ATM cards. CPU 350 is coupled to a media driver 354 which enables users to insert personalized media for acknowledgment or other functions as hereinafter discussed. IP Bank 302 also includes a primary local storage device 356 provided for the storage of all information masters selected for loading into IP Bank 302 and related index information. A secondary storage device 358 is provided to hold other programs, instructions and transaction related information. A buffer memory 360 is utilized to speed up downloading in order to accommodate high volume users during the peak seasons. A printer 362 is provided to print coupons on demand, receipts and various reports for the users. A power supply 364 provides power to printer 362. CPU 350, secondary storage device 358 and local storage 356. An uninterrupted power supply 366 coupled to primary power supply 364 assures continuous operation even during power down time.

CPU 350 is coupled to a network interface 368 to provide communication to central information bank 100, host fileserver 304 or an IP Bank subsystem, as hereinafter discussed. CPU 350 also is coupled to a wireless communication port 370, which in turn is coupled to an antenna 372. Wireless communication port 370 enables compatibility with an alternative communication medium in the event that such medium is required.

Figure 7 illustrates, in block diagram form, the structure for a user's personalized media storage cartridge 374. As explained hereinafter, a user inserts cartridge 374 into cartridge slot 348 for downloading of the information selected from IP Bank 302. The downloaded information is stored, in an encrypted format, on

cartridge 374 together with relevant basic index information copied from the electronic index contained in IP Bank 202 at the time of initial downloading. Cartridge 374 is compatible with readers to enable the user to view information stored on the cartridge. Cartridge 374 includes reading software 376 that, as explained in more detail hereinafter, performs sequential encryption and decryption of information. Registry correspondence segment 378 also is provided. an IP file registry 380 is created at the time of downloading information onto cartridge 374. Encrypted IP files 382 together with relevant electronic index information are stored on cartridge 374 as well as a non-erasable permanently marked serial number 384. Cartridge 374 also contains a commercial operating system environment 386 and free disk space 388.

In another embodiment of IP Bank 304, slot 348 is a part of a device connected to the user's computer being used to access the encrypted files.

Figure 8 illustrates the user process and component processing which occurs when a user utilizes the point-of-purchase system described above. Particularly, once the user enters the site 390, and if the user a first-time patron 392, the user will complete a user's application form 394. The user will then take the completed application and a picture I.D. to customer service station 306A, where the user will select and input a personal identification number (PIN) and a password 396. The customer service clerk will open an account for the customer 398. The user-selected password is automatically matched with a sequentially created customer account number within the central data banks. Using the keypad accompanying the cashier station, the clerk types in the name, address and social security number for the user. The written application will then be inserted into the printer slot accompanying the cashier station. While loading the customers' information, the central data bank reviews the information to determine if there are any prior problems with the customer or other discrepancies. When the verification process is completed, the designated customer account number for the user is printed on the user's application. Then, the clerk places, into the card slot, a user identification card (for example, a plastic card having dimensions of the standard credit card and containing a magnetic

strip on the back on which can be placed magnetic coded information). The card is then embossed with the user's name and account number and the magnetic strip is encoded with the applicable user codes (account number, card number and designated password information). The written application is then transmitted to the central data storage center for retention. The user now is able to use the issued card to make purchases or to rent the use of information titles. The machine used to emboss and encode the cards is a standard commercially available machine of the type currently being used in connection with the issuance of a bank's card, credit cards or debit cards. Using the password supplied by the user and the application registration number, a unique user identification number is electronically generated. This identification number is then magnetically stored on the magnetic strip on the back of the user's card and within the data files of the machine generating the embossed card. This information is stored in an encrypted file later transmitted to central information storage facility 100 using a suitable secure network. The unique number so generated is used to represent a personal signature (system registration number) of the user with regard to access to the system and encrypted files.

A user may also obtain a personal identification card by accessing the system, through a network, and by supplying the necessary information for the generation of a user identification file. Upon completion of the registration process, the user electronically receives a digital identification file and identification number (or password). At the time of registration, the user is assigned a unique account number that is associated with his or her data files within the system for accounting and tracking purposes. A digital file containing the assigned account number, the user identification number and the applicable selected password information represents the identification file. After the identification file has been stored, the user may obtain a printed copy of the relevant information contained in said file. At the same time as the identification file is being generated, the system programming generates a unique number using a combination of the user account number and the numeric equivalent of the user password. An electronic version of that unique number represents the electronic personal signature (registration number) for the user. A digital copy of the

unique number is stored with the user's identification file using a special encryption method. Only the public (non-hidden) portion of the data is available for printing to hard copy. The digital identification file is available to be used in tagging and encrypting data files and allowing access to the network and encrypted data files. A copy of the registration file, including a copy of the identification file, is stored at central information storage facility 100.

In addition to obtaining a personal identification card, a new enrollee purchases a reader/computer or other acceptable reading device (such as a special computerized interface, or an audio or video playback device). Each such device is assigned a unique serial number and a special code number. In one embodiment, the serial number is contained on a read only memory chip enclosed within the device. All of the many cartridges which accompany each such reading device is encoded in such a manner that information recorded on the cartridge can only be read by the related reading device. In one embodiment of the invention, this is accomplished by a simple program contained within the permanent memory of the device. In another embodiment, if the special code on the device is not the same as the number which the cartridge is seeking, then the cartridge will cause to be displayed in the reading device the words "cartridge cannot be read by this device" and to not allow any further access to any information contained on the cartridge by the particular reading device in question. If the numbers match, further access will be allowed. At the time a reading device is purchased, the clerk enters the serial number for the device into the central data bank through the cashier's station. The central data bank contains a list of the serial numbers of all approved reading devices and the corresponding special code number. The personal identification card for the customer purchasing the reading device is placed by the clerk into an appropriate slot on the cashier's station and the magnetic strip on the identification card is encoded with the applicable serial number for the reading device being purchased. Thereafter, whenever the user desires to obtain additional cartridges for reading by his or her reading device, the user needs only present his personal identification card and the cartridge to be properly coded to the clerk and by inserting the cartridge into the cartridge slot and the identification

card into the card slot and pressing the designated button on the cashiers station, the new cartridge will be correctly encoded to be readable by the user's reading device.

In another embodiment of the invention, the personal identification number given a user at the time of registration, the unique serial number of the user's storage medium being used to store the downloaded files and a hidden unique number given the content being copied for downloading are used to generate unique encryption and decryption algorithms and formulae. A copy of the algorithms and formulae so generated are stored in the central information storage facility. The encryption formula so generated is used to encrypt the data files as they are being downloaded and stored on the user's storage medium. The decryption formula and algorithms are stored in nonvolatile memory associated with the user's storage medium and are available for decryption when access to an encrypted file is requested. The memory unit and the housing thereof, where the decryption formula and algorithms are stored when downloaded to the user's storage medium, are configured so that any attempt to read the information so stored, other than through the use of the CCS software and related operating programming, renders the data unreadable in any manner. WORM memory is used in one embodiment of the memory unit.

The requirement that reading device codes and cartridge codes match, before access will be allowed, means that issued cartridges will not be readily readable by multiple reading devices. Multiple device reading will require special programming and the granting of special allowances, or approved purposes. In one embodiment, by purpose encoding the information, a publisher may expand or limit access to those users having a proper authorization or defined purpose. For example, a publisher may purpose encode a portion of a selected IP to be readable by any user, i.e., any purpose encoded. This type of purpose encoding may be used on, for example, promotional IP or some governmental publications. Other information may be purpose encoded to limit access to a single user to only view the information, i.e., classified material. The requirement reduces the possibility of unauthorized use of

information titles. Special allowances also can include various security code levels that allow publisher studios and content owners the ability to choose between a wide variety of security from completely open to the highest level in which the content of an IP can only be utilized by one individual. Security codes and special allowance rules for content owners could, for example, be determined by the market value of the content. The life cycle of the content can also be considered, so that new edition music, movie release, and new computer games have different security compared to content that has been in the market for an extended period of time. In staging security related to time value, market value, control level, type of intellectual property, or even legality, the highest level of security, in one embodiment, is the use of the highest security encryption algorithm with a hardware smart chip or smart storage device controller chip or processor.

In another embodiment in which IP Bank 302 is located either in proximity to central information storage facility 100 or in proximity to the user's computer, the user's storage medium is encoded with the applicable registration codes and numbers by use of the appropriate slot on either the user's computer or IP Bank 302, were a separate external unit is being used as such. With the appropriate software activated, the user inserts the applicable storage medium into the appropriate slot and using the appropriate software menu selects the appropriate instruction. The software programming then develops the appropriate registration information and encodes it on the storage medium in the appropriate manner. As part of the initiating process, the software programming checks the storage medium to make sure that it already contains the appropriate CCS. A user storage medium that does not already have a unique serial or identification number is given one and, and any user storage medium that does not accept electronically being tagged with a serial number in a appropriate manner is rejected and so marked. Only user storage media containing the proper software programs and registration information are usable to store data information files.

In operation, the customer takes the customer identification card to the IP display area for shopping 400. If the customer previously opened an account, the customer will not have to go through the above described process and can proceed directly to shopping area 400, where the customer will select an IP inventory card matching his IP selection. The IP inventory card has the IP ISBN and/or ISSN numbers, a bar code and information related to the particular information title, author, publisher, and edition date printed thereon. The customer brings the selected IP inventory card to the cashier's station 308A. The cashier magnetically reads the codes on the customer's I.D. card and scans or manually enters the bar codes on IP inventory cards 402. The customer then makes a proper payment, and the customer codes and information title bar codes are transmitted to host fileserver 304. Server 304 searches the existing customer account file to match the identification (i.e., pin and password) and will generate a downloaded IP list file based on the bar codes from the IP inventory cards or as manually loaded. Server 304 downloads the file to IP Bank 302 which electronically generates a portfolio of information titles ready to be downloaded on demand. The user can then proceed to IP Bank 302 at any later time and insert the identification card into the slot 340 of IP Bank 302 and a coded point-of-purchase cartridge 374 into IP Bank cartridge slot 348 to identify himself with a personal identification number, as illustrated in step 404. The user also enters a password 406 into the keypad 338. When IP Bank CPU 350 matches the personal identification number with a downloaded list portfolio, IP Bank CPU 350 starts downloading the requested information from local storage 356, through buffer memory 360, to media driver 354 which copies the information onto cartridge 374. As part of the downloading process, the data is dynamically encrypted to make the data uniquely readable only by authorized reading devices. The dynamic encrypting is described below in Section D. After downloading, the user removes cartridge 374 and then inserts cartridge 374 into his personal reader/computer to access the information acquired. The reader/computers are configured for long-term reading applications. The reading application software is stored on cartridges with the ability

to read the applicable software on the cartridges permanently stored within the memory of the reader/computers or other authorized reading device.

A user may select portions of selected information to combine and download. In one embodiment, the user may select at least one IP and select at least one portion of each selected IP. If more than one portion is selected, where each portion includes up to the entire selected IP, the portions are combined and downloaded to user's cartridge 374. For example, for a specific college course, a student may be required to download specific chapters from ten different IP. After selecting the ten IP and ten specific chapters of the selected IP, the selected information is combined, encrypted using the determined level of protection, and downloaded to the students cartridge 374. Similarly, a user may select individual tracks from music information to combine and download the selected tracks to a single cartridge 374 for playback at a later time.

In another embodiment in which IP Bank 302 comprises hardware and software in the user's computer and hardware and software located other than in proximity to the user's computer (e.g., an IP Bank located proximate to central information storage facility 100 or a stand alone IP Bank located apart from the user's computer and central information storage facility 100), after connecting to the IP Bank unit 100, using the appropriate browser software, the user will access the central storage facility 100 index of available titles and select the desired titles or parts of titles that the user wishes to purchase. From this information (from which a work is created that represents a portion of several different works), IP Bank 302 will cause to be generated a unique identification number for the newly created work. This unique identification number is generated in addition to any hidden code that is generated and serves as an index reference for the work. IP Bank 302 also uses the user's request to generate a shopping list with appropriate price information, which the user may accept as generated or change. When the list is accepted by the user, the menu requests payment information. Generally, at the IP Bank level, payment is via pre-generated credit voucher or via credit or debit cards. After payment approval, IP

Bank 302 causes copies of the requested titles to be generated, encrypted, and downloaded on to the user's storage medium. While connected to the user's computer and storage medium, IP Bank 302 examines the appropriate log files to determine whether there has been any misuse or unauthorized activities. If such a determination is made, the account is flagged for further investigation.

Point of Rental Delivery System

If a user is not interested in obtaining a permanent copy of a particular work but requires a copy for a period of time, e.g., a semester, the user may prefer to visit a point of rental site rather than a point of purchase site. A point of rental system is illustrated in Figure 9. The rental system is identical to the point of purchase system previously described herein (e.g., includes a host fileserver) except with respect to the differences pointed out below. In many instances, a single site or IP Bank may serve as a point of purchase system site and a point of rental system site and a point of delivery system for promotional or commercial information site or any combination thereof. As shown in Figure 9, the point of rental system includes IP Bank user terminal hubs 410A-B coupled to terminals 412A-E, and customer service station 414. User terminals 412A-E allow a customer to do an information title search and index search of the IP Bank memory and to transmit other information between IP Bank 302 and himself. Customer service station 414 combines the function of customer service as well as the cashier's station. For example, at customer service station 414, a credit customers' debit card can be credited and the ATM operation can be overridden, via ATM module 416, if necessary. Information can be printed out from customer service station 414 via printer 418.

Point of rental storage medium 420 is used in the rental system. Point of rental medium 420 is the same as point of purchase medium 374 except that medium 420 includes an automatic erasure mechanism that erases the information downloaded after the expiration of a preset time interval. More specifically, when information is downloaded from IP Bank 302 onto medium 420, IP Bank 302 also

downloads a "time stamp" (using a nonvolatile, time-encoded chip that cannot be manipulated) equal to the time period for which the user has paid to retain a copy of the information. The time stamp could take the form of a value loaded into a memory location on medium 420, which value corresponds to the rental time period. During usage, the actual usage time elapsed is subtracted from the electronically stamped time period. Once the user has consumed all of the usage hours or other time units authorized, the information title will self-destruct, i.e., be deleted from medium 420. This can be achieved simply by calling a stored program which erases the information associated with the memory location where the time value is stored. For example, when the value of the memory location where the authorized usage time is stored is zero, the stored erase program would be called upon to erase the information associated with the "zero" time usage authorization. In one embodiment, an advance alert feature is provided to allow the user time to pay for additional authorized use before the information is erased.

Another method for automatic erasure is for each rental or library cartridge to contain a real time clock and independent rechargeable power supply. When the cartridge is initially encoded for use, the real time clock mechanism is activated. As rented information titles are being downloaded, an expiration date is logged into the index information for each title. Any time after the real time clock on the cartridge reaches the designated expiration date, access to the relevant information title is denied. If use of the title is not extended, after the expiration of an additional number of days, use of the cartridge will cause a permanent erasure of the information title from the cartridge memory. With this method, if the real time clock falls to operate, the cartridge will become unreadable without repair. To repair a defective cartridge, the user need only bring the cartridge together with his personal identification card to the nearest service center where the real time clock will either be repaired or the relevant information titles will be loaded onto a replacement cartridge. The user will be credited for any lost time while the cartridge was unreadable. A service center could be located, for example, near each separate point of delivery site.

In one embodiment, GMT clock verification is added to the user device, to provide additional IP security in case the user travels with his device through time zones. An encrypted GMT code is added in another embodiment to compare and verify erasure dates and times with actual GMT.

The automatic erasure program could be created as an operating system module or as a separate executable program designed to be "terminate and stay resident" (TSR). A module integral with the operating system is preferred since such a structure ensures that if the operating system is viable, the automatic erasure module is viable.

If the user still needs more time with any particular information titles, the user may return to the point of rental site and "re-rent" the information. Alternatively, it is contemplated that the user could renew the rental via a modem coupled to the reader.

With respect to the user process for renting information, when a point of rental patron enters a point of rental site, the user will use a valid ATM card, bank card, credit card, or some other debit card and proceed directly to a user terminal 412A-E. Using such a terminal, the user can perform information title searching and download an order entry to IP Bank 302. When the download entry is complete, the user will go to IP Bank 302, insert a rental medium, an identification card, and a credit card, bank card, or debit card into IP Bank 302 for the transaction approval. If the transaction is not cleared or if the ATM system is not working properly, the patron can proceed to the customer service center and have the attendant manually override the ATM process, if appropriate. If the user does not have a valid ATM credit card or debit card, the user will go to the customer service center, pay the service clerk to receive credit on the IP Bank debit card. Then the customer may proceed to the user terminal where the user downloads the order entry.

After the transaction approval is cleared, the patron inserts point of rental medium 420 into IP Bank media driver 354, has his personal identification card

scanned and enters a password. The information is dynamically encrypted and downloaded from IP Bank 302 to medium 420 with an electronic stamp of the number of hours of usage authorized for each information title or an expiration date. After the downloading, the user will apply this medium on the personal reader/computer to access the information on the medium.

Typical examples for the point of rental site are libraries (commercial, education or public access) and IP rental shops. The information downloaded by the user may be free of charge to the users such as in the case of a library, or may incur certain rental fees at a predetermined rate, such as in the case of a rental shop or library charging on a per page use basis. Any given point of rental site may operate as a traditional library in allowing free use to library members for a limited period of time or may operate as a rental shop where fees are collected from users in accordance with the period of use allowed.

IP Bank Subsystem

A IP Bank subsystem couples to an IP Bank and host fileserver as described in more detail below. The central element of the subsystem is an IP Bank which is a modified version of the point-of-purchase IP Bank 302. The subsystem is specifically configured for the collective use by members or the staff of a commercial or business entity or a corporation. It delivers and it recalls information titles among authorized users within the business or corporate entity, and provides the capability of limiting the number of copies of a given work that may be distributed to other authorized users. If all of the licensed copies of any information titles have been checked out by the staff of an organization, then no other users may access the same information title within that particular subsystem until one or more of the licensed copies of the particular information is uploaded or recalled to the subsystem or additional copies are purchased. In one embodiment, the information is purpose encoded so as to limit a purpose to which access by a user to the information is allowed. For example, a central corporate library may allow specific users, i.e., R&D

personnel, to selected information, i.e., pending patent applications. All non-R&D personnel users without the proper purpose, or authorization code, are prevented from accessing the information.

Instead of purchasing the unlimited use of a limited number of copies, a commercial or business entity may lease the limited use of an unlimited number of copies or the use of a specified portion of a given information title. Under such circumstances, the commercial or business entity would be charged each time the subsystem is accessed from a participating work station for the portion of a specified information titled accessed and for the period of time the access occurs. By restrictions encoded on the interface between a participating work station and the subsystem, while accessing information from the subsystem, the ability of the work station to perform certain operations would be restricted. The restricted operations would be those related to the duplication or transmission of data related to information titles being accessed through the subsystem.

More specifically, and referring to Figure 10, IP Bank subsystem 422 contains a high resolution color graphic display 424 coupled to CPU 426 to display the instructions or status of subsystem 422. Subsystem 422 also includes a keyboard 428 with limited access to the system for keying selections for operating certain given functions such as product display. (In one embodiment, a voice recognition system is used for input in place of keyboard 428 or to provide additional input capability.) Subsystem 422 has a media driver 430 for the downloading of information and a local storage 432 which holds a portfolio of information the business entity has ordered for use. A secondary storage 434 is also provided to hold all the software programs that control and perform the functions of subsystem 422. Subsystem 422 further includes a power supply 436 and an uninterrupted power supply 438 to assure continuous operation during power failure downtime. A printer 440 is provided to print various reports.

IP Bank subsystem 422 has a network interface 442 that connects subsystem 422 to IP Bank 302 and host fileserver 304. Network interface 442 also may couple to the corporate or business entity network system 444. With such a structure, the corporate entity may transmit or download its own corporate proprietary information through IP Bank subsystem 422.

Media port extension interface 446 provides access by an adequate number of media drivers to the desired corporate terminals for corporate network stations. Media driver 430 is connected to the terminals or stations by a proprietor driver card. The corporate administration can utilize the dynamic encryption and the dynamic downloading function of IP Bank subsystem 422 to incorporate and accommodate the corporate proprietary information. The corporate proprietary information may be transmitted to IP Bank subsystem 422 using an encryption process and then downloaded selectively to the destination port and to the properly identified authorized personnel. IP Bank subsystem 422 is not only a customized corporate library of copyrighted and proprietary information, but also is a corporate document security device that encrypts and dispatches the corporate documents and the corporate confidential proprietary information in the corporate network system. As part of the network interface connection linking each participating work station to the subsystem and allowing access to encrypted information, a separate unit, e.g., a memory storage unit restricts certain operations which may be performed from the work station so long as the work station has access to encrypted information from the subsystem. The restrictions limit or prevent operations related to the duplication or transmission of data.

Promotional Delivery System

The promotional system is a point of delivery system for promotional and commercial information. It distributes promotional and commercial information in electronic format and users may either view the digitized promotional and commercial information at the site or download the information to their personalized

media for later viewing. User's call access the promotional and commercial information including the dynamic viewing electronically of advertising available discounts, commercials, special promotional events, software demos and product catalogs. Users may even shop electronically by manipulating the promotional and commercial information and placing orders through e-mail from a personal reader/computer or by ordering directly from an interactive promotional IP Bank. The promotional IP Bank has the same structure as IP Bank 302 for the point-of-purchase system.

A promotional system in accordance with the present invention is illustrated in block diagram form in Figure 11. As in the other point-of-sale systems, IP Bank 302 is networked to host fileserver 304. The promotional system further includes a number of promotional units 448A-D which electronically display and promote products. Unit 448A is coupled directly to central transactional database 104 and central information bank 100 while units 448B-D are coupled to host fileserver 304. Unit 448A receives information from merchant terminals 450-A-D and host fileserver 304, receives information via merchant terminals 450E-G. More specifically, host fileserver 304 receives advertising and special offer updates from the local businesses, national or regional advertisers, and corporate sponsors through merchants terminals (MT) 450E-G. The host fileserver 304 is also networked to a central transaction database which, in turn, provides a report to the publishers, advertisers, accounting, auditing firms, merchandise vendors, and others.

The promotional IP Bank allows selective downloading of promotional and commercial information to the user's point of rental medium (see discussion in Section B, System Architecture, for explanation of such downloading) for the user's private review and personal shopping at his convenience. The promotional and the commercial information downloaded will self-destruct (i.e., automatically erase) at the expiration of a pre-determined time interval as explained above with respect to point of rental delivery systems. The promotional IP Bank also provides a user interactive self-service vending feature. The user may order products or information

electronically via the network. Some of the promotional functions are: coupons on demand, virtual shopping, catalog sales, demos, subscription orders, electronic applications of credit cards, calling cards, or other types of services. Some public domain information distributed such as community events, ticket sales, institutional events or even public bulletins could also be distributed with the promotional information as a free or low cost service to the community.

The promotional and the commercial information flow is very similar to the information flow within the point-of-sale delivery system. However, rather than a publisher or copyright information owners, the information sources are local businesses, national or regional advertisers, and appropriate sponsors through advertising agents and other entities.

Information Tracking

In one embodiment, for each exchange, or download, of information, a tracking entry is transmitted, or stored, in an appropriate transactional database, for example central transactional database 104, to record movement, or transfer, of information from a first location to a second location. By reviewing these tracking entries, an information owner may monitor movement of the information and take appropriate action. For example, a tracking entry may be recorded in transactional database 104 each time any information is copied into, removed from, or copied from, IP Bank 302. Based upon the tracking entries, the information owner may charge the receiving or transferring party a fee as defined by the owner of the information. The fee charged may be based on a variety of factors including, but not limited to, an economic value of the information, use or purpose of the information, number of users, and the availability from other sources. The tracking entries may also include additional data so that the information owner may determine who transferred the information, the amount of information transferred, type of transfer, i.e., rental for specified period of time, and the time of transfer. Tracking entries, in one embodiment, are recorded for all transfers, i.e., IP Bank 302 to cartridge 374 and

central information bank 100 to IP Bank 302. Utilizing these entries, an information owner may also determine the type of information that is being transferred, the number of transfers, and the identification of the information receivers. For example, the information may be used to determine whether targeted users are receiving certain promotional materials are being received by targeted users, or to determine responses to different information pricing strategies.

D. Encryption

The above-described point-of-sale delivery systems have the capability of performing dynamic encryption of data as the data is downloaded onto a user's storage medium. Dynamic encryption refers to the process in which the IP Bank works together with the storage medium to perform a proprietary encryption of downloaded data. In one embodiment, different levels of encryption are utilized based on a series of factors or variables. These variables include, but are not limited to, an economic life, a market value, a general availability, a replacement cost, time sensitivity, and potential number of users, of the information. For example, today's TV listings may have a low level, or complexity, of encryption as a result of the low market value, low replacement cost, and general availability from many sources. Conversely, a multi-volume legal treatise may, for example, have a high, or complex, level of encryption as a result of the limited availability, replacement cost, and long economic life of the information. Based on the described factors, a source, i.e., publisher, of information may decide the appropriate level of encryption for each portion of information from the initial transmission to central information bank 100 to a user's cartridge 374. The level of information encryption, in any location, i.e., bank 100, may be higher or lower than another location, i.e., cartridge 374. More specifically, each time information is downloaded, or transmitted, the level of encryption may be independently altered, or determined. For example, the level of encryption at central information bank 100 may be different, i.e., higher or lower, than the level of encryption of an IP downloaded to a user's cartridge 374.

In addition to dynamic encryption, other encryption may be performed as illustrated in Figure 12. Figure 12 illustrates a three level encryption process. For example, prior to transmitting information on the network, the data may be encrypted. This facilitates preventing unauthorized users from accessing the transmitted information on the network. In addition to the pre-transport encryption, the data, may be encrypted prior to being placed in an IP bank. Publishers or other owners of the information may have approval authority over this level of encryption to provide such information owners with satisfaction that the data is adequately protected.

Once the data is stored in the IP bank, dynamic encryption techniques may be used when downloading the data onto storage media. The storage medium (Figure 7) includes a proprietary environment for building, reading, viewing and processing. The medium also has a commercial operating system environment for processing information files. An information file directory registry forms a part of the proprietary application, and a file directory pointer is contained in the operating system application.

The dynamic encryption process, in one form, uses the permanent serial numbers stored in the storage medium, the user's personal identification number, a personal signature code number, and a password to further encrypt the data stored in the IP bank as the data is downloaded to a user's storage medium. The personalized variables and codings are combined with various individualized information file variables to form an individualized data structure for the data downloaded to the user's personalized medium. As a result, those information files are individualized pertaining to the medium, the version of software, the information file itself, and other variables.

The dynamic encryption assists in reducing the possibility of the unauthorized use of proprietary or other information by causing all information downloaded through the point-of-sale delivery system to be readable and accessible by a selected number of user readers/computers. Specifically, data storage medium

accessible from one reader/computer will not be accessible using another reader/computer unless such access has been prearranged such as by providing the other reader/computer with an identical user identification number and password.

Examples of well-known encryption algorithms which may used in performing the above described three level encryption include the Z8068 Data Ciphering Processor (DCP). The DCP contains the structure to encrypt and decrypt data using National Bureau of Standards encryption algorithms. It may be used in a variety of environments including in dedicated controllers, communication concentrators, terminals and peripheral task processors in general processor systems. DCP provides a high throughput rate using cipher feedback, electronic code IP or cipher block chain operating modes. The provisions of separate ports for key input, clear data and enciphered data enhances security. The host system communicates with the DCP using commands entered in the master port or through auxiliary control lines. Once set up, data can flow through the DCP at high speeds because input, output and ciphering activities can be performed concurrently.

In alternative embodiments, encryption and decryption may be performed in dedicated hardware and/or software functions. For example, each reader and cartridge 374 may include a dedicated encryption integrated circuit (IC) and a dedicated decryption IC to maximize the transfer speed of the information. The level of encryption and decryption may be altered by adding additional functions and by enabling or disabling the additional levels.

With respect to dynamic encryption, the following describes one of many methods of dynamic encryption that could be used. Particularly, each regularly used alpha or numeric symbol is assigned a corresponding number as illustrated in Table 1.

Table 1

symbol	A	B	C	D	E	F	G	H	I	J	K
code	1	2	3	4	5	6	7	8	9	10	11
symbol	L	M	N	O	P	Q	R	S	T	U	V
code	12	13	14	15	16	17	18	19	20	21	22
symbol	W	X	Y	Z	0	1	2	3	4	5	6
code	23	24	25	26	27	28	29	30	31	32	33
symbol	7	8	9	.	,	;	:	+	-	x	
code	34	35	36	37	38	39	40	41	42	43	44

The serial number stored on the cartridge would be used to determine how many slots the code should shift to the left at the start the encrypting. For example, if the serial number ended with six, before starting of encryption, the code would be shifted to the left by six places. Table 2 illustrates the code table after the shift.

Table 2

symbol	A	B	C	D	E	F	G	H	I	J	K
code	7	8	9	10	11	12	13	14	15	16	17
symbol	L	M	N	O	P	Q	R	S	T	U	V
code	18	19	20	21	22	23	24	25	26	27	28
symbol	W	X	Y	Z	0	1	2	3	4	5	6
code	29	30	31	32	33	34	35	36	37	38	39
symbol	7	8	9	.	,	;	:	+	-	x	
code	40	41	42	43	44	1	2	3	4	45	6

The selected user password then is used to determine after how many symbols the code should again shift to the left. As an example, if the password were ROSE, then using the codes from Table 2, the numeric statement for rose would be 24212511. When the corresponding numbers are added together until reaching, a number between 1 and 10, the number reached in our example is 9 [18.9]. So after every 9th letter, the codes would be shifted another 6 spaces to the left. After the encrypting of 9 letters, the codes would be set as set forth in Table 3.

Table 3

symbol	A	B	C	D	E	F	G	H	I	J	K
code	13	14	15	16	17	18	19	20	21	22	23
symbol	*L*	M	N	O	P	Q	R	S	T	U	V
code	24	25	26	27	28	29	30	31	32	33	34
symbol	*W*	X	Y	Z	0	1	2	3	4	5	6
code	35	36	37	38	39	40	41	42	43	44	1
symbol	7	8	9	.	,	;	:	+	-	x	
code	2	3	4	5	6	7	8	9	10	11	12

Because the fact that the encrypting tables are constantly shifting, under this simple method, the phrase "My brown dog has fleas." would be encrypted as follows:

19 31 6 8 24 21 29 20 6

16 27 19 12 20 13 31 18 24

23 19 37 11

Decoding using only Table 1, the coded phrase would read as follows:

S 4 F H X U 2 T F

P 0 S L T M 4 R X

W S . K

Without knowing other information, it would be very difficult to find a pattern that would allow one to decode the symbols.

Knowing the placement of the codes relative to the symbols at the start of the encryption process and the number of symbols between shifts, decoding an encrypted phrase is simply a reversal of the process applying each of the tables in reverse.

There are any number of other methods of dynamic encryption that use different methods to vary encryption codes as one proceeds through the data to be encrypted. The objective, of course, is to make decoding difficult by avoiding obvious patterns associated with conventional language and number usage, so one would simply select a suitably difficult to decode dynamic encryption method.

In another embodiment, the unique dynamic encryption and decryption algorithms and formulae are generated using a unique combination of the personal signature (identification number) of the user, the serial (registration) of the storage medium and the registration number associated with the master data file being copied and encrypted using commercially available encryption programming or services. The electronic copies of the encryption and decryption formulae so generated are then stored at the central information storage facility for later use. With this embodiment of the invention, a unique set of encryption and decryption algorithms or formulae are so created for each separate copy of IP file requested. A copy of the unique decryption algorithm and formula is stored on nonvolatile memory on the cartridge (user storage medium) for later use. The nonvolatile memory on the cartridge (user storage medium) is so encased and configured that any attempt to read data stored thereon in an unauthorized manner will cause the memory to become totally unreadable and unusable. The unique encryption algorithm and formula so created are used to dynamically encrypt the selected IP file as the file is being copied and downloaded to the user cartridge (user storage medium) for storage and later use.

At the same time, a unique electronic header file is created for association with the encrypted copy of the requested IP file. The header file contains the user registration information, the rules and restrictions on use of the encrypted requested IP file and the electronic address for finding the appropriate decryption algorithm and formula and the location of the applicable CCS for decryption and use of the data within the encrypted IP file. After generation, the header file is wrapped within the encrypted IP file as the dynamically encrypted file is being generated, is attached to the electronic copy of the encrypted file as a separate part, or is written as a separate distinct data file that is electronically linked to the encrypted IP file. The manner in which the special header file is attached to the encrypted IP file depends upon the desired level of security and the hardware configuration.

E. Tamper Protection

In one embodiment of the present invention, access to the information is monitored, or recorded, to determine attempted unauthorized access to the information. If an unauthorized access is recorded, or stored, onto a user's medium, for example cartridge 374, the next time that user attempts to download additional information to cartridge 374, an unauthorized access message may be transmitted to notify the appropriate party, for example the cashier. As a result of the unauthorized access message, the cashier may revoke user's cartridge 374, notify the proper authorities, or record an entry into the user's account for future action. More specifically and in one embodiment, the unauthorized access is determined by first reading, or recording, the specific identification data from the information requester, or receiver. If the data provided by the information receiver is determined to not match, i.e., is unequal, predefined values, the unauthorized access message is recorded and information exchange is prevented. The data determination may be completed using known comparison hardware and/or software functions.

Additionally, the unauthorized access message may be generated if a user having an incorrect purpose, or authorization code, attempts to access

unauthorized information. For example, in a corporate environment, if a user attempted to access information for which the user did not have the proper authorization code, an unauthorized access message is generated and may be sent to, for example a system administrator or a security official. Different level of unauthorized access messages may also be generated. For example, a high level message may be generated if a user attempts to decrypt the information stored in various locations within the system, for example IP Bank 302 using an unauthorized device. A lower level message may be generated if a remote user has attempted to access data that is one level above that user's authorized level.

G. Other Embodiments

In another embodiment, IP Bank 302 may be configured to capture and exchange real-time information. For example, as a professor presents material to students in a classroom, the professor's presentation may be captured and converted into copyrighted text and exchanged with remote users. This conversion may be completed using known voice to text conversion systems using a known computer system. The professor's presentation may be supplemented with previously prepared, or concurrently prepared, written text. The text may be digitized and properly integrated into the text using known methods. Remote users may receive information from the professor's lecture in real-time as the material is presented or may receive the information at a later time. Remote users receive only that information which the remote user is authorized to receive from IP Bank 302 as described above.

In yet another embodiment IP Bank 302 is configured to receive audio, video, and/or computer software code. For example and in one embodiment shown in Figure 13, IP Bank 302 is coupled to a Video Cassette Recorder (VCR) 600, a stereo system 610 including a cassette recorder/player 620, a Compact Disk (CD) or DVD-X player and/or recorder 630, a television 640, and a computer 650. As described above, authorized information is received from IP Bank 302, and in one embodiment is stored to a storage device, for example a memory device 660. The

memory device 660 may be a plurality of memory cells, for example Read Access Ram (RAM), Read Only Memory (ROM), a rotating storage unit, i.e., a hard disk, a magnetic storage medium, i.e., magnetic tape, or other storage media, for example an optical storage medium. After the remote user has selected the appropriate information to receive from IP Bank 302, the information is stored in device 660. Device 660 is configured to transfer the stored information to the selected playback device, i.e., Video Cassette Recorder (VCR) 600, stereo system 610, cassette recorder/player 620, CD or DVD-X player/recorder 630, television 640, or computer 650. For example, in one embodiment, the remote user downloads, or receives, the entire contents of a top ten music album. The contents of the album are stored in device 660. As described above, the information may be permanently stored or may be stored for a fixed period of time or number of uses. After downloading the information to device 660, the remote user may transfer the information to stereo system 610 for listening. In another embodiment, the information may be transferred to, or through, device 660 to one of the other components, i.e., cassette recorder 620, VCR 610, CD or DVD-X recorder 630, or computer 650. To limit unauthorized copying or playback, the information may be playback using only those components, i.e., cassette recorder 620, VCR 610, CD recorder 630, or computer 650, coupled to device 660. For example, the remote user may download a feature movie by saving the movie on a tape using VCR 610. The remote user may then playback the movie as authorized as long as the tape is playback in VCR 610 that is coupled to device 660. Similarly, the remote user may download a software program so that the information is stored in device 660 or in a storage medium in computer 650. Depending upon the authorization code of the software, the program may be configured to execute only from computer 650 when computer 650 is coupled to device 660.

The above-described system facilitates controlled and monitored exchange of information between many types of information owners, distributors, and users. By using the described system, a user may obtain many types of authorized information. The user may, as determined by the information owner, purchase, rent,

or obtain without charge, the authorized information. The information, in one embodiment, is encrypted using various levels, or complexities, of encryption to prevent unauthorized access. The level of encryption depends upon a variety of factors or variables, for example, the economic life of the information. For example, IP Bank 302 may include information representing a reference dictionary and a top ten music album. Information from the reference dictionary and the album may have the same or different levels of encryption. In addition, students from a determined class may access the reference dictionary information without charge as the result of the school purchasing an unlimited use copy of the information, however, those same students would be required to purchase any information downloaded from the album. Additionally, the type of access may differ for different portions of the information. For example, a first track of the album information may be coded so that anyone may download the information without charge, however, the remaining tracks of the album information may be coded to require payment to download.

In still another embodiment, and referring to Fig. 14 and Fig. 15, the present invention relates to a storage and retrieval system that is vendor, product, and IP independent. This embodiment provides an object-based system that packages any type of data on a network. The type of data in the package is immaterial to this embodiment. However, accurate, timely, and secure delivery is ensured by the facilities provided by this embodiment of the present invention. Each separate IP is stored in digital form, and a consistent interface is used for its delivery. Various levels of standardized encryption are available. IP so encrypted is distributed and read in a uniform manner.

One of the facilities provided is one or more central data centers or central data storage facilities 100. Each central data center 100 acts in conjunction with any others present in the network to store and control delivery of IP from IP content providers 707. In some embodiments, additional data centers 702 are provided, each comprising one or more computers acting in conjunction with each

other. Computers 704 need not be in the same location. Each data center 702 either services one or more local clients 706 or acts as part of central data center 100.

Local public devices, or kiosks 708, are networked to and controlled by the data center. Kiosks 708 are examples of interfaces or access ports by which consumers have access to central data center 100 and by which they acquire desired IP's. Besides kiosks, other examples of access ports are computers (mainframe, desktop, and laptop), personal digital assistants (PDAs), electronic books or ebooks, modems, and other devices designed for accessing electronic networks such as telephone, Lansat, Internet, intranet, and cable networks for electronic transfer of digital data.

A security encryption compression (SEC) module 710 is provided in each consumer product or user device 712 to control access to, and use of IP's. To obtain an IP, a consumer must have a registered SEC module 710 and an authentication password that is recognized by kiosk 708. In addition, the consumer must have a data storage medium 714 to hold a requested IP after downloading. SEC module 710 will securely store the IP for use in an appropriate manner. SEC module 710, in one embodiment, provides its own user interface (for example, a screen and/or speakers, etc.). In another embodiment, SEC module 710 provides one or more external adapters 716A, 716B, to provide a signal for display by one or more other user devices 712A, 712B, 712C, 712D. External devices 712A, 712B, 712C, 712D, communicate with SEC module 710 through, for example, infrared ports, RCA plugs, headphone jacks.

SEC module 710 provides security through an "onion" approach, i.e., one that is made up of multiple protection layers that surround the IP. A first such layer is a hardware layer. SEC 710 comprises, in one embodiment, a chipset having a unique serial number, nonvolatile random access memory (NV-RAM) 718, read-only memory (ROM) 720, and a programmable logic controller/processor 722 with electrically erasable programmable read only memory (EEPROM) 724. The unique

serial number provides part of a public/private encryption key along with a user's access code stored in NV-RAM 718 and the IP being accessed. Each IP is specifically encoded to work only with a specified piece of registered hardware accessed by a specific user's security code. For some IP products and in some embodiments, transfer from one SEC module 710 to another is possible, but only through an authenticating interface such as a kiosk. Hardware protection provided by SEC module 710 is also the first authentication required by the network before it permits IP to be exchanged. If an SEC module 710 is bypassed to obtain direct access to storage medium 714, the physical interruption is archived and transmitted to the network during a subsequent communication, for example, the next communication.

A second layer of security is provided by SEC 710 firmware. This layer is provided by a programmed EEPROM 724 that decrypts stored IP flowing to an output device. Public/private key encryption is dynamically controlled by each IP. In one embodiment, this control is monitored as well as modified each time SEC module 710 communicates with the network.

A third layer of security is in the IP itself. It is encrypted and compressed while downloading to storage medium 714 in an SEC module 710.

Information obtained from a participating publisher 707 in digital form is enveloped with a uniform electronic signature that uniquely marks the version, creating a master copy for distribution. The signed master copy is then encrypted using a custom encryption algorithm and stored in a centralized data storage facility 100 or facilities (depending upon the size of the distribution network) for later retrieval. Each stored work has a title, which is added to a master table of contents or index 726.

Central storage facility 100 is electronically accessed by a user via an access port using secure, encrypted application-layer protocols. These protocols are independent of network transport and physical layer protocols, and thus, usable on any transmission network. Utilizing these application-layer protocols, the user

communicates with and receives instructions from central storage facility 100 for acquiring an electronic version of one or more selected IP's. In preparation for downloading the selected IP's, central storage facility 100 further encrypts 728 the IP's selected for distribution in accordance with a level of encryption selected by the publisher of the IP.

Encryption is provided in several different forms. For example, public/private encryption keys are used for data storage. Watermarking is used when sending data to an unsecured device such as a headphone, a television set, or a word processor. For example, a watermark code is injected by a digital-to-analog (D/A) converter into an analog output signal, which, when analyzed, identifies a source of the IP and its purchaser or licensee. The D/A converter code is determined by the serial number and personal ID code in NV-RAM 718 and from coding in the IP itself, thereby uniquely identifying a combination of user, device, and IP. The secure transfer of IP from the network is done, for example, via RSA encoding, which is described in detail in U.S. Patent No. 4,405,829. This secure transfer ensures that each packet is uniquely encrypted as well as monitored for possible hijacking and other data stream tampering.

In this embodiment, compression is provided both to maximize storage capabilities and to provide another layer of security for the IP.

SEC 710, in one embodiment, is an integrated storage device, with at least one of a hard disk, a RAM disk, an NV-RAM card (e.g., a compact flash memory card) or other storage medium as medium 714. One or more interfaces 716A, 716B, 716C, 716D are provided between the integrated storage device 714 and one or more output devices 712A, 712B, 712C, 712D. SEC 710 is designed to permit storage device manufacturers to incorporate the chipset into their storage devices, so that the storage devices can accept IP from the network.

In one embodiment, SEC 710 comprises a storage medium 714, an SEC chipset 718, 720, 722, 724, a USB (universal serial bus) port 730 to interface

with a kiosk 708 for IP transfer, an A/D converter 716A, and a headphone jack 712A. Another embodiment provides both a USB and a USB2 connection. In yet another embodiment, SEC 710 includes a infrared (IR) port and an RCA jack. In still other embodiments, SEC 710 is built into a device such as a personal digital assistant (PDA) or a single-function reader that serves as a device for reading digital IP or ebooks. In yet another embodiment, SEC 710 is built into a user device such as a laptop or desktop computer. For example, half of the hard drive of the computer can be set aside for a traditional operating system and the other half dedicated for storing IP, for example, user video, books, and audio. Still other embodiments include other wireless couplings, iLink, Firewire, and/or IEEE 1394 connections, for example.

Encryption ensures protection for IP by allowing downloaded IP to be read only on a particular SEC 710, and only by a user knowing a particular password. This limitation is provided, for example, utilizing a decryption formula provided by a key generation company 732 such as Verisign, Inc. The key generation company creates a formula that enables a decryption key to reverse the encryption 728 provided the network. This formula is sent during registration of an SEC module 710 and burned into NVRAM 718 or other limited access memory in one embodiment. In another embodiment, the formula is preprogrammed into SEC module 710. In one embodiment, the formula is stored in an encrypted form and decrypted by SEC 710 hardware only while it is being used. In one embodiment, the same formula is used by the network to encrypt IP at the central data center and by SEC 710 to decrypt the encrypted IP, using different keys determined by key generation company 732.

In one embodiment of the present invention, modern algorithms and appropriate key lengths are used to protect the IP when an IP file is initially created, during its distribution, and throughout its existence. In addition, system programming regularly and automatically updates and changes encryption and decryption code algorithms, keys, and formulae both to recapture IP for which such protection has been broken or compromised and to inhibit the cracking or compromising of IP protection. The frequency and/or number of times that algorithms, keys, and/or

formulae are changed depends, in one embodiment, on one or more factors such as the length of time a user has rights to an IP, a level of security that is assigned to the IP, and a preassigned schedule that is based upon the level of security assigned to the IP.

Employing key generation company 732 independent of the network operator to generate keys is not necessary, but may be preferred by owners of IP who believe that extra security and accountability is provided by such companies. It is not necessary to employ the services of a key generation company, however. Thus, in one embodiment of the present invention, key generation facilities are provided within the network itself.

In one embodiment, operation of the network proceeds as follows. Digital IP of various types and from various sources 708 is obtained and stored at central data center 100. Users (or potential users) of the digital IP obtain a SEC module 710 with a unique serial number (USN) permanently "burned" into the electronics of the module. Each user also chooses a personal password (PP) to identify himself or herself for registration. The user-selected PP, in some embodiments, is constrained within limits selected for security and practicality by the network operator. For example, in one embodiment, the PP is constrained to be no less than 6 characters and no more than 12.

A user with an SEC module 710 and a selected password then connects SEC module 710 to a kiosk 708 or other access port. SEC 710 transmits the USN and the user-selected PP to the network for registration, where the USN and PP are combined (for example, by concatenation or by a mathematical or logical operation) into a unique code (UC). In at least one embodiment, the UC is sent to a key generation company 732, for example, Verisign, Inc. The network itself also generates a hidden private key (HPK) and sends it to key generation company 732. Key generation company 732 then computes a private encryption key (PEK) from the HPK and the UC and sends the PEK to the network, where it is used for encryption of IP as it is sent to the user of SEC 710. The user then enters the PP into SEC 710,

where it is combined with the USN (as combined by the network) to decrypt the IP for the user.

In one embodiment, servers 734 at the centralized storage facility or facilities 100 determine a most efficient cost effective path to distribute IP, including, but not limited to, books, music, video, computer software and multimedia, and automatically tracks, verifies end user tampering, freezes the end users access, and regulates distribution method requirements. The regulation varies for each account and industry. For example, a retail establishment will use a different set of rules requiring different distribution limitations, retrieval and viewing methods, and security and encryption specifications when distributing IP than an institution or an individual is likely to require. Also, methods of electronic distribution and security requirements vary in accordance with the types of IP that are distributed. For example, digital music transfer and retrieval is performed differently from digitally retrieving, and reading an electronic IP, or transferring, retrieving and viewing video. A portion of the verification programming is devoted to checking and verifying the readability of the IP file after being encrypted and downloaded to the user's storage medium. The verification programming also attempts to correct correctable errors automatically and indicates detectable, but uncorrectable, errors. If uncorrectable errors are found during verification, the downloading and encryption process is repeated for a predetermined number of times or until no uncorrectable errors are detected within the predetermined number of attempts. If uncorrectable errors remain after the predetermined number of attempts is exhausted, the user is directed to a service representative to provide assistance. In one embodiment, CCS is also programmed or otherwise configured to automatically verify the readability of an encrypted IP during access for reading and automatically attempts to correct any errors discovered that do not relate to functioning of the security mechanism. For example, CCS is programmed to detect and correct errors resulting from media deterioration. Such errors can result from passage of time or from repeated access or use.

In one embodiment, the current invention consolidates distribution all forms of IP into one network that is transparent to the end consumer. The network includes transparent links that feeds to various IP web-sites 736 requested information. When an end user orders IP works via a web site 736, central repository 100 storing the requested data links the request and transmits the IP to a user's computer 738 (or other apparatus) with a secure dynamic encryption code that adds another layer of security to the secondary repositories. Participating IP retailers' web sites, kiosks, or other terminals, depending upon the needs of particular retailers, are also transparently linked to the centralized repository. Thus, a retail customer can continue to purchase IP content from a favorite retailer's web site or store, while the retailer links to the central IP data repository "behind the scenes."

Depending upon the most efficient and cost effective distribution method, the one or more centralized repositories 100 are locally based, regionally based or centrally based. Each central repository tracks IP data and credits sales 740, 742 to retailer sites. Key locking is provided so that individual retailers have access to their own sales tracking information at the central depository, irrespective of whether sales transfers occurred via Internet, microwave, infrared, satellite, cable, telephone, or other medium.

One embodiment of the present invention is linked to various web sites 736 in a manner transparent to consumers. When purchases are made through various web sites, stores or organizations, corporations, etc., an earmarked transactional fee is tracked, accounted for, and distributed to that particular entity through a universal tracking system 740, 742. In this embodiment, the central information storage facility is a grouping of separate servers or storage facilities at one or more physical locations that are physically operated by one or more persons and that are linked together via a network arrangement and by appropriate software such that they appear to the user to be a single functioning storage facility. Retail companies are thus provided with income, irrespective of whether their customers receive downloaded IP products via a web site 736 of the retailer, from a high-bandwidth kiosk 708 at a traditional store, or

recorded on media by the retailer on demand at a retail establishment. Using this embodiment of the present invention, retailers can provide a useful, profitable service by distributing IP through in-store kiosks or on media recorded by the retailer on demand to customers lacking high-speed Internet connections. Providing this service may be a decisive advantage to some retailers. The additional flexibility afforded retailers by the present invention may also help retailers avoid market base erosion as Internet bandwidth to the home continues to expand.

In one embodiment, IP's and/or links to IP's (i.e., pointers to another location where the IP is stored, such as on a server of an IP content provider 707) are obtained from content providers 707. Referring to Fig. 16, when IP is stored 800, a format conversion 1001 is performed. Format conversion 1001 includes "system encryption," so that the IP is unreadable on other systems. When a request is received, personalized encryption 1002 is applied, followed by dynamic encryption 1003 as the IP is transmitted over a network 804. (Encryptions 1002 and 1003 are configured so that they can be performed at essentially the same time, even though a temporary storage 802 is shown between them.) In the case of an IP link, format conversion 1001 is triggered by a user request. The result is stored in a temporary storage 812 (rather than a permanent storage in central data center 100, as is the case when the IP is stored, rather than linked). Personalized encryption 1002 and dynamic encryption 1003 is then applied before transmission over network 1003. (Temporary storage 814 is not necessarily different from temporary storage 802.) As the user downloads the IP, it is stored in user storage 714, where decryption of encryption algorithms 1003, 1002, and 1001 are performed, in the reverse order to which the encryption 1001, 1002, 1003 were applied. Decryption 1002 is performed from temporary storage 806 into a secure RAM 808, using SEC 710. The content of secure RAM 808 is displayed 810, after encryption 1001 is decrypted. Storage 806 and 808 are erased when the IP is no longer needed. User storage 714 is also erased under some conditions, such as when tampering is detected.

The registration, security, tagging (or watermarking), verification and use analysis programming produces a data trail logging the creation and use of a particular IP file by a user, thereby providing evidence of accountability in the event of an actual or attempted breach of the security features associated with the IP file. The registration, security, tagging verification and use analysis programming creates a permanent linking of a given copy of an IP file to a user requesting its creation and a user device used to create the copy in question. A permanent log of use of the IP file, in digital form, is stored in a separate memory storage area associated with the CCS programming and a copy of the log is periodically transferred to the central information data storage facility for storage and later analysis.

As illustrated in Figure 1, in one embodiment of the invention, security is enhanced through the use of an external authentication device [302A] affixed or connected to the user's device and a related identifier. Examples of suitable, currently available external authentication devices include parallel port software locks, iButtoms^{tm} (Dallas Semiconductor Corp. of Dallas, TX) and "Smart Cards" plugged in via a pc-card. In such embodiment, upon registering a user is physically given an identification code disk, card or button that is encoded with user identification information that is electronically readable when the code disk, card or button is inserted in an appropriate slot associated with the external authentication device. If the externally inserted electronic identification codes match the required codes for access, access to the encrypted files is allowed. If the externally inserted electronic identification codes do not match the required codes for access, access is denied.

In yet other embodiments of the present invention, user device 712 is a cellular or other wireless mobile telephone or wireless communications unit, for example, a cell phone with a wireless browser. With this type of user device, IP text can be downloaded directly to the wireless communications unit via the appropriate cellular or other type of wireless mobile network. In one embodiment, for example, the IP text is written text, pictures, or other visual content that is displayed in a browser window. In another embodiment, the IP text is music that is downloaded

directly to a cellular or other wireless mobile telephone. (One embodiment of a wireless telephone useful for this invention includes stereo headphones, or has a jack for such headphones.) The IP text undergoes dynamic encryption, as in other embodiments of the invention, and is decrypted, but not stored in the telephone or on other media. When a call is received, an audible or other type of indication is provided on the wireless telephone user device 712, and the music is interrupted when the call is answered. In yet another embodiment, music is selectively streamed to headphones or speakers, saved to a recording medium in the wireless telephone user device 712, or both. Telephone 712 is configured so that the music can be selectively saved via downloading to the recording medium when a call interrupts music being streamed for listening.

It will thus be seen that one or more embodiments of the present invention provide one or more IP consolidation features such as overall "mothership" IP encryption, open web selection and browser access, user screening ID and control, transfer verification, transaction purchase, attachment of secure rules, functionality and user allowances, auditing, and feedback and updating. While the present invention has been described with respect to specific embodiments, many modifications, variations, substitutions, and equivalents will be apparent to those skilled in the art. Accordingly, the invention is to be considered as limited only by the spirit and scope of the appended claims.

CLAIMS:

1. Apparatus for facilitating obtaining text of an IP, comprising:

a storage device having stored therein text of a plurality of IP, wherein the text includes electronically stored representations of at least one member of the group consisting of printed text works, movies, films, video presentations, television programming, music, audio works, audio presentations, radio programs, graphic material, art works, plays, operas, novels, writings, photographs, pictures, images, advertising copy, software, and portions and combinations thereof;

a processor connected to said storage device, said storage device further having stored therein a program for controlling said processor, said processor operative with the program to:

receive an IP selection request;

receive a user identification associated with the IP selection request; and

output encrypted text of the selected IP if the user identification and IP selection are valid utilizing a determined level of IP encryption.

2. Apparatus in accordance with Claim 1 wherein said processor is further operative with the program to purpose encode the text of the selected IP so as to limit a purpose which access by the user to the text is authorized.

3. Apparatus in accordance with Claim 1 wherein said processor is further operative with the program to encrypt the text of the selected IP utilizing an identifier associated with the user.

4. Apparatus in accordance with Claim 1 configured to electronically associate instructions provided by an IP supplier and relating to allowed

use, access, and pricing of IPs supplied by the supplier with the IPs supplied by the supplier.

5. Apparatus in accordance with Claim 4 further configured to receive use and access choices from the user associated with the IP selection request, and to use the instructions provided by the IP supplier and associated with the text of the selected IP to generate an electronic IP instruction file linked with the outputted text of the selected IP to regulate use of and access to the outputted text.

6. Apparatus in accordance with Claim 1 further configured to receive a determined level of security including the determined level of encryption from a supplier of the selected IP, the determined level of security being selected from a range of security levels.

7. Apparatus in accordance with Claim 1 wherein said determined level of IP encryption is based on at least an economic value of the selected IP.

8. Apparatus in accordance with Claim 1 wherein said processor is further operative with the program to encrypt the text of the selected IP utilizing a unique identification number associated with at least one of a user device and user data storage medium used to download or store the encrypted text of the selected IP.

9. Apparatus in accordance with Claim 1 wherein said processor is further operative with the program to cooperate with a user device being used to download and read a selected IP text, to restrict and regulate operation of the user device to limit reading and copying of the selected IP text.

10. Apparatus in accordance with Claim 9 wherein said processor is further operative with the program and the user device to determine whether IP texts downloaded to the user device have been used in an unauthorized manner, and to at least one of restrict or deny access to such IP texts when such unauthorized use is determined to have occurred.

11. Apparatus in accordance with Claim 10 wherein said processor is further operative with the program and the user device to cause the user device to permanently erase an IP text from memory of the user device when such unauthorized use is determined to have occurred.

12. Apparatus in accordance with Claim 9 wherein said processor is further operative with the program and the user device to cause the user device to at least one of temporarily restrict or deny access to an IP text in a memory of the user device.

13. Apparatus in accordance with Claim 1 further comprising at least one local unit communicatively coupled to said processor, said local unit comprising a memory for storing, in electronic form, information transmitted to said unit from said processor, and a local unit processor for controlling transfer of information stored in said unit to electronic storage media of system users, said local unit configured to encrypt the information when the information is to be transferred to the electronic storage media, said local unit configured to encrypt the information utilizing a determined level of information encryption.

14. Apparatus in accordance with Claim 13 wherein the information stored on a user's storage medium comprises a personal signature code number and serial number.

15. Apparatus in accordance with Claim 13 further configured to receive a determined level of security, including the determined level of encryption, from a supplier of the selected IP, the determined level of security being selected from a range of security levels.

16. Apparatus in accordance with Claim 15 wherein said determined level of information encryption and security is based upon at least an economic value of the selected IP

17. Apparatus in accordance with Claim 16 wherein the level of information encryption is not equal to the level of the IP encryption.

18. Apparatus in accordance with Claim 13 wherein the memory of the local unit is configured to regulate the use of information stored therein in encrypted or decrypted form so long as an electronic connection exists between the local unit and a user data storage medium, and to permanently erase selected files from the memory of the local unit whenever the electronic connection is no longer active.

19. Apparatus in accordance with Claim 1 wherein said processor is further operative with the program to determine attempted unauthorized access to the output copies of the text of an IP.

20. Apparatus in accordance with Claim 1 further configured to receive information identifying an establishment from which the IP selection request originated, and to tag the encrypted text of the selected IP with an identification of the establishment.

Auditing

21. Apparatus in accordance with Claim 20 wherein said processor is further operative with the program to utilize the tag to trace IP text to an establishment.

22. Apparatus in accordance with Claim 1 wherein said processor is further operative with the program to determine and store information relating to usage of the encrypted text by the user and which will cause said usage information to be encrypted and stored in a segregated section of the memory associated with the user storage medium or local unit and which will cause said usage information in encrypted form to be transmitted to the central data storage facility when said local unit and/or user data storage unit are later electronically connected to the central data storage facility.

23. A method for operating a computer to obtain text of an IP, comprising:

inputting into the computer an IP selection request;

inputting into the computer a user identification associated with the IP selection request;

inputting a unique identification number associated with at least one of a user device or user storage medium being used to at least one of download or store an encrypted copy of a selected IP corresponding to the IP selection request,

generating a unique encryption algorithm and a corresponding decryption algorithm, using the user identification and at least one member of the group consisting of: (a) a digital identification number associated with at least one of the user device or a user data storage medium, and (b) a digital identification associated with the selected IP;

communicating encrypted text of the selected IP to the user device for storage on the user storage medium using the unique encryption algorithm for encryption, wherein text of the selected IP comprises electronic representations of at least one member of the group consisting of: printed text works, movies, films, video presentations, television programming, music, audio works, audio presentations, radio programs, graphic material, art work, plays, operas, novels, writings, photographs, pictures, images, advertising copy, software, and portions and combinations thereof;

generating a header associated with the encrypted text of the selected IP file that contains, in digital form, a user identification, an identification of at least one of the user device or user data storage medium one which the encrypted digital copy of the selected IP is to be stored, a usage authorization indication, and an electronic address of the corresponding decryption algorithm;

validating the user identification, authorization indication, and IP selection, after the unique encryption algorithm and corresponding decryption algorithm have been generated; and

decrypting the digitally encrypted text of the selected IP using the corresponding decryption algorithm, conditioned upon said validation.

24. A method in accordance with Claim 23 wherein inputting to the computer an IP selection comprises the steps of:

selecting at least one of the IP in the memory storage; and

selecting the text of at least a portion of each the selected IP.

25. A method in accordance with Claim 24 wherein inputting to the computer an IP selection further comprises the steps of:

determining if more than one portion of an IP is selected; and

if more than one portion of an IP is selected, then combining the selected portions.

26. A method in accordance with Claim 23 further comprising the step of generating tracking information corresponding to the selected portions and the selected IP.

27. A method in accordance with Claim 23 further comprising the step of purpose-encoding the encrypted text of the selected IP so as to limit a purpose for which access by the user to the text is authorized.

28. A method in accordance with Claim 23 further comprising the step of determining a level of IP encryption.

29. A method in accordance with Claim 28 wherein determining a level of IP encryption comprises the step of selecting at least one of a plurality of levels of encryption code.

30. A method in accordance with Claim 29 wherein the step of generating the unique encryption algorithm and corresponding decryption algorithm comprises the step of obtaining the unique encryption algorithm and corresponding decryption algorithm from an independent key supplier.

31. A method in accordance with Claim 30 further comprising the step of storing copies of the obtained encryption and decryption algorithms in a memory of a central information storage facility for later use.

32. A method in accordance with Claim 30 wherein the at least one of the user device or user storage medium has a write once, read many times (WORM) memory unit contained therein, the WORM memory being configured to become inoperable when unauthorized access thereto is attempted;

said method further comprising the step of storing a digital representation of the unique encryption algorithm and of the corresponding decryption algorithms in the WORM memory for later use.

33. A method for operating a processor communicatively coupled to a network to obtain text of an IP, the network being coupled to a memory storage having stored therein text of a plurality of IP wherein the text of an IP comprises electronic representations of at least one member of the group consisting of: printed text works, movies, films, video presentations, television programming, music, audio works, audio presentations, radio programs, graphic material, art work, plays, operas, novels, writings, photographs, pictures, images, advertising copy, computer games, video games, software, and portions and combinations thereof; said method comprising:

determining an IP selection request;

inputting into the processor the IP selection request;

inputting into the processor a user identification associated with the IP selection request;

inputting a unique identification number associated with at least one of a user device or a user data storage medium in which an encrypted electronic copy of the text of the selected IP is to be stored;

outputting encrypted text of the selected IP if the user identification and IP selection request are valid utilizing a determined level of encryption.

34. A method in accordance with Claim 33 wherein determining the IP selection request comprises the steps of:

reviewing the IP in the memory storage;

selecting the text of at least a portion of at least one of the IP in the memory storage.

35. A method in accordance with Claim 33 wherein determining the IP selection request comprises the steps of:

selecting at least one IP in the memory storage;

selecting the text of at least a portion of each selected IP in the memory storage;

combining the selected portions of the selected IP.

36. A method in accordance with Claim 35 wherein selecting the text of at least a portion of each selected IP in the memory storage comprises the step of selecting the entire IP.

37. A method in accordance with Claim 35 wherein selecting the text of at least a portion of each selected IP in the memory storage comprises the step of selecting at least one word of each selected IP.

38. A method in accordance with Claim 35 wherein selecting the text of at least a portion of each selected IP in the memory storage comprises the step of selecting at least one section of each selected IP.

39. A method in accordance with Claim 35 further comprising the step of generating tracking information corresponding to the selected IP in the memory storage.

40. A method in accordance with Claim 39 further comprising the step of generating tracking information corresponding to the selected portions of the selected IP.

41. A method in accordance with Claim 39 further comprising the step of generating tracking information relating to usage of selected IP relative to a retail establishment location.

42. A method in accordance with Claim 33 further comprising the step of purpose encoding the encrypted text of the selected IP so as to define a purpose of use which access by the user to the text is authorized.

43. A method in accordance with Claim 33 further comprising the step of determining unauthorized access to the encrypted text.

44. A method in accordance with Claim 33 wherein the step of outputting encrypted text of the IP comprises the step of generating a unique encryption algorithm and a corresponding decryption algorithm using the user identification and at least one member of the group consisting of: (a) the digital identification number associated with at least one of the user device or the user data storage medium, and (b) a digital identification associated with the selected IP.

45. A method in accordance with Claim 44 wherein the step of generating the unique encryption algorithm and corresponding decryption algorithm comprises the step of obtaining the unique encryption algorithm and corresponding decryption algorithm from an independent key supplier.

46. A method in accordance with Claim 45 further comprising the step of storing copies of the obtained encryption and decryption algorithms in a memory of a central information storage facility for later use.

47. A method in accordance with Claim 46 wherein the at least one of the user device or user storage medium has a nonvolatile memory contained therein, the nonvolatile memory being configured to become inoperable when unauthorized access thereto is attempted;

said method further comprising the step of storing a digital representation of the unique encryption algorithm and of the corresponding decryption algorithms in the nonvolatile memory for later use.

48. A method in accordance with Claim 45 wherein said step of outputting encrypted text of the selected IP comprises the step of using the unique encryption algorithm to encrypt a digital copy of the selected IP as the copy is being made and downloaded to a user's data storage medium.

49. A method in accordance with Claim 33 further comprising the step of outputting, in association with the encrypted copy of the selected IP, information from which at least one member of the group consisting of: (a) access authorization codes, (b) usage authorization codes, and (c) a decryption algorithm for decrypting the encrypted text of the IP can be determined.

50. Apparatus for facilitating obtaining text of an IP, wherein the text includes electronic representations of at least one member of the group consisting of: printed text works, movies, films, video presentations, television programming, music, audio works, audio presentations, radio programs, graphic material, art work,

plays, operas, novels, writings, photographs, pictures, images, advertising copy, computer games, video games, software, and portions and combinations thereof; said apparatus comprising:

a storage device having stored therein text of a plurality of IP;

a processor connected to said storage device, said storage device further having stored therein a program for controlling said processor, said processor operative with the program to:

receive an IP selection request;

receive a user identification associated with the IP selection request; and

dynamically encrypt text of the selected IP as the text is output from the apparatus, using an encryption algorithm.

51. Apparatus in accordance with Claim 50 wherein said apparatus is configured to communicate via a network connection selected from the group consisting of: electronic cable connections, wired connections, commercial telephone connections, commercial cable network connections, cellular and other wireless mobile network connections, infrared connections, microwave connections, radio wave and other wireless connections, television wave connections, local device generated infrared signal connections, laser connections, and connections allowing for transfer of data in digital form from one point to another, and combinations thereof.

52. Apparatus in accordance with Claim 50 wherein said processor is further operative with the program to purpose encode the text of the selected IP so as to limit a purpose which access by the user to the text is authorized.

53. Apparatus in accordance with Claim 50 wherein said processor is further operative with the program to access encode the text of the selected IP so as to regulate access to the encrypted selected IP.

54. Apparatus in accordance with Claim 50 wherein said processor is further operative with the program to receive instructions from an IP supplier relating to permissible use and access to IP texts supplied by the IP supplier, to associate said instructions with corresponding IP texts, to receive usage and access choices from the user in conjunction with the IP request, and to use said instructions associated with said texts to generate an electronic instruction data file with the requested IP, the instruction data file having instructions readable by a user device to regulate use of and access to the selected IP after it is downloaded and stored on a user device or storage medium.

55. Apparatus in accordance with Claim 50 further comprising at least one local unit communicatively coupled to said processor, said local unit comprising a memory for storing, in electronic form, information transmitted to said unit from said processor, and a local unit processor for controlling transfer of information stored in said unit to electronic storage media of system users, said local unit configured to encrypt the information when the information is to be transferred to the electronic storage media, said local unit configured to encrypt the information utilizing a determined level of encryption.

56. Apparatus in accordance with Claim 50 further configured to:

communicate, to a key generation service via a communication network, information relating to at least one member of the group consisting of: (a) a digital identification number associated with at least one of a user device or a user data storage medium, and (b) a digital identification associated with the selected IP; and

receive an encryption algorithm corresponding to the selected level of encryption from the key generation service via the communication network.

57. Apparatus in accordance with Claim 50 wherein said processor is further operative with the program to encrypt the text of the selected IP utilizing a

unique identification associated with at least one of a user device or a user data storage medium being used to download or store the encrypted text.

58. Apparatus in accordance with Claim 50 wherein said processor is further operative with the program to restrict and regulate the operation of a user device and user data storage medium that receives the encrypted text of the IP by restricting ability of the user device to perform at least one operation selected from the list consisting of: (a) printing of the IP, (b) copying of the IP, and (c) permanently storing decrypted text of the IP to another storage medium; and to cause selected information to be permanently erased from memory associated with the user device when a connection between the user device and a user data storage medium on which encrypted data files are stored is broken.

59. Apparatus in accordance with Claim 58 wherein said processor is further operative with the program to periodically change encryption and decryption algorithms and formulae associated with a copy of a selected IP text as a function of at least one of passage of time and use of the selected IP text.

60. Apparatus in accordance with Claim 50 wherein said processor is further operative with the program to receive a user-provided access code and generate a system access code, and to restrict and regulate access to encrypted text of the IP by encrypting text of a selected IP using the user-provided access code and generated system access code so that both are required to decrypt the text of the selected IP.

61. Apparatus in accordance with Claim 50 wherein said processor is further configured to receive, in encrypted form, usage and access information relating to a user device for reading electronic texts of IP when the user device is communicating with the apparatus.

62. A method for distributing intellectual properties (IP's) in digital form, said method comprising the steps of:

obtaining information from a plurality of publishers, the information being selected from the group consisting of IP's, links to IP's, and combinations of IP's and links to IP's;

encrypting IP's obtained from the publishers;

storing the encrypted IP's in a storage facility;

indexing the stored, encrypted IP's and links to IP's;

transmitting at least selected portions of the index to customers via a network;

receiving selective electronic requests for the IP's from the customers via the network;

transmitting requested IP's to customers in response to their electronic requests, including dynamically encrypting the IP's; and

recording transaction fees for the transmissions.

63. A method in accordance with Claim 62 further comprising the step associating an electronic signature personalized to a requesting customer with the IP's transmitted to the requesting customer in response to the requesting customer's request.

64. A method in accordance with Claim 62 further comprising the steps of:

maintaining an accounting of IP's transmitted utilizing the electronic signatures;

billing customers for IP's transmitted in response to their requests, and crediting publishers for publisher's IP's transmitted to customers.

65. A method in accordance with Claim 62 wherein the IP's include compilations of portions of IP.

66. A method in accordance with Claim 62 wherein the selective electronic requests for the IP's include requests for information available as IP links, and wherein transmitting requested IP's to customers, including dynamically encrypting the IP's comprises the steps of:

retrieving a copy of an IP corresponding to an IP link;

processing the copy of the IP so as to make it readable on a device of the requesting customer; and

dynamically encrypting the processed copy of the IP.

67. A method in accordance with Claim 62 wherein transmitting the requested IP's comprises the step of transmitting instructions to corresponding requesting customers for acquiring the requested IP's.

68. A method in accordance with Claim 67 wherein transmitting instructions for acquiring the requested IP's includes transmitting one or more instruction selected from the set consisting of: how to buy the requested IP's, how to rent the requested IP's, how to pay for the requested IP's, and how to read the requested IP's.

69. A method in accordance with Claim 67 wherein transmitting instructions for acquiring the requested IP's includes transmitting hardware and software requirements for reading the requested IP's.

70. A method in accordance with Claim 62 further comprising the step of encrypting at least one of the requested IP's with a customer-specific algorithm to allow the at least one requested IP to be read on more than one user device having an electronic identification associated with the customer.

71. A method in accordance with Claim 62 wherein transmitting the encrypted IP's comprises further encrypting the selected, encrypted IP's in accordance with a level of encryption selected by a publisher of the IP.

72. A method in accordance with Claim 62 wherein said step of receiving selective electronic requests for the IP's from customers via a network comprises receiving selective electronic requests from end users of the IP's via a public network.

73. A method in accordance with Claim 72 wherein the public network comprises the Internet.

74. A method in accordance with Claim 62 wherein said step of receiving selective electronic requests for the IP's from customers via a network comprises receiving selective electronic requests transmitted from network terminals located in retail establishments.

75. A method in accordance with Claim 74 wherein said step of receiving selective electronic requests transmitted network terminals located in retail establishments comprises receiving electronic requests sent by at least one member of the group consisting of: employees, proprietors, and combinations thereof, of the retail establishments.

76. A method in accordance with Claim 75 and further comprising the steps of maintaining an accounting of IP's transmitted utilizing electronic signatures personalized to requesting customers, and crediting IP sales to the retail establishments.

77. A method for uniform storing, encryption, and electronic distribution of digitized intellectual property (IP) comprising:

creating an electronic index of a collection of digitally-stored intellectual property;

generating authorization information unique to a user when the user accesses a central digital IP storage facility;

regulating access to digitized IP at the central storage facility to authorized users;

transmitting digitized IP from the digital storage facility to authorized users, including dynamically encrypting the digitized IP;

generating a unique registration number for each transmitted digitized IP associating the authorized user downloading the digitized IP and a user storage device receiving the transmitted IP.

78. A method in accordance with Claim 77 further comprising the steps of recording and storing, as a data file stored on a central data storage file, information identifying and relating users requesting transmission of a digitized IP with the requested, digitized IP.

79. A method in accordance with Claim 78 further comprising the step of providing access to the information identifying and relating the users to the transmitted digitized IP only to selected individuals.

80. A method in accordance with Claim 78 further comprising the step of indexing the information identifying and relating the users to the transmitted digitized IP in accordance with one member of the group consisting of: date, user, and digitized IP identification.

81. A method in accordance with Claim 78 wherein dynamically encrypting the transmitted IP comprises generating and attaching a time code tag indicating at least one of: (a) a length of time that access to a transmitted IP will be allowed, and (b) a time period that access to a transmitted IP will be allowed.

82. A method in accordance with Claim 78 wherein dynamically encrypting the IP comprises changing assigned encryption and decryption algorithms

for a transmitted IP stored in a user device or on a storage medium, said changing occurring in accordance with a function of at least one of use of the transmitted, stored IP, and time.

83. A wireless telephone communications device configured to:

communicate over a wireless network;

receive, decrypt, and play an interruptible, encrypted stream of music; and

to interrupt the stream of music when a call is received and answered.

84. A device in accordance with claim 83 further having storage for recording the stream of music.

85 A device in accordance with Claim 84 further configured to selectively record music into storage when the stream of music is interrupted.

86. A device in accordance with Claim 83 configured for stereo playing of the music stream.

87. A method for transmitting IP to a mobile station comprising the steps of:

obtaining information from a plurality of publishers, the information being selected from the group consisting of IPs, links to IPs, and combinations of IPs and links to IPs,

receiving selective electronic requests for the IPs from customers via a mobile, wireless network;

transmitting requested IPs to customers in response to their requests via the mobile, wireless network, including dynamically encrypting the IPs; and

recording transaction fees for the transmissions.

88. A method in accordance with Claim 87 wherein the IPs comprise at least one member of the set consisting of movies, films, video presentations, television programming, music, audio works, audio presentations, and radio programs; and wherein the step of transmitting requested IPs to customers comprises the step of synchronously transmitting requested IPs to customers.

89. A method in accordance with Claim 87 wherein the IPs comprise at least one member of the set consisting of movies, films, video presentations, television programming, music, audio works, audio presentations, and radio programs; and wherein the step of transmitting requested IPs to customers comprises the step of asynchronously transmitting requested IPs to customers.

90. A method in accordance with Claim 87 wherein the wireless network is a cellular network.

91. A method in accordance with Claim 87 further comprising the step of transmitting instructions with the requested IPs containing limitations pertaining to the use of the requested IPs by the customers.

92. A method in accordance with Claim 91 wherein the instructions include instructions pertaining to the autoerasure of the requested IPs.

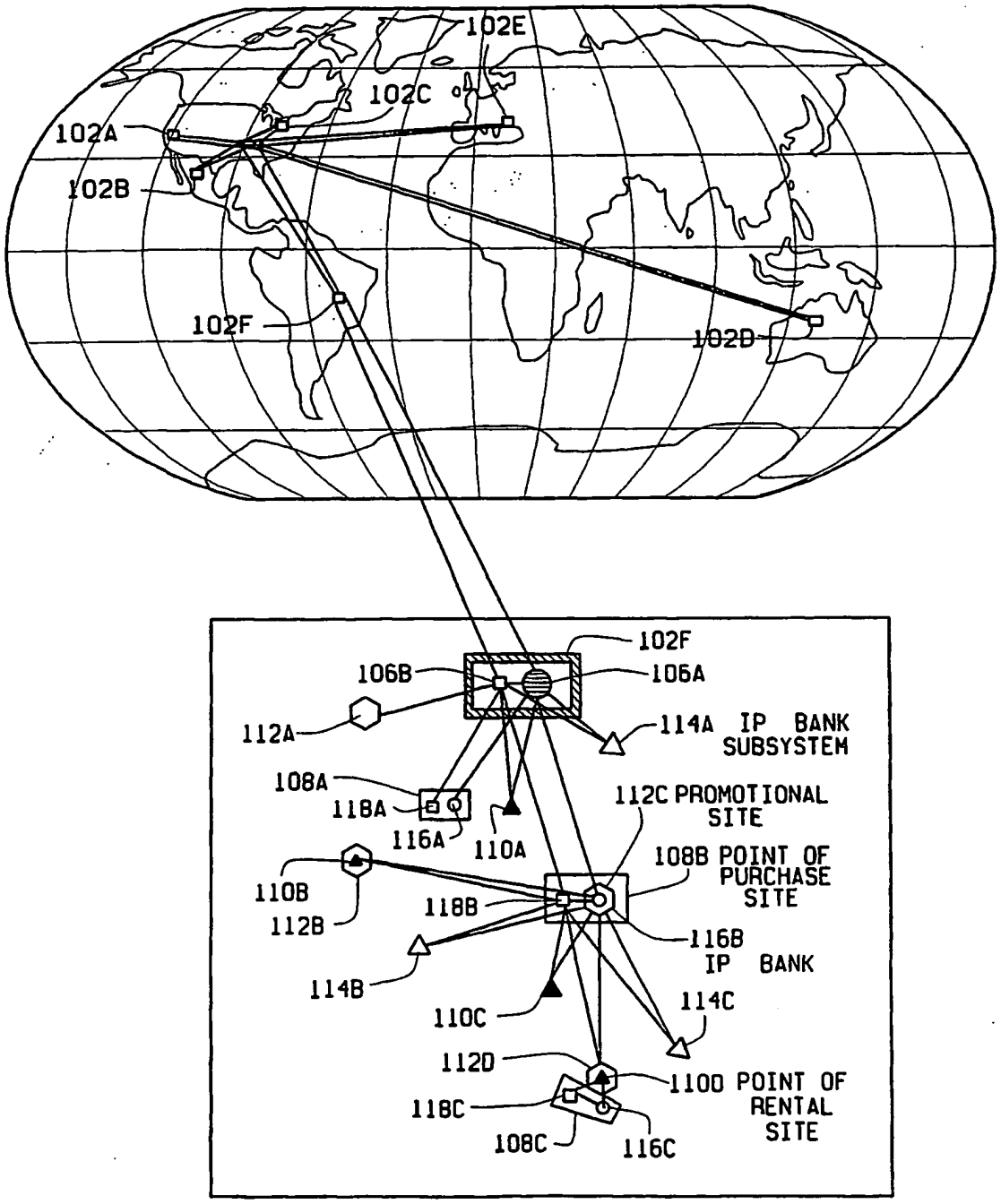

102E
102C
102A
102B
102F
102D
102F
106B
106A
112A
114A IP BANK SUBSYSTEM
108A
118A
116A
110A
112C PROMOTIONAL SITE
108B POINT OF PURCHASE SITE
110B
112B
118B
116B IP BANK
114B
110C
114C
112D
110D POINT OF RENTAL SITE
118C
108C
116C

FIG. 1

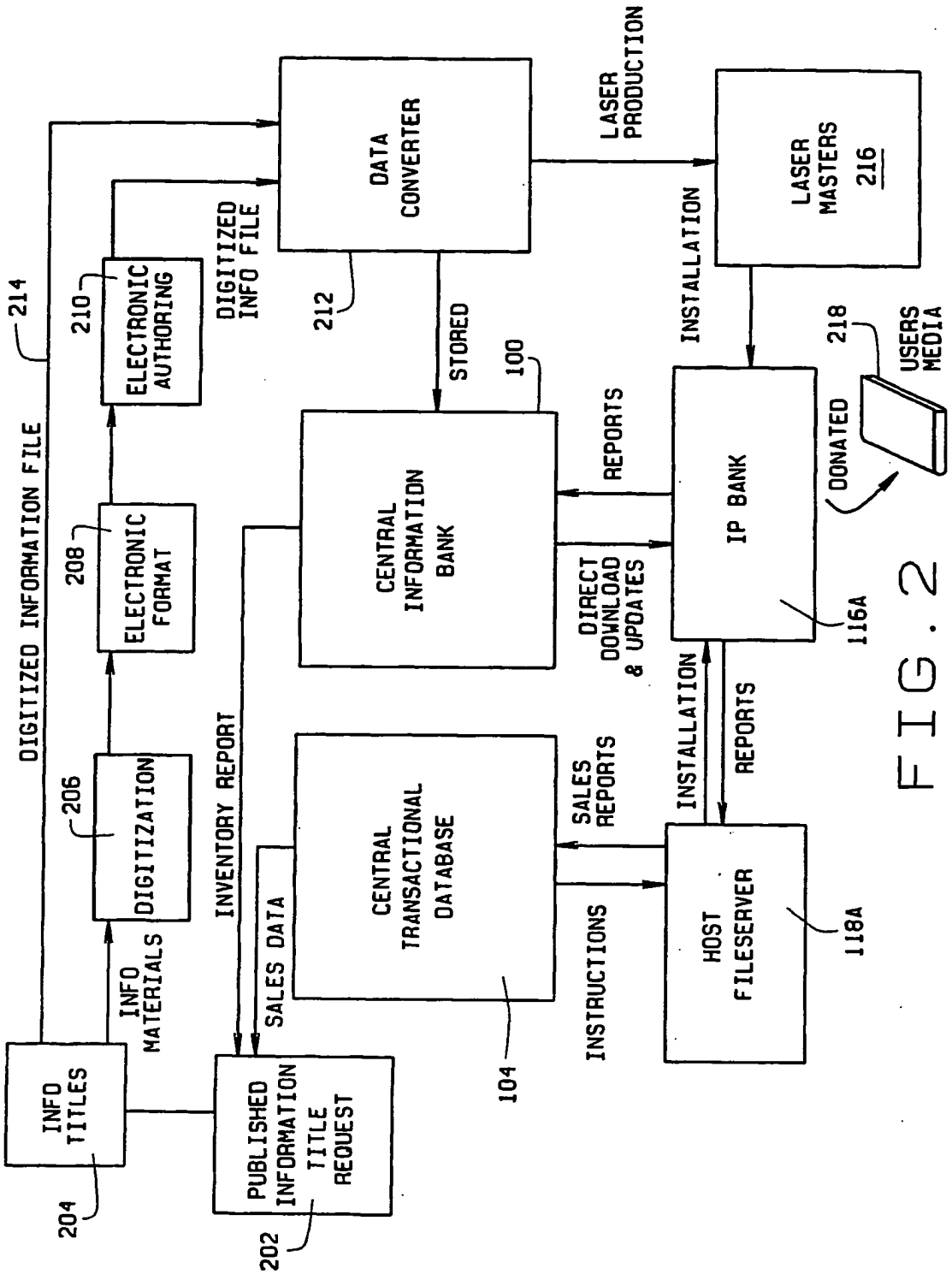

INFO TITLES
204
PUBLISHED INFORMATION TITLE REQUEST
202
INFO MATERIALS
DIGITIZATION
206
ELECTRONIC FORMAT
208
ELECTRONIC AUTHORING
210
DIGITIZED INFO FILE
212
DIGITIZED INFORMATION FILE
214
DATA CONVERTER
LASER PRODUCTION
LASER MASTERS 216
INSTALLATION
STORED
100
CENTRAL INFORMATION BANK
REPORTS
IP BANK
116A
DIRECT DOWNLOAD & UPDATES
DONATED
218
USERS MEDIA
INVENTORY REPORT
SALES DATA
CENTRAL TRANSACTIONAL DATABASE
104
SALES REPORTS
INSTALLATION
REPORTS
INSTRUCTIONS
HOST FILESERVER
118A

FIG. 2

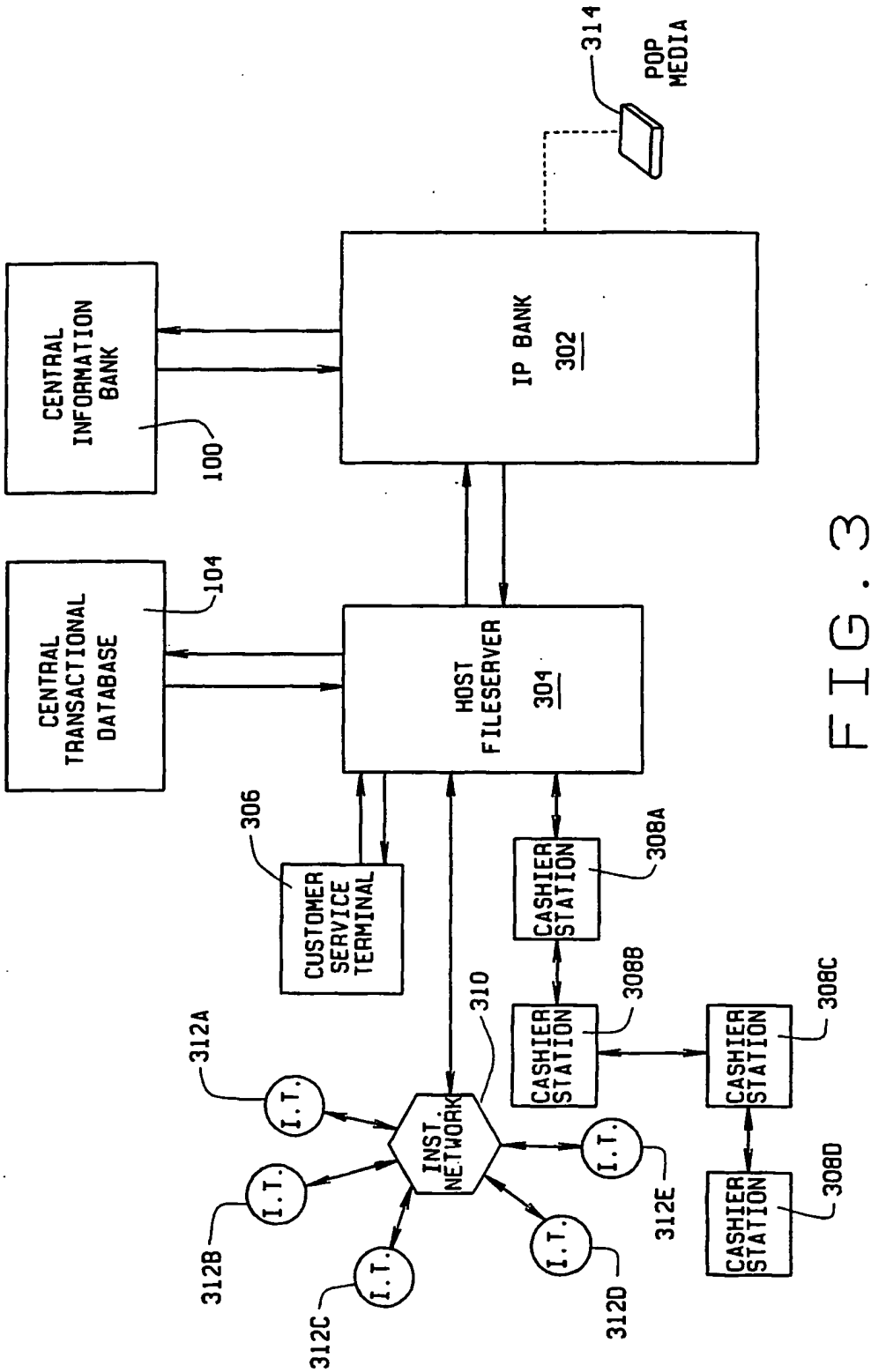

314
POP MEDIA
CENTRAL INFORMATION BANK
100
IP BANK
302
CENTRAL TRANSACTIONAL DATABASE
104
HOST FILESERVER
304
306
CUSTOMER SERVICE TERMINAL
CASHIER STATION
308A
CASHIER STATION
308B
CASHIER STATION
308C
CASHIER STATION
308D
310
INST. NETWORK
312A
312B
312C
312D
312E
I.T.

FIG. 3

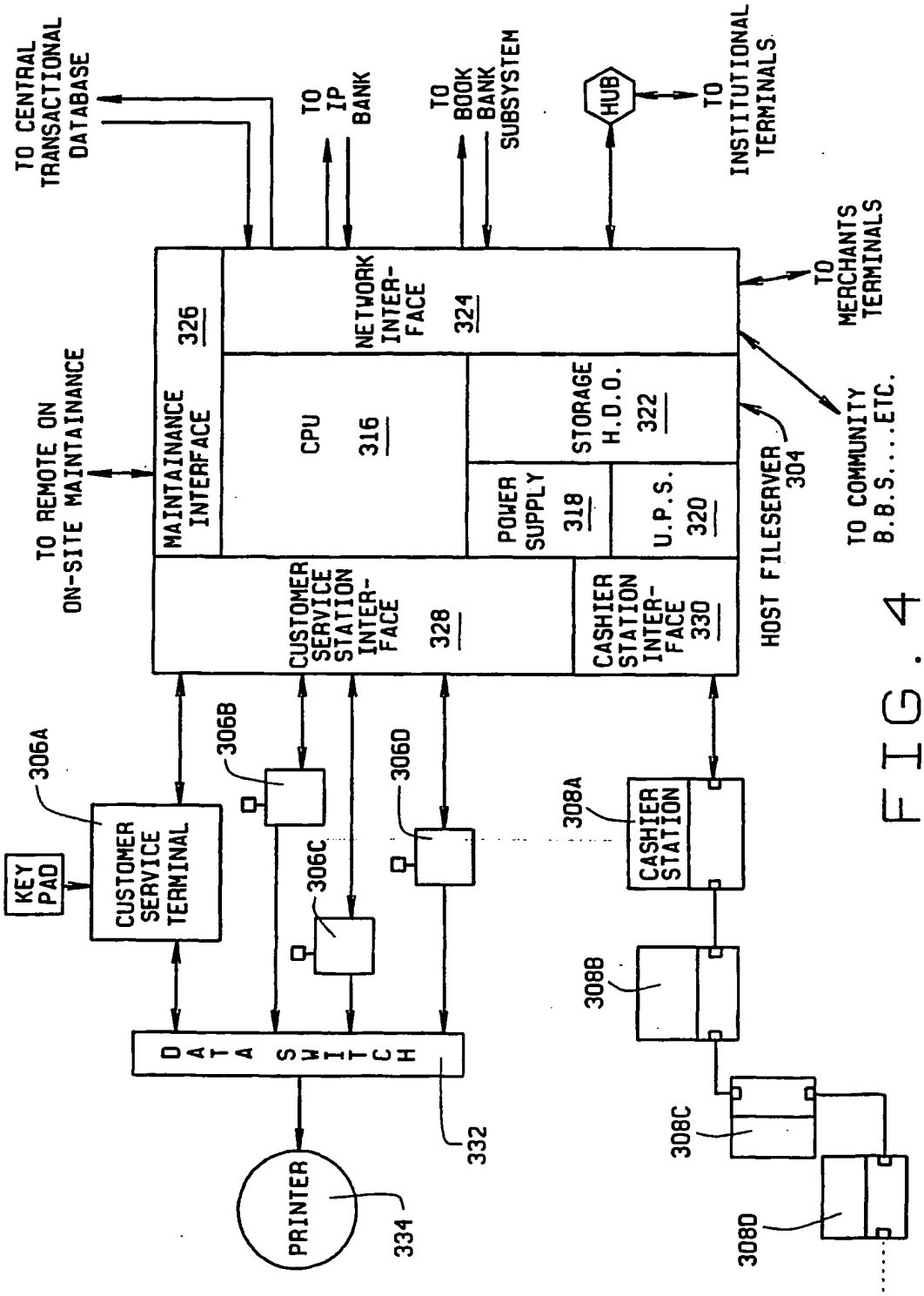

TO CENTRAL TRANSACTIONAL DATABASE
TO IP BANK
TO BOOK BANK SUBSYSTEM
HUB
TO INSTITUTIONAL TERMINALS
TO MERCHANTS TERMINALS
TO COMMUNITY B.B.S....ETC.
HOST FILESERVER 304
TO REMOTE ON ON-SITE MAINTAINANCE
MAINTAINANCE INTERFACE 326
NETWORK INTER-FACE 324
CPU 316
STORAGE H.D.O. 322
POWER SUPPLY 318
U.P.S. 320
CUSTOMER SERVICE STATION INTER-FACE 328
CASHIER STATION INTER-FACE 330
306A
306B
306C
306D
KEY PAD
CUSTOMER SERVICE TERMINAL
DATA SWITCH
332
PRINTER
334
308A
CASHIER STATION
308B
308C
308D

FIG. 4

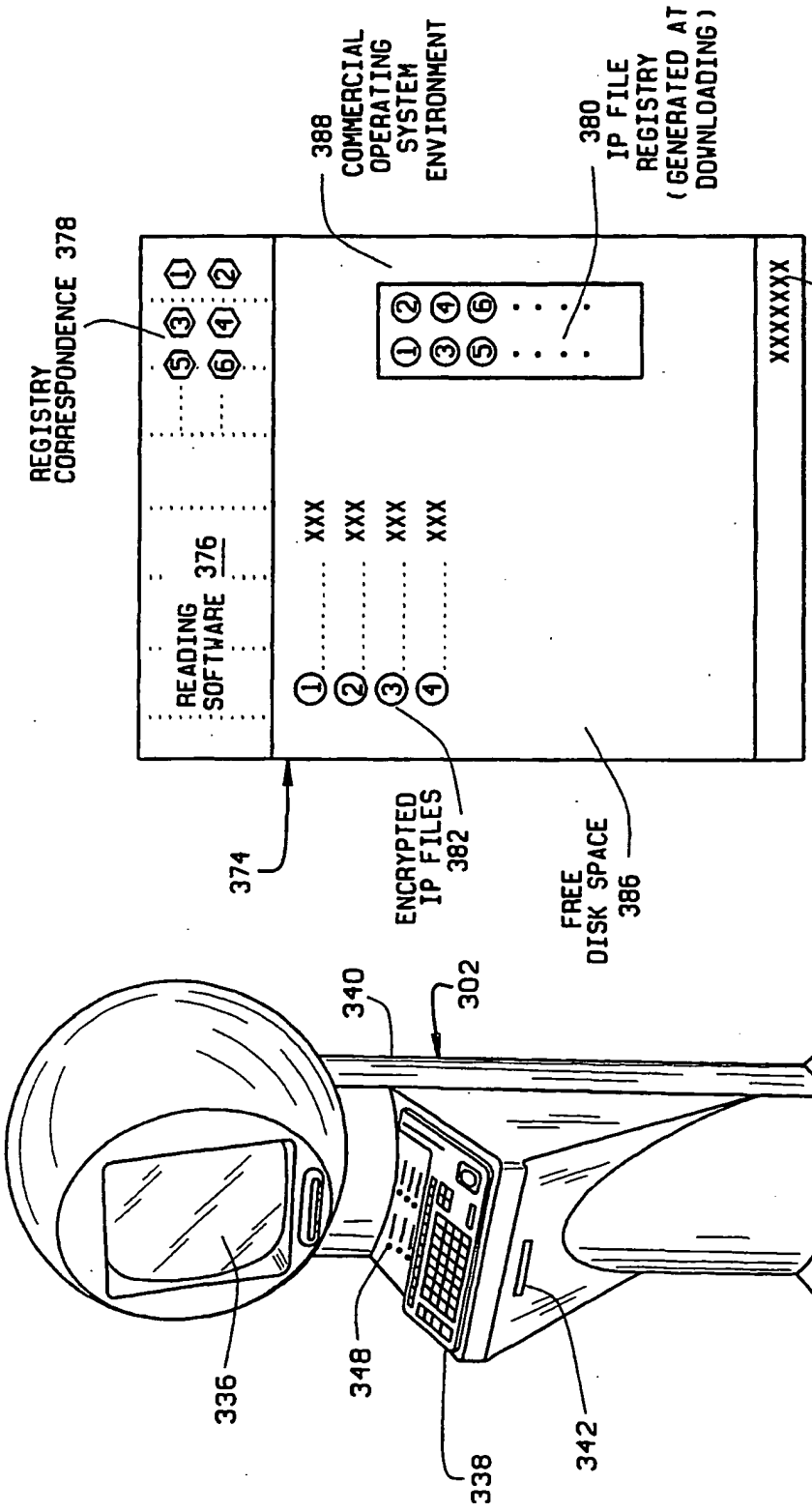

340
302
344
336
348
338
342
346

FIG. 5

REGISTRY CORRESPONDENCE 378
READING SOFTWARE 376
388 COMMERCIAL OPERATING SYSTEM ENVIRONMENT
380 IP FILE REGISTRY (GENERATED AT DOWNLOADING)
XXXXXXX
NON-ERASABLE PERMANENT MARKED SERIAL NO. 384
XXX
374
ENCRYPTED IP FILES 382
FREE DISK SPACE 386

FIG. 7

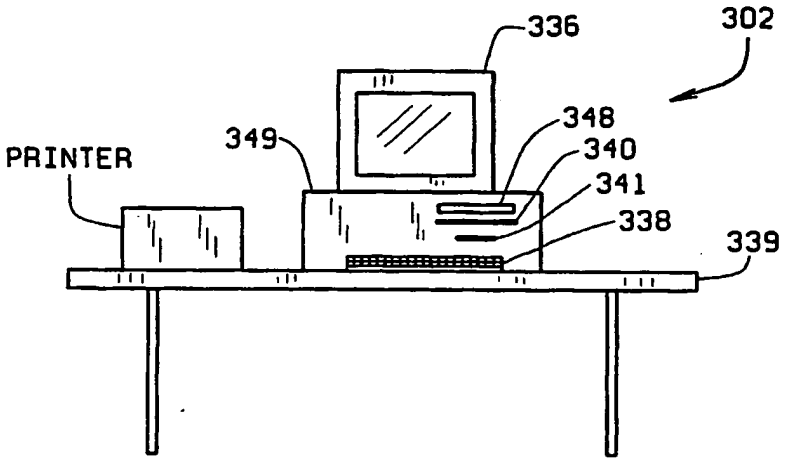

336
302
348
349
340
PRINTER
341
338
339

FIG. 5A

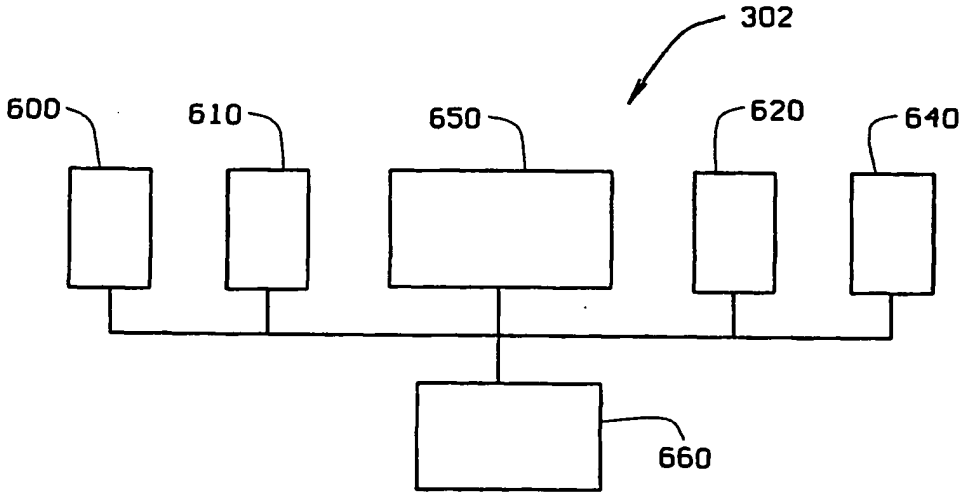

302
600
610
650
620
640
660

FIG. 13

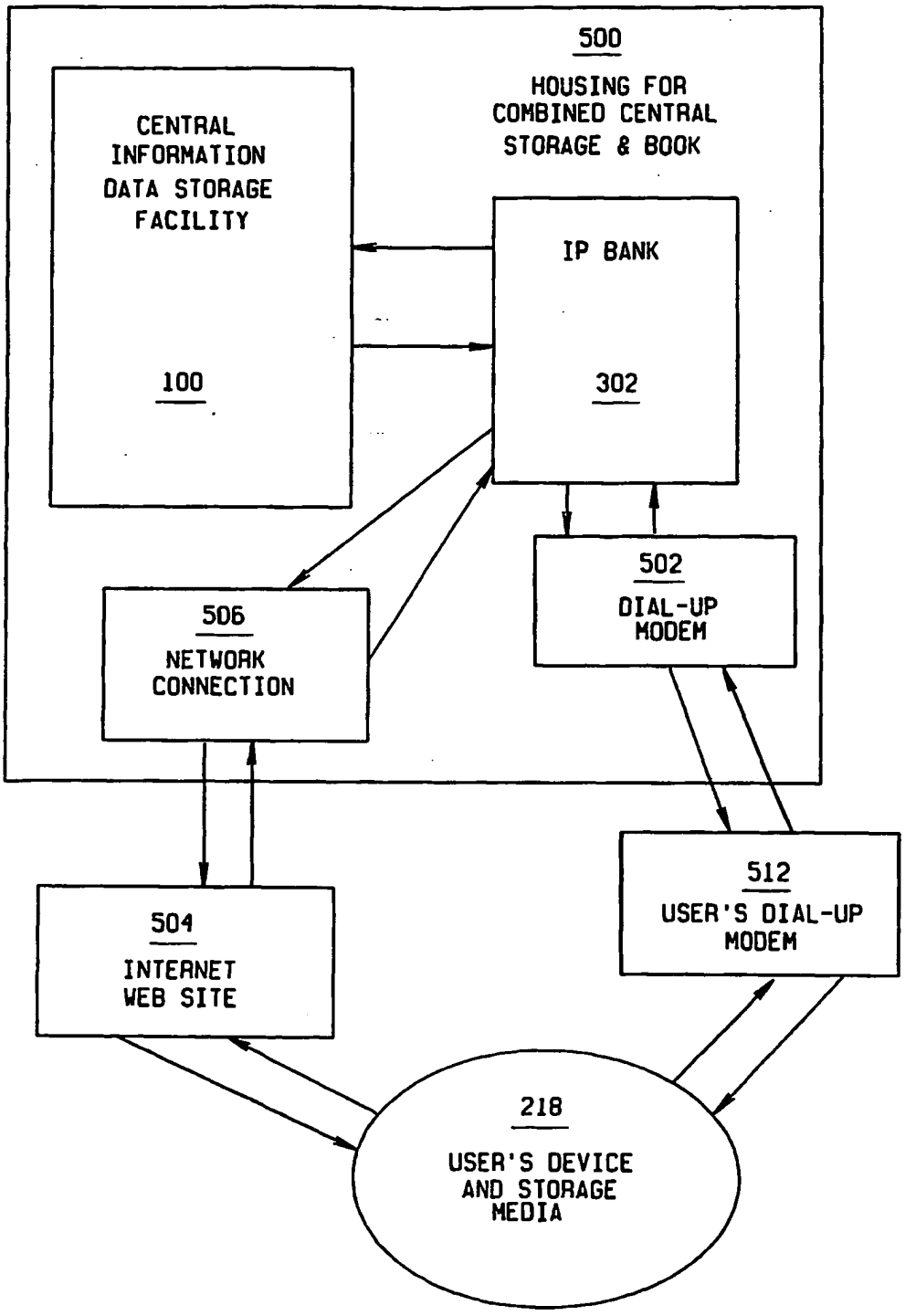

500
HOUSING FOR COMBINED CENTRAL STORAGE & BOOK
CENTRAL INFORMATION DATA STORAGE FACILITY
100
IP BANK
302
502
DIAL-UP MODEM
506
NETWORK CONNECTION
504
INTERNET WEB SITE
512
USER'S DIAL-UP MODEM
218
USER'S DEVICE AND STORAGE MEDIA

FIG. 5B

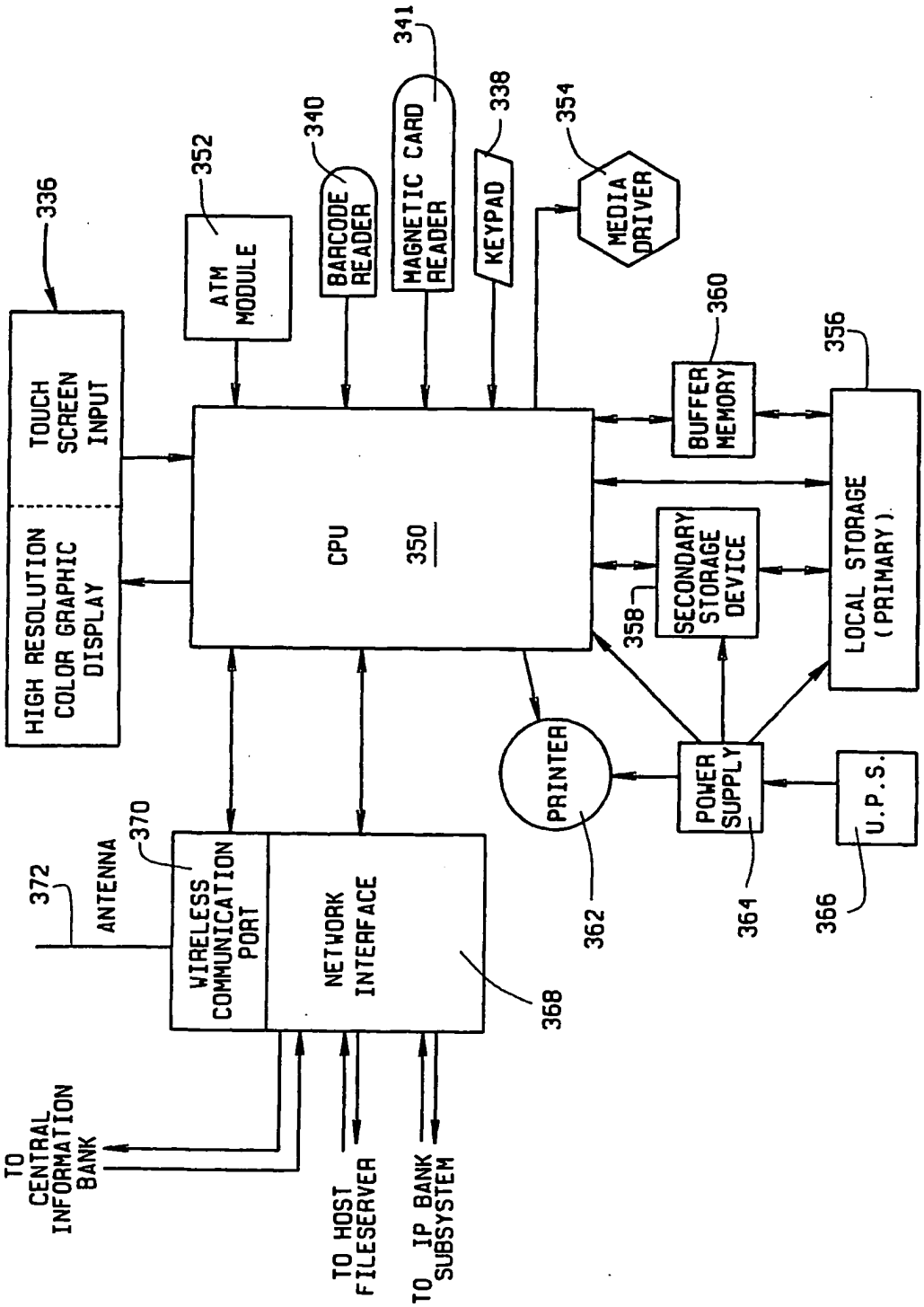

336
TOUCH SCREEN INPUT
HIGH RESOLUTION COLOR GRAPHIC DISPLAY
352
ATM MODULE
340
BARCODE READER
341
MAGNETIC CARD READER
338
KEYPAD
354
MEDIA DRIVER
CPU
350
360
BUFFER MEMORY
356
LOCAL STORAGE (PRIMARY)
358
SECONDARY STORAGE DEVICE
PRINTER
362
POWER SUPPLY
364
U.P.S.
366
370
372
ANTENNA
WIRELESS COMMUNICATION PORT
NETWORK INTERFACE
368
TO CENTRAL INFORMATION BANK
TO HOST FILESERVER
TO IP BANK SUBSYSTEM

FIG. 6

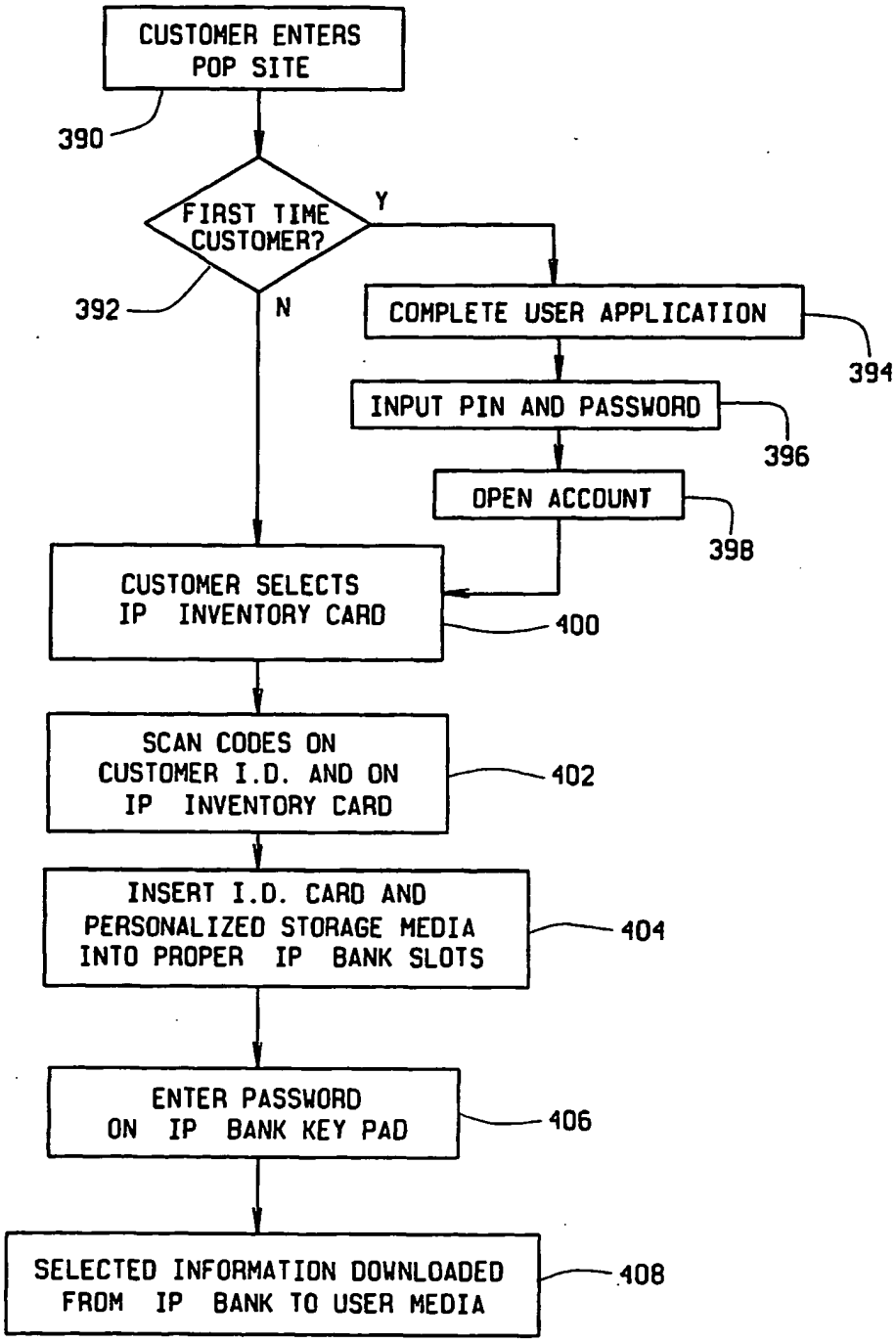

CUSTOMER ENTERS POP SITE
390
FIRST TIME CUSTOMER?
Y
N
392
COMPLETE USER APPLICATION
394
INPUT PIN AND PASSWORD
396
OPEN ACCOUNT
398
CUSTOMER SELECTS IP INVENTORY CARD
400
SCAN CODES ON CUSTOMER I.D. AND ON IP INVENTORY CARD
402
INSERT I.D. CARD AND PERSONALIZED STORAGE MEDIA INTO PROPER IP BANK SLOTS
404
ENTER PASSWORD ON IP BANK KEY PAD
406
SELECTED INFORMATION DOWNLOADED FROM IP BANK TO USER MEDIA
408

FIG. 8

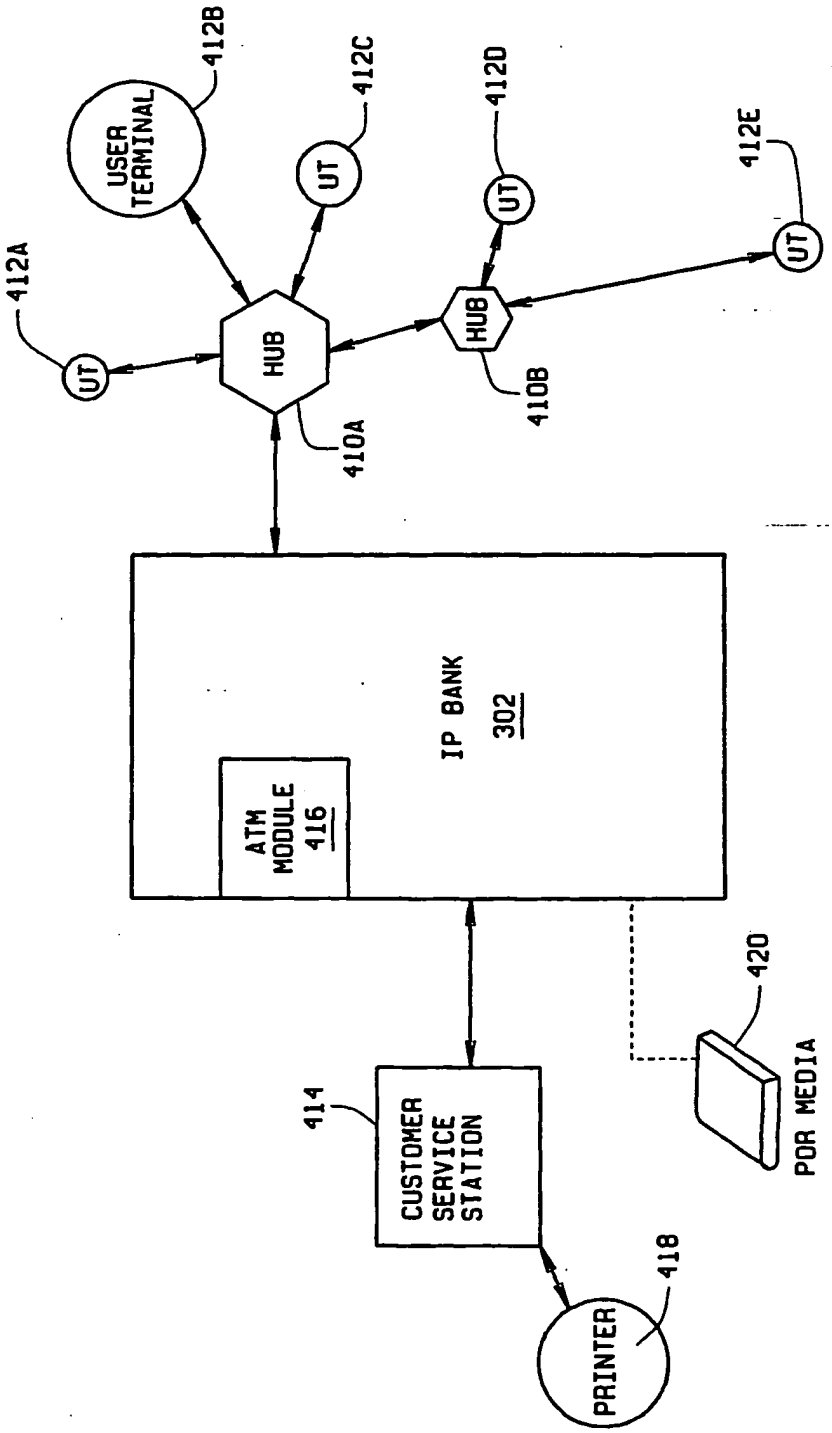

USER TERMINAL
412B
412C
412D
412E
412A
UT
HUB
410A
410B
IP BANK
302
ATM MODULE 416
420
POR MEDIA
414
CUSTOMER SERVICE STATION
PRINTER
418

FIG. 9

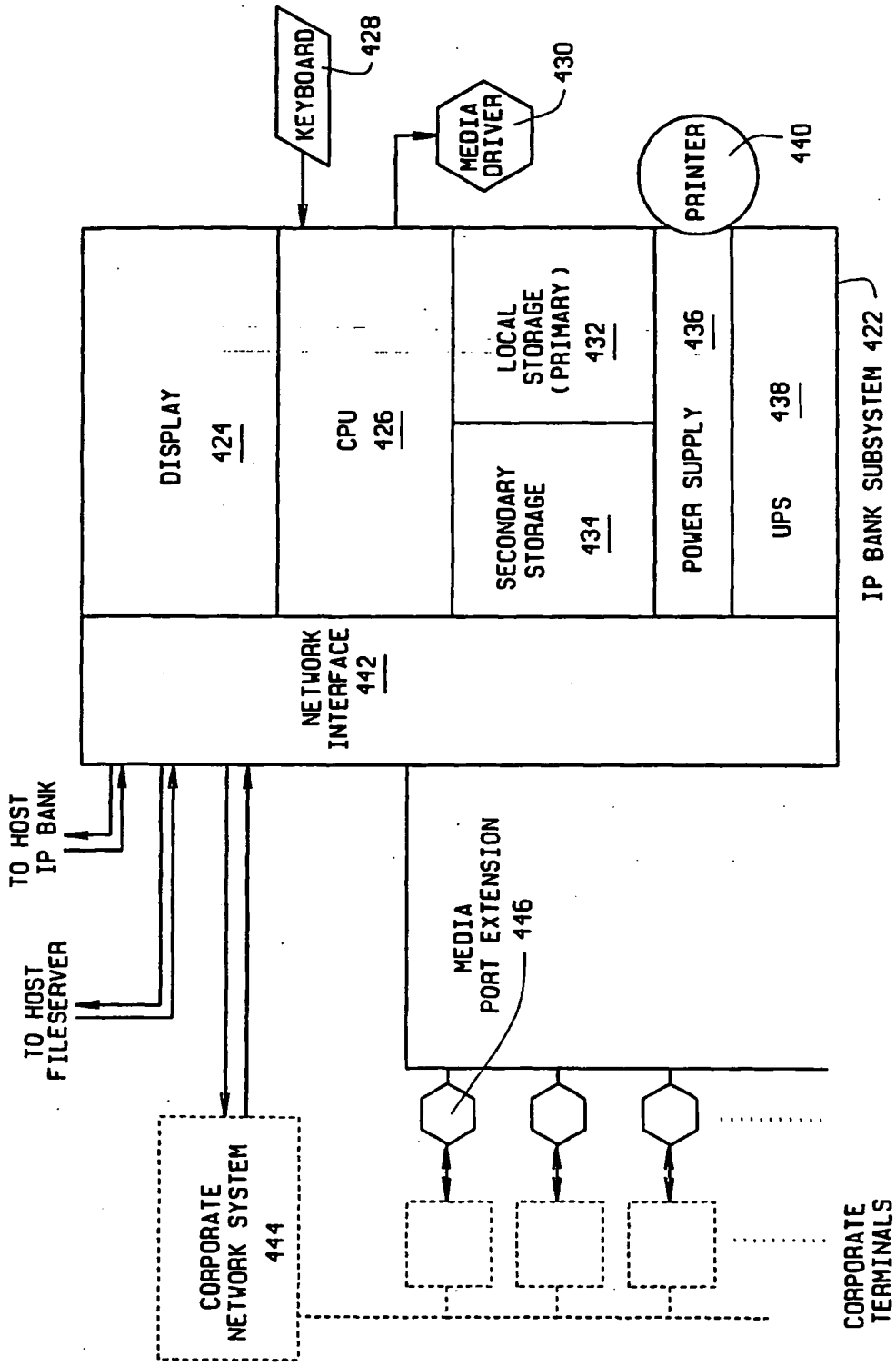

KEYBOARD
428
MEDIA DRIVER
430
PRINTER
440
DISPLAY
424
CPU
426
LOCAL STORAGE (PRIMARY)
432
SECONDARY STORAGE
434
POWER SUPPLY
436
UPS
438
IP BANK SUBSYSTEM 422
NETWORK INTERFACE
442
TO HOST IP BANK
TO HOST FILESERVER
MEDIA PORT EXTENSION
446
CORPORATE NETWORK SYSTEM
444
CORPORATE TERMINALS

FIG. 10

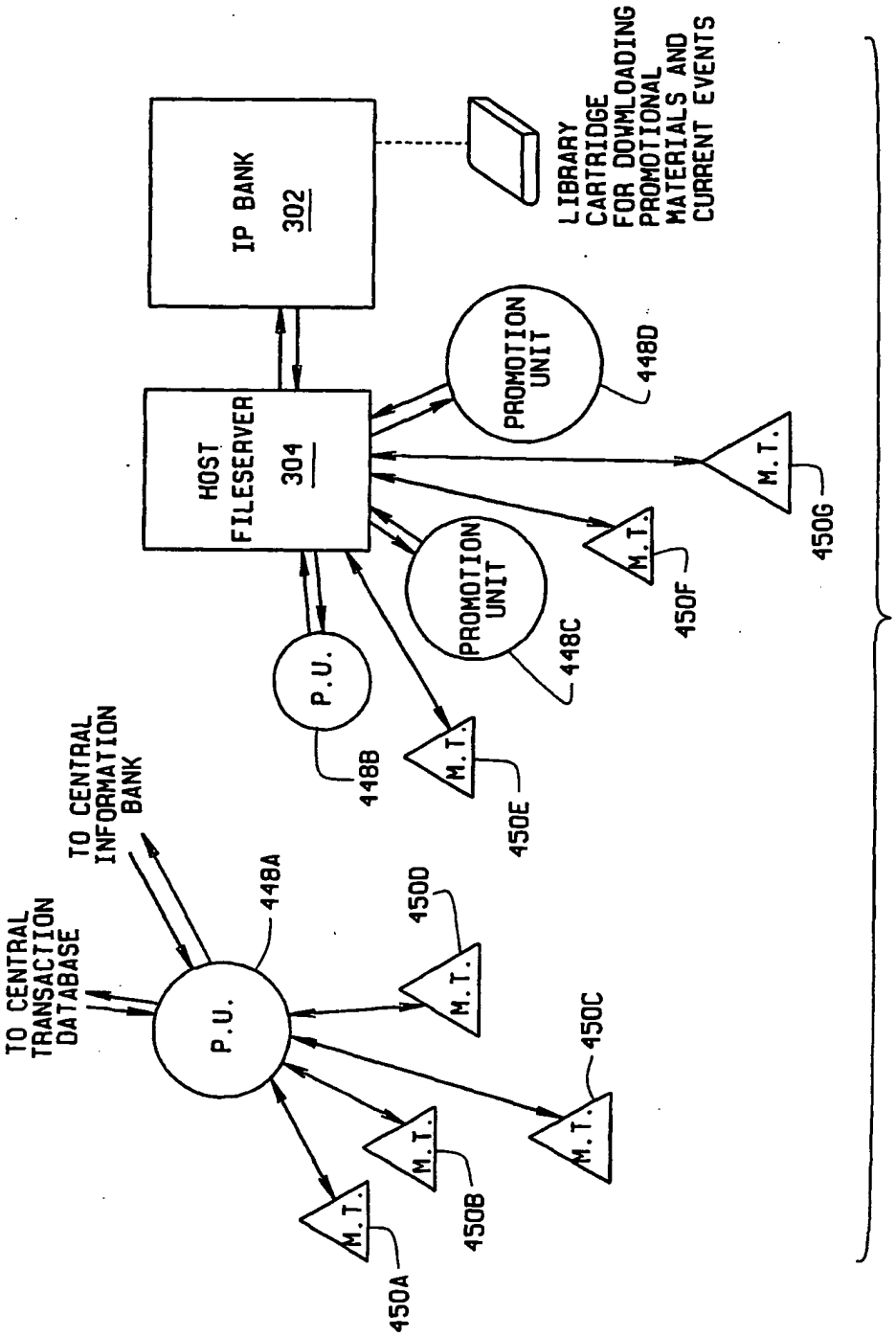

IP BANK
302
LIBRARY CARTRIDGE FOR DOWNLOADING PROMOTIONAL MATERIALS AND CURRENT EVENTS
HOST FILESERVER
304
PROMOTION UNIT
448D
PROMOTION UNIT
448C
P.U.
448B
P.U.
448A
TO CENTRAL INFORMATION BANK
TO CENTRAL TRANSACTION DATABASE
M.T.
450A
M.T.
450B
M.T.
450C
M.T.
450D
M.T.
450E
M.T.
450F
M.T.
450G

FIG. 11

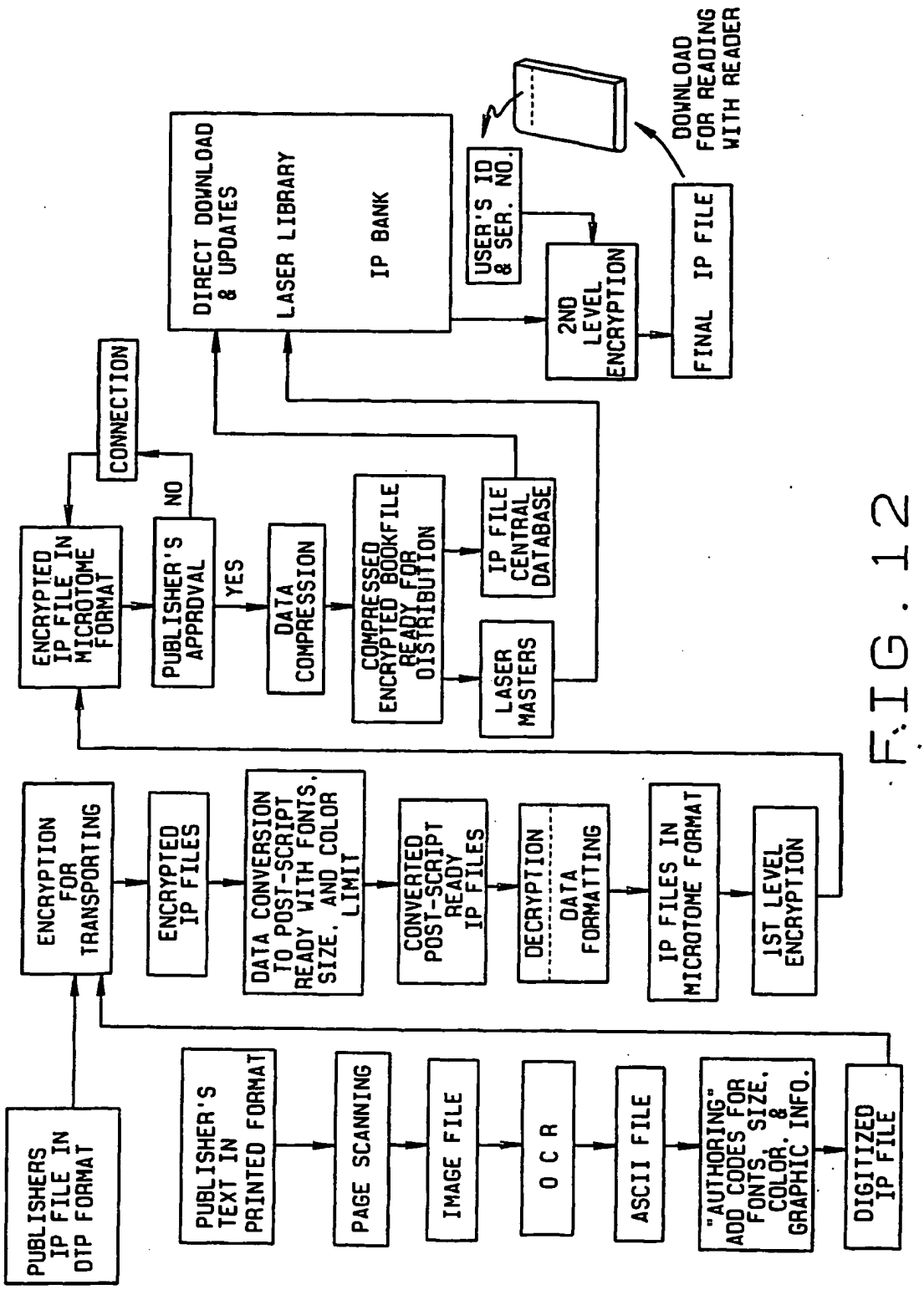

PUBLISHERS IP FILE IN DTP FORMAT
PUBLISHER'S TEXT IN PRINTED FORMAT
PAGE SCANNING
IMAGE FILE
O C R
ASCII FILE
"AUTHORING" ADD CODES FOR FONTS, SIZE, COLOR, & GRAPHIC INFO.
DIGITIZED IP FILE
ENCRYPTION FOR TRANSPORTING
ENCRYPTED IP FILES
DATA CONVERSION TO POST-SCRIPT READY WITH FONTS, SIZE, AND COLOR LIMIT
CONVERTED POST-SCRIPT READY IP FILES
DECRYPTION DATA FORMATTING
IP FILES IN MICROTOME FORMAT
1ST LEVEL ENCRYPTION
ENCRYPTED IP FILE IN MICROTOME FORMAT
PUBLISHER'S APPROVAL
NO
YES
CONNECTION
DATA COMPRESSION
COMPRESSED ENCRYPTED BOOKFILE READY FOR DISTRIBUTION
IP FILE CENTRAL DATABASE
LASER MASTERS
DIRECT DOWNLOAD & UPDATES
LASER LIBRARY
IP BANK
USER'S ID & SER. NO.
2ND LEVEL ENCRYPTION
FINAL IP FILE
DOWNLOAD FOR READING WITH READER

FIG. 12

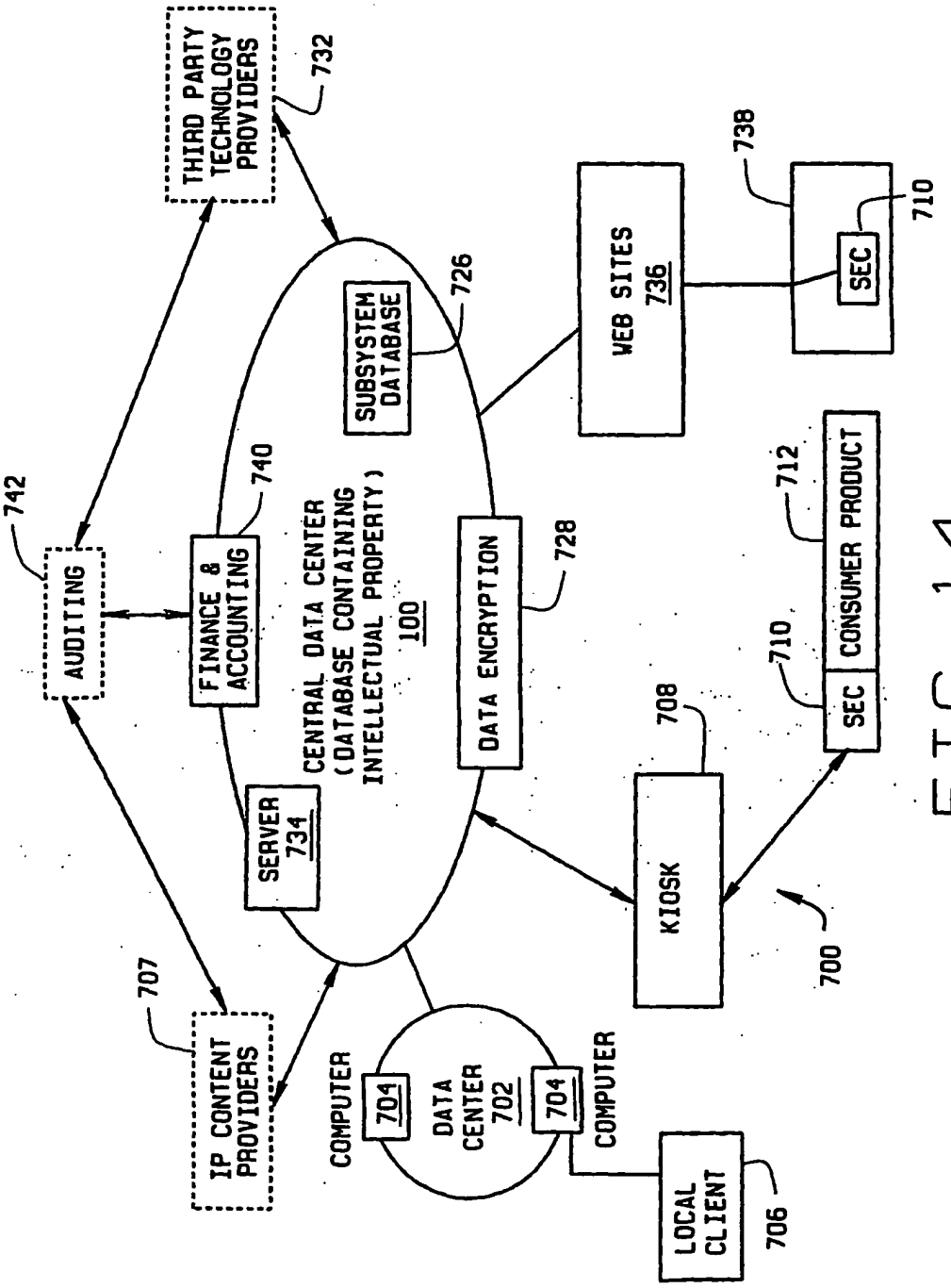

THIRD PARTY TECHNOLOGY PROVIDERS
732
738
WEB SITES 736
SEC
710
SUBSYSTEM DATABASE
726
742
AUDITING
740
FINANCE & ACCOUNTING
CENTRAL DATA CENTER (DATABASE CONTAINING INTELLECTUAL PROPERTY) 100
DATA ENCRYPTION
728
712
CONSUMER PRODUCT
SERVER 734
708
KIOSK
700
707
IP CONTENT PROVIDERS
COMPUTER
704
DATA CENTER 702
704
COMPUTER
LOCAL CLIENT
706

FIG. 14

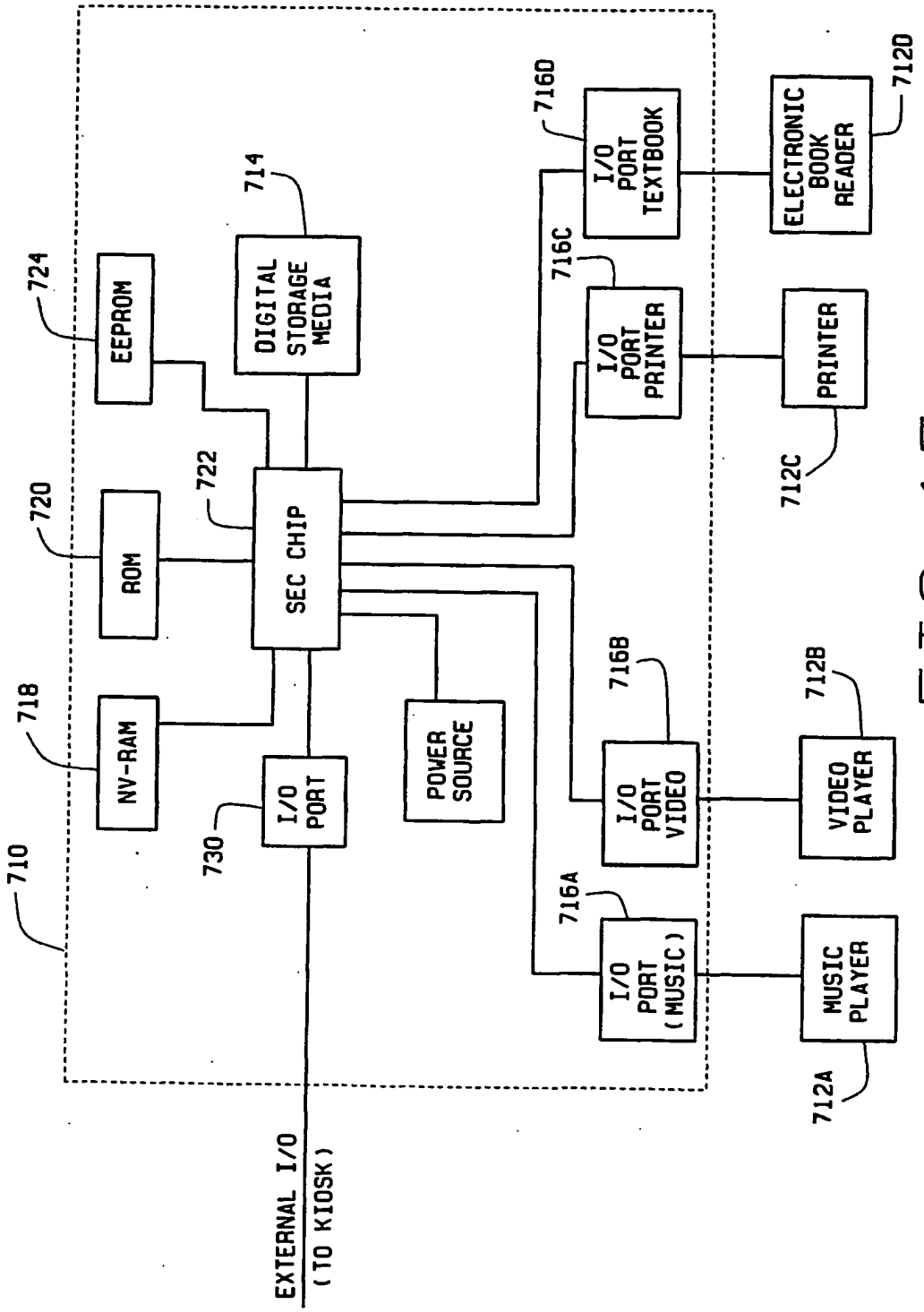

710
724
EEPROM
714
DIGITAL STORAGE MEDIA
716D
I/O PORT TEXTBOOK
712D
ELECTRONIC BOOK READER
716C
I/O PORT PRINTER
712C
PRINTER
720
ROM
722
SEC CHIP
718
NV-RAM
730
I/O PORT
POWER SOURCE
716B
I/O PORT VIDEO
712B
VIDEO PLAYER
716A
I/O PORT (MUSIC)
712A
MUSIC PLAYER
EXTERNAL I/O
(TO KIOSK)

FIG. 15

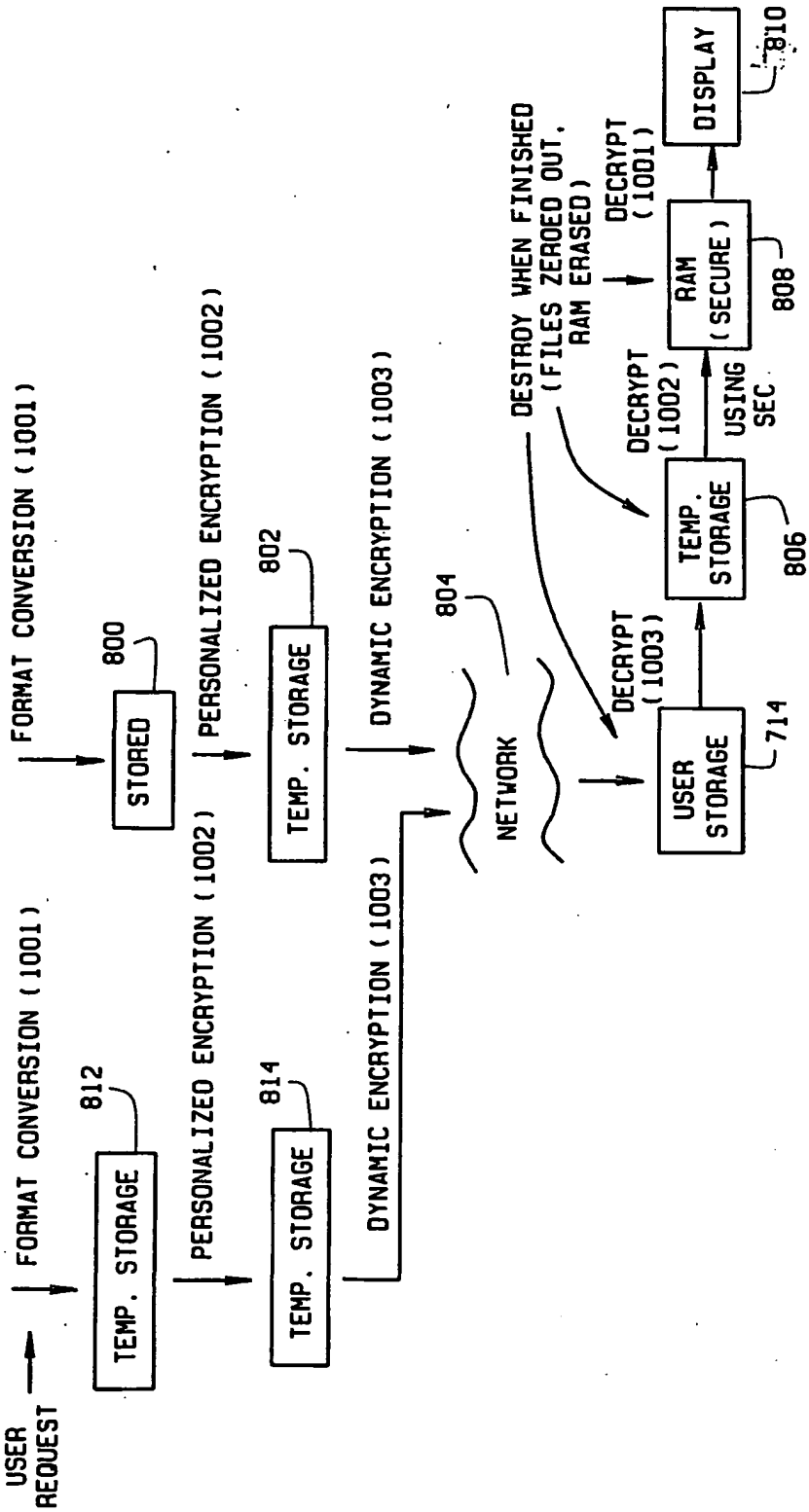

USER REQUEST
IP LINK
FORMAT CONVERSION (1001)
TEMP. STORAGE
812
PERSONALIZED ENCRYPTION (1002)
TEMP. STORAGE
814
DYNAMIC ENCRYPTION (1003)
IP STORED
FORMAT CONVERSION (1001)
STORED
800
PERSONALIZED ENCRYPTION (1002)
TEMP. STORAGE
802
DYNAMIC ENCRYPTION (1003)
804
NETWORK
DESTROY WHEN FINISHED (FILES ZEROED OUT, RAM ERASED)
DECRYPT (1003)
USER STORAGE
714
DECRYPT (1002)
TEMP. STORAGE
806
USING SEC
RAM (SECURE)
808
DECRYPT (1001)
DISPLAY
810

FIG. 16

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/05706

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 17/60
US CL : 705/50

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. : 705/50, 51, 52, 53, 54, 57

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,910,987 A (GINTER et al) 08 June 1999 (08.06.1999), See entire document	1-92
A	US 5,715,403 A (STEFIK) 03 February 1998 (03.02.1998), See entire document	1-92
A	US 5,388,196 A (PAJAK et al) 07 February 1995 (07.02.1995), See entire document	1-92
A	US 4,855,725 A (FERNANDEZ) 08 August 1989 (08.08.1989), See entire document	1-92
A	US 4,899,292 A (MONTAGNA et al) 06 February 1990 (06.02.1990), See entire document	1-92

☐ 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 application or patent 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	Date of mailing of the international search report
24 May 2001 (24.05.2001)	18 JUN 2001
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 Telephone No. (703)305-9700

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

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G06F 1/00 — A1

(11) International Publication Number: **WO 92/20022**

(43) International Publication Date: 12 November 1992 (12.11.92)

(21) International Application Number: PCT/US92/03812

(22) International Filing Date: 6 May 1992 (06.05.92)

(30) Priority data:

697,652	8 May 1991 (08.05.91)	US
723,456	28 June 1991 (28.06.91)	US
722,840	28 June 1991 (28.06.91)	US
723,457	28 June 1991 (28.06.91)	US

(71) Applicant: DIGITAL EQUIPMENT CORPORATION [US/US]; 146 Main Street, Maynard, MA 01754 (US).

(72) Inventor: WYMAN, Robert, Mark; 410 Second Avenue, South No. 108, Kirkland, WA 98033 (US).

(74) Agents: NATH, Ram, B. et al.; c/o Joyce D. Lange, Digital Equipment Corporation, 111 Powdermill Road, Maynard, MA 10754 (US).

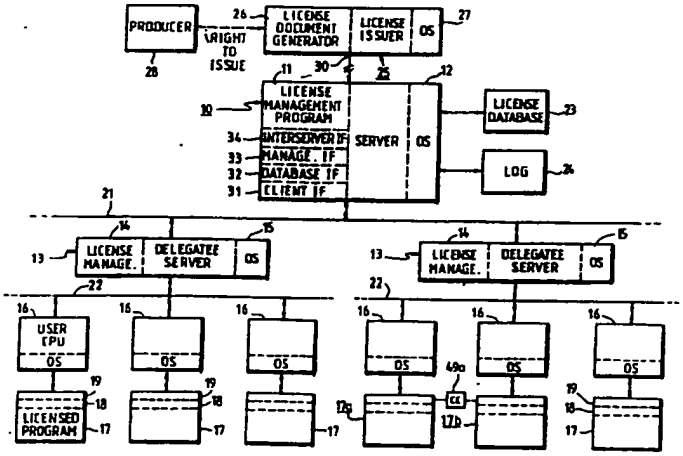
(81) Designated States: AT, AT (European patent), AU, BB, BE (European patent), BF (OAPI patent), BG, BJ (OAPI patent), BR, CA, CF (OAPI patent), CG (OAPI patent), CH, CH (European patent), CI (OAPI patent), CM (OAPI patent), CS, DE, DE (European patent), DK, DK (European patent), ES, ES (European patent), FI, FR (European patent), GA (OAPI patent), GB, GB (European patent), GN (OAPI patent), GR (European patent), HU, IT (European patent), JP, KP, KR, LK, LU, LU (European patent), MC (European patent), MG, ML (OAPI patent), MR (OAPI patent), MW, NL, NL (European patent), NO, PL, RO, RU, SD, SE, SE (European patent), SN (OAPI patent), TD (OAPI patent), TG (OAPI patent).

Published

With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: MANAGEMENT INTERFACE AND FORMAT FOR LICENSE MANAGEMENT SYSTEM



(57) Abstract

A distributed computer system employs a license management system to account for software product usage. A management policy having a variety of alternative styles and contexts is provided. Each licensed product upon start-up makes a call to a license server to check on whether usage is permitted, and the license server checks a database of the licenses, called product use authorizations, that it administers. If the particular use requested is permitted, a grant is returned to the requesting user node. The product use authorization is structured to define a license management policy allowing a variety of license alternatives by values called "style", "context", "duration" and "usage requirements determination method". The license administration may be delegated by the license server to a subsection of the organization, by creating another license management facility duplicating the main facility. The license server must receive a license document (a product use authorization) from an issuer of licenses, where a license document generator is provided. A mechanism is provided for one user node to make a call to use a software product located on another user node; this is referred to as a "calling card", by which a user node obtains permission to make a procedure call to use a program on another node. A management interface allows a license manager at a server to modify the license documents in the database maintained by the server, within the restraints imposed by the license, to make delegations, assignments, etc. The license documents are maintained in a standard format referred to as a license document interchange format so the management system is portable and can be used by all adhering software vendors. A feature of the database management is the use of a filter function.

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.

AT	Austria	FI	Finland	ML	Mali
AU	Australia	FR	France	MN	Mongolia
BB	Barbados	GA	Gabon	MR	Mauritania
BE	Belgium	GB	United Kingdom	MW	Malawi
BF	Burkina Faso	GN	Guinea	NL	Netherlands
BG	Bulgaria	GR	Greece	NO	Norway
BJ	Benin	HU	Hungary	PL	Poland
BR	Brazil	IE	Ireland	RO	Romania
CA	Canada	IT	Italy	RU	Russian Federation
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland			SN	Senegal
CI	Côte d'Ivoire	KR	Republic of Korea	SU	Soviet Union
CM	Cameroon	LI	Liechtenstein	TD	Chad
CS	Czechoslovakia	LK	Sri Lanka	TG	Togo
DE	Germany	LU	Luxembourg	US	United States of America
DK	Denmark	MC	Monaco		
ES	Spain	MG	Madagascar		

MANAGEMENT INTERFACE AND FORMAT FOR LICENSE MANAGEMENT SYSTEM

BACKGROUND OF THE INVENTION

This invention relates to methods of operation of computer systems, and more particularly to a method and system for managing the licensing of software executed on computer systems.

In U.S. Patent 4,937,863, issued to Robert, Chase and Schafer and assigned to Digital Equipment Corporation, the assignee of this invention, a Software Licensing Management System is disclosed in which usage of licensed software may be monitored in a computer system to determine if a use is within the scope of a license. The system maintains a database of licenses for software products,

delivering the license document may be in the form of a network, or may be a phone line using modems, or may include physical delivery by disks or CD ROMs, for example. Likewise, the method of delivery of the software products being licensed, i.e., the applications programs 17 to be executed on the CPUs 16, is not material to the license management facility of the invention; the products are delivered by some appropriate means, e.g., the communications link 30 and the networks 21 and 22, by CD ROMs or disks physically distributed, etc.

Although shown in Figure 1 as operating on a distributed system, in the simplest case the license management facility of the invention may be operated on a single CPU. The license management program 11 and the applications program 17 may be executing on the same CPU, in which case the license document would be stored in a database 23 as before, on this CPU, and the calls from the unit 18 to the license server would be local instead of RPCs. As in the distributed system, however, the licensed product would still not have access to the license document, but instead could only make inquires to the server program, even if all are executing on the same CPU.

In operation of the distributed system of Figure 1, the producer 28 gives the issuer 25 authority to grant licenses on its behalf (the producer and issuer can be a single entity or multiple entities). The license document generator program 26, under control of a user (a person), generates a license (usually the result of negotiation between the user of program 26 and a user of the server 10). This license is called a product use authorization, and it is transmitted by the link 30 to the server 10. The license management program in the server 10 stores the product use authorization in the database 23, and, if delegation is an authorized option, may distribute parts of the authorized use to the delegatee servers 13,

where it is likewise stored in a database. Thereafter, administration of the license is only in response to inquiries from user nodes 16. When execution of a program 17 begins, the unit 18 is invoked to check on the availability of a license for this particular node. The unit 18 sends (as by an RPC) a request to the license management program 14 (or 11 if there is no delegatee), where the product use authorization stored in database 23 is checked to see if use is authorized. If so, a return is sent to the user node 16, granting permission to continue. When the program 17 has finished executing, the unit 18 again is invoked to signal to the license management program, again by an RPC, that the authorization is released, so the license management program can take appropriate action, e.g., log the use in log 24, etc.

To implement these operations, the license management program 11 or 14 contains several functions, including a client interface 31, a database interface 32, a management interface 33, and an interserver interface 34 for communicating with the delegatees 13 (if any). The client interface 31, as described below, handles the requests received from the user nodes 16, and returns resulting from these requests. The database interface 32 handles the storing and retrieval of license information in the database 23, and logging license usage activity to log 24, and retrieval of this data. The management interface 33 handles the tasks of receiving the product use authorizations from the issuer 25 and maintaining the database 23 via the database interface 32. The interserver interface 34 handles the task of communicating with the delegatee servers 13, including transmitting the assigned parts of the product use authorizations, or communicating with other license servers that may be separately executing the license management function; for example, calls for validating calling cards may be made to another such server.

If there are no delegatees or no other license servers, then of course the interserver interface 34 has no function, and is idle.

The license document or "product use authorization" forming the basis for the license management activity of the program 11 on the server 10 may be illustrated as a data structure containing the information set forth in Figure 2; in actual practice the product use authorization is preferably a more abstract data arrangement, not in such a rigidly structured format as illustrated. For example, the product use authorization as well as similar documents stored in the database 23, or passed between components of the system of Figure 1, may be of the so-called tag-length-value data format, where the data structure begins with an identifying tag (e.g., PUA or product use authorization) followed by a field giving the length, followed by the value itself (the content). One type of data treatment using this tag-length-value format is an international standard referred to as ASN.1 or Abstract Syntax Notation. In any event, the document 35 illustrated in Figure 2 is merely for discussing the various items of data, rather than representing the way the information is stored. Some of the fields shown here exist at some times and not others, and some are optional; the product use authorization may also include additional fields not shown or discussed here. Also it should be noted that copies of parts of this type of document are made for the delegatees, so this representation of Figure 2 is a composite of several documents used in the system of Figure 1. The document 35 includes fields 36 identifying the software product by product name, producer, version numbers, release date, etc. The issuer 25 is identified in field 37, and the licensee (usually the owner of the license server 10) identified in field 38. The essential terms of the license grant are then defined in fields 40-46. The start date and end date are specified in fields 40; these store the exact time (date, hour, minute, second, etc.) when the license becomes valid and

when it ends, so licenses may be granted to start at some future time and to end at a particular time. Note that the previous practice has been to specify only the ending date, rather than also a start date as employed here. Each of the nodes, including issuer 25, servers 10 and 13, and user nodes 16, maintain a time value by a local clock referenced to a standard, so inherent in the license management facility is the maintaining of a time standard to compare with the start and end date information in the fields 40. The units granted are specified in field 41; the units are an arbitrary quantitative measure of program usage. In a delegatee server 13, the units field 41 will have some subset of the units field in the original product use authorization. As units are granted to users 16 or delegated, the remaining units available for grant are indicated in a subfield 42 in the copy of the document used by the server. The management policy occupies fields 43-46, and includes style, context, duration and LURDM (license use requirements determination method), as will be explained. The style field 43 specifies whether the licensed units are controlled by an "allocative" style or "consumptive" style, or some other "private" algorithm, where styles are ways used to account for the consumption or allocation of the units. The context field 44 specifies the location and environment in which product use or license management occurs, i.e., a CPU or an individual user or a network, etc. Duration field 45 indicates whether the license granted to a user is by assignment, by transaction, or immediate. The LURDM field 46 indicates the license use requirements determination method, in some cases using a license use requirements table (LURT) seen as field 47, as will be described.

Additional fields 48-54 in the product use authorization 35 of Figure 2 define features such as delegation authorization, calling authorization, overdraft

authorization, combination authorization, token, signature, checksum, etc. These will be described in the following paragraphs.

If the delegation field 48 is true, a license server 10 may distribute license units to multiple servers 13. A time limit may be imposed, i.e., units can be delegated to other hardware systems until they time out. Delegation allows an administrator to distribute units to improve response time and increase the resilience of the system. For example, the communication network 21 may include a satellite link to a remote facility where the local server 13 has a number of clients or users 16, in which case the calls to the server 13 would be completed much quicker than would be the case if calls had to be made to the server 10. Also, delegation may be used as a method of allocating licensed units within a budget for administrative purposes. Usually the delegation authorization is a feature that is priced by the issuer, i.e., a license granting 1000 units with delegation authorization is priced higher than without this authorization.

The field 49 contains a calling authorization and/or a caller authorization. If the caller authorization in field 49 is true, the product is permitted to receive calls from other named products requesting use of the product, and if conditions are met (identified caller is authorized) the server can grant a calling card, as described below. If the calling authorization is true, the product can make calls to other products. If neither is true, then the product can neither make or receive calls using the calling card feature. Referring to Figure 1, if product 17a wishes to make a remote procedure call to a feature of product 17b running on a different user node 16, it makes a call to its server 13 including a request for a calling card, and, if permitted, the return to product 17a includes a calling card 49a. The product 17a then makes a call to product 17b in the usual manner of

RPCs, sending along the calling card 49a, which the product 17b then verifies by a call to its server 13 before executing the called procedure and issuing its return to product 17a. The feature of calling cards is important for distributed applications. For example, if a product is able to execute faster in a distributed system by assigning tasks to other CPUs, then the issue is presented of which license policy is needed, i.e., does every node executing a part of the task have to be licensed and consume or receive allocation of a unit, or just the one managing the task? This is resolved for most applications by use of this calling card concept. The product use authorization for such a product has the calling authorization field 49 enabled, so calling cards can be issued. This feature is typically separately priced.

The combination authorization field 50 of Figure 2 determines whether or not license requests from a user node 16 can be satisfied by combining units from multiple product use authorizations. It may be advantageous to purchase licenses with different policy values, and use units from certain product use authorizations only for overflow or the like. Or, for other reasons, it may be advantageous to "borrow" and "lend" units among delegated servers or user nodes. This function is permitted or denied by the content of field 50.

The overdraft field 51 determines whether or not a requested allocation from a user node 16 will be nevertheless granted, even though the units available field 42 is zero or too small to permit the requested use. Overdrafts can be unlimited, or a specific overdraft pool can be set up by a server 10, for a customer's internal administrative purposes. That is, the overdraft value may be unlimited in the original license, but limited or zero for internally distributed copies of the license. Thus, the product use authorization sent by the issuer 25 to

the customer may have overdrafts permitted by the field 51, but the customer may deny overdraft permission for its own budgeting purposes. In any event, if overdraft is permitted, additional fees have to be paid to the issuer at some accounting period, when the logged usage from log 24 indicates the available units have been exceeded. If overdraft is denied, then the units 18 of the user nodes making request allocations are structured to inform the products 17 that a license grant is not available. The intent is not to prevent the application program from running; the license server merely informs the application whether or not the license manager determines that it is authorized to run. The application can itself be structured to shut itself down if not authorized to run, or it can be structured to shut down certain functions (e.g., ability to save files, ability to print, etc.), or it can be structured to continue in a fully functional manner. The purpose of the license management facility is not that of enforcement, nor that of "copy protection", but instead is merely that of license management.

An optional token field 52 is available in the product use authorization 35 of Figure 2. This field can contain comments or other information desired by the issuer or user. For example, a telephone support number may be included in the token field, then when the product 17 shows its "help screen" the number is inserted. This number would be part of the argument, i.e., data transmitted to the user node 16, when the server 10 makes a return following a request allocation message from the user. This field may also be used to store information used in a "private" style, where the information from this field returned to the user node is employed by the application program 17 or the stub 19 to determine if the application can be activated.

The signature field 53 in the product use authorization 35 is a part of a validation mechanism which provides important features. This field contains a digital signature encoded to reflect the data in the license itself, as well as other encoding methods not known to customers, so it cannot be duplicated unless the encoding algorithm is known. In a preferred embodiment, a so-called "public/private key" system of encoding is used for the signature field 53. The encoding algorithm used to generate the signature 53 is known to the issuer 25, using a private key, and anyone knowing the public key can decode the signature to determine if it is valid but cannot determine the encoding algorithm so it cannot produce a forged signature. So, if the server 10 knows the public key which is unique to the issuer 25, it can determine if a license document 35 is genuine, but it cannot itself generate license documents. However, if the server possesses a valid license document that gives it the right to delegate, then it will be assigned its own private key (different from all other issuers or servers) and its delegatees 13 will be able to determine if a valid delegated license is delivered to them as they will be given the public key for the servers 13. The field 53 will thus contain both the original signature from the issuer 25 and the license server's signature when delivered to a delegatee 13. The decoding algorithm using a public key for any signatures is thus used by the license server 10 or delegatee 13 to make sure a product use authorization 35 is authentic before it is stored in the database 23. Related to the digital signature 53 is a checksum field 54, which merely encodes a value related by some known algorithm to the data in the product use authorization 35 itself. This field may be used merely to check for corruption of the data as it is stored, recalled, and transmitted within the system. That is, the checksum is used for data validation rather than security.

Two concepts central to the license management system implemented using the license document or product use authorization 35 of Figure 2 are the "license units", specified in field 41 or 42 and the "context", specified in field 44. License units are an abstract numerical measure of product use allowed by the license. When a product 17 (or a function or feature of a product) makes a license-checking request, the license management program 11 on server 10 computes how many license units are required to authorize this particular use of the product, and this is the license units requirement, in some cases using the LURDM field 46. A "context" is a set of tagged values which define the location and environment in which product use or license management occurs. Context values may be specified in field 44 of the product use authorization 35 of Figure 2 to restrict the environments in which the license may be managed and in which product use may occur. A context template may also be specified in the field 44 to indicate which parts of the complete context of product use (sub-contexts) are significant in differentiating product uses for the purposes of unit allocation; when this is specified, it allows separate product uses to share license units in a controlled way.

The two general types of policies specified in field 43 are allocative and consumptive. An allocative policy grants to the holder a specific number of license units (field 41) and specifies the policy which must be used to account for the allocation of these units. A software product 17 which is being managed by an allocative license will require verification that the appropriate number of license units have been allocated to it prior to performing services to the user. Typically, this allocation of units occurs either at the time of activation of the product 17 or at the time that product use is enabled on a particular platform (user CPU 16). The units typically remain allocated to the product 17 throughout the period that the product is running or is enabled to run. Upon termination of

processing or disabling, the allocated units are deallocated and made available for allocation to other instances of the software product 17 (other users 16 activating the product). In general, as long as any license units remain unallocated in field 42, the holder of the license is contractually authorized to increase his utilization of the licensed product. The usage does not deplete the license, however, as the units are returned to the units-available field 42 after a user is finished, and can be granted again to another user.

A consumptive unit based license, indicated in policy field 43, grants to the holder a specific number of initial license units (from field 42) and specifies the policy used to account for the consumption of those units. A software product 17 which is being managed by a consumptive license will cause an appropriate number of license units to be consumed to reflect the services provided by the product. Once consumed, units cannot be reused. Thus, the number of units available for future use declines upon every use of the licensed software product 17. This may also be referred to as a "metered" policy, being conceptually similar to measured consumption of electricity, water, etc. When the number of available units in field 42 reaches zero, the license may require that further use of the product is prohibited, or, the agreement may permit continued decrementing of the number of available units; the result is the accumulation of a negative number of available units in the field 42. It is anticipated that most consumptive unit based licenses will consider negative units to represent an obligation of the license holder to pay the license issuer 25. The transaction log 24 maintains an audit trail for providing a record of the units used in a consumptive license.

Referring to Figure 3, the major elements of the management policy are set forth in a table, where the possible entries for the fields 43, 44, 45 and 46 are

listed. For the style entry 43, the possibilities are allocative and consumptive as just described, plus a category called "private" which represents a style of management undefined at present but instead to be created especially for a given product, using its own unique algorithm. It is expected that most licenses may be administered using the named alternatives of Figure 3, but to allow for future expansion to include alternatives not presently envisioned, or to permit special circumstances for unique software, the "private" choices are included, which merely mean that the product 17 will generate its own conditions of use. It is important to note that, except for the "private" alternative, the license management is totally in control of the license management program 11 on the license server 10 (or delegatee 13), rather than at the product 17. All the product 17 does, via the unit 18, is to make the request inquiry to the server 10 via the client interface 31, and report when finished.

The context field 44 specifies those components (sub-contexts) of the execution-context name which should be used in determining if unit allocations are required. License data is always used or allocated within, or for the benefit of, some named licensing context, and context can include "platform contexts" and "application contexts". Platform contexts are such things as a specific network, an execution domain, a login domain, a node, a process ID or a process family, a user name, a product name, an operating system, a specific hardware platform, as listed in Figure 3. Applications contexts are information supplied from the application (the product 17), such as may be used in a "private" method of determining license availability. The context name can use several of these, in which case the context name is constructed by concatenating the values of all subcontexts into a single context name, e.g., a VAX 3100 platform using VMS operating system.

The duration field 45 defines the duration of an allocation of license units to a specific context or the duration of the period which defines a valid consumptive use. For durations of type "Assignment," the specification of a reassignment constraint is also provided for, as discussed below. There are three types of duration, these being "transaction," "assignment" and "immediate" as seen in Figure 3.

The transaction duration type, when specified for an allocative policy, indicates that license units should be allocated to the specified context upon receipt of a license request and that those units should be deallocated and returned to the pool of available units upon receipt of a corresponding license release from a user node 16. Abnormal termination of the process or context having made the original license request will be semantically equivalent to a license release. On the other hand, when specified for a consumptive policy, this duration type indicates that license units should be allocated to the specified context upon receipt of a license request and permanently removed from the available units pool (field 42) upon receipt of a license release which reflects successful completion of the transaction. Upon receipt of a license release which carries an error status or upon abnormal termination of the processor context having made the original license request, the allocated units will be deallocated and returned to the pool of available units (field 42).

The assignment duration type in Figure 3 (field 45 of Figure 2) imposes the constraint that the required units must have been previously assigned to a specific context. The sub-contexts which must be specified in the assignment are those given in the context-template. A "reassignment constraint" may be imposed, and this is a limitation on how soon a reassignment can be made. For example, a

reassignment constraint of 30-days would require that units assigned to a specific context could not be reassigned more often than every 30-days; this would prevent skirting the intent of the license by merely reassigning units whenever a user of another context made a request allocation call for the product. Related to this assignment constraint, a "reallocation limit" may also be imposed, to state the minimum duration of an allocation; where there is a context template of process, the intent is to count the number of uses of the software product at a given time, but where software runs in batch rather than interactive mode it may run very quickly on a powerful machine, so a very few concurrent uses may permit almost unlimited usage - by imposing a reallocation constraint of some time period, this manner of skirting the intent of the license may be constrained.

The immediate duration type (field 45 of Figure 2) is used to indicate that the allocation or consumption of an appropriate number of license units from the pool of available units (field 42) should be performed immediately upon receipt of a license request. Receipt of license release or abnormal terminations will then have no impact on the license management system. When specified as the duration for an allocative policy, the effect will be simply to check if an appropriate number of license units are available at the time of a license request. When specified as the duration for a consumptive policy, the effect will be to deduct the appropriate number of license units from the available pool at the time of a license request, and, thereafter, abnormal termination, such as a fault at the user CPU 16 or failure of the network link, will not reinstate the units.

The LURDM or license unit requirement determination method, field 46, has the alternatives seen in Figure 3 and stores information used in calculating the number of units that should be allocated or consumed in response to a license

request. If this field specifies a table lookup kind, this means license unit requirements are to be determined by lookup in the LURT (field 47) which is associated with the current license. If a constant kind is specified, this indicates that the license units requirements are constant for all contexts on which the licensed product or product feature may run. A private LURDM specifies that the license unit requirements are to be determined by the licensed product 17, not by the license management facility 11. The license unit requirements tables (LURTs) provide a means by which issuers of licenses can store information describing the relation between context (or row selector) and unit requirements. The license units requirements determination method (LURDM) must specify "table lookup" for the LURT to be used, and if so a row selector must be specified, where a valid row selector is any subcontext, e.g., platform ID, user name, time of day, etc. An example of an LURT fragment is shown in Figure 4, illustrating the license unit requirements table mechanism. In this example, the row selector is "platform-ID" so the platform-ID value determines which row is used. The issuer of this LURT of Figure 4 has established three unit requirement tiers for use in determining the unit requirements for that issuer's products. The reason for the tiers is not mandated by the license management system, but the issuer 25 (actually the user of the program 26) would probably be establishing three pricing tiers, each reflecting a different perspective on the relative utility of different platforms in supporting the use of various classes of product 17. The first column in Figure 4, Column A, specifies the use requirements for a class of products whose utility is highly sensitive to the characteristics of the specific platform on which they are run. This can be seen by observing that the unit requirements are different for every row in Column A. Products which use the second column (Column B) appear to have a utility which is more related to the class of platform on which they run. This is indicated by the fact that all the PC

platforms share a single value which is different from that assigned to the VAX platform. The final column (Column C) is for use with a class of products which is only supported on the VAX platform. Figure 4 is of course merely an example, and the actual LURT created by the license document generator 26 and stored in the license database 23 (as field 47 of the product use authorization 35) can be of any content of this general format, as desired by the license issuer.

Instead of always selecting the rows in LURT tables according to the platform ID of the execution platform, in order to handle the breadth of business practices that need to be supported by the license management facility, the LURT mechanism is extended by providing a "row selector" attribute in the LURT class structure. No default is provided although it is expected that the normal value for the row selector attribute will be "platform ID."

In the system of patent 4,937,863, a concept similar to that of the LURT of Figure 4 was provided, with rows selected by the platform ID and columns selected by some arbitrary means, typically according to product type. The system of this invention allows flexibility in the selection of both LURT row and column while continuing to provide backwards compatibility for licenses defined within the constraints of patent 4,937,863.

Some examples will illustrate potential uses for the row selector attribute. A customer may only want to pay for the use of a product during one or two months of the year; the product may be FORTRAN and the reason for this request may be that the company has a fairly stable set of FORTRAN subroutines that are given regular "annual maintenance" only during the months of May and June. To handle this customer's needs, the FORTRAN product would generate

an application subcontext which would contain a value representing the month of the year. Then, a LURT table would be defined with twelve rows, one for each month of the year. In some column, probably column A, a negative one (-1) would be placed in each month except for May and June. These two months would contain some positive number. The product use authorization would then have a LURDM field specifying a LURT for use to determine the units requirement, and would name this custom LURT table. The effect would be that the PUA could only be used during the months of May and June since negative one is interpreted by license managers to mean "use not authorized." This mechanism could also be used to do "time of day" charging. Perhaps charging fewer units per use at night than during the day. Also, if a subcontext was used that contained a year value, a type of license would be provided that varied in its unit requirements as time passed. For instance, it might start by costing 10-units per use in 1991 but then cost one unit less every year as time passed, eventually getting to the point where the unit requirement was zero.

Another example is font names. A specific customer may purchase a license giving it the right to concurrent use of 100-units of a large font collection; some of the fonts may cost more to use than others. For instance, Times Roman might cost 10-units per use while New Century Schoolbook costs 20-units per use. The problem is, of course, making sure that charges are properly made. The solution is to build a LURT table with a specified application subcontext as its row selector. A row is then created for each font in the collection and in Column A of the LURT, the number of units required to pay for use of the font would be specified. The print server would then specify the name of a font as the value of the application subcontext whenever it does an *lm_request_allocation()* call. This will allow charges to be varied according to font name.

A further example is memory size. Some products are more or less valuable depending on the size of memory available to support them. A software vendor wishing to determine unit requirements based on memory size will be able to do so by building LURT tables with rows for each reasonable increment of memory (probably 1-megabyte increments). Their applications would then sense memory size (using some mechanism not part of the license management facility) and pass a rounded memory size value to the license manager in a private context.

Other examples are environment and operating system. Some products may be valued differently depending on whether they are being run in an interactive mode or in batch. This can be accomplished by building LURT rows for each of the standard platform subcontexts that specify environment. Regarding operating system, it has been considered desirable by many to have a single product use authorization permit the use of a product on any number of operating systems, this conflicts with some vendors policies who do not want to have to create a single price for a product that applies to all operating systems. Thus, if an operating system independent license were offered for a C compiler, the price would be the same on MS-DOS, VMS, and/or UNIX. Clearly, it can be argued that the value of many products is, in part, dependent on the operating system that supports them. By using a row selector of operating system (one of the standard platform subcontexts), license designers could, in fact, require different numbers of units for each operating system. However, it might be more desirable to base the row selection on a private application subcontext that normally had the same value as the operating system subcontext. The reason for this is that the license designer might want to provide a default value for operating system names that were unknown at the time the LURT rows were defined. If this is the case, the product would contain a list of known operating systems and

pass the subcontext value of "Unknown" when appropriate. The LURT row for "Unknown" would either contain a negative one (-1) to indicate that this operating system was unsupported or it would contain some default unit requirement.

Another example is variable pricing within a group. One of the problems with a "group" license is that there is only one unit requirements field on the PUA for a group. Thus, all members of the group share a single unit requirement. However, in those cases were all members of the group can be appropriately licensed with a constant unit requirement yet it is desired to charge different amounts for the use of each group member, a LURT can be built that has rows defined for each group member. The row selector for such a group would be the standard platform subcontext "product name."

Many different types of license can be created using different combinations of contexts, duration and policy from the table of Figure 3. As examples, the following paragraphs show some traditional licensing styles which can be implemented using the appropriate values of the product use authorization fields 43-46.

A "system license" as it is traditionally designated is a license which allows unlimited use of a product on a single hardware system. The correct number of units must be allocated to the processor in advance and then an unlimited product use is available to users of the system. The product use authorization would have in the context field 44 a context template for a node name, the duration field would be "assignment" and the policy style field 43 would be "allocative".

A "concurrent use" license is one that limits the number of simultaneous uses of a licensed product. Concurrent use license units are only allocated when the product is being used and each simultaneous user of the licensed product requires their own units. In this case the context template, field 44, is a process ID, the duration field is "transaction" and the policy style 43 is "allocative".

A "personal use" license is one that limits the number of named users of a licensed product. This style of licensing guarantees the members of a list of users access to a product. Associated with a personal use type of product use authorization there is a list of registered users. The administrator is able to assign these users as required up to the limit imposed by the product use authorization; the number of units assigned to each user is indicated by the LURDM. It may be a constant or it may vary as specified in a LURT. The context template is "user name", the duration is "assignment", and the policy is "allocative".

A "site license" is one that limits the use of a licensed product to a physical site. Here the product use authorization contains for the context template either "network name" or "domain name", the duration is "assignment" and the policy style field 43 is "allocative".

Generally, a license to use a software product is priced according to how much benefit can be gained from using the product, which is related to the capacity of the machine it will run on. A license for unlimited use on a large platform such as a mainframe, where there could be thousands of potential users at terminals, would be priced at a high level. Here the style would be "allocative", the context template = "node", the duration = "assignment" and the LURDM may be "Column A" - the units, however, would be large, e.g., 1000. At the other end

of the scale would be a license for use on a single personal computer, where the field values would be the same as for the mainframe except the units would be "1". If a customer wanted to make the product available on the mainframe but yet limit the cost, he could perhaps get a license that would allow only five users at any given time to use the product; here the fields in the product use authorization would be: units = 5; style = allocative; context template = process; duration = transaction; LURDM = constant, 1-unit. This would still be priced fairly high since a large number of users may actually use the product if a session of use was short. A lower price would probably be available for a personal use license where only five named persons could use the product, these being identified only in the license server 10, not named by the license issuer 25. Here the fields in the product use authorization are: units = 5; style = allocative; context template = user name; duration = transaction; LURDM = constant, 1-unit.

An additional feature that may be provided for in the product use authorization 35 is license combination. Where there are multiple authorizations for a product, license checking requests sent by user nodes 16 may be satisfied by combining units from multiple authorizations. Individual product use authorizations may prohibit combined use. Thus, a licensee may have a license to use a product 17 on an allocative basis for a certain number of units and on a consumptive basis for another number of units (this may be attractive from pricing standpoint); there might not be enough units available for a particular context from one of these licenses, so some units may be "borrowed" from the other license (product use authorization), in which case a combination is made.

The interface between the program executing on the client or user 16 and the license server 10 or its delegatees 13 includes basically three procedure calls:

a request allocation, a release allocation and a query allocation. Figure 5 illustrates in flow chart form some of the events occurring in this client interface. The request allocation is the basic license checking function, a procedure call invoked when a software product 17 is being instantiated, functioning to request an allocation of license units, with the return being a grant or refusal to grant. Note that a product may use request allocation calls at a number of points in executing a program, rather than only upon start-up; for example, a request allocation may be sent when making use of some particular feature such a special graphics package or the like. The release allocation call is invoked when the user no longer needs the allocation, e.g., the task is finished, and this return is often merely an acknowledge; if the style is consumptive, the caller has the opportunity via the release allocation call to influence the number of units consumed, e.g., decrease the number due to some event. The query allocation call is invoked by the user to obtain information about an existing allocation, or to obtain a calling card, as will be described.

The request allocation, referred to as *lm_request_allocation()*, is a request that license units be allocated to the current context. This function returns a grant or denial status that can be used by the application programmer to decide whether to permit use of the product or product feature. The status is based on the existence of an appropriate product use authorization and any license management policies which may be associated with that product use authorization. License units will be allocated or consumed, if available, according to the policy statement found on the appropriate product use authorization. The product would normally call this function before use of a licensed product or product feature. The function will not cause the product's execution to be terminated should the request fail. The decision of what to do in case of failure to obtain allocation of license

units is up to the programmer. The arguments in a request allocation call are the product name, producer name, version, release date, and request extension. The product name, producer name, version and release date are the name of the software product, name of producer, version number and release date for specifically identifying the product which the user is requesting an allocation be made. The request extension argument is an object describing extended attributes of the request, such as units required, LURT column, private context, and comment. The results sent back to the calling node are a return code, indicating whether the function succeeded and, if not, why not, and a grant handle, returned if the function completes successfully, giving an identifying handle for this grant so it can be referred to in a subsequent release allocation call or query allocation call, for example.

The release allocation, referred to as *lm_release_allocation()*, is an indication from a user to the license manager to release or consume units previously allocated. This function releases an allocation grant made in response to a prior call to request allocation. Upon release, the license management style 38 determines whether the units should be returned to the pool of available units or consumed. If the caller had specified a request extension on the earlier call to request allocation which contained a units-required-attribute, and the number of units requested at that time are not the number of units that should be consumed for the completed operation, the caller should state with the units-consumed argument how many units should be consumed. The arguments of the release allocation are: grant handle, units consumed, and comment. The grant handle identifies the allocation grant created by a previous call to request allocation. The units-consumed argument identifies the number of units which should be consumed if the license policy is consumptive; this argument should only be used

in combination with an earlier call to request allocation which specified a units requirement in a request extension. Omission of this argument indicates that the number of units to be consumed is the same as the number allocated previously. The comment argument is a comment which will be written to the log file 24 if release units are from a consumptive style license or if logging is enabled. The result is a return code indicating if the function succeeded, and, if not, why not.

The query allocation, or *lm_query_allocation()*, is used by licensed products which have received allocations by a previous request allocation call. The query is to obtain information from the server 10 or delegatee server 13 about the nature of the grant that has been made to the user and the license data used in making the grant, or to obtain a calling card (i.e., a request that a calling card be issued). Typically, the item read by this query function is the token field 52 which contains arbitrary information encoded by the license issuer and which may be interpreted as required by the stub 19 for the licensed product software 17, usually when a "private" allocation style or context is being employed. The arguments in this procedure call are the grant handle, and the subject. The grant handle identifies the allocation grant created by a previous call to request allocation. The subject argument is either "product use authorization" or "calling card request"; if the former then the result will contain a public copy of the product use authorization. If this argument is a calling card request and a calling card which matches the previous constraints specified in that request can be made available, the result will contain a calling card. If the subject argument is omitted, the result will contain an instance of the allocation. The results of the query allocation call are (1) a return code, indicating whether the function succeeded, and, if not, why not, and (2) a result, which is either an allocation, a product use authorization or a calling card, depending on type and presence of the subject argument.

Referring to Figure 5, the flow chart shows the actions at the client in its interface with the server. When the software product 17 is to be invoked, the unit 18 is first executed as indicated by the block 60, and the first action is to make a request allocation call via the stub 19, indicated by the block 61. The client waits for a return, indicated by the loop 62, and when a return is received it is checked to see if it is a grant, at decision block 63. If not, the error code in the return is checked at block 64, and if a return code indicates a retry is possible, block 65, control passes back to the beginning, but if no retry is to be made then execution is terminated. If the policy is to allow use of the product 17 without a license grant, this function is separately accounted for. If the decision point 63 indicates a grant was made, the grant handle is stored, block 66, for later reference. The program 17 is then entered for the main activities intended by the user. During this execution of product 17, or before or after, a query allocation call can be made, block 67, though this is optional and in most cases not needed. When execution of the program 17 is completed, the grant handle is retrieved, block 68, and a release allocation call is made, block 69. A loop 70 indicates waiting for the return from the server, and when the return received it is checked for an error code as before, and a retry may be appropriate. If the release is successfully acknowledged, the program exits.

Referring to Figure 6, the actions of the server 10 or delegatee server 13 in executing the license management program 11 or 14, for the client interface, are illustrated in flow diagram form. A loop is shown where the server program is checking for receipt of a request, release or query call from its clients. The call would be a remote procedure call as discussed above, and would be a message communicated by a network, for example. This loop shows the decision blocks 71, 72 and 73. If a release allocation call is received, a list of products for which

authorizations are stored is scanned, block 74, and compared to the product identity given in the argument of the received call, block 75. If there is no match, an error code is returned to the client, block 76, and control goes back to the initial loop. If the product is found, the authorization is retrieved from the database 23, block 77 (there may be more than one authorization for a given product, in which case all would be retrieved, but only one will be referred to here) and all of the information is matched and the calculations made depending upon the management policy of Figures 3 and 4, indicated by the decision block 78. If a grant can be made, it is returned as indicated at block 79, or if not an error code is returned, block 80. If a release allocation call is received, indicated by a positive at the decision block 72, the grant handle in the argument is checked for validity at block 81. If no match is found, an error code is returned, block 82, and control passes back to the initial loop. If the handle is valid, the authorization for this product is retrieved from the database 23 at block 83, and updated as indicated by the block 84. For example, if the license management style is allocative, the units are returned to the available pool. Or, in some cases, no update is needed. The authorization is stored again in the database, block 85, and a return made to the client, block 86, before control passes back to the initial loop. If the decision block 73 indicates that a query allocation call is received, again the grant handle is checked at block 87, and an error code returned at block 88 if not valid. If the grant handle matches, the authorization is retrieved from the database 23, at block 89, and a return is made to the client giving the requested information in the argument, block 90.

The basic allocation algorithm used in the embodiment of the license management system herein described, and implemented in the method of Figures 5 and 6, is very simple and can handle a very large proportion of known license

unit allocation problems. However, it should be recognized that a more elaborate and expanded algorithm could be incorporated. Additions could be made in efforts to extend the allocation algorithm so that it would have specific support for optimizing unit allocation in a wider variety of situations. Particularly, sources of non-optimal allocations occurring when using the basic allocation algorithm are those that arise from combination and reservation handling.

The first step is formation of full context. The client stub 19 is responsible for collecting all specified platform and application subcontexts from the execution environment of the product 17 and forwarding these collected subcontexts to the license management server 13 or 10. The collection of subcontexts is referred to as the "full context" for a particular license unit allocation request.

The next step is retrieval of the context template. When the license manager receives an *lm_request_allocation()*, it will look in its list of available product use authorizations (PUA) to determine if any of them conform to the product identifier provided in the *lm_request_allocation()* call. The product identifier is composed of: product name, producer, version, release date. If any match is found, the license manager will extract from the matching PUA the context template. This template is composed of a list of subcontexts that are relevant to the process of determining unit requirements. Thus, a context template may indicate that the node-ID subcontext of a specific full context is of interest for the purposes of unit allocation. The context template would not specify any specific value for the node-ID; rather, it simply says that node-ID should be used in making the allocation computation.

The next step is masking the full context. Having retrieved the context template, the license manager will then construct an "allocation context" by filtering the full context to remove all subcontexts which are not listed in the context template. This allocation context is the context to be used in determining allocation requirements.

Then follows the step of determining if the request is new. The license manager maintains for each product use authorization a dynamic table which includes the allocation contexts of all outstanding allocations for that PUA (i.e., allocations that have been granted but have not yet been released). Associated with each entry in this table is some bookkeeping information which records the number of units allocated, the full context, etc. To determine if a recent *lm_request_allocation()* requires an allocation of units to be made, the license manager compares the new allocation context with all those allocation contexts in the table of outstanding allocations and determines if an allocation has already been made to the allocation context. If the new allocation context does not already exist in the table, an attempt will be made to allocate the appropriate number of units depending on the values contained in the LURDM structure of the PUA and any LURTs that might be required. If an allocation context similar to that specified in the new allocation request does exist in the table, the license manager will verify that the number of units previously allocated are equal to or greater than the number of units which would need to be allocated to satisfy the new allocation request. If so, the license manager will return a grant handle to the application which indicates that the allocation has been made (i.e., it is a "shared allocation" - the allocated units are shared between two requests.) If not, the license manager will attempt to allocate a number of units equal to the

difference between the number previously allocated and the number of units required.

The step of releasing allocations (Fig. 6, blocks 84-85) occurs when the license manager receives an *lm_release_allocation()* call; it will remove the record in its dynamic allocation table that corresponds to the allocation to be released. Having done this, the license manager will then determine if the allocation to be removed is being shared by any other allocation context. If so, the units associated with the allocation being released will not be released. They will remain allocated to the remaining allocation contexts. Some of the units might be released if the license manager determines that the number of allocated units exceeds the number needed to satisfy the outstanding allocation contexts. If this is the case, the license manager will "trim" the number of allocated units to an appropriate level.

In summary, the two things that make this algorithm work are (1) the basic rule that no more than one allocation will be made to any single allocation context, and (2) the use of the context template to make otherwise dissimilar full contexts appear to be similar for the purposes of allocation.

The license designer's task, when defining basic policy, is then to determine which contexts should appear to be the same to the license manager. If the license designer decides that all contexts on a single node should look the same (context template = node-ID), then any requests that come from that node will all share allocations. On the other hand, a decision that all contexts should be unique (i.e., context template = process-ID) will mean that allocations are never shared.

and stores a unit value indicating the number of licensing units for each product. When a user wishes to use a licensed product, a message is sent to the central license management facility requesting a license grant. In response to this message, the facility accesses the database to see if a license exists for this product, and, if so, whether units may be allocated to the user, depending upon the user's characteristics, such as the configuration of the platform (CPU) which will execute the software product. If the license management facility determines that a license can be granted, it sends a message to the user giving permission to proceed with activation of the product. If not, the message denies permission.

While the concepts disclosed in the patent 4,937,863 are widely applicable, and indeed are employed in the present invention, there are additional functions and alternatives that are needed in some applications. For example, the license management system should allow for simultaneous use of a wide variety of different licensing alternatives, instead of being rigidly structured to permit only one or only a few. When negotiating licenses with users, vendors should have available a wide variety of terms and conditions, even though a given vendor may decide to narrow the selection down to a small number. For example, a software product may be licensed to a single individual for use on a single CPU, or to an organization for use by anyone on a network, or for use by any users at terminals in a cluster, or only for calls from another specific licensed product, or any of a large number of other alternatives. A vendor may have a large number of products, some sold under one type of license and some under others, or a product may be a composite of a number of features from one or more vendors having different license policies and prices; it would be preferable to use the same license management system for all such products.

Distributed computing systems present additional licensing issues. A distributed system includes a number of processor nodes tied together in a network of servers and clients. Each node is a processor which may execute programs locally, and may also execute programs or features (subparts of programs) via the network. A program executing on one node may make remote procedure calls to procedures or programs on other nodes. In this case, some provision need be made for defining a license permitting a program to be executed in a distributed manner rather than separately on a single CPU, short of granting a license for execution on all nodes of a network.

In a large organization such as a company or government agency having various departments and divisions, geographically dispersed, a software license policy is difficult to administer and enforce, and also likely to be more costly, if individual licenses are negotiated, granted and administered by the units of the organization. A preferred arrangement would be to obtain a single license from the software producer, and then split this license into locally-administered parts by delegation. The delays caused by network communication can thus be minimized, as well as budgetary constraints imposed on the divisions or departments. Aside from this issue of delegation, the license management facility may best be operated on a network, where the licensing of products run on all nodes of the network may be centrally administered. A network is not necessary for use of the features of the invention however, since the license management can be implemented on a single platform.

Software products are increasingly fragmented into specific functions, and separate distribution of the functions can be unduly expensive. For example, a spreadsheet program may have separate modules for advanced color graphics, for

accessing a database, for printing or displaying an expanded list of fonts, etc. Customers of the basic spreadsheet product may want some, none or all of these added features. Yet, it would be advantageous to distribute the entire combination as one package, then allow the customer to license the features separately, in various combinations, or under differing terms. The customer may have an entire department of the company needing to use the spreadsheet every day, but only a few people who need to use the graphics a few days a month. It is advantageous, therefore, to provide alternatives for varied licensing of parts or features of software packages, rather than a fixed policy for the whole package.

Another example of distribution of products in their entirety, but licensing in parts, would be that of delivering CD ROMs to a customer containing all of the software that is available for a system, then licensing only those parts the customer needs or wishes to pay fees for rights to use. Of course, the product need not be merely applications programs, operating systems, or traditional executable code, but instead could also include static objects such as printer fonts, for example, or graphics images, or even music or other sound effects.

As will be explained below, calling and caller authorizations are provided in the system according to one feature of the invention, in order to provide technological support for a number of business practices and solve technical problems which require the use of what is called "transitive licensing." By "transitive licensing" is meant that the right to use one product or feature implies a right to use one or more other products or features. Transitive licenses are similar to group licenses in that both types of license consist of a single instrument providing rights of use for a plurality of products. However, transitive licenses differ from group licenses in that they restrict the granted rights by specifying that

the licensed products can only be used together and by further specifying one or more permitted inter-product calling/caller relationships. Some examples may help to clarify the use and nature of a transitive license: the examples to be explained are (1) two products sold together, (2) a give-away that results from narrow choices of licensing alternatives, (3) a client licensing method in a client/server environment, (4) impact of modular design, and (5) the impact of distributed design.

A software vendor might have two products for sale: the first a mail system, and the second a LEXIS™-like content-based text retrieval system. Each of these products might be valued at $500 if purchased separately. Some customers would be satisfied by purchasing the rights to use only one of these products. others might find that they can justify use of both. In order to increase the likelihood that customers will, in fact, purchase both products, it would not be surprising if the software vendor offered his potential customers a volume discount, offering the two products for a combined price of $800. The customers who took advantage of this combined offer would find that they had received two products, each of which could be exploited to its fullest capabilities independently from the other. Thus, these customers would be able to use the content based retrieval system to store and retrieve non-mail documents. However, from time to time, the vendor may discover that particularly heavy users of mail wish to be able to use the content based retrieval system only to augment the filing capabilities provided by the standard mail offering. It is likely that many of these potential customers would feel that $800 is simply too much to pay for an extended mail capability. The vendor might then consider offering these customers a license that grants mail users the right to use the content-based retrieval system only when they are using mail and prohibits the use of content

based retrieval with any other application that might be available on the customers system. This type of license is referred to below a "transitive license," and it might sell for $600.

Another example is a relational database product (such as that referred to as Rdb™) designed for use on a particular operating system, e.g., VMS. This relational database product has two components: (1) A user interface used in developing new databases, and (2) a "run-time" system which supports the use of previously developed databases. The developers of the database product might spend quite a bit of effort trying to get other products made by the vendor of the database product to use it as a database instead of having those other products build their own product-specific databases. Unfortunately, the other product designers may complain that the cost of a run-time license for the database product, when added to the cost of licenses for their products, would inevitably make their products uncompetitive. Thus, some mechanism would be needed that would allow one or another of the vendor's products to use the run-time system for the relational database product in a "private" manner while not giving unlicensed access to products of other vendors. No such mechanism existed, prior to this invention; thus, the vendor might be forced to sell the right to use its run-time system for the database product with its proprietary operating system license. Clearly, this combined license would make it possible for the vendor's products to use its database product without increasing their prices; however, it also would make it possible for any customers and third-parties to use the database product without paying additional license fees. However, had the system of the invention been available, the vendor could have granted transitive licenses for the run-time component of its database product to all the vendor's products. Essentially, these licenses would have said that the database run-time could be used without an

additional license fee if and only if it was used in conjunction with some other of the vendor's products. Any customer wishing to build a new relational database application or use a third-party application that relied on the vendor's database product would have had to pay the vendor for its database run-time license.

A proposed client/server licensing method provides yet another example of a problem which could be solved by transitive licensing. Typically, a client is only used by one user at a time, while a server can support an arbitrary number of clients depending on the level of client activity and the capacity of the machine which is supporting the server. While traditionally, server/client applications have been licensed according to the number of clients that a server could potentially support, this may not be the most appropriate method for licensing when the alternatives afforded by the invention are considered. The business model for the proposed client/server method requires that each client be individually licensed and no explicit licensing of servers is required to support properly licensed clients. Such a licensing scheme makes it possible to charge customers only for the specific number of clients they purchase. Additionally, it means that a single client can make use of more than one server without increasing the total cost of the system. The solution to this transitive licensing problem would be to provide a mechanism that would allow the clients to obtain license unit allocations and then pass a "proof" of that allocation to any servers they may wish to use. Servers would then support any clients whose proofs could be verified to be valid. On the other hand, if a client that had not received a proof of allocation attempted to use a server, the server would obtain a license allocation for that client session prior to performing any services. Such a solution has not been heretofore available.

As the complexity and size of the software systems provided to customers increases, it is found that the actual solution provided to customers is no longer a single product. Rather, customers are more often now offered solutions which are built up by integrating an increasing number of components or products, each of which can often stand alone or can be part of a large number of other solutions. In fact, a product strategy may rely almost exclusively on the vendor's engineering and selling a broad range of specialized components that can only be fully exploited when combined together with other components into a larger system. Such components include the relational database runtime system mentioned above, mail transport mechanisms, hyperinformation databases, document format conversion services, time services, etc. Because these components are not sold on their own merits, but rather on their ability to contribute to some larger system, it is unlikely that any one customer will be receiving the full abstract economic value of any one of the components once integrated into a system. Similarly, it can be observed that the value of any component once integrated into a larger system varies greatly from system to system. Thus, it may be found that a mail transport mechanism contributes a large part of a system whose primary focus is mail, however, it will contribute proportionally less of the value of a system that provides a broader office automation capability. As a result of these observations, the job of the business analyst who is attempting to find the "correct" market price for each component standing on its own, is more complex. In reality, the price or value of the component can only be determined when considering the contribution of that component to the full system or solution in which it is integrated. Attempting to sell the components at prices based on their abstract, independent values will simply result in overpriced systems.

Transitive license styles are particularly suited to dealing with pricing of modular components, since component prices can be clearly defined in relation to the other components or systems which they support. Thus, a vendor can charge a price of $100 for the right to use a mail transport system in conjunction with one product, yet charge $200 for the use of the same mail transport system when used by another product.

In addition to the "business" reasons for wanting to support transitive licensing, there is also a very good technical reason that arises from the growing tendency of developers to build "distributed products" as well as the drive toward application designs that exploit either tightly or loosely coupled multiprocessor systems; the availability and growing use of remote procedure calls has contributed to this tendency. This technical problem can be seen to arise when considering a product which has a number of components, each of which may run in a different process space and potentially on a different computer system. Thus, there might be a mail system whose user interface runs on one machine, its "file cabinet" is supported by a second machine and its mail transport system runs on yet a third machine. The simple question which arises is: "Which of the three components should check for licenses?" Clearly it must be ensured that no single component can be used if a valid license is not present. Thus, the answer to the question will probably be that all three components should check for licenses. However, the question is then presented: "Where are the licenses to be located?". This can become more complex.

Increasingly, the distributed systems being built are being designed so that it is difficult to predict on which precise machine any particular component will run. Ideally, networks are supposed to optimize the placement of functions

automatically so that the machine with the most available resource is always the one that services any particular request. This dynamic method of configuring the distribution of function servers on the network makes it very difficult for a system or network manager to predict which machines will run any particular function and thus very difficult for him to decide on which machines software licenses should be loaded.

Even if a system manager could predict which machines would be running the various application components and thus where the license units should be loaded, the situation would still be less than ideal. The problem arises from the fact that each of the components of the application would be independently making requests for license unit allocations. This behavior will result in a difficult problem for anyone trying to decide how many license units are required to support any one product. Given the mail example, the problem wouldn't exist if it were assumed that all three components (i.e., user interface, file cabinet, and transport system) were required by the design of the mail system to be in use simultaneously. If this were the case, it could be simply assumed that supporting a single activation of the mail system would require three units. However, in a real mail system, it will be inevitably discovered that many users will only be using just the user-interface and file-cabinet components of the system at one time. Thus, there will be some unused units available which could be used to authorize additional users. This situation might not be what is desired by the software vendor.

The problem of providing license support to multi-component products which are dynamically configured could be solved by viewing each of the product components as a distinct licensable product and by treating the problem as one

of transitive licensing, but a mechanism for accomplishing this has not been available. Essentially, a single license document would be created that stated that if any one of the components had successfully obtained a license to run, it could use this grant to give it the right to exploit the other components. Thus, in the example above, the user might start the mail system by invoking its user interface. This user interface code would then query the license management facility for a license allocation and once it has received that allocation, it would pass a proof of allocation to the other mail components that it uses. Each of the other components would request that the license management system validate that the "proof" is valid prior to performing any service; however, none of the other components would actually require specific allocations to be made to them. In this way, the complexity of licensing and managing networks of distributed applications can be significantly reduced.

SUMMARY OF THE INVENTION

In accordance with one embodiment of the invention, a license management system is used to account for software product usage in a computer system. The system employs a license management method which establishes a management policy having a variety of simultaneously-available alternative styles and contexts. A license server administers the license, and each licensed product upon start-up makes a call to the license server to check on whether usage is permitted, in a manner similar to that of patent 4,937,863. The license server maintains a store of the licenses, called product use authorizations, that it administers. Upon receiving a call from a user, the license server checks the product use authorization to determine if the particular use requested is

permitted, and, if so, returns a grant to the requesting user node. The license server maintains a database of product use authorizations for the licensed products, and accesses this database for updating and when a request is received from a user. While this license management system is perhaps of most utility on a distributed computer system using a local area network, it is also operable in a stand-alone or cluster type of system. In a distributed system, a license server executes on a server node and the products for which licenses are administered are on client nodes. However, the license management functions and the licensed products may be executing on the same processor in some embodiments.

The product use authorization is structured to define a license management policy allowing a variety of license alternatives by components called "style", "context", "duration" and "usage requirements determination method". The style may be allocative or consumptive. An allocative style means the units of the license may be allocated temporarily to a user when a request is received, then returned to the pool when the user is finished, so the units may be reused when another user makes a request. A consumptive style means the units are deducted from an available pool when a user node makes a valid request, and "consumed", not to be returned for reuse. The context value defines the context in which the use is to be allowed, such as on a particular network, by a particular type of CPU, by a particular user name, by a particular process, etc. The duration value (used in conjunction with the style component) concerns the time when the license units are to be deducted from the available pool of units, whether at the time of request, after a use is completed, etc. A usage requirements determination method may be specified to define or provide information concerning the number of license units charged in response to a license request from a user node; for example, some CPU platforms may be charged a larger number of license units

than others. A table may be maintained of usage requirements, and the determination method may specify how to access the table, for example. The important point is that the user node (thus the software product) can only make a request, identifying itself by user, platform, process, etc., and the license management facility calculates whether or not the license can be granted (that is, units are available for allocation), without the user node having access to any of the license data or calculation. There is a central facility, the license server, storing the license documents, and, upon request, telling the licensed products whether they can operate under the license terms.

An important feature of one embodiment is that the license administration may be delegated to a subsection of the organization, by creating another license management facility duplicating the main facility. For example, some of the units granted in the product use authorization may be delegated to another server, where the user nodes serviced by this server make requests and receive grants.

The license management facility cannot create a license itself, but instead must receive a license document (a product use authorization) from an issuer of licenses. As part of the overall license management system of the invention, a license document generator is provided which creates the product use authorizations under authority of the owner of the software, as negotiated with customers. Thus, there are three distinct rights in the overall license management facility of the invention: (1) the right to issue licenses, (2) the right to manage licenses, and (3) the right to use the licensed products. Each one of these uses the license document only in prescribed ways. The license issuer can generate a license document. The license manager (or license server as referred to herein) can grant products the right to use under the license, and can delegate parts of the

licensed units for management by another server, as defined by the license document; the way of granting rights to products is by responding to certain defined calls from the products. And, the licensed products can make certain calls to the license server to obtain grants of rights based upon the license document, inquire, or report, but ordinarily cannot access the document itself.

As explained above, transitive licensing is an important feature of one embodiment. This is the provision of a mechanism for one user node to get permission to use another software product located on another user node; this is referred to as a calling authorization and a caller authorization, using a "calling card," and these are examples of the optional features which must be specifically permitted by the product use authorization. A user node must obtain permission to make a procedure call to use a program on another node; this permission is obtained by a request to the license server as before, and the permission takes the form of a calling card. When a calling card is received by a second node (i.e., when the procedure call is made), a request is made by the second node to the license server to verify (via the product use authorization) that the calling card is valid, and a grant sent to the user node if allowed. In this manner, all nodes may have use of a program by remote calls, but only one consumes license units.

Another important feature of one embodiment is a management interface which allows a license manager to modify the license policy components of a license document maintained by at a license server in its database. Usually the license manager can only make modifications that restrict the license policy components to be more restrictive than originally granted. Of course, the management interface is used to make delegations and assignments, if these are authorized.

The license document interchange format is an important feature, in that it allows the license management system to be used with a wide variety of software products from different vendors, so long as all follow the defined format. The format uses data structures that are defined by international standards.

An important function is the filter function, used in the management interface and also in the client interface to select among elements in the data structures.

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 other features and advantages thereof, will be best understood by reference to the detailed description of specific embodiments which follows, when read in conjunction with the accompanying drawings, wherein:

Figure 1 is a diagram in block form of a distributed computer system which may be used to implement the license management operations according to one embodiment of the invention;

Figure 2 is a diagram of the content of a license document or "product use authorization" generated by the license document generator and stored by the license server in the system of Figure 1;

Figure 3 is a diagram of the alternatives for license style, context and duration making up the license management policy implemented in the system of Figure 1, according to one embodiment of the invention;

Figure 4 is a diagram of an example of a fragment of a license use requirements table (LURT) used in the system of Figure 1, according to one embodiment of the invention;

Figure 5 is a logic flow chart of a program executed by a user node (client), in the system of Figure 1, according to one embodiment of the invention;

Figure 6 is a logic flow chart of a program executed by a license server, in the system of Figure 1, according to one embodiment of the invention; and

Figure 7 is a diagram of the calls and returns made in an example of use of calling cards in the system of Figure 1.

Figure 8 is a diagram of an LDIF document identifier, according to an standard format;

Figure 9 is a syntax diagram of an LDIF document;

Figure 10 is a diagram of an LDIF document structure;

Figures 11, 13, 15, 17, 18, 19, 21-28 and 31-43 are syntax diagrams for elements of various ones of the LDIF data structures;

Figure 16 is a diagram of a license data structure;

Figures 12, 14 and 20 are examples of descriptions of data elements using a standard notation;

Figures 29 and 30 are examples of context templates used in the license management system;

Figures 44 and 45 are tables of attributes specific to filter and filter item type; and

Figure 46 is notation in a standard format for an example of a filter.

DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

Referring to Figure 1, a license management facility according to one example embodiment of the invention is centered around a license server 10, which typically includes a CPU located in the customer's main office and executing a license management program 11 as will be described, under an operating system 12. The license server 10 communicates with a number of delegatees 13 which likewise include CPUs in departments or divisions of the company or organization, each also executing a license management program 14 under an operating system 15. The license management program 14 is the same as the program 11 executing on the main server 10; the only difference in the functions of server 10 and servers 13 is that the latter have a delegated subset of the license units granted to the server 10, as will be described. The CPUs 13 are in turn servers for a number of

users 16, which are CPU nodes where the licensed programs 17 are actually executed. The programs 17 executing on the user CPUs 16 are applications programs (or operating systems, etc.) which have added to them units 18 and 19, according to the invention, allowing them to make inquiry to the their server 13 (or 10) before executing and to report back after executing, using a client stub 19 in the manner of remote procedure calls, in one embodiment. A user node 16 may have many different programs 17 that may be executed, and the various user nodes 16 would usually each have a set of programs 17 different from the other user nodes, all of which would be administered by the license management program 14 or 11. The terms "program" and "licensed product" are used in reference to the element 17, but it is understood that the products being administered may be segments of programs, or functions or features called by another program, or even merely data (such as printer fonts), as well as complete stand-alone applications programs. The license server 10 communicates with the delegatee servers 13 by a network 21, as is usual in large organizations, and the delegatee servers 13 each communicate with their user nodes 16 by networks 22; these networks may be of the Ethernet, token ring, FDDI types or the like, or alternatively, the user nodes 16 may be merely a cluster of terminals on a multiuser system with the delegatee being a host CPU. The particular hardware construction of the user nodes, server nodes, communication networks, etc., and the operating systems 12 or 15, are of no concern regarding the utility of the features of the invention, the only important point being that the user CPUs 16 of the software products 17 in question can communicate readily and quickly with their respective server nodes 13 or 10. In one embodiment, remote procedure calls (RPCs) are used as the communication medium for the interfaces between components of the system, handling the inquiries and grants as will be described.

A remote procedure call is similar to a local procedure call but is made to a procedure located on a remote node, by way of a communications network.

The function of the unit 19 is that of a client stub, in a remote procedure call sense. The calls to the license server 10 are made through this stub 19, and returns are received by the stub 19 and passed on to the program 17. The stub 19 is responsible for obtaining the network addresses of other nodes on the network, such as the server 10. Also, the stub 19 is responsible for determining the context (as defined below) for passing on to the server 10. The unit 18 functions to execute a "private" type of license availability determination if this is used, rather than this task being done by the application program 17, but if the ordinary method of determination is employed (using the license server) as is usually the case, the unit 18 is merely code that starts the execution and passes calls and returns back and forth between the program 17 and the unit 19.

The license server 10, using the license management program 11, maintains a license data file 23 comprising a number of license documents or licenses (product use authorizations), and also maintains a log 24 which is a record of the usage activity of all of the user CPUs 16 of each of the licensed programs. The delegatee servers 13 would maintain similar license databases and logs. The license server 10 has no authority to originate a license, but instead must receive a license from a license issuer 25. The issuer 25 is again a CPU executing a license document generator program 26 under an operating system 27. The license issuer 25 may be under control of the producer 28 of the programs or software products being licensed, or may be controlled by a distributor who has received the authority to grant licenses from the producer or owner 28. The communications link 30 between the license issuer 25 and the license server 10 for

This mechanism permits the system of the invention to dispose of the cumbersome, explicit support of license types having different scope such as the cluster licenses, node licenses, and process licenses found in prior license management systems including that of patent 4,937,863. Instead of defining a limited set of scopes (cluster, node, etc.), the system of this invention provides a general mechanism which allows an effectively unlimited range of allocation scopes to be defined.

Transitive licensing, as referred to above, is supported by the system of the invention by (1) calling authorizations, which are statements made in field 49 of the product use authorization 35 for one product (the "caller") to permit that product to call another product (the "callee"), and, (2) caller authorizations, which are statements made in field 49 of the product use authorization for one product (the "callee") to permit it to be called by another product (the "caller").

If calling or caller authorizations are to be exploited by products, then whenever one product calls another product, it must pass the callee a calling card 49a. This calling card 49a is an encoding of an identification of the caller as well as a statement by the license management system that a license unit allocation has been made to the caller which is passing the calling card. This calling card is then passed by the callee to the license management system for validation and, if the either the product use authorization of the caller carries an appropriate calling authorization or the product use authorization of the callee carries an appropriate caller authorization, the use of the callee by the caller will be authorized without requiring any additional license unit allocations.

Referring to Figure 7, the intercomponent interactions that occur when either calling or caller authorizations are being used are illustrated. This figure shows a license management server 10, a caller product 17a named "Product-1" and a callee product 17b named "Product-2". When Product-1 starts to run, it will make an *lm_request_allocation()* call to the license management server 10 to obtain a grant handle for an allocation of some number of units of the Product-1 license. Either immediately, or at some later time, but always prior to making a call to Product-2, Product-1 will call *lm_query_allocation()*, passing the grant handle received earlier and specifying that it wants a calling card for the product named "Product-2." If the field 49 of the product use authorization 35 used to satisfy the grant represented by the grant handle carries a calling authorization in field 49 naming "Product-2," the license manager will create a calling card 49a which includes the statement that a calling authorization exists and pass this calling card back to Product-1. If the calling authorization does not exist, the calling card passed to Product-1 will contain a statement to that effect.

Once Product-1 has successfully obtained a calling card 49a from the license manager, it will then make a call to Product-2, passing the calling card along with any other initialization parameters that would normally be used when starting Product-2. Product-2 will then pass that calling card to the license manager as part of its *lm_request_allocation()* call and the license manager will determine if the calling card is valid. Note that calling cards become invalid once the process which received the calling card makes an *lm_release_allocation()* call or terminates abnormally. If the calling card is valid, and it indicates that a calling authorization is present, the license manager will verify this statement and if found to be true, will return a grant handle to Product-2. If, on the other hand, the calling card carries an indication that no calling authorization is present, the

license manager will attempt to find a product use authorization for Product-2 that contains a caller authorization naming Product-1 as an authorized caller. If the caller authorization is found, a grant handle will be passed back to Product-2. If not, the license manager will ignore the calling card and proceed with the normal *lm_request_allocation()* logic.

The requirement to be passing calling cards between products requires that both the caller and the callee be "aware" of the fact that calling and caller authorizations may be used. This is one of the few examples of a requirement for a product 17 to become actively involved in the licensing problem when using the licensing management system of the invention. However, since the use of calling/caller authorizations if a fairly "sophisticated" and powerful feature, it is considered acceptable to impose this burden on application coders.

MANAGEMENT INTERFACE

Referring to Figure 1, the license management program 11 executing on a server 10 includes a license management interface 33 which functions to allow a user at a console for the server 10 CPU or at a remote terminal to implement certain necessary operations. The management interface 33 is essentially the tools or mechanisms available to the license manager at the licensee's site to (a) load the various licenses received from issuers 25 into the database 23 and make them available for request allocation calls from the users, (b) remove the licenses from the machine when expired, (c) to make delegations if permitted, (d) to make assignments, (e) to make reservations, etc. Whatever the license manager is allowed to do to modify the license for his special circumstances (within the

original grant, of course), he does it by the mechanism of the management interface 33. Some licenses are not modified at all, but merely loaded. In a multiple machine environment, as on a network, there is considerable modification, as it is necessary to make sure the correct number of units are distributed onto the correct machines, the right people have access, other people don't have access, etc. Thus, in a network environment, there is extensive use of the management interface 33.

In reference to the terminology used in describing the management interface, as well as the license management system in general, it is helpful to note that the documentation conventions, data declarations, macro declarations, etc., for the object management used in one embodiment of the invention are according to the standards set forth in *OSI Object Management API Specification, Version 2.0*, X.400 API Association and X/Open Company Limited, 24 August 1990, a published document.

The specific operations available to the management interface 33 are to allow a manager to open and close a management session, register (load) objects in the license database 23, obtain a list of objects in the license database 23, and control a cursor (a cursor is a movable pointer to a member of a list of items). Once an object in the license database 23 is identified with the cursor, certain changes may be made in the object by a write function. For example, certain fields of a license document of Figure 2 or an LURT of Figure 4 may be changed in only specified ways as will be explained.

The operation of opening a session goes by the name of *lm_open_session()* and is used to establish a license management service session between a

management client and the service. Opening a session also creates a workspace to contain objects returned as a result of functions invoked within the session. Object management Objects can be created and manipulated within this workspace. Objects created within this workspace, and only such objects, may be used as Object arguments to the other license management service management functions used during the session established by a call to this function. More than one session may exist simultaneously.

The arguments that go with a *lm_open_session()* call are (a) the binding handle, which is binding information that defines one possible binding (a client-server relationship), and (b) a comment which will be inserted in the log file 24 if logging is enabled. The results from a *lm_open_session()* call are (a) a return code indicating whether the function succeeded, and, if not, why not, (b) a session, which is an established license management session between the management client and the license management service, and (c) a workspace that will contain all objects returned as a result of functions invoked in the session.

The close session call is referred to by *lm_close_session()* and functions to terminate the lm session. This function terminates the license service management session and makes the argument unavailable for use with other interface functions. The arguments that go with a *lm_close_session()* call are (a) the session which identifies the established lm session between the management client and the license management service, and (b) a comment which will be inserted in the log file if logging is enabled. The result of the call is a return code indicating whether the function succeeded, and, if not, why not.

The list function returns a set of selected objects in the license database 23, and uses the name *lm_list_licenses()*. This function is used to search the license database 23 and return a cursor which represents the first of one or more objects which match the specified filter. The specified filter will be applied to each object in the license database 23; all objects for which the filter evaluates true will be included in the object list accessible by the set_cursor function. The arguments that go with *lm_list_licenses()* are (a) session which identifies an established session between the management client and the license management service, and (b) a filter which is an object used to select license database 23 objects; license database objects will only be included in the object list headed by the cursor if they satisfy the filter - the constant no-filter may be used as the value of this argument if all license data objects are to be included in the object list. The results of the *lm_list_licenses()* call are (a) a return code indicating whether the function succeeded, and, if not, why not, and (b) a license list upon successful completion of this call containing a cursor which represents the first of one or more objects in the current license database 23 for which the specified filter evaluates true.

The register function is to register objects in the license database 23, and uses the name *lm_register()*. This function is used to register (i.e., load or create) new objects, or modify existing objects, in the license database 23; the objects which may be registered include only those which are subclasses of the license data class or history objects. The arguments are (a) session, which identifies an established session between the management client and the license management service, (b) license data object which is to be registered; if this argument is omitted, the comment argument is a required argument and a history object containing the comment will be registered in the license database 23, and (c)

comment, which will be inserted in the log file if logging is enabled. The result is a return code indicating whether the function succeeded, and, if not, why not. The errors possible when it does not succeed include data-expired, duplicate-object, no-such-session, memory-insufficient, network-error, etc., indicated by this return code.

The set cursor function establishes a new cursor, and is called by *lm_set_cursor()*. The arguments are (a) session, which identifies an established session between the management client and the license management service, (b) forward, which is a boolean value indicating if the direction in which the cursor is to be moved is forward or reverse, (c) filter which is used to eliminate cursors from the search for the next cursor that are not wanted; a new cursor will only be set if it satisfies the filter - the constant no-filter may be used as the value of this argument if any cursor is to be considered as the target cursor, and (d) the cursor which is to be used as the starting point in searching for the new cursor. The results are (a) a return code indicating whether the function succeeded, and, if not, why not, and (b) next-cursor, which is the requested cursor. The error codes in the return code may be end-of-list, not-a-cursor, etc.

After a session is opened, and an object such as a product use authorization or a LURT has been identified by the cursor, using the functions explained above, the management interface 33 is able to execute certain object management interface functions such as write or copy. By this mechanism, the management interface can modify certain limited attributes. None of these attributes can be modified in such a way that they reduce constraints established by corresponding attributes in the license data objects. The more important attributes which can be modified by the management interface 33 using this mechanism are:

(a) assignment: an assignment of some or all of the units granted on the associated product use authorization;

(b) reservation: a reservation of some or all of the units granted on the associated product use authorization;

(c) delegation: a delegation of the right to manage some or all of the units granted on the associated product use authorization, or if the associated license data is not a product use authorization, the delegation is of the right to use that license data;

(d) backup delegation: a statement of the right to manage some or all or the units granted on the associated product use authorization; this right is only active at times when the delegating server is not available;

(e) allocation: an allocation of units to a specific context;

(f) allocation period: the minimum duration of a single allocation - all allocated units cannot be allocated to a new context until a time period equal to the allocation period has passed since the units were last allocated;

(g) termination date: a date which is to override the value specified as the end date of the product use authorization 40 - this date must be earlier than specified;

(h) delegation permitted: an override of the delegation permitted flag of the associated license data;

(i) overdraft: the current overdraft level;

(j) overdraft logging: an override of the overdraft logging attribute of the associated product use authorization;

(k) comment: a comment created by the licensee;

(l) extended info: information not defined by the architecture which may be of use in managing the license data.

It will be noted that an assignment and a reservation are identical, the only difference being that a reservation is something optional, while an assignment is something that is required. If the duration is Assignment in the policy declaration of Figure 3, the license manager must assign some or all of the units before units can be allocated. Reservations, on the other hand, are made by the license manager using the management interface, regardless of the policy.

Thus, there are certain attributes that can be changed by a license administrator using the management interface at the server 10, but none of these can result in obtaining more extensive rights to use than granted by the product use authorization. In each case, the license administrator can limit the rights which will be allocated to users in some way that may be appropriate for the administrator for control purposes.

LICENSE DOCUMENT INTERCHANGE FORMAT

The major structural components of an ASN.1 encoded document which conforms to the specifications for the license management system discussed above will be described. The object identifier that is assigned to this data syntax, according to one embodiment, is that specified in ASN.1 as seen in Figure 8. The International Standards Organization or ISO, as it is referred to, defines how bit patterns are chosen to uniquely identify an object type, so the bit pattern set forth in Figure 8 would preceed each document used in the license management system so the document could be identified as being a document conforming to the prescribed License Document Interchange Format.

A document encoded according to this format is represented by a value of a complex data type called "license document interchange format document" of LDIFDocument, in this embodiment. A value of this data type represents a single document. This self-describing data structure is of the syntax defined in the international standard ASN.1 referred to above. The X/Open standard referred to above defines the conventions that must be used in employing this syntax, while the syntax itself is described in an OSI (Open Systems Interconnect, a standard administered by ISO) document identified as X.409 (referenced in the X/Open document identified herein).

The LDIFDocument data type consists of an ordered sequence of three elements: the document descriptor, the document header, and the document itself. Each of these elements are in turn composed of other elements. The overall structure of the LDIFDocument data type will be described, and the nature of the document descriptor and document header types. Then, the document content elements will be described in detail, as well as the various component data types used in the definition of the descriptor, the header and the content.

The LDIFDocument represents a single license document, with the syntax being shown in Figure 9 and the high-level structure of an LDIF document in graphical form being seen in Figure 10. The DocumentDescriptor of Figure 9 is a description of the document encoding, the DocumentHeader contains parameters and processing instructions that apply to the document as a whole, and the DocumentContent is the content of the document, all as explained below.

Referring to Figure 9, what this says is that an LDIFDocument is composed of (::= means "is composed of") a number of elements, the first thing in an

LDIFDocument is a bit pattern (tag) according to an international standard, indicating a certain type of document follows, which is indicated here to be "private" or vendor selected, the number 16373 in this case. Following the bit pattern which functions as a "starting delimiter" it is "implicit" that a "sequence" of elements must follow, where a sequence is distinguished from a set. A sequence is one or more of the elements to follow, whereas a set is exactly one of the elements to be listed. Implicit means that any file identified as LDIFDocument must have a sequence data type, rather than some other type. In the case of Figure 9, the sequence is document-descriptor, document header and document content; the document-content is mandatory, whereas the first two are optional. If an element in the sequence begins with a "0" it is a document-descriptor, "1" means a document-header, and "2" means it is a document-content. Again, it is implicit that the data following is of the format DocumentDescriptor, etc., in each case, and these are defined in Figure 11, Figure 13 and Figure 15.

Each file is in the tag-length-value format mentioned above, and also each element of a file containing multiple elements is of the tag-length-value format. The data stream could be examined beginning at any point, and its content determined by first looking for a tag, which will tell what data structure this is, then a length field will say how long it is, then the content will appear. These structures are nested within one another; a document containing several product-use-authorizations would be an LDIFDocument of the format of Figure 9, with a number of DocumentContent elements of Figure 15 following, with the length given for the LDIFDocument spanning the several PUAs, and the length given for each PUA being for the one PUA.

In Figure 11, the elements major-version and minor-version are seen to be "implicit integer". This means that because the element is of the type major-version, etc.. it must be an integer. Various other implicit types are given in other syntax diagrams, such as character-string, boolean, etc.

In Figure 15, the license body is identified as being of the type "choice" meaning it can be one of PUA, LURT, GroupDefinition, KeyRegistration, etc. Thus, knowing this is a license-body does not mean the data type of the object is known; it is a bit further where the kind of a license-body becomes known. The defintion of a license body is not implicit, but instead is a chioce type.

The contents of the various data elements will now be described in detail with reference to Figures 11-43. Using these detailed descriptions, the exact format of each of the elements used in the LDIF can be interpreted.

The license document descriptor or DocumentDescriptor consists of an ordered sequence of four elements which specify the version level of the LDIF encoding and identify the software that encoded the document, with the syntax being shown in Figure 11. An example of the way a product called PAKGEN V1.0 is expressed in the DocumentDescriptor encoding is shown in Figure 12. The fields in the DocumentDescriptor syntax are major-version, minor-version, encoder-identifier and encoder-name. The major-version field is the primary indicator of compatibility between LDIF processors and the encoding of the present document; this major-version field is updated if changes are made to the system encoding that are not backward compatible. The minor-version field is the revision number of the system encoding. The encoder-identifier field is a registered facility mnemonic representing the software that encoded the document;

the encoder-identifier can be an acronym or abbreviation for the encoder name - this identifier is constant across versions of the encoder. The encoder-identifier should be used as a prefix to Named Value Tags in Named Value Lists to identify the encoder of the named value. The encoder-name field is the name of the product that encoded the document; the encoder-name string must contain the version number of the product.

The document header or DocumentHeader contains data that pertains to the document as a whole, describing the document to processors that receive it; the syntax is shown in Figure 13. An example of a document header is shown in Figure 14, using the hypothetical product PAKGEN V1.0 of Figure 12. The private-header-data contains the global information about the document that is not currently standardized; all interpretations of this information are subject only to private agreements between parties concerned, so a processor which does not understand private header data may ignore that data. The Title field is the user-visible name of the document. The Author field is the name of the person or persons responsible for the information content of the document. The Version field is the character string used to distinguish this version of the document from all other versions. The Date filed is the date associated with this document. Note that the nature and significance of the Title, Author, Version, and Date fields can vary between processing systems.

The content of an LDIF document is represented by a value of a complex data type called DocumentContent. An element of this type contains one or more LicenseData content element using a syntax as shown in Figure 15. There are no restrictions on the number, ordering or context of LicenseData elements. The structure of a LicenseData element is represented in Figure 16. No restrictions

are made on the number, ordering, or context of LicenseData elements. The license-data-header field of Figure 16 specifies that data, common to all types of license data, which describes the parties to the licensing agreement, the term of the agreement, and any constraints that may have been placed on the management of the license data encoded in the license body. The license-body is an element that contains one content element, including: product use authorizations, license unit requirements tables, group definitions, key registrations, and various forms of delegations. The Management-Info is an element that contains information concerning the current state of the license data; this element is not encoded by Issuers.

The license data header, called LicenseDataHeader, is represented as a syntax diagram in Figure 17. The license-id field provides a potentially unique identification of the encoded license data, so issuers of license data can generate unique license-ids to distinguish each issuance of license data; however, the architecture does not require this to be the case, since the only architectural restriction is that no two objects in any single license management domain may have the same value for license-id. The licensee field identifies the party who has received the rights reflected in the license data; there are at least two parties involved in all transfers of license data, first, the issuer of the license data, and second, the licensee or recipient of that data - it is anticipated that individual licensees will specify to those issuing them licenses what the licensee fields on their license data should contain. the term field identifies the term during which the license data may be used; the validity of license data can be limited by issuers to specific time ranges with given starting and ending dates, which are carried in the term element - attempts to use license data or products described by that data either before the start date or after the end date will result in conforming license

managers denying access to the license. Management-constraints identifies constraints placed on the right to manage the associated license data; these constraints can include (a) limiting the set of contexts permitted to manage the data, (b) limiting the set of platforms which may benefit from that management, and (c) limiting the right to backup and delegate the managed data. The signature provides the digital signature used by the issuer to sign the license data and identifies the algorithm used in encoding the signature. Issuer-comment is a comment provided by the issuer and associated with the license data.

The IssuerComment is of an informational nature and does not impact the process of authorizing product or feature use. This field is not included in the fields used to generate the signature for a license, thus, even if specified by an issuer, the IssuerComment can be omitted from a license without invalidating the license. If specified, the IssuerComment should be stored in the appropriate license data base with the associated license data. The IssuerComment can be retrieved by products which use the system and may be of particular utility to products in the "Software Asset Management" domain which are intended to extend or augment the administrative or accounting facilities or basic system components. Some examples of potential uses for this field are order information, additional terms and conditions, and support information. For order information, some issuers may wish to include with their loadable license data some indication of the purchase order or orders which caused the license data to be issued; licensees may find it useful to include this data in their license databases to assist in the license management process. For additional terms and conditions, the system will never provide automatic means for the management of all possible license terms and conditions, and so some issuers may wish to include summaries of non-system managed terms and conditions in the comment as a reminder. For

support information, the IssuerComment could be used to record the phone numbers or addresses of the responsible individuals within the issuing organization who should be contacted if there are problems with the data as issued.

A product use authorization as previously discussed in reference to Figure 2 is used to express the issuance of a right to use some product, product feature, or members of some product group. As such, it records the identity of the product for which use is authorized and specifies the means that will be used by the license manager to ensure that the licensee's actual use conforms to the terms and conditions of the license. Figure 18 illustrates a syntax diagram for a ProductUseAuthorization. Product-id identifies the name of the producer of the product or product feature of which usage rights are being granted as well as the name of that product; in addition, issuers of product use authorizations may specify a range of versions and/or releases whose use is controlled by the specific product use authorization. Units-granted - Contains the number of units of product use which are granted by the license. Management-policy defines the policy which is to be used in managing the granted software usage rights; this definition specifies the Style, Context-Template, Duration, and License Unit Requirements Determination Method which must be used. The calling-authorizations and caller-authorizations are as explained above in reference to calling cards. The execution-constraints field identifies constraints placed on the characteristics of execution contexts which may be authorized to benefit from the units granted by this Product Use Authorization. The product-token filed contains product specific data not interpreted in any way by any processors conformant with the architecture; software product producers 28 use this array to augment the capabilities of conformant license managers.

Some anticipated uses of the token field include language support, detailed feature authorizations, and product support number. For language support, a token could be constructed which contains a list of local language interface versions whose use is authorized; thus, if a product were available in English, German, French and Spanish, a token could be constructed listing only English and German as the authorized languages. For detailed feature authorizations, some license issuers will wish to have very fine control over the use of features in a complex product; however, they may not wish to issue a large number of individual Product Use Authorizations to "turn on" each feature, so these vendors could construct tokens which contain lists of the features authorized or whose use is denied. For product support number, some issuers may wish to include on the product use authorization, and thus make available to the running product, some information concerning the support procedures for the product; for example, an issuer might include the telephone number of the support center or a support contract number, and the product could be designed to retrieve this data from the license manager and display it as part of Help dialogues.

The LURT's or license use requirements tables of Figure 4 provide a means by which issuers of licenses, whose LURDM is dependent on the type of platform on which the product is run, can store information describing the relationship between the platform type and unit requirements. A syntax diagram for a LURT is shown in Figure 19. In Figure 20, an example of how the LURT of Figure 4 might be encoded is illustrated. Lurt-name specifies the name by which the LURT is to be known to conforming license managers. The rows field models a list of multicolumn lurt rows. Platform-id identifies the platform for which this LurtRow provides license unit requirements. The lurt-columns field provides a list of one or more lurt column values; the first value provided is

assigned to column-1 of the lurt-row, the second value provided is assigned to column-, etc. A lurt column value of -1 indicates that use of the product or feature is not authorized, while a lurt column value of 0 or greater indicates the number of units that must be allocated in order to authorize product use on the platform described by this lurt-row. All unspecified columns (e.g., columns whose number is greater than the number of column values provided in the lurt columns element) will be considered to contain the value -1.

In reference to Figure 19, to use the row-selector feature mentioned above, the platform-ID element would be replaced with *row-selector* which would be implicit of Context. Also, in Figure 34 described below, in the lurdm-kind element, *row-selector* would be included if the row-select feature is to be used.

As discussed above, Figure 4 provides an example of a hypothetical LURT, illustrating the LURT mechanism, where the issuer of this LURT table has established three unit requirement tiers for use in determining the unit requirements for that issuer's products. Figure 20 provides an example of how the LURT presented in Figure 4 might be encoded.

A group definition is used to define and name a license group. Once so defined, the name of this group can be used on product use authorizations in the same manner as a product name. Since a single product use authorization specifies the management policy for all members of the group, the members of that group must be compatible in their licensing styles, i.e., a personal use type product can not be mixed with a concurrent use product in the same group. Figure 21 shows a group definition syntax diagram. Group-name is the name which must appear on Product Use Authorizations for this group. Group-version

specifies the current version of this group; the requirements for matching between the version information on a product use authorization and that on a specified group definition are the same as those rules which require matching between produce use authorizations and the Release Date data provided by products. Group-members lists those products or features which are components of the named group.

A key registration is used by a producer 28 or issuer 25 who have been registered as authorized license issuers and provided with an appropriate public and private key pair. The key registration identifies the public key which is to be used by conforming license managers 10 in evaluating signatures 53 created by the named issuer 25 or producer 28. A key registration syntax diagram is shown in Figure 22. Key-owner-name provides the name which must be used in either of, or both, of the Producer and Issuer fields of license data generated by the issuer; the key-owner-name must be identical to that specified in the Issuer field of the header record. Key-algorithm identifies the registered algorithm that is to be used when producing digital signatures with this key. Key-value identifies the public key.

An issuer delegation is typically issued by a producer 28 and authorizes the named issuer 25 to issue licenses for products produced by the producer. An issuer delegation syntax diagram is shown in Figure 23. Delegated-issuer-name identifies the name which must appear in the Issuer field of any Product Use Authorization generated using the License Issuer Delegation. Delegated-product-id identifies the products whose licenses the named issuer is authorized to issue. Delegated-units-granted, if specified, indicates that the use of this IssuerDelegation is to be managed in the style of a consumptive license; the value of this attribute

gives the number of units for which license documents may be generated (i.e., if granted 1000 units by a Producer, an Issuer can only issue 1000 units.) Template-authorization provides a "template" Product Use Authorization whose attribute values must be included on any Product Use Authorization generated using this IssuerDelegation; in the case of attributes which have a scalar value (i.e., Version, Release Date, etc.), the Issuer may issue licenses with more restrictive values than those specified on the Template Authorization. Sub-license-permitted indicates whether the Issuer identified on this IssuerDelegation may issue an IssuerDelegation for the delegated-product-id.

A license delegation, as shown in a syntax diagram of Figure 24, is used to delegate the right to manage license data. Such delegations are created by the licensee (by the license manager), if authorized by the issuer 28. A backup delegation, also shown in Figure 24, is used by one license management facility to authorize another to manage the delegated rights in the case that the delegating license manager is not running. The delegated-units field specifies the number of units whose management is being delegated; this may only be specified when a product use authorization is being delegated. Delegation-distribution-control defines the mechanisms by which the distribution and refreshing of the delegation will be accomplished. Delegatee-execution-constraints identifies any constraints which are placed on the execution-context of the Delegatee; these constraints are applied in addition to those which are a part of the delegated License Data. Assignment-list identifies any assignments of the delegated units that must be respected by the delegatee. Delegated-data stores a copy of the LicenseData received from the issuer that is the subject of the delegation; the delegated data is not provided when the LicenseDelegation element is included in a DelegationList.

The management information or ManagementInfo element records information concerning the current state of the LicenseData with which it is associated. A syntax diagram of the ManagementInfo element is shown in Figure 25. The assignments field identifies a list of one or more assignments which may be outstanding for the units on the associated product use authorization. Reservations identifies a list of one or more reservations which may be outstanding for the units on the associated product use authorization. Delegations identifies a list of all outstanding delegations. Backup-delegations identifies all outstanding backup delegations. the allocations field provides detailed information about outstanding allocations which involve units from the associated product use authorization. Registration-date is the date on which the LicenseData was registered in the license database. Registrar is the context which caused the LicenseData to be registered. Local-comment is a comment field. Termination-date means a license defined date after which the license data may not be used; this date must be earlier than the end-date specified in the license data's term record. The extended-info field allows additional information concerning the state of the LicenseData and its handling by the license manager that is not standardized.

The defined types of elements will now be described. These defined type are:

Allocation	ManagementPolicy
Assignment	Member
Context	NamedValue
DistributionControl	NamedValueList
ExecutionConstraints	ProductID
IntervalTime	Signature

LicenseID	Term
LUDRM	Version
ManagementConstraints	

The allocation element records the information concerning a single unit allocation, and is shown in a syntax diagram in Figure 26. Allocation-context specifies the context to which the allocation was made. The allocation-lur field specifies the license unit requirement which applies to the allocation-context; this license unit requirement is calculated without consideration of any allocation sharing which may be possible. The allocation-group-id field identifies the "allocation-group" for the current allocation, in which an unshared allocation will always have an allocation group id of 0; allocations which utilize shared units will have an allocation group id which is shared by all other allocations sharing the same units.

The assignment element is shown in syntax diagram in Figure 27. Assigned-units identifies the number of units which are assigned. Assignment-term identifies the start and end of the assignment period. Assignee identifies the context to which the assignment is made.

The context element is shown in syntax diagram in Figure 28. The SubContext-type field identifies the type of subcontext, and this type can be either standard or private; if standard, the type value will be taken from the standard-subcontext-type enumeration: (a) network-subcontext means the subcontext value identifies a network; (b) execution-domain-subcontext means the subcontext value is the name of the management domain within which the caller is executing; (d) login-domain-subcontext means the subcontext value is the name of the

management domain within which the user of the caller was originally authenticated or "logged in"; (d) node-subcontext means the subcontext value is the name of a node; (e) process-family-subcontext means the subcontext value is an implementation specific identifier for a group of related processes; (f) process-ID-subcontext means the subcontext value is an implementation specific process identifier; (g) user-name-subcontext means the subcontext value is a user name; (h) product-name-subcontext means the subcontext value is the same as the product name found on the Product Use Authorization; (i) operating-system-subcontext means the subcontext value is a character string representation of the name of the operating system; (j) platform-ID-subcontext means the subcontext value is an identifier that describes the hardware platform supporting the context. The subcontext-value field is the value of the subcontext.

As discussed above, license data is always used or allocated within, or for the benefit of, some named licensing context. This context name is constructed by concatenating the values of all subcontexts into a single context name. A Context Template specifies those components of the context name which should be used in calculating license unit requirements. The management system determines the need to perform a unit allocation each time license units are requested. The full context on whose behalf the allocation should be made is obtained for each requested authorization. The system will mask the full context to exclude all sub-contexts not specified in the context template and then determine if the resulting context already has units allocated to it. If not, units will be allocated according to the specification of the LURDM, otherwise, the units previously allocated will be shared by the new context. Thus, if a given product authorization contains a context specification of NODE + USER_NAME, each context which requests license unit allocations and which has a unique pair

of NODE + USER_NAME subcontext values will require an explicit grant of license units to be made. On the other hand, any contexts which share the same pair of NODE and USER_NAME subcontext values will be able to "share" a single allocation of license units. The requirement for specific allocations of units and the ability to share units is exhibited in Figure 29 which attempts to provide a "snapshot" of the units allocated for the product FOOBAR V4.1 at a particular instance. It is seen from the figure that although presented with five unique full contexts, only four of them are unique when looking only at those portions of each context which are described by the Context Template (ie: NODE + USER_NAME). A unit allocation must be made for each of the four instances of unique contexts, when masked by the Context Template. The fifth context can share allocated units with another context. Thus, the total requirement to support product use as described in this example would be 40-units (ie: four allocations of ten units each). Significant changes in the unit requirements can be achieved by making small modifications to the Context Template. Figure 30 shows the same contexts as in Figure 29 but a Context_Template of NODE. The total unit requirement for this example would be three units (three allocations of ten units each) rather than the forty units required in the previous example.

The distribution control element defines the mechanism that will be used for distributing the subject delegation and records some status information concerning the distribution of that delegation. A syntax diagram of the distribution control element is shown in Figure 31. Distribution-method identifies the means by which the delegation will be distributed, and the alternatives are refresh-distribution, initial=distribution-only, and manual-distribution. Refresh-distribution means the license manager shall be responsible for the initial distribution of the delegation and for ensuring that refresh delegations are

properly distributed. Initial-distribution-only means the license manager shall be responsible for the initial distribution of the delegation, however, distribution of refresh delegations will be made by some other means. Manual-distribution means the distribution of the delegation will be under the control of some other mechanism (perhaps a license asset manager). Current-start-date is the time that the last successful initial or refresh delegation distribution was performed. Current-end-date identifies the last date on which the most recent delegation distribution was performed. Refresh-interval identifies the period of time between attempts to refresh the delegation; the refresh-interval may not be longer than the maximum-delegation-period and should normally be less than that in order to ensure that refresh delegations are distributed prior to the expiration of the previous delegations that they are replacing. Retry-interval identifies the amount of time to wait for an unsuccessful distribution attempt to try again. Maximum-retry-count identifies the maximum number of times that an unsuccessful distribution attempt may be retried. Retries-attempted records the number of unsuccessful retry attempts which have been made since the last successful initial or refresh delegation distribution was performed.

The execution constraints elements place limits on the environments and contexts which may receive allocations. A syntax diagram of the execution constraints element is shown in Figure 32. Operating-system contains a list of zero or more operating systems on which the use of the subject license is authorized; if no operating systems are specified, it is assumed that license use is authorized on all operating systems. Execution-context specifies a list of zero or more full or partial context names which identify the contexts within which products described by the license data may be executed; if no context names are specified, the licensed products may be executed in any context controlled by the licensee.

Environment-list identifies those environments within which the licensed product may be used.

The interval time element is defined by the syntax IntervalTime ::= UTCTime.

The license ID element uniquely identifies the license data it is associated with, and is described by the syntax diagram of Figure 33. Here issuer uniquely identifies the issuer of the license data as well as the name space within which the LicenseID Number is maintained. While the issuer name will typically be the same as the name of the issuer's company or personal name, this is not a requirement. For instance: The issuer name for Digital Equipment Corporation is "DEC," an abbreviation of the corporate name. Valid contents of the Issuer field are maintained in the an Issuer Registry. The serial-number provides a unique identification or serial number for the license data. The amendment field is an integer which is incremented each time license data is amended by its issuer, with the first version of any license data carries the amendment number 0; an amendment can only be applied to license data if that license data has identical Issuer and Number values and an amendment number less than the number of the amendment to be applied.

The license units requirements determination method or LURDM element is shown in syntax diagram in Figure 34. The combination-permitted field indicates whether conforming license managers are permitted to combine together into a common pool the units from different product use authorizations if those produce use authorizations have the same product record value; for example, if combination is permitted and a single license manager discovers in its database

two 500-unit authorizations for the use of DEC Cobol, the license manager would be permitted to combine these two authorizations into a logical grant of 1000 units. The overdraft-limit modifies the behavior of a conforming license management facility in those cases where it is found that there are zero or fewer license units available for use at the time of a request for the allocation or consumption of additional license units. Operation of overdraft is different depending upon whether allocative, or consumptive style is being used. In using with allocative style, an allocation is granted even though the remaining units are zero or less, up to the overdraft-limit. In using with consumptive style, the license is authorized to accumulate a negative balance of license units, up to the overdraft-limit. Overdraft-logging-required indicates whether all license grants which are the result of overdraft use must cause a log record to be generated. When the allocation-size field is non-zero, then all unit allocations and delegations must be made in sizes which are whole number multiples of the allocation-size value. Lurdm-kind identifies the method by which license unit requirements will be calculated once the requirement for an allocation has been discovered, the permitted alternatives being (a) LURT which specifies that license unit requirements are to be determined by lookup in the LURT which is associated with the current license, (b) Constant which specifies that license unit requirements are constant for all platforms on which the licensed product or product feature may run, and (c) Private-LURDM which specifies that license unit requirements are to be determined by the licensed product, not by the license management facility. The named-lurt-id specifies the name of the LURT table to be used in determining license unit requirements if the LURDM-kind is specified as LURT; if the LURDM-kind is specified as LURT and no table is explicitly named, the name of the table to be used is constructed from the issuer name on the product use authorization. Lurdm-value specifies the LURT column to be

used when LURDM-kind = LURT; however, when LURDM-kind = Constant, the Lurdm-value field contains the precise number of units to be allocated or consumed. Default-unit-requirement specifies the unit requirement value to be used when the appropriate LURT does not have a row corresponding to the appropriate platform ID; when specified on a product use authorization with Style = Allocative, the context template will change to Process + Product_Specific and the Duration will change to Transaction in cases of unrecognized Platform ID's.

The management constraints element is shown in a syntax diagram in Figure 35. The management-context field specifies a list of zero or more partial context names which identify the specific contexts within which the license data may be managed. If no management contexts are specified, the license data may be managed within any context controlled by the licensee. The contexts used in specifying Management Context Constraints may only contain the Network, Domain, and Node subcontexts. Specifying a list of management contexts does not effect whether or not the license data can be used within other contexts. For example, unless otherwise restricted, license data with a specified management context can be remotely accessed from or delegated to other nodes in a network. The management-scope field defines the maximum permitted size of the license management domain within which the license data may be managed or distributed, these being single-platform, management-domain, or entire-network. Single-platform constrains the license management domain for the subject license data to be no larger than a single platform. Management-domain constrains the license management domain for the subject license data to be no larger than a single management domain. Entire-network constrains the license management domain for the subject license data to be no larger than a single wide area network; that

network which contains the platform on which the license units were initially loaded. Although technology may not exist to detect the interorganizational boundaries of a wide area network (i.e., what is on the Internet as opposed to being on a company's own network), the assumption is that interorganization and internetwork sharing of licenses will normally be considered a violation of license terms and conditions. The backup-permitted field indicates if the Issuer has authorized the use of backup delegations for this data. Delegation-permitted indicates if the Issuer has authorized the licensee to delegate this data. Maximum-delegation-period identifies the longest interval during which a delegation may be valid; by default, delegations have a life of 72-hours.

The major elements of the management policy specification are shown in Figure 3, as previously discussed. A syntax diagram for the management policy element is shown in Figure 36. For the Style field, three fundamental styles of license management policy are supported, allocative, consumptive, and private-style, as explained above. Only one of these styles may be assigned to any single product use authorization. The Context-template specifies those components (sub-contexts) of the execution-context name which should be used in determining if unit allocations are required. The Duration defines the duration of an allocation of license units to a specific context or the duration of the period which defines a valid consumptive use. For durations of type "Assignment," the specification of a Reassignment Constraint is also provided for. Three types of Duration_Kind are supported, these being Transaction, Assignment and Immediate, as explained above. The lur-determination-method stores information used in calculating the number of units that should be allocated or consumed in response to a license request. The allocation-sharing-limit identifies the largest number of execution contexts that may share an allocation made under this management policy; an

allocation-sharing-limit of 0 indicates that the number of execution contexts that may share an allocation is unlimited. The reassignment-constraint specifies a minimum duration of assignment; although there is normally no constraint placed on how frequently granted units may be reassigned, an issuer may constrain reassignment by specifying this minimum duration of an assignment, in which case reassignment of assigned units will not be supported until the amount of time specified in the Reassignment Constraint has passed. If an assignment of some particular set of units has been delegated and the delegation period for that delegation has not terminated, cancellation of the delegation must be performed prior to reassignment.

The member element identifies a specific licensed product which may be part of a calling authorization or group definition, and is shown in syntax diagram in Figure 37. Member-product identifies the product which is a member. Member-signature is constructed from the product and token fields of the called member structure as well as the product and issuer fields of the calling product. Member-token provides the data which should be used as the product token for this member.

Named values are data elements with a character string tag that identifies the data element, and have a syntax as shown in Figure 38, which also shows the syntax for ValueData and named value list. A named value list models a list of named values, with an example being shown in Figure 39. In Figure 38, Value-Name uniquely identifies the value; no standard value names are defined, and the period character can be used as a part of the value name to form a hierarchical tag registry at the discretion of the issuer. Value-data is the data that has been named; data types are selected from the possible Value Data types, seen in the

Figure. Value-boolean means the named data is a boolean value. Value-integer means the named data is an integer value. Value-text means the named data is a StringList value. Value-general means the named data is a stream of bytes in any format. Value-list means the named data is a list of named data values.

The product ID explicitly identifies the product which is the subject of the license data with which it is associated, with the syntax for ProductID being shown in Figure 40. The version and release date fields provide a mechanism for defining which specific instances of the licensed product are described in the associated license data. The Producer field is a registered name which identifies the producer of the licensed feature; in the case of Group Names, the Producer is always also the Issuer of the group. The Product-name identifies a licensed software feature. The First-version identifies the earliest version of the product whose use is authorized. The Last-version identifies the latest version of the product whose use is authorized. The First-release-date identifies the earliest release of the product whose use is authorized. The Last-release-date identifies the latest release of the product whose use is authorized. Conforming license managers are required to interpret the contents of these fields in the most restrictive way possible. Thus, if a license is issued with Last-version = 3.0 and a Last-release-Date of 1-Jan-1991, then the use of version 2.0 of the licensed product would be unauthorized if it had a release date of 2-Jan-1991. If either a First-version or First-release-date is specified without a matching Last-version or Last-release-date, use of the produce is authorized for all versions or release dates following that specified. Similarly, if either a last-version or Last-release-date is specified without a matching First-version or First-release-date, use of the produce is assumed to be authorized for all versions or release dates prior to that specified. Issuers should typically only specify one of either First-version or First-release-

date. This is the case since it is anticipated that these fields will typically refer to events which occurred prior to the moment of license data issuance. Thus, it should normally be possible for the issuer to state unambiguously with only one of these two fields which is the oldest implementation of the product that is to be authorized. The architecture does permit, however, both fields to be used in a single product authorization.

The signature element is used to establish the integrity and authorship of the license data with which it is associated. A syntax diagram for the signature element is shown in Figure 41. The Signature-algorithm field identifies the registered algorithm that was used to produce the digital signature. Signature-parameters are the values of the algorithm's parameters that are to be used; the need for and syntax of parameters is determined by each individual algorithm. Signature-value is an enciphered summary of the information to which the signature is appended; the summary is produced by means of a one-way hash function, while the enciphering is carried out using the secret key of the signer (Issuer).

The term element defines an interval during which the license data is valid, and is shown in syntax diagram form in Figure 42. The fields are start-date and end-date. Start-date identifies the first date of the term; if not specified, the license data is considered valid on any date prior to the end-date. End-date identifies the last date of the term; if not specified, the license data is considered valid on any date after the Start-date. While the Start-date is always either omitted or specified as an absolute date, the End-date can be either absolute or relative. If the End-date is specified as a relative or "interval" date and the Start-date has been omitted, the date of license registration will be used as the effective

start date in computing the valid term of the license data. It should be noted that the system does not specify the mechanism by which system dates are maintained by platforms supporting system components. Instead, the system always accepts that system time returned to it as correct. Thus, the reliability of the management of license data which specifies terms is dependent on the time management function of the underlying platform.

The version element identifies a four-part version of the licensed software product or feature. A syntax diagram of the version element is shown in Figure 43. The schematics of each of the four parts is not detailed, but it is required that producers who wish to permit version ranges to be specified on product use authorizations ensure that the collating significance of the four parts is maintained. When comparing versions, Part-1 is considered first, then Part-2, then Part-3, and finally, Part-4. Part-1 identifies a major modification to the versioned object. Part-2 identifies a modification to the versioned object which is less significant than a modification which would cause a change in the Part-1 value. Part-3 identifies a modification to the versioned object which is less significant than a modification which would cause a change in the Part-2 value. Part-4 identifies a modification to the versioned object which is less significant than a modification which would cause a change in the Part-3 value.

FILTERS

An important feature is the use of filters in the license management program 11, including the client interface 31 and the management interface 33. A filter is used is select items in the license database 23, for example. Various

selection mechanisms are used in picking out or doing lookups in database technology; filters are one of them. The filter engine used in the license management system 11 of Figure 1 is generally of a known construction, with the exception of the select filter item type as will be described, which allows a complex rather than a flat data format to be selected from. The feature that is of importance to this embodiment is the way of specifying items as an input to the filter function , rather than the filter function itself. Thus, there is described below a template for specifying input to the filter engine. This is as if a form were used as the input, with blanks on the form; by filing in certain blanks these would be the items selected on, the blanks not filled in would be "don't care".

An instance of the class *filter* is a basis for selecting or rejecting an object on the basis of information in that object. At any point in time, a filter has a value relative to every object - this value is false, true or undefined. The object is selected if and only if the filter's value is true. This concrete class has the attributes of its superclass - *Object* - and the specific attributes listed in the table of Figure 44.

A filter is a collection of simpler filters and elementary filter-items together with a Boolean operation. The filter value is undefined if and only if all the component filters and filter-items are undefined. Otherwise, the filter has a Boolean value with respect to any object, which can be determined by evaluating each of the nested components and combining their values using Boolean operation (components whose value is undefined or ignored). The attributes specific to *filter* as shown in Figure 44 are (a) *filter items* which are a collection of assertions, each relating to just one attribute of an object, (b) *filters* which are a

collection of simple filters, and (c) *filter type* which is the filter's type, of one of the following values: And, Or, Not.

An instance of the class *filter item* is a component of a *filter*. It is an assertion about the existence or values of a single attribute of a license data object or one or its subobjects. This concrete class has the attributes of its superclass - *object* - and the specific attributes listed in the table of Figure 45.

The value of a filter item is undefined if: (a) the Attribute Types are unknown, or (b) the syntax of the Match Value does not conform to the attribute syntax defined for the attribute type, or (c) a required Attribute is not provided. The attributes specific to *filter item* as shown in Figure 45 are (a) *filter item type* which identifies the type of filter item and thereby the nature of the filter, and its value must be one of

equality	less
inequality	present
greater or equal	select
less or equal	request candidates
greater	simulate request

(b) *attribute type* which identifies the type of that attribute whose value or presence is to be tested; the value of All Attributes may be specified, (c) *match value* which is the value which is to be matched against the value of the attribute, (d) *filter* which identifies the filter to be used in evaluating a selected subobject of the current object; the filter is ignored if the *filter item type* is not *select* or if the specified attribute type is not present in the object, and upon evaluation of the *filter* the value of *filter item* will be set to that of the *filter*, (e) *initial substring*, if present, this is the substring to compare against the initial portion of the value of

the specified attribute type, (f) *substring*, if present, this is the substring(s) to compare against all substrings of the value of the specified attribute type, (g) *final substring*, if present, this is the substring to compare against the final portion of the value of the specified attribute type, and (h) *license request*, if present, this is license request against which the appropriate license data objects should be evaluated; this attribute may only be specified if the value of the filter item type is either Request Candidates or Simulate Request.

An instance of enumeration syntax *Filter Type* identifies the type of a filter. Its value is chosen from one of the following: (a) *And* means the filter is the logical conjunction of its components; the filter is true unless any of the nested filters or filter items is false, or if there are no nested components, the filter is true; (b) *Or* means the filter is the logical disjunction of its components; the filter is false unless any of the nested filters or filter items is true, or, if there are no nested components, the filter is false; (c) *Not* means the result of the filter is reversed; there must be exactly one nested filter or filter item, and the filter is true if the enclosed filter or filter item is false, and is false if the enclosed filter or filter item is true.

An instance of enumeration syntax *Filter Item Type* identifies the type of a filter item. Its value is chosen from one of the following: (a) *Equality* which means the filter item is true if the object contains at least one attribute of the specified type whose value is equal to that specified by Match Value (according to the equality matching rule in force), and false otherwise; (b) *Inequality* which means the filter item is true if the object contains at least one attribute of the specified type whose value is not equal to that specified by Match Value (according to the equality matching rule in force), and false otherwise; (c) *Greater*

or Equal which means the filter item is true if the object contains at least one attribute of the specified type whose value is equal to or greater than the value specified by Match Value (according to the matching rule in force), and false otherwise; (d) *Less or Equal* which means the filter item is true if the object contains at least one attribute of the specified type whose value is equal or less than the value specified by Match Value (according to the matching rule in force), and false otherwise; (e) *Greater* which means the filter item is true if the object contains at least one attribute of the specified type whose value is greater than the value specified by Match Value (according to the matching rule in force), and false otherwise; (f) *Less* which means the filter is true if the object contains at least one attribute of the specified type, whose value is less than the value specified by Match Value (according to the matching rule in force),and false otherwise; (g) *Present* which means the filter item is true if the object contains at least one attribute of the specified type, and false otherwise; (h) *Select* which means the filter item is true if the object contains at least one attribute of the specified type which has an object syntax and when the Filter is evaluated against the attributes of that object the Filter is true, and false otherwise; (i) *Request Candidates* which means the filter item is true if the object against which it is evaluated is one which could be used to provide some or all of the units requested by the specified License Request; the evaluation is made independently of any outstanding allocations or preallocations; and (j) *Simulate Request* which means the filter item is true if the object against which it is evaluated is one which would be used to provide some or all of the units requested by the specified License Request.

The Request Candidates and Simulate Request filter item types are of special use in testing and prototyping of systems by a license manager at a

licensee's site. For example, the license manager can simulate the effect of potential assignments, the effect of a population of certain types on a network, etc.

As an example, Figure 46 shows how a filter may be constructed to identify "All Product Use Authorizations issued by Digital for the Product 'Amazing Graphics System' which contains a calling authorization for Digital's 'Amazing Database' Product". This example is in the international standard format referred to as X.409 as mentioned above.

Filters can also used in a request allocation, being specified in a request extension as explained above. That is, a filter is one of the optional items in a request extension. For example, if a user wanted to use a version of WordPerfect with French language extension, and there were version with and without on the network, his request allocation would have a request extension that specified a filter for "French" in the token field. In this manner, a product can describe itself more richly. The filter in the request extension can be a Required filter or a Preferred filter, meaning the feature such as "French" is either absolutely necessary, or merely the preferred.

While this invention has been described with reference to specific embodiments, this description is not meant to be construed in a limiting sense. Various modifications of the disclosed embodiments, as well as other embodiments of the invention, will be apparent to persons skilled in the art upon reference to this description. It is therefore contemplated that the appended claims will cover any such modifications or embodiments as fall within the true scope of the invention.

WHAT IS CLAIMED IS:

1. A method of managing use of licensed software items, said software items separately executable on a computer system or accessible by said computer system, the computer system including a processor and one or more nodes, comprising the steps of:

maintaining by said processor a store of license authorizations for said software items; each license authorization including an indication of license management policy for a software item, said indication having a plurality of sets of policy components, said sets of policy components granting alternatives of specified restrictive rights to selectively access and execute said software items in said system; said indication of license management policy being in the format of an encoded document of a data type consisting of an ordered sequence of elements;

accessing said store by said processor to modify in said store one or more of said specified restrictive rights of said policy components of an identified license authorization;

accessing said store by said processor using a filter to obtain information from said license authorization for a selected software item, in response to a request from a node, and

comparing an identification of said node and said software item with said information, to produce and send to said node a grant or refusal of said request.

2. A method according to claim 1 including the step of receiving said license authorizations , for storing in said store,

from a license grantor external to said processor, and wherein said step of accessing said store to modify in said store one or more of said specified restrictive rights employs management functions executable on said processor but not on said nodes or said license grantor to identify a license authorization in said store.

3. A method according to claim 1 wherein said indication is in the format of an encoded document of a data type consisting of an ordered sequence of three elements, the three elements including a document descriptor, a document header and the document content.

4. A method according to claim 1 wherein said filter specifies one or more of said attributes and a Boolean operator for each selected attribute.

5. A method according to claim 2 wherein said step of accessing said store to modify one or more of said policy components is to allow grant of rights to use which are more restrictive than said specified restrictive rights.

6. A method according to claim 2 including the steps of:

sending a request by a user of one of said software items to obtain permission to use said software item; said request identifying the user and said software item;

accessing said store to obtain information from said license authorization for said software item, in response to said request, and comparing said identification of said user and said software item with said information, to produce a grant or refusal of said request for sending to said user.

7. A method according to claim 6 wherein said store is maintained by a license server, and said request is sent to said server and wherein said request is in the form of a remote procedure call, and said grant or refusal sent to said user is a return of said procedure call.

8. A method according to claim 7 wherein said license authorization is a data arrangement specified as a product use authorization, and said product use authorization is received by said server from an issuer, and wherein said server and said users are nodes on a computer network.

9. A method according to claim 2 wherein said policy components include a termination date, and said management functions can modify said termination date to an earlier termination date and wherein said policy components include a right of delegation of a right to grant said requests to another server, and said management functions can modify said right of delegation to remove said right of delegation.

10. A method according to claim 2 including storing in association with said license authorization a number of management attributes, and said management functions being able to modify said management attributes.

11. A method according to claim 10 wherein said management attributes include a reservation of units of license use granted by said license authorization so that said units will not be granted to a user in response to said request, and wherein said management attributes include an allocation of units of license use to a specific context.

12. A method according to claim 10 wherein said management attributes include an allocation period which is the minimum duration of an allocation of units, and wherein said management attributes include permission to enable a backup delegation of the right to grant said requests.

13. A system for managing use of licensed software products, comprising: means for maintaining a store of license documents, one for each said product; each license document including an indication of license policy having plurality of sets of policy components granting specified restrictive rights to use said software products, said policy components in each set providing alternatives;

a management interface for accessing said store to modify

selected ones of said components of an identified license authorization.

14. A system according to claim 13 including:

means for sending a request from a user of one of said products to obtain permission to use said product; said request identifying the user and said product;

means for accessing said store to obtain information from said license document for said product, in response to said request, and for comparing said identification of said user and said product with said information, and with constraints imposed by said policy components, to produce a grant or refusal of said request and send said grant or refusal to said user.

15. A system according to claim 13 wherein said management interface can modify said selected ones of said components to allow grant of rights to use which are more restrictive than said specified restrictive rights and wherein said means for maintaining, and said means for accessing and sending to said user are all located at a server on a distributed network, and said means for sending a request is located at a user node on said network.

16. A system according to claim 14 wherein said request is in the form of a remote procedure call, and said grant or refusal sent to said user is a return of said procedure call, and wherein said

license document is a data arrangement specified as a product use authorization, and said product use authorization is received by said server from a license issuer.

17. A system according to claim 13 wherein said policy components include a termination date, and said management functions can modify said termination date to an earlier termination date, and wherein said policy components include a right of delegation of a right to grant said requests to another server, and said management functions can modify said right of delegation to remove said right of delegation.

18. A system according to claim 15 including means for storing in association with said license authorization a number of management attributes, wherein said management functions are able to modify said management attributes and wherein said management attributes include a reservation of units of license use granted by said license authorization so that said units will not be granted to a user in response to said request.

19. A system according to claim 18 wherein said management attributes include an allocation of units of license use to a specific context.

20. A system according to claim 18 wherein said management attributes include an allocation period which is the minimum

duration of an allocation of units, and include permission to enable a backup delegation of the right to grant said requests.

21. A method according to claim 3 wherein said document descriptor includes an encoding method version number, and encoder-identifier and an encoder-name, and wherein said document-header includes a title, an author, a version and a date for the software item.

22. A method according to claim 3 wherein said document content includes at least one of the following:

a product-use-authorization;

a license-use-requirements-table;

a group-definition;

a key-registration;

a delegation.

23. A method according to claim 3 wherein said document-content includes a license-data-header, and said license-data-header describes the parties to the license document, the term of the agreement and constraints that may have been placed on management of the license data.

24. A method according to claim 3 wherein said document-content includes management-info, where the management-info may include at least one of the following:

an assignment;

a reservation;

a delegation;

a backup delegation;

an allocation;

a registration date;

a registrar;

a comment;

a termination-date.

25. A method according to claim 3 wherein:

said document descriptor includes an encoding method version and a date for the software item;

said document content may include at least one of the following: a product-use-authorization, a license-use-requirements-table, a group-defination, a key-registration, and a delegation;

said document-content selectively includes a license-data-header, and said license-data-header describes the parties to the license document, the term of the agreement and constraints that may have been placed on management of the license data;

said document-content may have been placed on management of the license data;

said document-content selectively includes management-info, where the management-info may include at least one of the following: an assignment, a reservation, a delegation, a backup delegation, an allocation, a registration date, a registrar, and a

comment.

26. A method according to claim 3 wherein said store is maintained by a license server, and said request is sent to said server, and wherein said server and said users are nodes on a computer network.

27. A method according to claim 3 wherein said request is in the form of a remote procedure call, and said grant or refusal sent to said user is a return of said procedure call, and wherein said license authorization is received by said server from an issuer.

28. A method according to claim 3 including the steps of: sending a request by a user of one of said software items to obtain permission to use said software item; said request identifying the user and said software item;

sending said grant or refusal to said user.

29. Apparatus for managing use of licensed software items, comprising:

means for maintaining a store of license authorizations for said software items; each license authorization including an indication of license management policy for a software item, said indication being in the format of an encoded document of a data type consisting of an ordered sequence of three elements, the three elements including a document descriptor, a document header and the

document content;

means for sending a request by a user of one of said software items to obtain permission to use said software item; said request identifying the user and said software item;

means for accessing said store to obtain information from said license authorization for said software item, in response to said request, and comparing said identification of said user and said software item with said information, to produce a grant or refusal of said request;

means for sending said grant or refusal to said user.

30. Apparatus according to claim 29 wherein said document descriptor includes an encoding method version number, and an encoder-identifier and an encoder-name, and wherein said document-header includes a title, an author, a version and a date for the software item.

31. Apparatus according to claim 29 wherein said document content includes at least one of the following:

a product-use-authorization;

a license-use-requirements-table;

a group-definition;

a key-registration;

a delegation.

32. Apparatus according to claim 29 wherein said document-content includes a license-data-header, and said license-data-header describes the parties to the license document, the term of the agreement and constraints that may have been placed on management of the license data.

33. Apparatus according to claim 29 wherein said document-content includes management-info, where the management-info may include at least one of the following:

an assignment;

a reservation;

a delegation;

a backup delegation;

an allocation;

a registration date;

a registrar;

a comment;

a termination-date.

34. Apparatus according to claim 29 wherein:

said document descriptor includes an encoding method version number, and encoder-identifier and an encoder-name;

said document-header includes a title, an author, a version and a date for the software item;

said document content may include at least one of the following: a product-use-authorization, a license-use-requirements-

table, a group-definition, a key-registration, and a delegation;

said document-content may include a license-data-header, and said license-data-header describes the parties to the license document, the term of the agreement and constraints that may have been placed on management of the license data;

said document-content may include management-info, where the management-info may include at least one of the following: an assignment, a reservation, a delegation, a backup delegation, an allocation, a registration date, a registrar, and a comment.

35. Apparatus according to claim 29 wherein said store is maintained by a license server, and said request is sent to said server, and wherein said request is in the form of a remote procedure call, and said grant or refusal sent to said user is a return of said procedure call.

36. Apparatus according to claim 29 wherein said license authorization is received by said server from an issuer, and wherein said server and said users are nodes on a computer network.

37. A method of storing license documents by a server for a license management system, comprising the steps of:

maintaining a store of license documents for software items; each license document including an indication of license management policy for a software item, said indication being in the format of an encoded document of a data type consisting of an ordered

sequence of three elements, the three elements including a document descriptor, a document header and the document content;

accessing said store to obtain information from a selected one of said license documents for a software item, in response to a request, and referencing said indication of license management policy, to produce a grant or refusal of said request.

38. A method according to claim 37 wherein said document descriptor includes an encoding method version number, an encoder-identifier and an encoder-name, and wherein said document-header includes a title, an author, a version and a date for the software item.

39. A method according to claim 37 wherein said document content includes at least one of the following:

a product-use-authorization;

a license-use-requirements-table;

a group-definition;

a key-registration;

a delegation.

40. A method according to claim 4 wherein said step of selecting by a filter may select on one or more of the attributes: issuer, producer, product name, product use authorization, calling authorization, and wherein said store is maintained by a license server, and said request is sent to said server.

41. A method according to claim 4 wherein said request is in the form of a remote procedure call, and said grant or refusal sent

to said user is a return of said procedure call.

42. A method according to claim 40 wherein said license authorization is a data arrangement specified as a product use authorization, and said product use authorization is received by said server from an issuer, and wherein said server and said users are nodes on a computer network.

43. Apparatus for managing use of licensed software items, comprising:

means for maintaining a store of license authorizations for said software items; each license authorization including an indication of license management policy for a software item, said indication being an encoded document containing a number of attributes defining said license policy;

filter means for selecting from said store, said filter means specifying one or more of said attributes and a Boolean operator for each selected attribute;

means for sending a request by a user of one of said software items to obtain permission to use said software item; said request identifying the user and said software item;

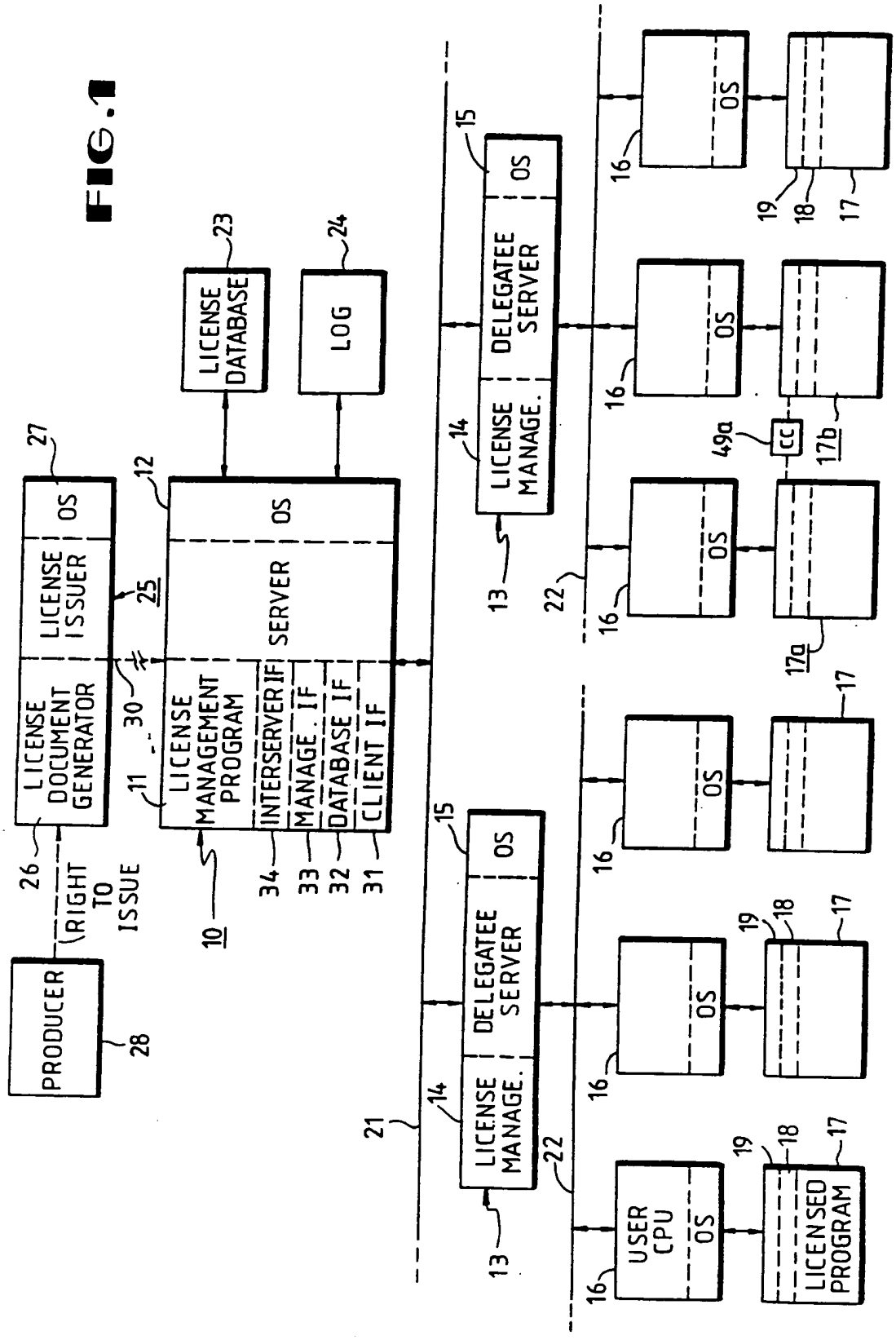
means for accessing said store to obtain information from said license authorization for said software item, in response to said request, and comparing said identification of said user and said software item with said information, to produce a grant or refusal of said request; and

means for sending said grant or refusal to said user.

44. Apparatus according to claim 43 wherein said filter means may select on one or more of the attributes: issuer, producer, product name, product use authorization, calling authorization, and wherein said store is maintained by a license server, and said request is sent to said server, and wherein said request is in the form of a remote procedure call, and said grant or refusal sent to said user is a return of said procedure call.

45. Apparatus according to claim 43 wherein said license authorization is a data arrangement specified as a product use authorization, and said product use authorization is received by said server from an issuer, wherein said server and said users are nodes on a computer network.

FIG.1



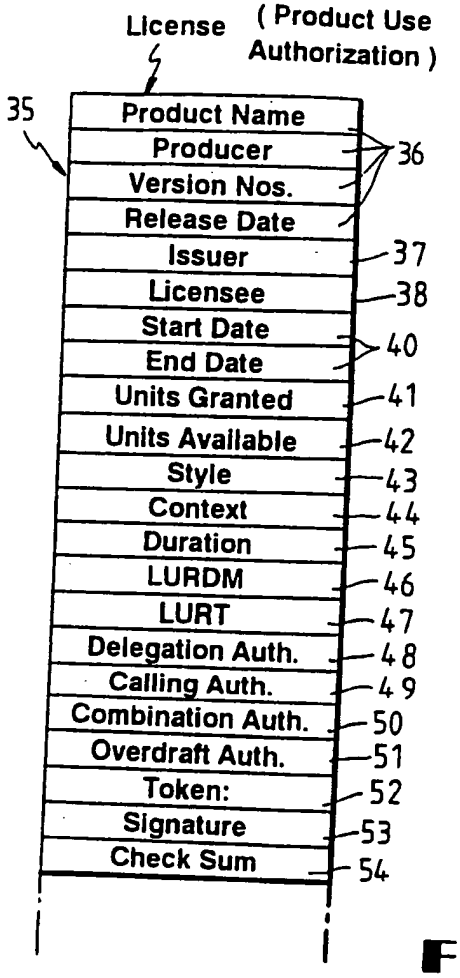


FIG.2

License Unit Requirements Table			
Row Selector	Columns		
Platform ID	A	B	C
PC-0	10	230	-1
PC-1	12	230	-1
VAX 6210	158	300	150

FIG.4

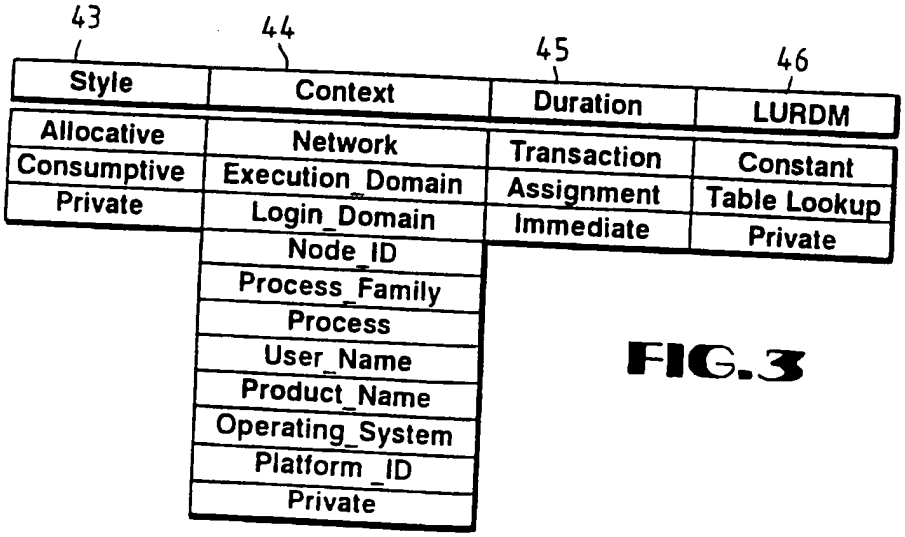


FIG.3

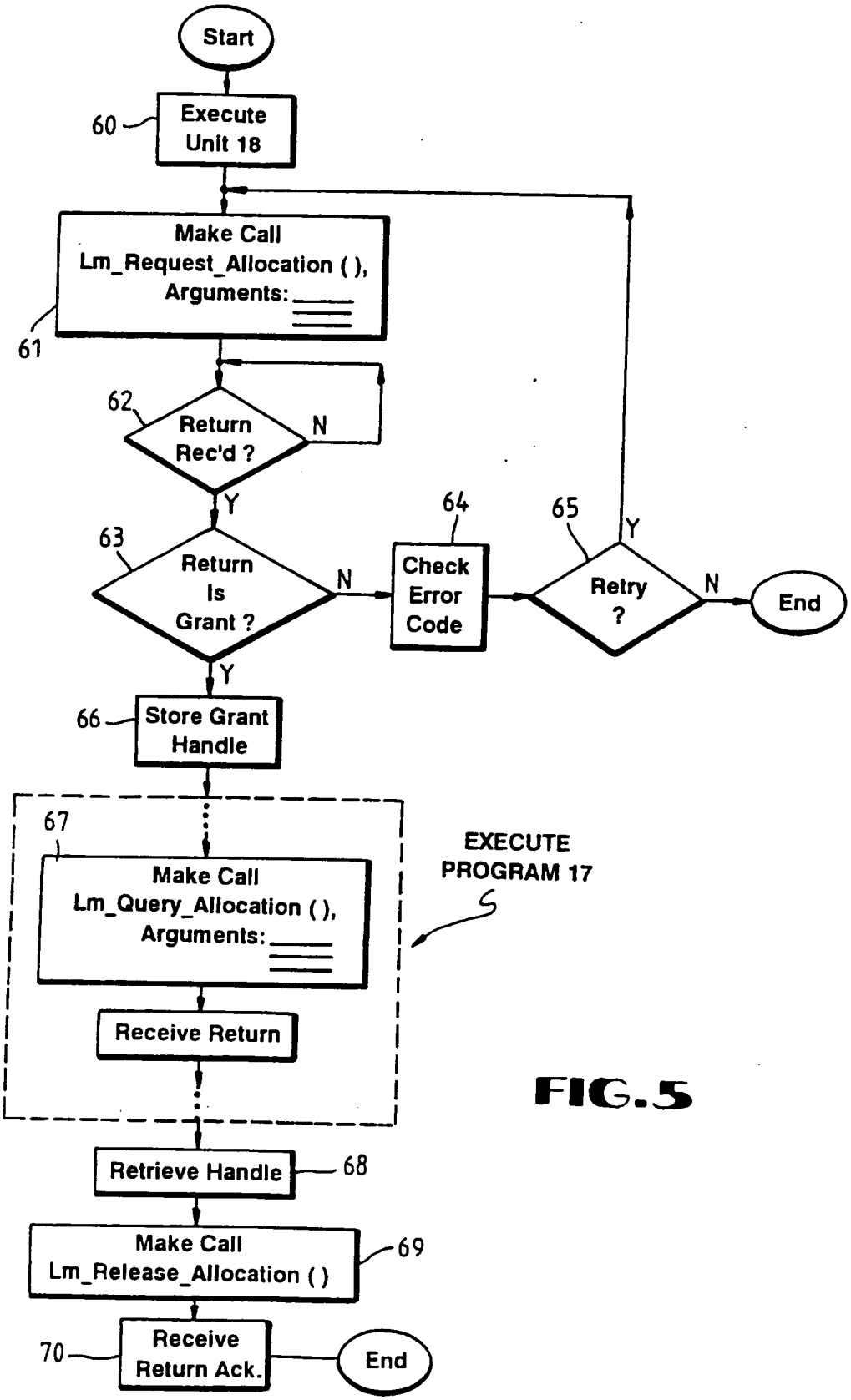


FIG.5

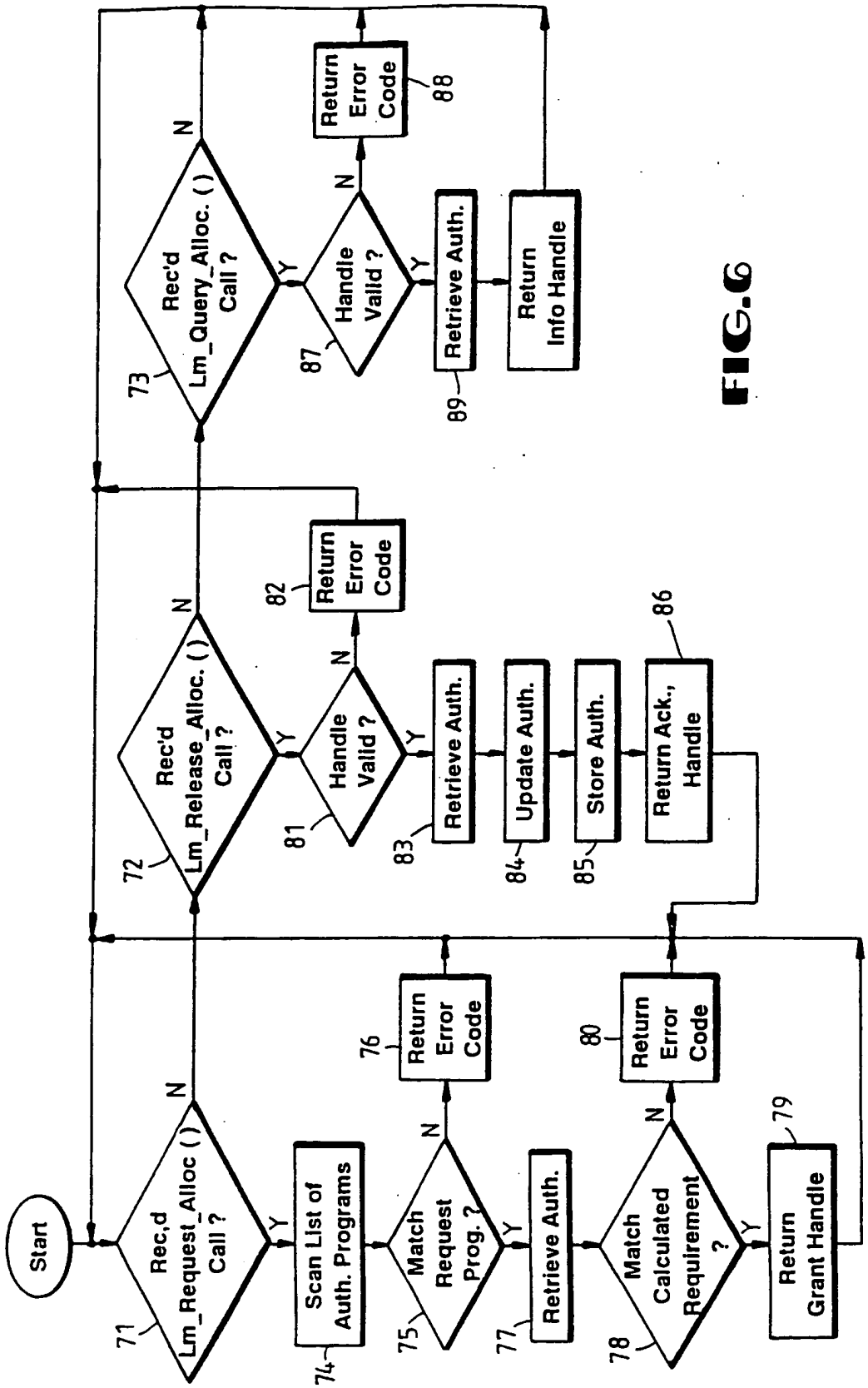


FIG.6

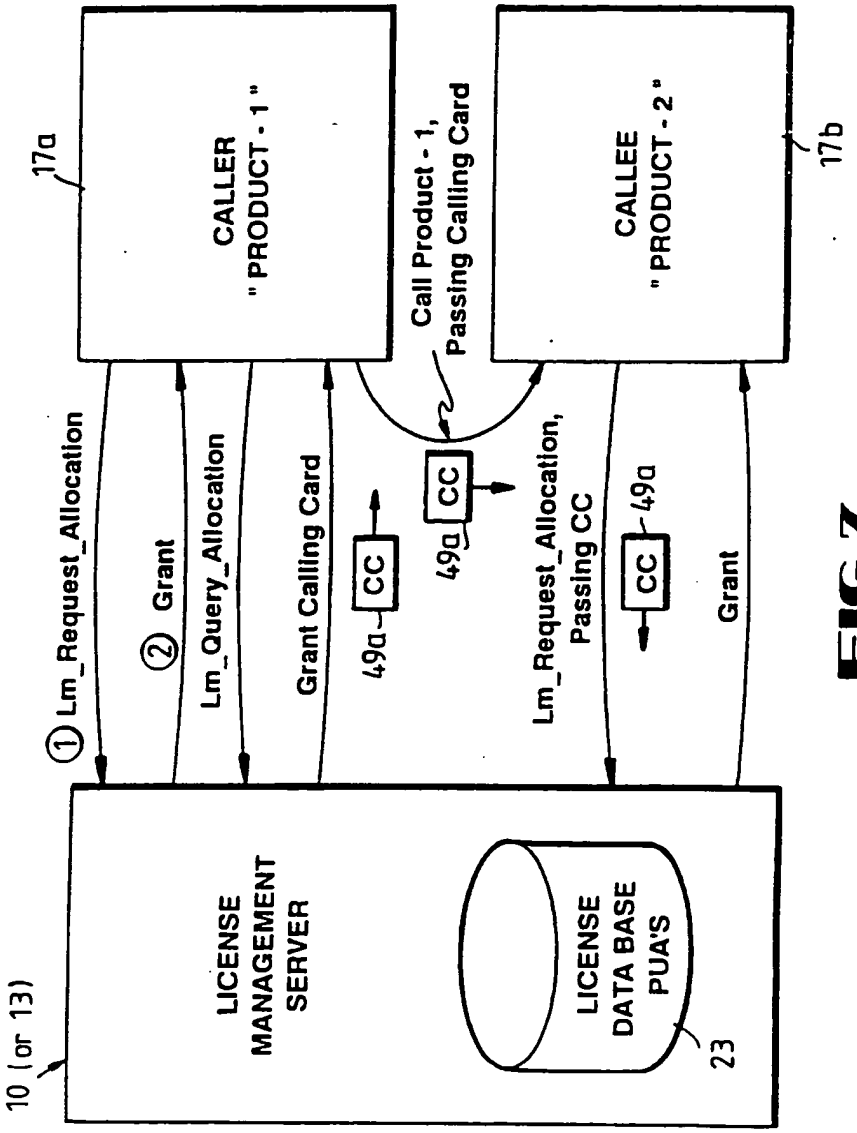


FIG.7

```
Object Identifier Value ::= {
                              iso(1)
                              identified-organization(3)
                              icd-ecma(12)
                              member-company(2)
                              dec(1011)
                              data-syntaxes(1)
                              cda(3)
                              ldif(17)
                            }
Object Identifier Encoding ::= {
                                 0x6, 0x8, 0x2B, 0xC, 0x2,
                                 0x87, 0x73, 0x1, 0x3, 0x11
                               }
```

FIG. 8 LDIF Object Identifier

```
LDIFDocument        ::=    [PRIVATE 16373] IMPLICIT SEQUENCE {
  document-descriptor          [0] IMPLICIT DocumentDescriptor OPTIONAL,
  document-header              [1] IMPLICIT DocumentHeader OPTIONAL,
  document-content             [2] IMPLICIT DocumentContent
                               }
```

FIG. 9 LDIF Document Syntax Diagram

FIG. 10 LDIF Document Structure

```
DocumentDescriptor      ::=     SEQUENCE {
  major-version                   [0] IMPLICIT INTEGER OPTIONAL,
  minor-version                   [1] IMPLICIT INTEGER OPTIONAL,
  encoder-identifier              [2] IMPLICIT Character-String OPTIONAL,
  encoder-name                    [3] IMPLICIT Character-String OPTIONAL
                                }
```

FIG. 11 Document Descriptor Syntax Diagram

```
Pakgen DocumentDescriptor  ::= {
      major-version 1,
      minor-version 0,
      encoder-identifier "PAKGEN",
      encoder-name {Character-String "PAK Generator V1.0"}
      }
```

FIG. 12 Document Descriptor Example

```
DocumentHeader      ::=     SEQUENCE {
  private-header-data           [0] IMPLICIT NamedValueList OPTIONAL,
  title                         [1] IMPLICIT Character-String OPTIONAL,
  author                        [2] IMPLICIT Character-String OPTIONAL,
  version                       [3] IMPLICIT Character-String OPTIONAL,
  date                          [4] IMPLICIT UTCTime OPTIONAL
                        }
```

FIG. 13 Document Header Syntax Diagram

```
example-header document-header ::= {
    title {Character-String "PAKGEN Licenses with Associated LURT data"}
    author {Character-String "Tom Jones, Foobar, Inc. License Department"}
    version {Character-String "V0.1"}
    date "198801021100-0500"
    }
```

FIG. 14 Document Header Example

```
Document Content        ::= SEQUENCE OF LicenseData

LicenseData             ::= SEQUENCE {
   license-data-header          [0] IMPLICIT LicenseDataHeader,
   license-body                 [1] CHOICE {
      product-use-authorization       [0] IMPLICIT ProductUseAuthorization,
      license-units-requirements-table [1] IMPLICIT LURT,
      group-definition                 [2] IMPLICIT GroupDefinition,
      key-registration                 [3] IMPLICIT KeyRegistration,
      issuer-delegation                [4] IMPLICIT IssuerDelegation,
      license-delegation               [5] IMPLICIT LicenseDelegation,
      backup-delegation                [6] IMPLICIT BackupDelegation
                                    },
   management-info              [2] IMPLICIT ManagementInfo OPTIONAL
                             }
```

FIG. 15 Document Content Syntax Diagram

FIG. 16 License Data Structure

```
LicenseDataHeader      ::= SEQUENCE {
    license-id              [0] IMPLICIT LicenseID,
    licensee                [1] IMPLICIT Character-String OPTIONAL,
    term                    [2] IMPLICIT Term OPTIONAL,
    management-constraints  [4] IMPLICIT ManagementConstraints OPTIONAL,
    signature               [5] IMPLICIT Signature,
    issuer-comment          [6] IMPLICIT NamedValueList OPTIONAL
```

FIG. 17 License Data Header Syntax Diagram

```
ProductUseAuthorization    ::= SEQUENCE {
    product-id                     [0] IMPLICIT ProductID,
    units-granted                  [1] IMPLICIT INTEGER,
    management-policy              [2] IMPLICIT ManagementPolicy,
    calling-authorizations         [3] IMPLICIT SEQUENCE OF Member OPTIONAL,
    caller-authorizations          [4] IMPLICIT SEQUENCE OF Member OPTIONAL,
    execution-constraints          [5] IMPLICIT ExecutionConstraints OPTIONAL,
    product-token                  [6] IMPLICIT NamedValueList OPTIONAL
                               }
```

FIG. 18 Product Use Authorization Syntax Diagram

```
LURT                       ::= SEQUENCE {
    lurt-name                      [0] IMPLICIT Character-String,
    rows                           [1] IMPLICIT RowList
                               }
RowList                    ::= SEQUENCE OF LurtRow

LurtRow                    ::= SEQUENCE {
    platform-id                    [0] IMPLICIT Character-String,
    lurt-columns                   [1] IMPLICIT SEQUENCE OF INTEGER
                               }
```

FIG. 19 License Unit Requirement Table Syntax Diagram

```
Example LURT ::= {
    lurt-name { Character-String "Example LURT"}
    rows {
        LurtRow {
            {Character-String "PC-0"}
            {{10} {230} {-1}}
        LurtRow {
            {Character-String "PC-1"}
            {{12} {230} {-1}}
        LurtRow {
            {Character-String "VAX 6210"}
            {{158} {300} {150}}
    }
}
```

FIG. 20 Example Encoding of LURT

```
Group Definition        ::= SEQUENCE {
    group-name                  [0] IMPLICIT Character-String,
    group-version               [1] IMPLICIT Version,
    group-release-date          [2] IMPLICIT UTCTime,
    group-members               [3] IMPLICIT SEQUENCE OF Member
                            }
```

FIG. 21 Group Definition Syntax Diagram

```
KeyRegistration         ::= SEQUENCE {
    key-owner-name              [0] IMPLICIT Character-String,
    key-algorithm               [1] IMPLICIT Character-String,
    key-value                   [2] IMPLICIT OCTET STRING
                            }
```

FIG. 22 Key Registration Syntax Diagram

```
IssuerDelegation        ::= SEQUENCE {
  delegated-issuer-name       [0] IMPLICIT Character-String,
  delegated-product-id        [1] IMPLICIT SEQUENCE OF Member,
  delegated-units-granted     [2] IMPLICIT INTEGER OPTIONAL,
  template-authorization      [3] IMPLICIT ProductUseAuthorization OPTIONAL,
  sub-license-permitted       [4] IMPLICIT BOOLEAN DEFAULT FALSE
                          }
```

FIG. 23 Issuer Delegation Syntax Diagram

```
LicenseDelegation       ::= SEQUENCE {
  delegated-units             [0] IMPLICIT INTEGER OPTIONAL
  delegated-distribution-control  [1] IMPLICIT DistributionControl,
  delegatee-execution-constraints [2] IMPLICIT ExecutionConstraints OPTIONAL,
  assignment-list             [3] IMPLICIT AssignmentList OPTIONAL,
  delegated-data              [4] IMPLICIT LicenseData OPTIONAL
                          }
```

FIG. 24 License Delegation & Backup Delegation Syntax Diagrams

```
ManagementInfo ::= SEQUENCE {
    assignments          [0] IMPLICIT AssignmentList OPTIONAL,
    reservations         [1] IMPLICIT AssignmentList OPTIONAL,
    delegations          [2] IMPLICIT DelegationList OPTIONAL,
    backup-delegations   [3] IMPLICIT DelegationList OPTIONAL,
    allocations          [4] IMPLICIT AllocationList OPTIONAL,
    registration-date    [5] IMPLICIT UTCTime,
    registrar            [6] IMPLICIT Context,
    local-comment        [7] IMPLICIT NamedValueList OPTIONAL,
    termination-date     [8] IMPLICIT UTCTime OPTIONAL,
    extended-info        [9] IMPLICIT NamedValueList OPTIONAL
}
```

FIG. 25 ManagementInfo Syntax Diagram

```
AllocationList      ::= SEQUENCE OF Allocation

Allocation          ::= SEQUENCE {
    allocation-context      [0] IMPLICIT Context,
    allocation-lur          [1] IMPLICIT INTEGER,
    allocation-group-id     [2] IMPLICIT INTEGER OPTIONAL
                        }
```

FIG. 26 Allocation Syntax Diagram

```
AssignmentList      ::= SEQUENCE OF Assignment

Assignment          ::= SEQUENCE {
    assigned-units          [0] IMPLICIT INTEGER,
    assignment-term         [1] IMPLICIT Term,
    assignee                [2] IMPLICIT Context
                        }
```

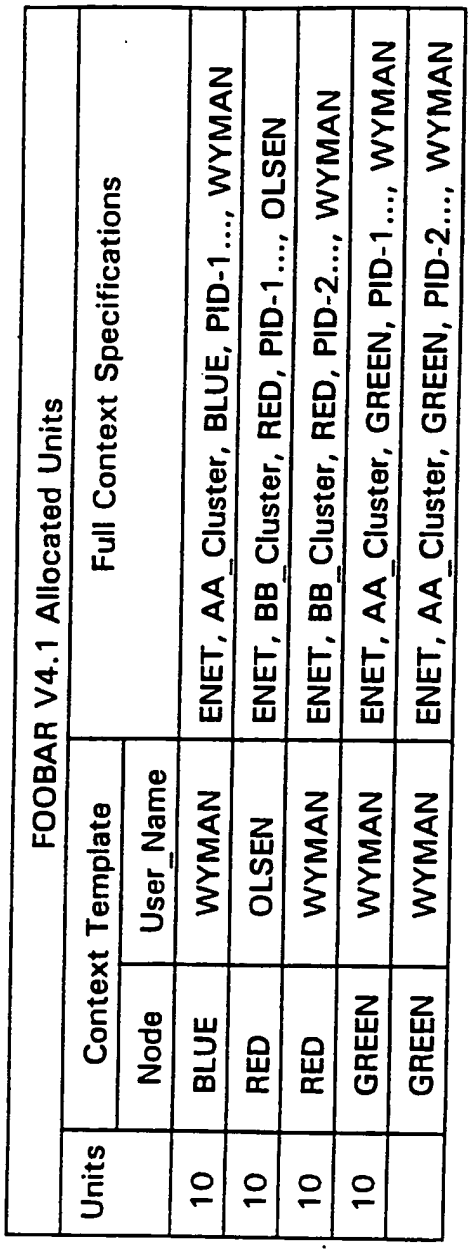
FIG. 27 Assignment Syntax Diagram

```
ContextList          ::= SEQUENCE OF Context

Context              ::= SEQUENCE OF Subcontext

SubContext           ::= SEQUENCE {
    sub-context-type            [0] SubContextType,
    subcontext-value            [1] ValueData
                        }
SubContextType       ::= CHOICE {
    standard-subcontext-type    [0] IMPLICIT INTEGER {
        network-subcontext(1),
        execution-domain-subcontext(2),
        login-domain-subcontext(3),
        node-subcontext(4),
        process-family-subcontext(5),
        process-id-subcontext(6),
        user-name-subcontext(7),
        product-name-subcontext(8),
        operating-system-subcontext(9),
        platform-id-subcontext(10)
                        }
    private-subcontext          [1] IMPLICIT INTEGER {first(0),last(255)}
                        }
```

FIG. 28 Context Syntax Diagram



FOOBAR V4.1 Allocated Units			
Units	Context Template		Full Context Specifications
	Node	User_Name	
10	BLUE	WYMAN	ENET, AA_Cluster, BLUE, PID-1...., WYMAN
10	RED	OLSEN	ENET, BB_Cluster, RED, PID-1...., OLSEN
10	RED	WYMAN	ENET, BB_Cluster, RED, PID-2...., WYMAN
10	GREEN	WYMAN	ENET, AA_Cluster, GREEN, PID-1...., WYMAN
	GREEN	WYMAN	ENET, AA_Cluster, GREEN, PID-2...., WYMAN

FIG. 29 Only unique contexts require explicit unit allocations.

FOOBAR V4.1 Allocated Units		
Units	Context Template Node	Full Context Specifications
10	BLUE	ENET, AA_Cluster, BLUE, PID-1...., WYMAN
10	RED	ENET, BB_Cluster, RED, PID-1...., OLSEN
	RED	ENET, BB_Cluster, RED, PID-2...., WYMAN
10	GREEN	ENET, AA_Cluster, GREEN, PID-1...., WYMAN
	GREEN	ENET, AA_Cluster, GREEN, PID-2...., WYMAN

FIG. 30 Modification of Context_Template impacts units requirements.

```
DistributionControl          ::= SEQUENCE {
    distribution-method              [0] IMPLICIT INTEGER {
            refresh-distribution(1),
            initial-distribution-only(2),
            manual-distribution(3)
                                     },
    current-start-date               [1] IMPLICIT UTCTime OPTIONAL
    current-end-date                 [2] IMPLICIT UTCTime OPTIONAL,
    refresh-interval                 [3] IMPLICIT IntervalTime OPTIONAL,
    retry-interval                   [4] IMPLICIT IntervalTime OPTIONAL,
    maximum-retry-count              [5] IMPLICIT INTEGER OPTIONAL,
    retries-attempted                [6] IMPLICIT INTEGER OPTIONAL
                                 }
```

FIG. 31 Distribution Control Syntax Diagram

```
ExecutionConstraints  ::= SEQUENCE {
    operating-system      [0] IMPLICIT SEQUENCE OF Character-String OPTIONAL,
    execution-context     [1] IMPLICIT ContextList OPTIONAL,
    environment-list      [2] IMPLICIT SEQUENCE OF EnvironmentKind OPTIONAL
                          }
EnvironmentKind       ::= INTEGER {
        batch(1),
        interactive(2),
        local(3),
        network(4),
        remote(5)
                          }
```

FIG. 32 Execution Constraints Syntax Diagram

FIG. 33 License ID Syntax Diagram

```
LURDM                    ::= SEQUENCE {
    combination-permitted        [0] IMPLICIT BOOLEAN DEFAULT TRUE,
    overdraft-limit              [1] IMPLICIT INTEGER DEFAULT 0,
    overdraft-logging-required   [2] IMPLICIT BOOLEAN DEFAULT FALSE,
    allocation-size              [3] IMPLICIT INTEGER OPTIONAL,
    lurdm-kind                   [4] IMPLICIT INTEGER {
        lurt(1),
        constant(2),
        private-lurdm(3)
                                 },
    named-lurt-id                [5] IMPLICIT Character-String OPTIONAL,
    lurdm-value                  [6] IMPLICIT INTEGER OPTIONAL,
    default-unit-requirement     [7] IMPLICIT INTEGER OPTIONAL
                                 }
```

FIG. 34 License Unit Requirements Determination Method Syntax Diagram

```
ManagementConstraints    ::= SEQUENCE {
    management-context          [0] IMPLICIT ContextList OPTIONAL,
    management-scope            [1] IMPLICIT INTEGER {
        single-platform(1),
        management-domain(2),
        entire-network(3)
                                } OPTIONAL,
    backup-permitted            [2] IMPLICIT BOOLEAN DEFAULT TRUE,
    delegation-permitted        [3] IMPLICIT BOOLEAN DEFAULT TRUE,
    maximum-delegation-period   [4] IMPLICIT IntervalTime OPTIONAL
                                }
```

FIG. 35 Management Constraints Syntax Diagram

```
ManagementPolicy    ::= SEQUENCE {
  style                    [0] IMPLICIT INTEGER {
      allocative(1),
      consumptive(2),
      private-style(3)
                           },
  context-template         [1] IMPLICIT SEQUENCE OF SubcontextType
                               OPTIONAL,
  duration                 [2] IMPLICIT INTEGER {
      transaction(1),
      assignment(2),
      immediate(3)
                           } OPTIONAL,
  lur-determination-method [3] IMPLICIT LURDM OPTIONAL,
  allocation-sharing-limit [4] IMPLICIT INTEGER OPTIONAL,
  reassignment-constraint  [5] IMPLICIT IntervalTime OPTIONAL
                           }
```

FIG. 36 Management Policy Syntax Diagram

```
Member          ::= SEQUENCE {
    member-product          [0] IMPLICIT ProductID,
    member-signature        [1] IMPLICIT Signature,
    member-token            [2] IMPLICIT NamedValueList OPTIONAL
                    }
```

FIG. 37 Member Syntax Diagram

```
NamedValue      ::= SEQUENCE {
    value-name          Character-String,
    value-data          ValueData
                    }

ValueData       ::= CHOICE {
    value-boolean           [0] IMPLICIT BOOLEAN,
    value-integer           [1] IMPLICIT INTEGER,
    value-text              [2] IMPLICIT SEQUENCE OF Character-String
    value-general           [3] IMPLICIT OCTET STRING,
    value-list              [4] IMPLICIT SEQUENCE OF ValueData
                    }

NamedValueList  ::= SEQUENCE OF NamedValue
```

FIG. 38 Named Value, Value Data & Named Value List Syntax Diagrams

```
ExampleList NamedValueList ::= {
    NamedValue {
        value-name {Character-String "Purchase Order"}
        value-data {INTEGER 154493}
    }
    NamedValue {
        value-name {Character-String "Telephone Support #"}
        value-data {Character-String {+1 (999) 555-1234}
    }
}
```

FIG. 39 Named Value List Example

```
ProductID ::= SEQUENCE {
    producer                [0] IMPLICIT Character-String,
    product-name            [1] IMPLICIT Character-String,
    first-version           [2] IMPLICIT Version OPTIONAL,
    last-version            [3] IMPLICIT Version OPTIONAL,
    first-release-date      [4] IMPLICIT UTCTime OPTIONAL,
    last-release-date       [5] IMPLICIT UTCTime OPTIONAL
}
```

FIG. 40 Product ID Syntax Diagram

```
Signature       ::= SEQUENCE {
    signature-algorithm     [0] IMPLICIT Character-String,
    signature-parameters    [1] IMPLICIT NamedValueList OPTIONAL,
    signature-value         [2] IMPLICIT OCTET STRING
                    }
```

FIG. 41 Signature Syntax Diagram

```
Term            ::= SEQUENCE {
    start-date              [0] IMPLICIT UTCTime OPTIONAL,
    end-date                [1] IMPLICIT UTCTime OPTIONAL,
                    }
```

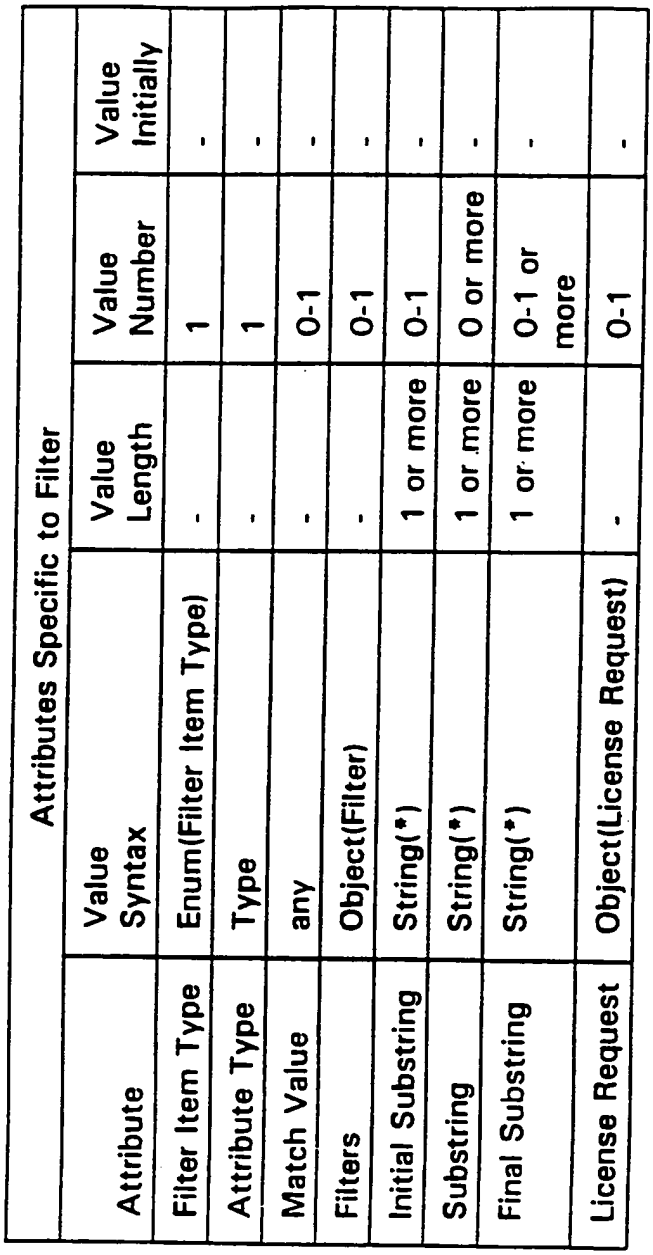
FIG. 42 Term Syntax Diagram

```
Version ::= SEQUENCE {
    part-1      [0] IMPLICIT INTEGER,
    part-2      [1] IMPLICIT INTEGER DEFAULT 0,
    part-3      [2] IMPLICIT INTEGER DEFAULT 0,
    part-4      [3] IMPLICIT INTEGER DEFAULT 0
                }
```

FIG. 43

Attributes Specific to Filter				
Attribute	Value Syntax	Value Length	Value Number	Value Initially
Filter Items	Object(Filter Item)	-	0 or more	-
Filters	Object(Filter)	-	0 or more	-
Filter Type	Enum(Filter Type)	-	1	-

FIG. 44



Attributes Specific to Filter				
Attribute	Value Syntax	Value Length	Value Number	Value Initially
Filter Item Type	Enum(Filter Item Type)	-	1	-
Attribute Type	Type	-	1	-
Match Value	any	-	0-1	-
Filters	Object(Filter)	-	0-1	-
Initial Substring	String(*)	1 or more	0-1	-
Substring	String(*)	1 or more	0 or more	-
Final Substring	String(*)	1 or more	0-1 or more	-
License Request	Object(License Request)	-	0-1	-

FIG. 45

```
Filter {
    Filter-Type AND
    Filter-Item {
        Filter-Item-Type SELECT
        Attribute-Type Product-Use-Authorization
        Filter {
            Filter-Type AND
            Filter-Item{
                Filter-Item-Type SELECT
                Attribute-Type Calling-Authorization
                Filter{
                    Filter-Type AND
                    Filter-Item {
                        Filter-Item-Type EQUALITY
                        Atribute-Type Producer
                        Match-Value "Digital"
                    }
                    Filter-Item {
                        Filter-Item-Type EQUALITY
                        Attribute-Type Producer
                        Match-Value "Amazing Database"
                    }
                }
            }
            Filter-Item {
                Filter-Item-Type EQUALITY
                Attribute-Type Producer
                Match-Value "Digital"
            }
            Filter-Item{
                Filter-Item-Type EQUALITY
                Attribute-Type Issuer
                Match-Value "Digital"
            }
            Filter-Item {
                Filter-Item-Type EQUALITY
                Attribute-Type Product-Name
                Match-Value "Amazing Graphics System"
            }
        }
    }
}
```

FIG. 46 Example Filter Value Notation

INTERNATIONAL SEARCH REPORT

PCT/US 92/03812

International Application No

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all)[6]

According to International Patent Classification (IPC) or to both National Classification and IPC

Int.Cl. 5 G06F1/00

II. FIELDS SEARCHED

Minimum Documentation Searched[7]

Classification System	Classification Symbols
Int.Cl. 5	G06F

Documentation Searched other than Minimum Documentation to the Extent that such Documents are included in the Fields Searched[8]

III. DOCUMENTS CONSIDERED TO BE RELEVANT[9]

Category °	Citation of Document, [11] with indication, where appropriate, of the relevant passages [12]	Relevant to Claim No.[13]
Y	EP,A,0 332 304 (DIGITAL EQUIPMENT CORPORATION) 13 September 1989 cited in the application	1-3, 6-19,22, 24, 26-29, 31-33, 35-37,39
Y	see figure 1 cited in the application	43-45
A	see column 3, line 31 - column 7, line 55	5,15,21, 25,30

-/--

° Special categories of cited documents :[10]

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

"E" earlier document but 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

"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

IV. CERTIFICATION

Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report
09 SEPTEMBER 1992	17. 09. 92
International Searching Authority	**Signature of Authorized Officer**
EUROPEAN PATENT OFFICE	WEISS P.

Form PCT/ISA/210 (second sheet) (January 1985)

III. DOCUMENTS CONSIDERED TO BE RELEVANT (CONTINUED FROM THE SECOND SHEET)

Category °	Citation of Document, with indication, where appropriate, of the relevant passages	Relevant to Claim No.
Y	IBM TECHNICAL DISCLOSURE BULLETIN. vol. 31, no. 8, 1 January 1989, NEW YORK US pages 195 - 198; 'METHOD FOR MANAGING CLIENT/SERVER RELATIONSHIP IN THE AIX OPERATING SYSTEM'	1-3, 6-19,22, 24, 26-29, 31-33, 35-37,39
Y	see the whole document	43-45
A	---	21

ANNEX TO THE INTERNATIONAL SEARCH REPORT ON INTERNATIONAL PATENT APPLICATION NO. US 9203812 SA 60557

This annex lists the patent family members relating to the patent documents cited in the above-mentioned international search report. The members are as contained in the European Patent Office EDP file on 09/09/92
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP-A-0332304	13-09-89	US-A- 4937863	26-06-90
		JP-A- 2014321	18-01-90

EPO FORM P0479

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G06F 11/34, H04L 9/00

A1

(11) International Publication Number: WO 93/01550

(43) International Publication Date: 21 January 1993 (21.01.93)

(21) International Application Number: PCT/US92/05387

(22) International Filing Date: 30 June 1992 (30.06.92)

(30) Priority data:
724,180 1 July 1991 (01.07.91) US
907,934 29 June 1992 (29.06.92) US

(71) Applicant: INFOLOGIC SOFTWARE, INC. [US/US]; 1223 Peoples Avenue, Suite 5405, Troy, NY 12180 (US).

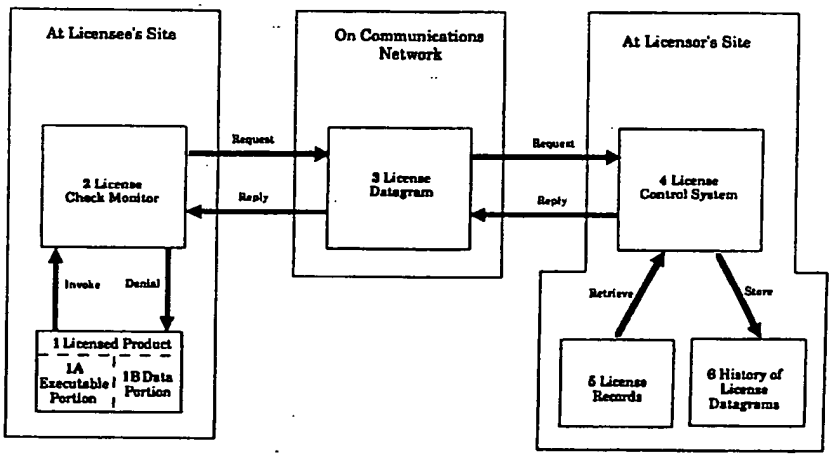
(72) Inventor: GRISWOLD, Gary, N. ; 1937 Regent Street, Schenectady, NY 12309 (US).

(74) Agents: LAZAR, Dale, S. et al.; Cushman, Darby & Cushman, Ninth Floor, 1100 New York Avenue, N.W., Washington, DC 20005-3918 (US).

(81) Designated States: AT, AU, BB, BG, BR, CA, CH, CS, DE, DK, ES, FI, GB, HU, JP, KP, KR, LK, LU, MG, MN, MW, NL, NO, PL, RO, RU, SD, SE, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IT, LU, MC, NL, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, SN, TD, TG).

Published
With international search report.

(54) Title: LICENSE MANAGEMENT SYSTEM AND METHOD



(57) Abstract

A license management system and method for recording (6) the use of licensed product (1), and for controlling (4) its use. A licensed product invokes a license check monitor (2) at regular time intervals. The monitor generates request datagrams (3) which identifiy the licensee and the product and sends the request datagrams over a communications facility to a license control system (4). The license control system maintains a record (6) of the received datagrams, and compares the received datagrams to data stored in its licensee database (5). Consequently, the license control system (4) transmits reply datagrams with either a denial or an approval message. The monitor (2) generates its own denial message if its request datagrams are unanswered after a predetermined interval of time. The datagrams are counted at the control system to provide billing information.

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.

AT	Austria	FI	Finland	ML	Mali
AU	Australia	FR	France	MN	Mongolia
BB	Barbados	GA	Gabon	MR	Mauritania
BE	Belgium	GB	United Kingdom	MW	Malawi
BF	Burkina Faso	GN	Guinea	NL	Netherlands
BG	Bulgaria	GR	Greece	NO	Norway
BJ	Benin	HU	Hungary	PL	Poland
BR	Brazil	IE	Ireland	RO	Romania
CA	Canada	IT	Italy	RU	Russian Federation
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland			SN	Senegal
CI	Côte d'Ivoire	KR	Republic of Korea	SU	Soviet Union
CM	Cameroon	LI	Liechtenstein	TD	Chad
CS	Czechoslovakia	LK	Sri Lanka	TG	Togo
DE	Germany	LU	Luxembourg	US	United States of America
DK	Denmark	MC	Monaco		
ES	Spain	MG	Madagascar		

LICENSE MANAGEMENT SYSTEM AND METHOD

BACKGROUND

Field of the Invention

The present invention generally relates to systems for managing licenses of products such as computer software, video games, CD-ROM information, movies and other video products, music and other audio products, multimedia products, and other systems for up-to-date recording of actual usage of such a licensed product to enable efficient billing therefor.

Description of Related Art

Licenses for information products such as computer software, music, video products and the like usually provide licensees with limited rights. The licenses may restrict sites of use, duration of use, or number of concurrent uses of the products. The licenses also may limit the use of the products depending on currentness of licensee's payments. However, enforcing the conditions of the licenses is difficult, because, in general, the licensed products may be easily copied or "pirated" and used without the licensor's knowledge.

Compliance with limited license rights has been encouraged with copy protection. Known methods of computer software copy protection include putting a

physical hole or mark on the diskette containing a product, or placing data on the diskette in a location where no data is expected. A disk with an illegally copied software product usually would not contain the marks. At the beginning of its operation, a copy-protected, but illegally copied software product would search its own diskette for the marks. Upon failing to detect the marks, the software would abort from its normal procedures.

Most software products sold today do not have such copy protection, partly because copy protection renders legitimate duplication of copy protected software difficult, but not impossible. Copy protection frustrates the making of legitimate copies, while not eliminating unauthorized copying. Many software publishers have experienced higher sales by eliminating copy protection schemes.

Another method for enforcing limited licensing rights of computer software is described in U.S. patent No. 4,932,054 to Chou. Chou describes a "coded filter" hardware device which is plugged into a port of a computer. The "coded filter" outputs an authorization control code when a predetermined control code is sent to it. The licensed software functions properly only if the "coded filter" transmits the correct authorization control code to the software.

While devices such as described by Chou have existed for several years, they have not been well accepted by the market. Since the device is attached to the outside of a computer, it can easily be lost or stolen, preventing the use of licensed software. In addition, if a licensee purchased a number of software

products, each of which used Chou's protection scheme, the licensee would collect a stack of "coded filters."

Hershey, in U.S. patent No. 4,924,378, describes a method for limiting the number of concurrent uses of a licensed software product. Each workstation of a network has a license storage area in its local memory. License Management System (LMS) daemons are provided in the network in a number corresponding to the permissible number of concurrent uses of the software product. To use the software, a work station stores a daemon in its license storage area. If all daemons are in use, no further work stations may use the software.

Robert et al., in U.S. patent No. 4,937,863, describe a similar invention. This invention includes a license management facility which accesses a database of license information related to licensed computer software programs. When a user attempts to use a licensed program, the license management facility first checks the database. Access to the licensed product is prevented if licensing conditions related to the product are not satisfied (e.g., expiration of licensing dates, etc).

While the Robert et al. and Hershey patents show effective techniques for controlling licensed computer software, each also reveals components that cannot be easily managed by an average user. A system manager, or someone with special access privileges to the internals of a machine, must install the licensed software. This hinders the distribution of the software.

Licensable products other than computer software have not generally been copy-protected. For example,

video tapes can be easily copied by anyone with two VCR machines, and audio tapes and music CDs can be easily copied to tape. Computer CD-ROMs can be copied to magnetic disk; however, their large information storage capacity relative to that of magnetic disks makes this a very expensive proposition. The introduction of digital audio tape is being delayed, because some view its ability to easily produce very high quality copies as a threat to music royalties.

Hellman, in U.S. patent No. 4,658,093, describes means to bill by usage. This is accomplished via communication of an encrypted authorization code from a licensor to a base unit at the licensee's site. The encrypted authorization code contains information related to an identification of the base unit, a number of uses requested, and a random or non-repeating number; however, implementation of Hellman's scheme requires a "base unit", such as a computer, video game unit, record player, video recorder, or video disk player, with a unique identification number. The requirement is difficult to satisfy, because, at the present, only a fraction of such systems on the market have an internally readable serial number for identification. In addition, vendors of these systems provide no guarantees for the uniqueness of any given device's serial number. Furthermore, an internal serial number can change when hardware maintenance is performed on the device. Also, Hellman's approach requires that an identical copy of each software product be stored at the authorization site. These copies are used in the generation of unique keys. The unstated assumption that all copies of a specific version of a software

product are identical is unrealistic. Minor bug fixes to software are often made without generating a new version of the product. Also, some software products, such as those which run on Macintosh computers, are self-modifying.

While Hellman's invention counts each use of the software, it does not monitor the duration of use. Thus, Hellman's system would not be able to bill for extensive use of licensed software if the software were continuously operated. Finally, while Hellman suggests the inclusion of an automated communication system as part of his invention, he does not disclose how this communication system could be implemented. Instead, he mentions non-automated use of telephone and mail. In summary, Hellman's patent is an interesting discussion of cryptographic techniques, but it does not provide a practical, real-world implementation of those techniques.

Shear, in U.S. Patent No. 4,977,594, describes a system and method to meter usage of distributed databases such as CD-ROM systems. The method describes a hardware module which must be part of the computer used to access the distributed database. This module retains records of the information viewed. Once the module storage is filled, the module must be removed and delivered to someone who will charge for the usage recorded therein and set the module back to zero usage. Like Hellman's method, this method requires a hardware module which must be incorporated within the computer so the system can control user access. No database publisher will be able to use this method until there are a very large number of units containing such modules. Hardware manufacturers

will be hesitant to include the module in the design of their computers until there is sufficient demand from customers or publishers for this system. The method and apparatus according to the present invention can be implemented entirely in software and hence does not require special, dedicated computer subsystems.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a license management system and method which can ensure that a licensed product is used only on machines under which it is licensed.

It is another object of the present invention to provide a license management system and method which may terminate access to a licensed product once its license has expired.

It is yet another object of the present invention to provide a license management system and method which may terminate access to a licensed product when payment for a license is overdue.

It is a further object of the present invention to provide a license management system and method which can limit the number of concurrent uses of a licensed product.

It is yet another object of the present invention to provide a license management system and method which can bill licensees for the duration of actual usage of a licensed product.

The present invention provides an advantageous feature of quickly and effectively implementing license agreements between a licensor and licensee.

The present invention provides another advantageous feature of allowing logic used to control licenses to be easily changed.

The present invention provides yet another advantageous feature of detecting, at the licensor's site, many types of attempts to alter the license management system.

The present invention provides a further advantageous feature of permitting anyone without special access privileges to install a licensed product.

In the present invention, a licensed product generates request "datagrams," messages transmitted over a communications network. The request datagrams are sent to the licensor's site. At the licensor's site the datagram is compared to information stored in a license database. After the comparison, a reply datagram is sent to the licensee. Upon receiving the reply datagram, the licensed product reacts in accordance with the instructions therewithin. For example if a reply datagram contained a "denial," the licensed product would display an appropriate message to the user and then suspend further execution of its programs.

In the present invention, the licensed product is implemented on a network node attached to a communications network that includes the licensor. The network node may be a computer, a CD-ROM player, a tele-computer or other multimedia machine, or any other appropriate device. The node may also be an intelligent type of consumer electronic device used for presenting information, such as an intelligent television, VCR, videodisk player, music CD player,

audio tape player, telephone or other similar device. Further, the communications network may be any two-way network such as a computer network, telephone network, a cellular telephone network or other wireless network, a two-way cable TV network, or any other equivalent system.

Should the user detach the node from the network, the licensed product will fail to receive reply datagrams. Upon several failures to receive reply datagrams, the licensed product will generate its own denial.

After a request datagram has been sent out, a user may be permitted to use the licensed product for a limited duration. This feature may be necessary because of the delays in network communications. When networks are sufficiently fast, use of a licensed product can be postponed until the reply datagram is received.

In the preferred embodiment of the present invention, licensees' network addresses are used to identify the licensees. Other embodiments may use a licensed product serial number or hardware serial numbers for the identification.

A licensed product as in the present invention generates a request datagram after each period of product use. The number of request datagrams received by the licensor can be used to bill the licensee. For example, if datagrams are sent after every hour of product use, the licensee will be billed for the amount equal to the number of request datagrams received by the licensor multiplied by the hourly rate.

The embodiments of the present invention may incorporate a query system at a licensor's site for reporting on problem datagrams. This would allow the licensors to take appropriate actions in accordance with problems associated with each datagram.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other objects and advantages of this invention will become more apparent and more readily appreciated from the following detailed description of the presently preferred exemplary embodiment of the invention, taken in conjunction with the accompanying drawings, of which:

FIGURE 1 is a general block diagram of the preferred exemplary embodiment of the present invention;

FIGURE 2 shows representative diagrams of the contents and formats of data at licensee's site, contained in datagrams, and at licensor's site;

FIGURE 3 illustrates a sequence of representative operations executed at the licensee's site and at the licensor's site, together with required inputs for the execution of the operations and with outputs produced therefrom;

FIGURE 4 illustrates a sequence of representative operations to send a request datagram, together with required inputs for the execution of the operations and with outputs produced therefrom;

FIGURE 5 illustrates a sequence of representative operations when a reply datagram is overdue, together with required inputs for the execution of the operations and with outputs produced therefrom;

FIGURE 6 shows a sequence of representative operations to process a reply datagram, together with required inputs for the execution of the operations and with outputs produced therefrom;

FIGURE 7 shows a sequence of representative operations to generate an authorization code, together with required inputs for the execution of the operations and with outputs produced therefrom; and

FIGURE 8 shows a sequence of representative operations to send a reply datagram, together with required inputs for the execution of the operations and with outputs produced therefrom.

DETAILED DESCRIPTION OF THE PRESENTLY PREFERRED EXEMPLARY EMBODIMENT

As shown in FIGURE 1, a licensed product 1 is located at a licensee's site. Product 1 may include a data portion 1B and a functional portion 1A such as computer software product or any other kind of information product used to control use of data portion 1B. If data portion 1B is CD-ROM database information, functional portion 1A should enable the licensee to search indexes and display text. If data portion 1B is video information, functional portion 1A should control the display of the video information. For audio information, functional portion 1A should play the audio information. If data portion 1B is an electronic book, functional portion 1A should display and turn pages. The above examples show some of the ways functional portion 1A can control data portion 1B; however, they are hardly exhaustive.

By including in product 1 both information and software which controls the information, product 1 is

an executable product. Non-software information in product 1 is preferably encrypted so that it cannot be easily extracted from the product.

License check monitor 2 sends license datagrams 3 to the licensor and also receives license datagrams 3 from the licensor. License check monitor 2 also prevents further use of product 1 when a datagram 3 containing a "denial" message is received.

License datagrams 3 are messages that describe information related to the use of licensed product 1. Datagrams 3 are sent over a communications network between the licensee and licensor. Initially, the licensee sends a request datagram 3 over the network to the licensor. The licensor then returns a reply datagram containing either an approval or denial. It is also possible to implement the present invention by having the licensor transmit a reply datagram only for approvals.

At the licensor's site, license control system 4 makes licensing decisions by comparing request datagram 3 with license records 5. After the comparison, control system 4 stores information related to request datagram 3 into history of license datagram record 6. It is noted that request datagrams 3 are periodically sent while product 1 is in use. Thus, the history of license datagrams in record 6 provides means for measuring the duration of use of product 1.

Representations of data and records stored at the licensee's site, contained in datagrams, and stored at the licensor's site are illustrated in FIGURE 2. At the licensee's site, network service 7, which handles delivery and transmission of datagrams 3, supplies

network address 8. It is by this address that license control system 4 identifies a location of use of product 1.

Licensed product record 9 is contained within monitor 2. Within the license product record 9 is an identification record 10, which contains the following two items: licensor's network address 11, and product model number 12 that identifies product 1. When a licensor has only one product, or uses different licensor network addresses 11 for each product, product model number 12 may not be needed.

Datagram sent record 13 stores information about the last sent datagram 3. It includes a datagram number 14, which uniquely identifies the last transmitted datagram 3, and the date and time 15 when the last datagram 3 was sent from the licensee's site.

Licensed product record 9 also contains control parameters record 16, which is used for controlling the timing of key events in the communication of license check monitor 2 with license control system 4. Send interval 17 specifies a time interval between each transmission of a new datagram 3 from the licensee to the licensor.

Wait interval 18 is the length of time that monitor 2 waits to receive a reply datagram 3 before resending the same request datagram 3. The duration of this interval depends on the speed of the communications network being used to deliver datagrams 3.

Disconnect allowed interval 19 is the duration of time that monitor 2 allows product 1 to be used without a reply datagram 3 from the licensor. The duration of this interval depends on the reliability

of the communications network. The interval must be long enough to take into consideration network downtime. For example, suppose a message was sent from the licensor and the network went down just afterwards. Disconnect allowed interval 19 should be long enough to allow the network to resume its normal operation and successfully deliver datagrams 3 from the licensor; otherwise, the licensee would be forced to stop using product 1 until the network was operational.

License datagram 3 contains header 20. Header 20 is used during execution of low level communication protocols within the network. Source network address 21 is the network address from where datagram 3 is sent. Destination network address 22 is the network address to where datagram 3 is sent. Additional data may be included in header 20 if required by low level protocols used in delivering datagrams 3.

Data 23, a part of datagram 3, conveys a message, and contains a number of fields. Product model number 24 and datagram number 25 identify product 1 and datagram 3, respectively. It is noted that retransmitted datagrams have an identical datagram number. Duplicate datagrams must be identified at a licensor's site so that they do not all contribute in billing a licensee.

Each datagram number 25 is unique for each request datagram 3 transmitted from the licensee, except for retransmitted datagrams. This allows a reply datagram 3 received by a licensee to be verified as an actual reply to a request datagram 3 from that licensee, as explained below.

Number of processes running 26 is the number of concurrent uses of product 1 at the time datagram 3 is sent. Authorization code 27 is used on reply datagrams 3 to indicate an approval or a denial. Message text 28 contains a message which will be displayed to the user upon a denial.

License database 29 at the licensor's site holds records of information about customers, licenses, and license usage. The types of information within license database 29 of the present embodiment are shown in FIGURE 2. However, a specific license management system may require its license database to hold types of information other than those in FIGURE 2. For example, licensee name and address may be incorporated as a part of a license database 29.

License record 5 contains information on licenses. Licensee network address 30 identifies a precise network node which is licensed to use product 1. If request datagrams are received which do not originate from known licensee network addresses 30, reply datagrams containing denial messages are transmitted. Product model number 31 is the model number of a licensed product. Termination date 32 is the expiration date of a license. When the license of a product is issued for an unlimited duration, termination date 32 should reflect a date very far into the future, relative to the licensing date.

The present embodiment allows licenses to be paid for in a lease-like or rental fashion. If a licensee were to rent or lease product 1, paid through date 33 would reflect the date through which the licensee has paid for using the product. Grace period 34 is the time interval for which the licensee is allowed to be

delinquent before services are disconnected. Grace period 34 would reflect a very large time interval if the license is not of a lease-like or rental type. When the license provides for a limit on the number of concurrent uses of a product 1, number of processes licensed 35 contains the limiting number. When the license does not provide for such a limit, number of processes 35 should be a very large number.

History of license datagrams 6 is an archive of datagrams 3 received from the licensee.

FIGURE 3 illustrates operations executed at the licensee's site and at the licensor's site. An overview of the processing at the licensee's site is described by steps 101.0 to 106.0, and an overview of the processing at the licensor's site is described by steps 107.0 to 110.0.

At the licensee's site, at step 101.0, product 1 invokes monitor 2. This is accomplished by first establishing monitor 2 as a handler for a timer expiration interrupt signal and for received datagrams 3. Next, a timer is set with a very short time to cause an initial call to monitor 2. At step 102.0, monitor 2 computes a time 36 since the last datagram was sent by determining the difference between the current date and sent time and date and time 15 that a datagram was last sent from the licensee's site. When product 1 commences execution, datagram sent date and time 15 is set to "null." Thus, time since send 36 is very large at the beginning of the monitor's execution. At step 103.0, time since send 36 is compared to send interval 17. If time since send 36 is greater than send interval 17, then a request datagram is transmitted, per the steps described in

FIGURE 4. Step 104.0 first checks if a reply to the last datagram has arrived and if wait interval 18 has expired. If a reply has not arrived and the wait interval has expired, steps 104.1-104.3 (FIGURE 5) are executed. Step 105.0 processes authorization code 27 in a reply when the reply is received, in accordance with steps 105.1 to 105.5 (FIGURE 6). At step 106.0, product 1 resumes normal execution of its programs until the next interrupt signal is generated.

At the licensor's site, license control system 4 receives and processes datagram 3, in accordance with steps 107.0 to 110.0. Step 107.0 receives request datagram 3. Step 108.0 generates authorization code 27, per steps 108.1 to 108.8 (FIGURE 7). Step 109.0 creates reply datagram 3 and transmits the datagram to the licensee via steps 109.1 to 109.5 (FIGURE 8).

FIGURE 4 shows the procedure which monitor 2 follows for sending request datagram 3 to the licensor. Step 103.1 sets source network address 21 in datagram 3 to the network address 8 of the licensee's location on the network. Step 103.2 sets destination network address 22 to licensor's network address 11. Step 103.3 encrypts product model number 12 for datagram 3. Step 103.4 assigns a unique number to datagram 3, encrypts the number, and stores it as datagram number 14. This number is altered when an entirely new datagram 3 is sent. Datagrams which are retransmitted have the same datagram number 25 as the original. As already explained, this allows license control system 4 to identify duplicate datagrams.

Step 103.5 counts the number of processes using product 1, currently running, encrypts the count, and stores the encryption as the number of processes

running 26. In the UNIX operating system, this procedure could be performed using the command "ps" to obtain a list of current processes, the command "grep" to extract the processes of product 1, and "wc" to count the number of processes. Step 103.6 sets authorization code 27 to number 255 and encrypts the number.

Number 255 indicates that datagram 3 is a request for authorization. Such an indication is needed to guard the present system against the following steps for circumventing the present invention: intercepting outgoing datagrams; and inputting the intercepted datagrams to monitor 2.

Step 103.7 stores the current date and time as sent date & time 15. This date is needed to compute when to send the next datagram 3. Step 103.8 assigns a value to send interval 17, which sets an alarm for invoking monitor 2 to send the next datagram 3. Step 103.9 sends datagram 3.

In the present embodiment a datagram is transmitted via a connectionless datagram service. Methods for transmission are well documented for some networking systems. For example, TCP/IP (Transport Control Protocol/Internet Protocol) includes a connectionless protocol called UDP (User Datagram Protocol). A method for sending a datagram using UDP protocol from a SUN Microsystem computer is documented in a SUN manual titled, Network Programming Guide, in section 9 titled "Transport Level Interface Programming."

Step 103.10 sets another alarm using wait interval 18 for retransmitting datagram 3, if no reply datagram has been received. The alarm causes monitor

2 to be invoked for checking whether a reply datagram 3 has been received. Monitor 2 will transmit a duplicate of the previously transmitted datagram, if no reply has been received. After the execution of step 103.10, "Send License Datagram" procedure returns system control to step 104.0 in FIGURE 3.

FIGURE 5 shows the operation of the "Reply Datagram is Overdue" procedure. Step 104.1 compares time since the last datagram was sent 36 to disconnect allowed interval 19, which, as described above, is the interval that product 1 is allowed to operate even if a reply is overdue. If time since send 36 is smaller than disconnect allowed interval 19, datagram 3 is retransmitted via executing step 103.9 in FIGURE 4. Step 104.2 "disconnects" product 1 from further service, if time since send 36 is greater than disconnect allowed interval 19.

Step 104.2 comprises a sequence of sub-steps 104.2.1-104.2.3. Step 104.2.1 assigns number 5 to authorization code 27 in the current datagram being processed. Value 5 is interpreted by monitor 2 as a denial. Step 104.2.2 sets message text 28 to the following: "A reply from licensor to numerous authorization requests was never received. This product must be connected to a communications network in order to function." Step 104.2.3 transfers system control to step 105.3 in FIGURE 6. Step 105.3 processes the current denial datagram 3 as if it were just received.

Through the execution of steps 104.1-104.3, the present system permits the use of product 1 for a prescribed period of time. After the prescribed

period of time has elapsed, the present system generates a denial.

FIGURE 6 illustrates the steps which monitor 2 follows in processing a reply datagram 3. Step 105.1 decrypts all encrypted data in the received datagram. Step 105.2 compares datagram number 25 with datagram number 14 associated with the last datagram. If datagram number 25 is not equal to datagram number 14, step 105.2 ignores the current datagram and transfers procedural control to step 103.9 (FIGURE 4) in order to resend the last transmitted datagram. After disconnect allowed interval 19 elapses, monitor 2 generates a denial.

In essence, step 105.2 guards against the circumvention of the present invention via: (1) intercepting a reply datagram 3 (from the licensor) containing an approval (2) storing the reply datagram 3; and (3) inputting the stored datagram to monitor 2.

If the execution of step 105.2 does not transfer its procedural control to step 105.3, and if authorization control 27 is not zero (indicating an unqualified authorization has not been received), step 105.3 processes authorization code 27 via steps 105.3.1 to 105.3.3. Step 105.3.1 retrieves message text 28 from datagram 3. If message text 28 is null, then the current datagram 3 is ignored, and monitor 2 resends the last transmitted datagram 3. Step 105.3.1 further protects the present system from attempts to generate fake datagrams and to feed the fake datagrams to monitor 2 by checking for a proper authorization code of zero.

If message text 28 is not null, step 105.3.2 presents the message 28 to the user on an output

device such as a CRT screen. Step 105.3.3 terminates the current use of product 1. This step may be implemented by subroutine or function call to a simple exit that saves any current user data to a file. Alternatively, product 1 may be designed so that, upon being directed to terminate further execution, it first gives the user an opportunity to save their data.

If authorization code 27 is zero, step 105.4 allows further use of product 1. Step 105.5 returns procedural control to 106.0 on FIGURE 3.

FIGURE 7 shows a sequence of operations within the "Generate Authorization Code" procedure. The procedure produces appropriate authorization code 27 when a request datagram 3 is received at the licensor's site.

Step 108.1 decrypts all encrypted data in the received datagram 3. Using source network address 21 and product model number 24 in the datagram 3, step 108.2 searches the license database 29 for matching licensee network address 30 and product model number 31. If license database 29 does not contain a record of product model number 24 of the product 1 being licensed to the licensee, step 108.3 sets authorization code 27 of its reply datagram 3 to 1 (i.e., the sending node is not a registered address) and authorization is denied.

Step 108.3 prevents copies of product 1 from being installed on multiple nodes independently of whether they are within or outside the licensee's organization. Step 108.3 also prevents the licensee from transporting product 1 from one node to another node without the licensor's approval. This is

important because the two nodes may have different processing capacities, and they may be billable at different rates.

If the date a request datagram is received is later than license termination date 32, step 108.4 sets authorization code 27 to number 2 (i. e., license has expired). Step 108.4 allows the licensor to fix licensing periods, or to determine free trial periods for the use of the product. The licensing period may be extended by resetting license termination date 32 at the licensor's site.

If the date when the datagram is received is later than the paid through date 33 as extended by the grace period 34, step 108.5 sets authorization code 27 to 3 (i.e., payment is past due).

If the number of processes running 26 exceeds a licensed number of concurrent uses of product 1 (at a particular node), then step 108.6 sets authorization code 27 to 4 (i.e. concurrent process usage limit is exceeded).

Step 108.7 sets authorization code 27 to 0 indicating processing can continue. It is noted that steps 108.3-108.7 are a part of a

```
IF (x1) then (y1)
ELSE if (x2) then (y2)
ELSE if (x3) then (y3) ...
```

statement of a procedure (e. g., FORTRAN, PASCAL, C, etc). Thus, only one of the steps 108.3-108.7 is executed. Step 108.7 sets authorization code 27 to 0 (indicating approval of further use) only if steps 108.3-108.6 do not execute the THEN portion of each step. Step 108.7 also stores the received datagram 3 in history of license datagrams 6.

Step 108.8 is the last of authorization processing rules 108.1-108.7. After the execution of steps 108.3-108.7, step 108.8 returns procedural control to step 109.0 in FIGURE 3.

FIGURE 8 illustrates the steps which license control system 4 follows to send reply datagram 3 to the licensee.

Step 109.1 encrypts authorization code 27 and writes the encrypted code into datagram 3. Next, step 109.2 writes message text 28 corresponding to authorization code 27 into datagram 3.

Step 109.2 may be replaced with the following method for relaying proper messages to a product user. Proper messages corresponding to each authorization code is stored in monitor 2 at each licensee's site. Upon reception of a reply datagram 3, monitor 2 would locate within itself the proper message corresponding to the authorization code, and use the message for various purposes. This method would reduce the size of reply datagrams 3. However, if the licensor wanted to implement new denial codes, each product would need to somehow incorporate the new message associated with the new denial code into itself. The list of messages, one of which may be written as message text 28, are as follows:

AUTHORIZATION CODE	TEXT MESSAGE
1	This product is not licensed to run at this location. Please contact the licensor to either license this product, or move an existing license of your organization to this location. Use of this product at this

	location is discontinued until this problem is resolved.
2	Your license on this product has expired. Please contact licensor in order to have your license extended. Use of this product is discontinued until this problem is resolved.
3	Payment on this licensed product is over due and past your grace period. Please have your accounting department send payment in order to continue your license. Use of this product is discontinued until this problem is resolved.
4	Your current use of this licensed product exceeds limits for the number of uses your organization has licensed. Please try again later.
5	A reply from licensor to numerous authorization requests was never received. This product must be connected to a communications network in order to function.
0	Authorization is OK. There is no message.

Step 109.3 swaps source network address 21 and destination network address 22. Step 109.4 transmits datagram 3 back to monitor 2.

At step 109.5, a communications network delivers datagram 3 to monitor 2. Subsequently, procedural control returns to step 107.0 in FIGURE 3 to process the next datagram 3.

Although only a few exemplary embodiments of this invention have been described in detail above, those skilled in the art will readily appreciate that many

modifications are possible in the preferred embodiments without materially departing from the novel teachings and advantages of this invention. For example, product 1 was described as sometimes consisting of information as well as software which controls the information. This approach provides the greatest flexibility, but it is also possible to include the software which controls the information in the networked machine at the licensee's site. In this case, product 1 is split, with part of it on media and part on the licensee's machine. By doing this, some space can be saved on the media containing product 1, but the capabilities of these products will be limited by the standard functions available on these machines.

Also, the presently described embodiment includes a product 1 which is at the licensee's site. This implies that product 1 is on some physical media such as diskette, tape, or CD. However, product 1 can be electronically delivered over communications lines to the licensee and therefore might exist in the memory of the licensee's machine, rather than any physical media. In the case of a product such as music, radio programs and the like, product 1 may even be broadcast to the licensee's site for playback; thus, the product 1 would not even be "resident" in the licensee's machine.

The presently described embodiment allows the licensee to access the licensed product concurrent with the sending and receiving of datagram 3. In this way, the present invention does not inconvenience the legitimate licensee; however, for sensitive licensed products such as confidential information, the license

check monitor 2 can prevent access to the product 1 until an authorization reply datagram 3 is received.

Further, monitor 2 could be realized as an integral part of product 1. Monitor 2 could also be implemented as: 1) a separate process which is the parent process of product 1 (Such a parent process would have the authority to cancel the use of product 1); 2) a single system level task which controls license checking of all products at the licensee's site; and 3) custom logic in a digital integrated circuit (the present invention could be implemented as hardware instead of software).

Also, though the above embodiment has been described as being implemented on a computer system network where operator messages are provided on a CRT monitor or the like, the invention may be practiced on other hardware platforms by incorporating appropriate changes known to those of ordinary skill in the art. For example, in an alternative hardware embodiment such as a music or video playback device, monitor 2 is invoked by the licensee's action of pushing the "play" or similar button, and in a broadcast music application or similar system, the monitor may be invoked simply by turning the device on. The processing of monitor 2 is as described in the presently described embodiment. However, when a denial message is received or generated, monitor 2 must be able to switch "play" to "off".

The presently described embodiment is designed to be used in conjunction with a connectionless UDP (User Datagram Protocol) in the TCP/IP protocol suite as an underlying protocol. However, the present invention could also be realized using a slower,

connectionless protocol such as electronic mail or a variety of connection protocols (e. g., File Transfer Protocols (FTP), Telnet).

It is noted that protocol suites quite different from TCP/IP could be used, such as ISO (International Standards Organization) protocol. In addition, datagrams 3 could be sent over telephone systems with communications protocols such as those specified by CCITT (Consultative Committee on International Telephony and Telegraphy). In this case, telephone numbers could serve as network addresses 21, 22. Communications protocols for wireless communications such as cellular telephone can also be used to send the datagram 3.

Accordingly, all such modifications are intended to be included within the scope of this invention as defined by the following claims.

WHAT IS CLAIMED IS:

1. A method for monitoring the use of a licensed product, comprising the steps of:

generating, at regular time intervals, datagrams including an address in a communications facility, said facility address identifying a licensee;

automatically sending said datagrams from at least one licensee's site over said facility to a licensor's site while said licensed product is in use;

receiving said datagrams at said licensor's site;

storing an indication of receipt of each of said datagrams; and

counting said datagrams from each licensee as an indication of the use by the licensee of said licensed product.

2. A method as in claim 1 further wherein:

said generating step includes the step of incorporating a model number of said product in said datagrams; and

said counting step includes the step of separately counting datagrams for each product model number for each licensee.

3. A method as in claim 1, wherein said generating step includes the step of automatically obtaining said facility address that identifies said licensee from said facility without any data being provided by said licensee.

4. A method for controlling use of a licensed product comprising the steps of:

generating a request datagram including an address in a communications facility, said facility address identifying a licensee;

automatically sending said request datagram from at least one licensee's site over said facility to a licensor's site while said licensed product is in use;

receiving said request datagram at said licensor's site;

comparing said received request datagram with rules and license data at said licensor's site to determine if use of said licensed product is authorized;

sending a reply authorizing datagram to said licensee's site if use of said licensed product is approved; and

receiving said reply authorizing datagram at said licensee's site and denying the use of said product when no reply authorizing datagram is received.

5. A method as in claim 4, wherein:

said generating step includes the step of incorporating a model number of said product in said datagram;

said comparing step includes the step of comparing said rules and license data for a particular model number; and

said sending step includes the step of transmitting said reply datagram for each product model number.

6. A method as in claim 4, wherein said generating step includes the step of automatically obtaining said facility address that identifies said licensee from said facility without any data being provided by said licensee.

7. A method as in claim 4 further comprising the step of sending a reply denial datagram if use of said licensed product is not approved as determined in said comparing step, said step of automatically sending said request datagram from a licensee's site including the step of resending said request datagram if neither a reply authorizing datagram nor a reply denial datagram is received from said licensor's site within a predetermined time from sending said request datagram from said licensee's site.

8. A method as in claim 4, wherein said step of automatically sending said request datagram from said licensee's site includes the step of sending a request datagram at regular time intervals.

9. A method as in claim 4, wherein:

said generating step includes the step of providing a datagram identification code within said datagram;

said reply datagram sending step includes the step of inserting the same datagram identification code in said reply datagram; and

said reply receiving step rejects said reply authorizing datagram if the datagram identification code included in said reply authorizing datagram does

not match the datagram identification code included in said request datagram.

10. A method as in claim 4, wherein:

said comparing step includes the step of comparing said facility address that identifies said licensee with a list of valid licensee addresses to determine if said facility address is a valid address; and

said reply authorizing datagram is not sent if said facility address that identifies said licensee is not valid.

11. A method as in claim 10 further comprising the step of sending a reply denial datagram if said facility address that identifies said licensee is not valid.

12. A method as in claim 4, wherein:

said comparing step includes the step of comparing a license expiration date with a date at which said datagram is received; and

said reply authorizing datagram is not sent if the license expiration date is later than the date at which said datagram is received.

13. A method as in claim 12, further comprising the step of sending a reply denial datagram if the license expiration date is later than the date at which said datagram is received.

14. A method as in claim 4, wherein:

said comparing step includes the step of checking currentness of payments from said license; and

said reply authorizing datagram is not sent if payment is overdue.

15. A method as in claim 14, further comprising the step of sending a reply denial datagram if payment is overdue.

16. A method as in claim 4, wherein:

said generating step includes the step of incorporating in said datagram data indicative of the number of processes currently using said product at said licensee's site;

said comparing step includes the step of comparing the number of processes using said product at the licensee's site to an authorized number; and

said reply authorizing datagram is not sent if said number of processes using said product exceeds said authorized number.

17. A method as in claim 16, further comprising the step of sending a reply denial datagram if said number of processes using said product exceeds said authorized number.

18. A method as in claim 4, wherein said sending step includes the steps of sending said reply authorizing datagram when use of said product is approved and sending a reply denial datagram when use of said product is not approved, said receiving step

denying use of said product when said reply denial datagram is received.

19. A method as in claim 18, wherein said receiving and denying step denies use of said product when neither a reply authorizing datagram nor a reply denial datagram is received within a predetermined time after said request datagram is sent.

20. A method as in claim 18, further comprising the step of indicating, at a licensee's site, a reason for denial when said reply denial datagram is received.

21. A method as in claim 4, wherein:

said licensed product comprises an executable portion and a data portion; and

said method further comprises a step of controlling use of said data portion with said executable portion.

22. A method as in claim 4 further comprising a step of allowing use of said licensed product before a reply datagram is received.

23. A system for controlling licensed product comprising:

a communications facility to which at least one licensee having a license for operating a licensed product from the licensor is connected;

monitoring means, connected to said facility at a site of each said licensee, for generating a request datagram including an address of said licensee

on said facility and transmitting said request datagram over said facility to a site of said licensor, and for receiving and processing a reply datagram; and

controlling means, connected to said facility at said licensor's site, for receiving said request datagram, comparing said request datagram with rules and license data to determine if use of said licensed product is authorized and sending a reply authorizing datagram to said licensee's site if use of said product is approved; and

said monitoring means including means for denying use of said licensed product when no reply authorizing datagram is received.

24. A system as in claim 23, wherein:

said monitoring means sends request datagrams at regular time intervals during use of said licensed product; and

said controlling means further comprises means for counting said request datagrams received at said controlling means and means for computing an amount to be billed to said licensee in response to said counting.

25. A system as in claim 23 wherein:

said monitoring means incorporates a model number for said product in said request datagram; and

said controlling means comprises means for counting datagrams for each product model number for each licensee, in order to compute an amount to be billed to each licensee.

26. A system as in claim 23, wherein said monitoring means automatically obtains said facility address of said licensee from said facility without any input from said licensee.

27. A system as in claim 23, wherein:

said controlling means sends a reply denial datagram to said licensee's site if use of said product is not approved; and

said monitoring means resends said request datagram if no reply authorizing datagram and no reply denial datagram is received within a predetermined period of time after said requesting datagram is sent.

28. A system as in claim 23, wherein said monitoring means transmits request datagrams at predetermined time intervals.

29. A system as in claim 23, wherein:

said monitoring means incorporates a unique identification code in said request datagram;

said controlling means incorporates the same request datagram identification code in said reply authorizing datagram; and

said monitoring means rejects any reply authorizing datagram which does not include the same identification code as included in said request datagram.

30. A system as in claim 23, wherein said controlling means compares said facility address of said licensee with a list of valid licensee facility addresses and does not generate a reply authorizing

datagram if said facility address of said licensee is not valid.

31. A system as in claim 30, wherein said controlling means sends a reply denial datagram when said facility address is not valid.

32. A system as in claim 23, wherein said controlling means compares an expiration date of a license of said product with a date at which said request datagram is received by said controlling means, and does not generate a reply authorizing datagram, thus denying use of said product, if the license expiration date is earlier than the date at which said request datagram is received.

33. A system as in claim 32, wherein said controlling means sends a reply denial datagram if the license expiration date is earlier than the date at which said request datagram is received.

34. A system as in claim 23, wherein said controlling means generates a reply authorizing datagram, thus denying use of said product, if a payment for the use of said product is overdue.

35. A system as in claim 34, wherein said controlling means sends a reply denial datagram if payment for the use of said product is overdue.

36. A system as in claim 23, wherein:

said monitoring means includes in said request datagram data indicative of the number of

processes, at a licensee's site, currently using said product; and

said controlling means does not generate a reply authorizing datagram, thus denying a use of said product, if more than a predetermined number of processes using said product are running at the licensee's site.

37. A system as in claim 36, wherein said controlling means sends a reply denial datagram if more than said predetermined number of processes using said product are running at the licensee's site.

38. A system as in claim 23, wherein said controlling means sends a reply denial datagram if use of said product is not approved.

39. A system as in claim 38, wherein said monitoring means denies use of said licensed product when no reply authorizing datagram and no reply denial datagram is received within a predetermined time from the sending of said request datagram.

40. A system as in claim 38, further comprising means for indicating, at a licensee's site, a reason for denial when said reply denial datagram is received.

41. A system as in claim 23, wherein:

said licensed product comprises an executable portion and a data portion; and

said system further comprises means for controlling use of said data portion with said executable portion.

42. A system as in claim 41, wherein said data portion controlling means is disposed within said executable portion.

43. A system as in claim 41, wherein said data portion controlling means comprises a first partial controlling means disposed within said executable portion and a second partial controlling means disposed within said monitoring means.

44. A system as in claim 23, wherein said monitoring means includes means for permitting use of said licensed product before a reply datagram is received.

45. A system for monitoring product comprising:

a communications facility to which at least one licensee having a license for operating a licensed product from a licensor is connected;

monitoring means, connected to said facility at a site of each said licensee, for generating datagrams including an address of said licensee on said facility and transmitting said datagrams at periodic intervals over said facility to a site of said licensor; and

control means, connected to said facility at said licensor's site, for receiving said request datagrams, storing an indication of receipt of each of said datagrams and counting said datagrams from each

licensee as an indication of the use by the licensee of said licensed product.

46. A system as in claim 45, wherein said monitoring means automatically obtains said facility address of said licensee from said facility without any input from said licensee.

47. A system as in claim 45, wherein:

said monitoring means incorporates a product model number in said request datagrams; and

said controlling means separately counts request datagrams for each product model number for each licensee.

48. A method for monitoring the use of a licensed product comprising the steps of:

generating, at regular time intervals, datagrams including an address in a communications facility, said facility address identifying a licensee; and

automatically sending said datagrams from at least one licensee's site over said communications facility to a licensor's site while said licensed product is in use.

49. A method as in claim 48 further wherein:

said generating step includes the step of incorporating a model number of said product in said datagrams.

50. A method as in claim 48, wherein said generating step includes the step of automatically

obtaining said facility address that identifies said licensee from said communications facility without any data being provided by said licensee.

51. A method for controlling use of a licensed product comprising the steps of:

generating a request datagram including a facility address that identifies a licensee in a communications facility;

automatically sending said request datagram from a licensee's site over said communications facility to a licensor's site while said licensed product is in use; and

receiving a reply authorizing datagram at said licensee's site and denying the use of said product when no reply authorizing datagram is received.

52. A method as in claim 51 wherein:

said generating step includes the step of incorporating a model number of said product in said datagram.

53. A method as in claim 51, wherein said generating step includes the step of automatically obtaining said facility address that identifies said licensee from said communications facility without any data being provided by said licensee.

54. A method as in claim 51, wherein:

said reply datagram is one of at least a reply authorization datagram and a reply denial datagram; and

said step of automatically sending said request datagram from a licensee's site includes a step of resending said request datagram if neither a reply authorizing datagram nor a reply denial datagram is received within a predetermined time from sending said request datagram from said licensee's site.

55. A method as in claim 51, wherein said step of automatically sending said request datagram from said licensee's site includes the step of sending a request datagram at regular time intervals.

56. A method as in claim 51, wherein:

said generating step includes the step of providing a datagram identification code within said datagram; and

said reply receiving step rejects said reply authorizing datagram if the datagram identification code included in said reply authorizing datagram does not match the datagram identification code included in said request datagram.

57. A method as in claim 51, wherein:

said generating step includes the step of incorporating in said datagram data indicative of the number of processes currently using said product at said licensee's site.

58. A method as in claim 51, further comprising the steps of:

receiving a reply denial datagram; and

displaying, at a licensee's site, a reason for denial when said reply denial datagram is received.

59. A method as in claim 51, wherein:

said licensed product comprises an executable portion and a data portion; and

said method further comprises a step of controlling use of said data portion with said executable portion.

60. A method as in claim 51 further comprising a step of allowing use of said licensed product before a reply datagram is received.

61. A system for controlling a licensed product comprising:

a communications facility to which at least one licensee is connected;

monitoring means, connected to said communications facility at a site of each said licensee, for generating a request datagram including an address of said licensee on said communications facility and transmitting said request datagram over said communications facility, and for receiving and processing a reply authorizing datagram; and

means for denying use of said product when no reply authorizing datagram is received.

62. A system as in claim 61, wherein:

said monitoring means sends request datagrams at regular time intervals during use of said licensed product.

63. A system as in claim 61 wherein:

said monitoring means incorporates a model number for said product in said request datagram.

64. A system as in claim 61, wherein said monitoring means automatically obtains said facility address of said licensee from said communications facility without any input from said licensee.

65. A system as in claim 61, wherein:

said monitoring means resends said request datagram if no reply authorizing datagram and no reply denial datagram is received within a predetermined period of time after said requesting datagram is sent.

66. A system as in claim 61, wherein said monitoring means transmits request datagrams at predetermined time intervals.

67. A system as in claim 61, wherein:

said monitoring means incorporates a unique identification code in said request datagram; and

said monitoring means rejects any reply authorizing datagram which does not include the same identification code as included in said request datagram.

68. A system as in claim 61, wherein:

said monitoring means includes in said request datagram data indicative of the number of processes, at a licensee's site, currently using said product.

69. A system as in claim 61, wherein:

said monitoring means denies use of said licensed product when no reply authorizing datagram and no reply denial datagram is received within a predetermined time from the sending of said request datagram.

70. A system as in claim 61, further comprising means for indicating, at a licensee's site, a reason for denial when a reply denial datagram is received.

71. A system as in claim 61, wherein:

said licensed product comprises an executable portion and a data portion; and

said system further comprises means for controlling use of said data portion with said executable portion.

72. A system as in claim 71, wherein said data portion controlling means is disposed within said executable portion.

73. A system as in claim 71, wherein said data portion controlling means comprises a first partial controlling means disposed within said executable portion and a second partial controlling means disposed within said monitoring means.

74. A system as in claim 61, wherein said monitoring means includes means for permitting use of said licensed product before a reply datagram is received.

75. A system for monitoring a licensed product comprising:

a communications facility to which at least one licensee is connected;

monitoring means, connected to said communications facility at a site of each said licensee, for generating datagrams including an address of said licensee on said communications facility and transmitting said datagrams at periodic intervals over said communications facility.

76. A system as in claim 75, wherein said monitoring means automatically obtains said communications facility address of said licensee from said communications facility without any input from said licensee.

77. A system as in claim 75, wherein:

said monitoring means incorporates a product model number in said request datagrams.

78. A method for monitoring the use of a licensed product comprising the steps of:

receiving datagrams at a licensor's site on a communications facility having at least one licensee's site thereon, said datagrams being generated at regular time intervals and including a facility address that identifies a licensee in said communications facility;

storing an indication of receipt of each of said datagrams; and

counting said datagrams as an indication of the use of said licensed product.

79. A method as in claim 78 further wherein:

said datagrams include a model number of each product; and

said counting step includes the step of separately counting datagrams for each product model number for each licensee.

80. A method for controlling use of a licensed product comprising the steps of:

receiving a request datagram at a licensor's site on a communications facility having at least one licensee's site thereon, said request datagram including a facility address identifying a licensee and being automatically sent over said communications facility to said licensor's site while said licensed product is in use;

comparing said received request datagram with rules and license data at said licensor's site to determine if use of said licensed product is authorized; and

sending a reply authorizing datagram if use of said licensed product is approved.

81. A method as in claim 80 wherein:

said datagrams include a model number of said product;

said comparing step includes the step of comparing said rules and license data for a particular model number; and

said sending step includes the step of transmitting said reply datagram for each product model number.

82. A method as in claim 80 further comprising the step of sending a reply denial datagram if use of said licensed product is not approved as determined in said comparing step.

83. A method as in claim 80, wherein:

said datagrams include a datagram identification code; and

said reply datagram sending step includes the step of inserting the same datagram identification code in said reply datagram.

84. A method as in claim 80, wherein:

said comparing step includes the step of comparing said facility address that identifies said licensee with a list of valid licensee addresses to determine if said facility address is a valid address; and

said reply authorizing datagram is not sent if said facility address that identifies said licensee is not valid.

85. A method as in claim 84 further comprising the step of sending a reply denial datagram if said facility address that identifies said licensee is not valid.

86. A method as in claim 80, wherein:

said comparing step includes the step of comparing a license expiration date with a date at which said datagram is received; and

said reply authorizing datagram is not sent if the license expiration date is later than the date at which said datagram is received.

87. A method as in claim 86, further comprising the step of sending a reply denial datagram if the license expiration date is later than the date at which said datagram is received.

88. A method as in claim 80, wherein:

said comparing step includes the step of checking currentness of payments from said license; and

said reply authorizing datagram is not sent if payment is overdue.

89. A method as in claim 88, further comprising the step of sending a reply denial datagram if payment is overdue.

90. A method as in claim 80, wherein:

said datagrams include data indicative of the number of processes currently using said product at said licensee's site;

said comparing step includes the step of comparing a number of processes using said product to an authorized number; and

said reply authorizing datagram is not sent if said number of processes using said product exceeds said authorized number.

91. A method as in claim 90, further comprising the step of sending a reply denial datagram if said

number of processes using said product exceeds said authorized number.

92. A method as in claim 80, wherein said sending step includes the steps of sending said reply authorizing datagram when use of said product is approved and sending a reply denial datagram when use of said product is not approved.

93. A system for controlling a licensed product comprising:

a communications facility to which at least one licensee and a licensor are connected at a licensee's site and at a licensor's site, respectively; and

controlling means, connected to said communications facility at said licensor's site, for: receiving a request datagram, said request datagram including an address of said licensee on said communications facility and being transmitted over said communications facility to a site of said licensor; comparing said request datagram with rules and license data to determine if use of said licensed product is authorized; and sending a reply authorizing datagram to said licensee's site if use of said product is approved.

94. A system as in claim 93, wherein:

said request datagrams are sent at regular time intervals during use of said licensed product; and

said controlling means comprises means for counting said request datagrams received at said

controlling means and means for computing an amount to be billed to said licensee in response to said counting.

95. A system as in claim 93 wherein:

said datagrams include a model number for said product; and

said controlling means comprises means for counting datagrams for each product model number for each licensee, in order to compute an amount to be billed to each licensee.

96. A system as in claim 93, wherein:

said controlling means sends a reply denial datagram to said licensee's site if use of said product is not approved.

97. A system as in claim 93, wherein:

said datagrams include a unique identification code; and

said controlling means incorporates the same request datagram identification code in said reply authorizing datagram.

98. A system as in claim 93, wherein said controlling means compares said facility address of said licensee with a list of valid licensee facility addresses and does not generate a reply authorizing datagram if said facility address of said licensee is not valid.

99. A system as in claim 98, wherein said controlling means sends a reply denial datagram when said facility address is not valid.

100. A system as in claim 93, wherein said controlling means compares an expiration date of a license of said product with a date at which said request datagram is received by said controlling means, and does not generate a reply authorizing datagram, thus denying use of said product, if the license expiration date is earlier than the date at which said request datagram is received.

101. A system as in claim 100, wherein said controlling means sends a reply denial datagram if the license expiration date is earlier than the date at which said request datagram is received.

102. A system as in claim 93, wherein said controlling means generate a reply authorizing datagram, thus denying use of said product, if a payment for the use of said product is overdue.

103. A system as in claim 102, wherein said controlling means sends a reply denial datagram if payment for the use of said product is overdue.

104. A system as in claim 93, wherein:

said datagrams include data indicative of the number of processes, at a licensee's site, currently using said product; and

said controlling means does not generate a reply authorizing datagram, thus denying a use of said

product, if more than a predetermined number of processes using said product are running at the licensee's site.

105. A system as in claim 104, wherein said controlling means sends a reply denial datagram if more than said predetermined number of processes using said product are running at the licensee's site.

106. A system as in claim 93, wherein said controlling means sends a reply denial datagram if use of said product is not approved.

107. A system as in claim 93, wherein:

said licensed product comprises an executable portion and a data portion; and

said system further comprises means for controlling use of said data portion with said executable portion.

108. A system as in claim 107, wherein said data portion controlling means is disposed within said executable portion.

109. A system for monitoring a licensed product comprising:

a communications facility to which at least one licensee and a licensor are connected at a licensee's site and at a licensor's site, respectively; and

control means, connected to said communications facility at a licensor's site, for: receiving request datagrams, said request datagrams

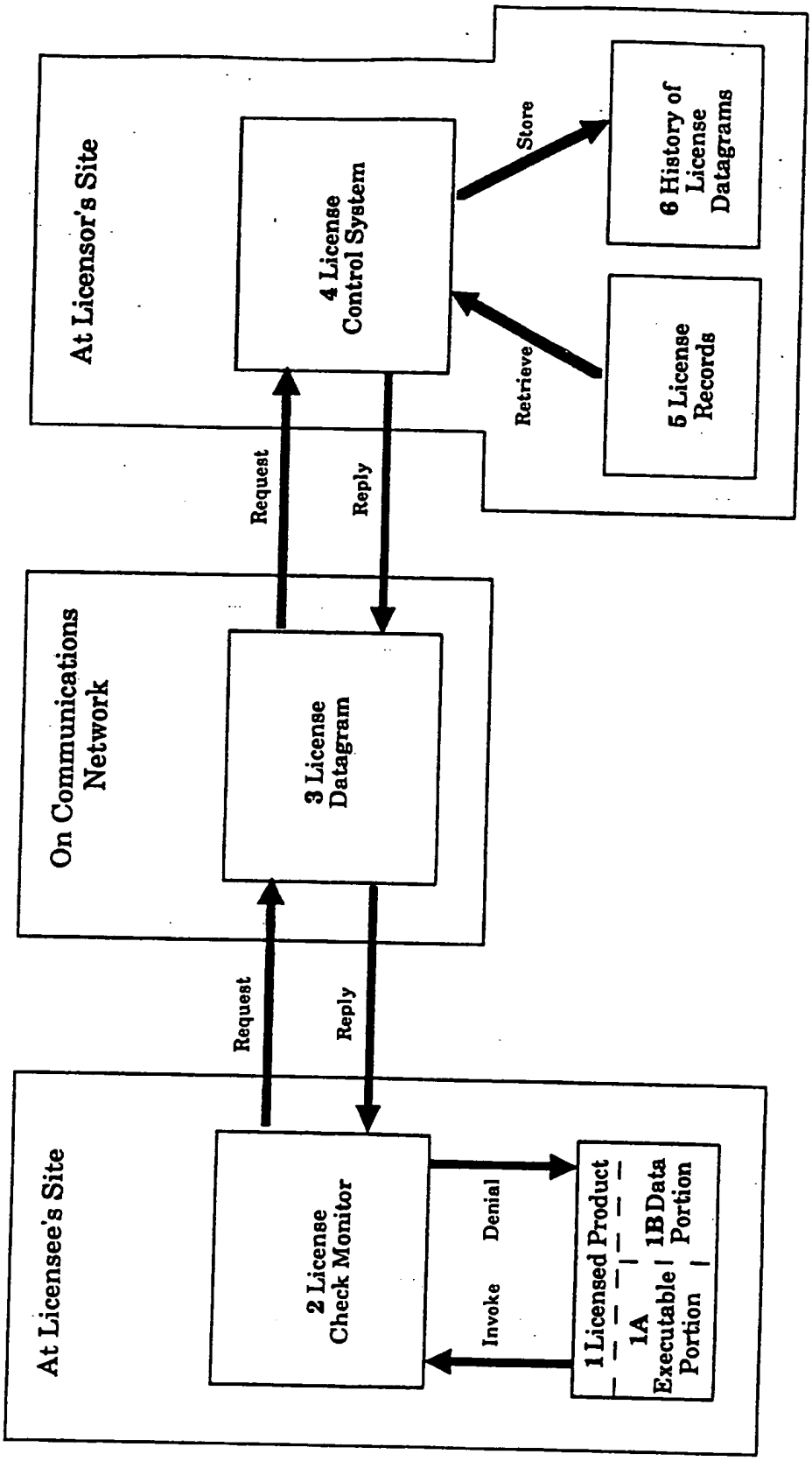
including an address of said licensee on said communications facility and being transmitted at periodic intervals over said communications facility to said licensor's site; storing an indication of receipt of each of said datagrams; and counting said datagrams from each licensee as an indication of the use by the licensee of said licensed product.

110. A system as in claim 110, wherein:

said request datagrams include a product model number; and

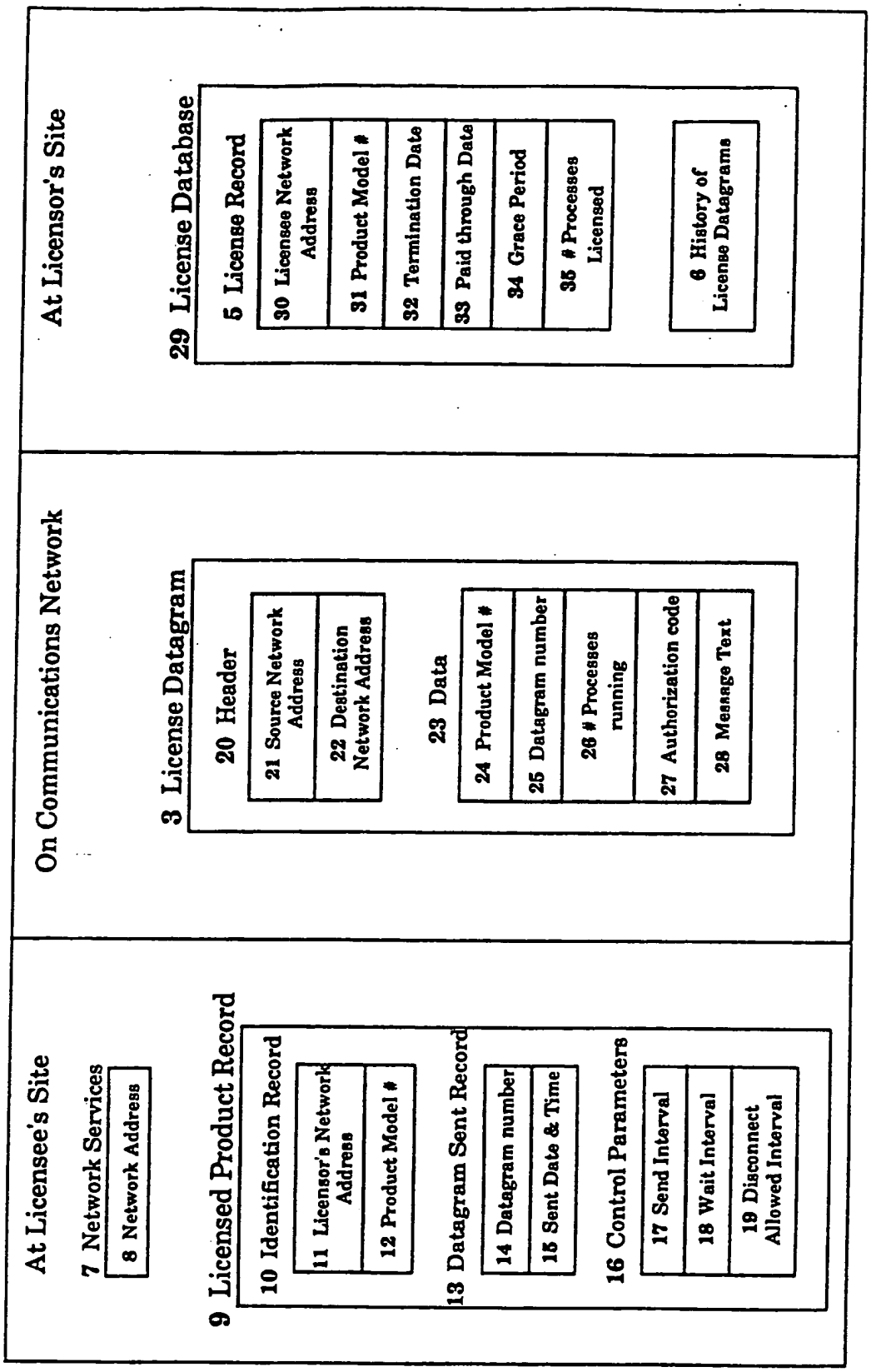
said controlling means separately counts request datagrams for each product model number for each licensee.

FIG. 1



SUBSTITUTE SHEET

FIG. 2



SUBSTITUTE SHEET

FIG. 3

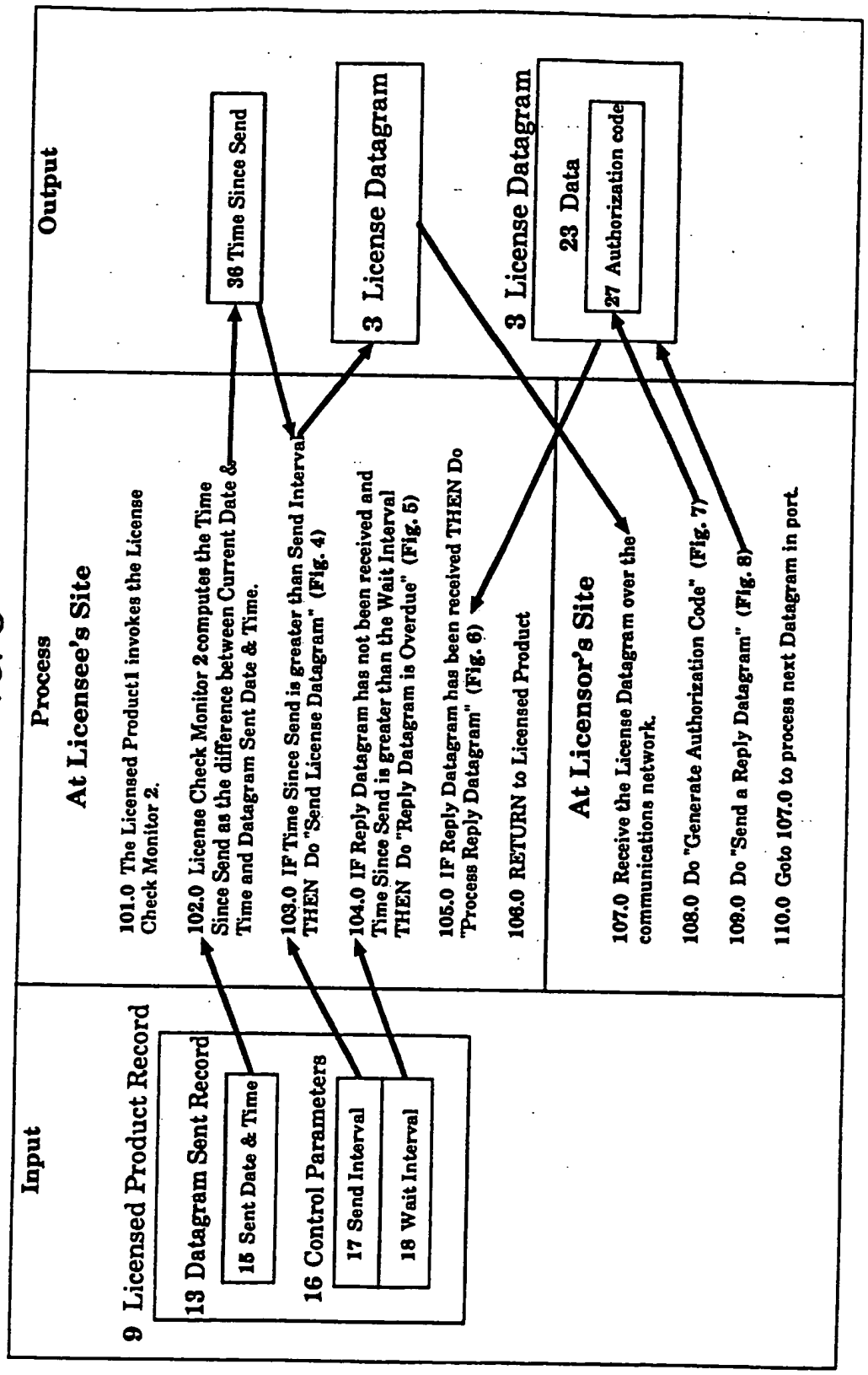


FIG. 4

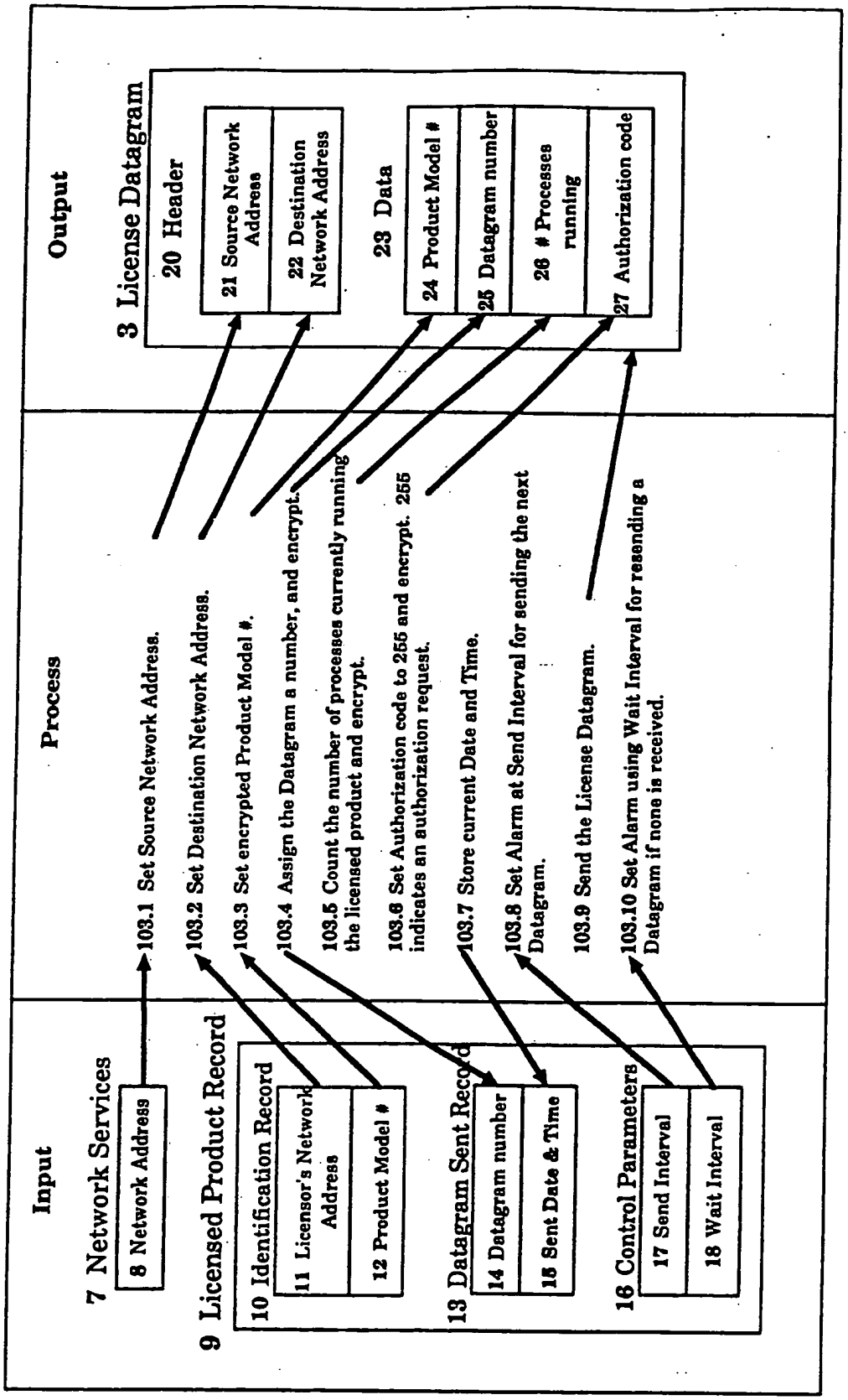


FIG. 5

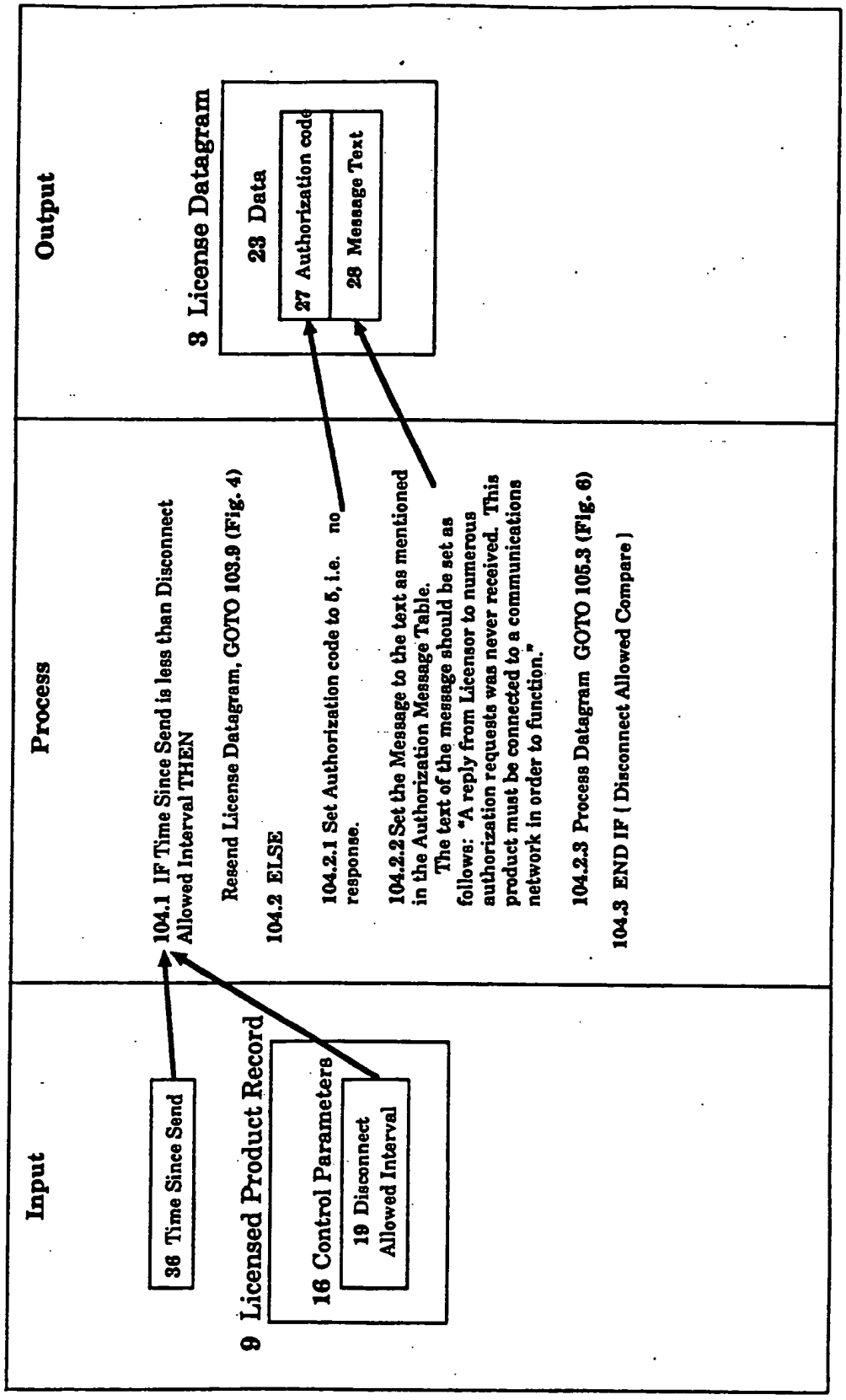


FIG. 6

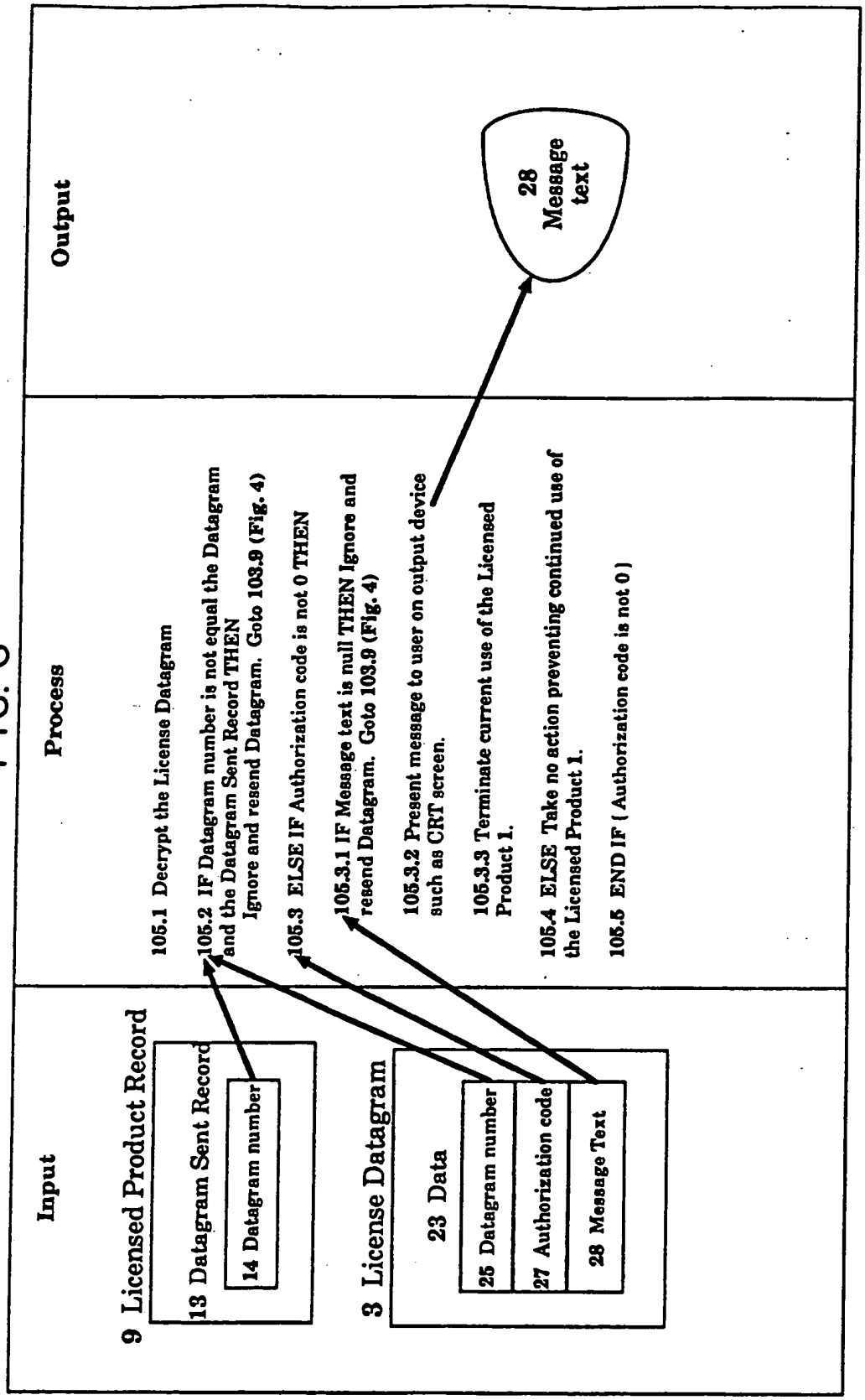


FIG. 7

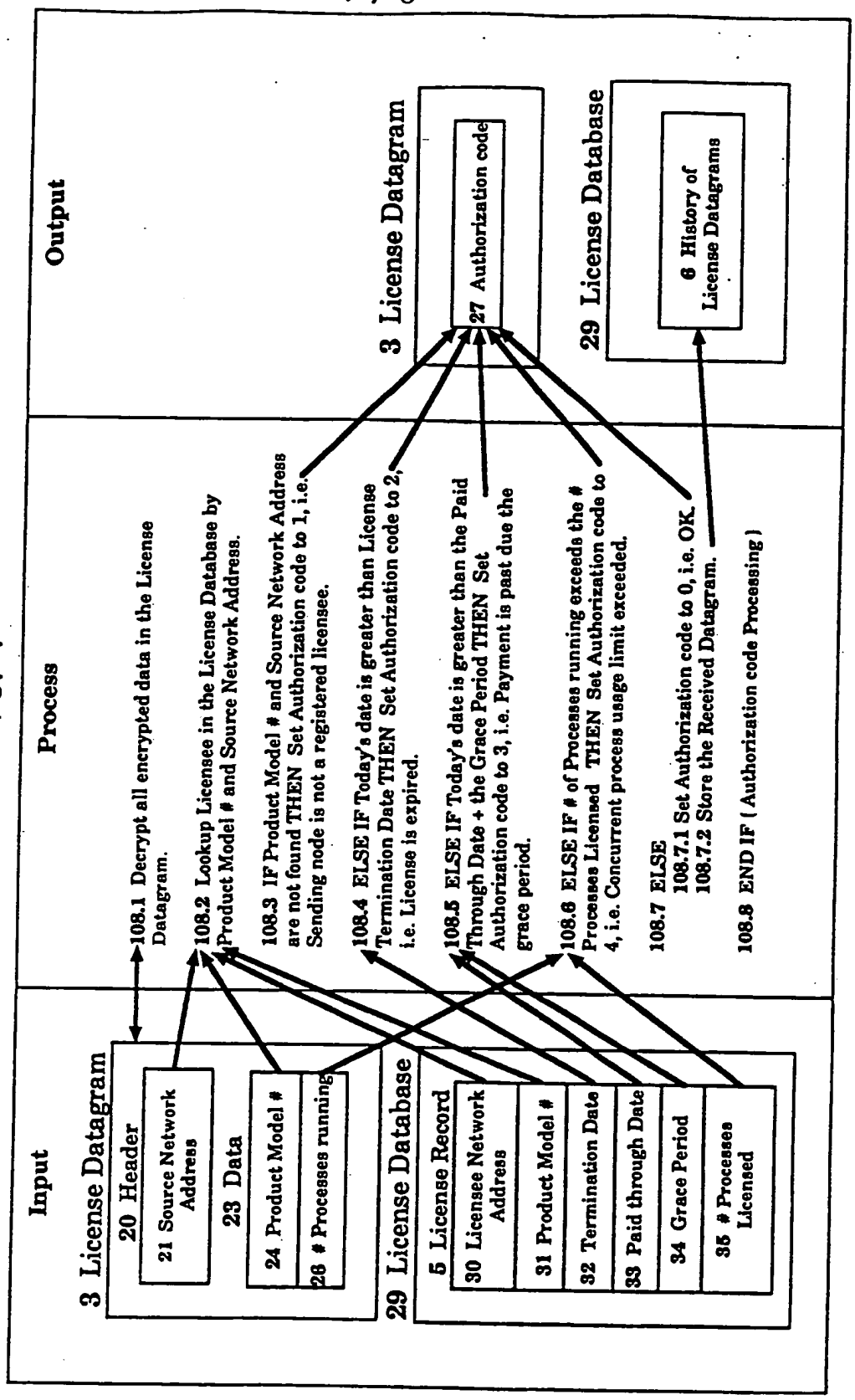
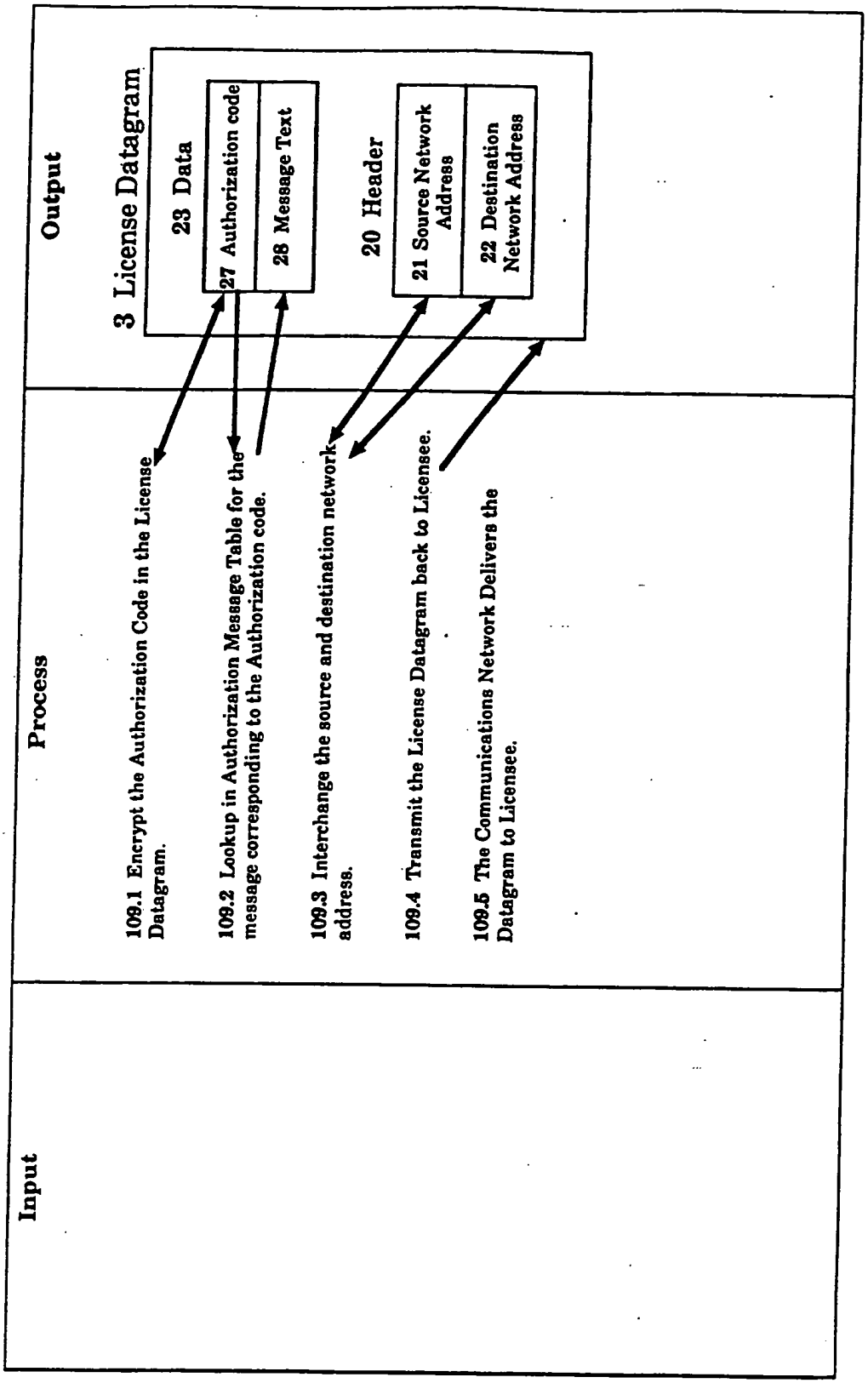


FIG. 8



INTERNATIONAL SEARCH REPORT

International application No.
PCT/US92/05387

A. CLASSIFICATION OF SUBJECT MATTER

IPC(5) :G06F 11/34; H04L 9/00
US CL :395/725; 380/4
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. : 364/406; 380/25

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

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

APS DATABASE: Software#, information, usage, monitor?, Licens?

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y,P	US,A, 5,103,476 (WAITE ET AL) 07 APRIL 1992 See entire text.	1-110
Y,P	US,A, 5,050,213 (SHEAR) 17 SEPTEMBER 1991 See column 6, lines 27-51.	1-6,9-21,23-26,29-43,45-53,56-59,61-64,67-73,75-110
Y,P	US,A, 5,047,928 (WIEDEMER) 10 SEPTEMBER 1991 See col. 6, lines 16-54.	1-6,9-21,23-36,29-43,45-53,56-59,61-64,67-73,75-110
Y	US,A, 5,023,907 (JOHNSON ET AL) 11 JUNE 1991 See entire document.	1-110

[X] 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 part 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	Date of mailing of the international search report
05 AUGUST 1992	04 NOV 1992
Name and mailing address of the ISA/ Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Authorized officer Debbie Williams KENNETH S. KIM
Facsimile No. NOT APPLICABLE	Telephone No. (703) 308-1634

Form PCT/ISA/210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US92/05387

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,A, 5,014,234 (EDWARDS, JR.) 07 MAY 1991 See col. 3, lines 4-16.	1-110
Y	US,A, 5,010,571 (KATZNELSON) 23 APRIL 1991 See entire document.	1-6,9-21,23-26,29-43,45-53,56-59,61-64,67-73,75-110.
Y	MACMILLAN Publishing Company, 1985, WILLIAM STALINGS, Data and Computer Communications. p199-203.	1-110
Y,P	US,A, 5,113,519 (JOHNSON ET AL) 12 MAY 1992 See col. 6, lines 36-68.	1-110
Y	US,A, 4,937,863 (ROBERT ET AL) 26 JUNE 1990 See col. 3, lines 25-40.	1-6,9-21,23-26,29-43,45-53,56-59,61-64,67-73,75-110

Form PCT/ISA/210 (continuation of second sheet)(July 1992)*

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 9/32 | A1

(11) International Publication Number: **WO 99/49615**

(43) International Publication Date: 30 September 1999 (30.09.99)

(21) International Application Number: PCT/US98/22238

(22) International Filing Date: 21 October 1998 (21.10.98)

(30) Priority Data:
09/049,321 27 March 1998 (27.03.98) US
09/175,559 20 October 1998 (20.10.98) US

(63) Related by Continuation (CON) or Continuation-in-Part (CIP) to Earlier Application
US Not furnished (CIP)
Filed on Not furnished

(71) Applicant *(for all designated States except US)*: MICROTOME, INC. [US/US]; 150 S. Price Road, St. Louis, MO 63124 (US).

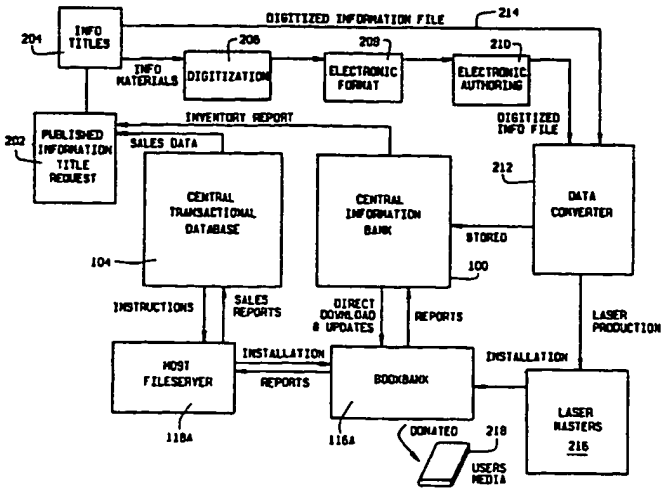
(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: SAIGH, Michael, M. [US/US]; 150 S. Price Road, St. Louis, MO 63124 (US).

(74) Agents: BEULICK, John, S. et al.; Armstrong, Teasdale, Schlafly & Davis, Suite 2600, One Metropolitan Square, St. Louis, MO 63102-2740 (US).

(81) Designated States: AL, AM, AT, AT (Utility model), AU (Petty patent), AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), EE, EE (Utility model), ES, FI, FI (Utility model), GB, GE, GH, GM, 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, SK (Utility model), SL, TJ, TM, TR, TT, UA, UG, US, 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
With international search report.

(54) Title: INFORMATION DISTRIBUTION SYSTEM



(57) Abstract

An information distribution system, in accordance with one form of the present invention, includes a central information bank (100) and a central transactional data base (104) coupled to point-of-sale delivery systems (202). Information flows between each point-of-sale delivery system and the central information bank and central transactional data base via communication network such as the telephone network, a satellite network, or any other network suitable for the transfer of information. The point-of-sale delivery systems may take one of many forms including a point of purchase delivery system, a point of rental delivery system, a "book bank" (116A) subsystem, a promotional delivery system, or any combination of such systems. Information exchanged within the system is controlled using various levels of encryption and is monitored to prevent unauthorized exchange.

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	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

INFORMATION DISTRIBUTION SYSTEM

Field of the Invention

The present invention relates to a system for distributing information in electronic form and more particularly, relates to a communication network for transmittal information between a central information bank and a user interface.

Background of the Invention

With the current information publishing and distribution system, information usually is delivered as fixed printed images on paper or fixed in other media forms such as tapes, diskettes, cartridges, laser disk, or compact disk. Prior to and at various points in the delivery process, the information usually is warehoused. Eventually, the information is delivered to retail outlets scattered throughout a distribution territory. Upon receipt of the information, the retailers either store the information or display the materials for resale.

The present publishing and distribution system has many disadvantages. One disadvantage is the amount of time and labor required for preparing, printing, producing and distributing information. Another disadvantage of the current system is the lack of control over the production quantity of information, i.e., the number of copies made and sold. The current distribution system is further disadvantaged by the time, work and costs required in publishing and distributing information updates.

In an attempt to improve the dissemination of some types of information, bulletin board networks have been established. Networks, such as Internet, also have been or are being established. Known networks generally utilize a telephone network or some other network as a communication media and can be accessed using commercially available software and almost any type of computer. As presently operated, however, such networks are unsuitable for the distribution of proprietary information and information which is intended for limited copying. The

free transfer of information using such networks provides little or no protection for copyright and proprietary information owners.

Summary of the Invention

The present invention, in one aspect, is an information distribution system which overcomes disadvantages and shortcomings of the current information publishing and distribution system. An information distribution system, in accordance with one form of the present invention, includes a central information bank and a central transactional data base coupled to point-of-sale delivery systems. The central information bank and central transactional data base do not necessarily have to be co-located and can be implemented on different, but coupled, computer systems. Information flows between each point-of-sale delivery system and the central information bank and central transactional data base via a communication network such as the telephone network, a satellite network, or any other network suitable for the transfer of information.

More specifically, information obtained from publishers is digitized, i.e., converted to an electronic form, to create a master copy in a uniform electronic format. Information obtained from publishers in digitized format simply is converted into the uniform electronic format. The master copies are stored in the central information bank.

The central transactional data base performs a record keeping function. Particularly, the transactional data base records and stores information related to each transaction performed at each point-of-sale site. Upon request, the transactional data base transmits sales data to a requesting publisher.

The point-of-sale delivery systems may take one of many forms including a point of purchase delivery system, a point of rental delivery system, a "book bank" subsystem, a promotional delivery system, or any combination of such systems. In the point of purchase delivery system, information is downloaded, i.e., copied, onto a user's storage media for later access by the user. The point of rental

delivery system is similar to the point of purchase system except that in addition to downloading information, an automatic erasure time period designation is downloaded. As explained in more detail hereinafter, the time period designation is utilized so that upon expiration of the designated time, the downloaded information is automatically erased from the user's storage media. The book bank subsystem is a sub-network established between authorized users, such as employees of a corporation. Each user within the sub-network with the proper authorization, or approval, can access designated information stored within the sub-network. Such a configuration enables sharing of information. In the promotional delivery system, promotional and other commercial information can be accessed for viewing and ordering of products.

A most important element of each point-of-sale delivery system is the user interface, sometimes referred to herein as the "Book Bank". The term Book Bank, as used herein, refers to the interface between the network and the user. Although the term Book Bank may imply "booktype" material, such term is not so limited. The material may be of many types such as movies, music, video, images, text, audio, and computer software material.

The Book Bank is a self-service, user interactive information vending device. Each Book Bank contains a high volume, local memory storage having, a customized portfolio of the most demanded information products for the particular site at which the Book Bank is located. Other information is transferred, via commercial communication networks (i.e., telephone networks, cable systems, satellite or cellular system or other similar communication networks), to a Book Bank for supplemental, secondary and less demanded purposes. A central processing unit contained within the Book Bank and coupled to the Book Bank local memory, storage controls downloading and dynamic encryption of the information.

In one embodiment, a user may select portions or entire contents of one or more books. The selected information may then be combined and downloaded to

the user's storage device, for example a cartridge. The cartridge, in one embodiment, includes a unique identification number and a determined amount of memory for storing the selected information.

Widespread use of the present invention should greatly simplify, and reduce the costs associated with, the publication and distribution of information. Particularly, the present architecture reduces the amount of time and resources required for the distribution of information. Further, information updates can easily be made simply by updating the master copy stored in the central information bank and then either writing over the copies stored in each Book Bank with the updated master copy or downloading the updated master copy and storing both the old and updated versions in each Book Bank. Each Book Bank contains an electronic index of the various information titles accessible from the Book Bank. In addition, the number of production quantities of a particular work can be readily controlled using the central transactional data base to track the number of copies made and sold, within the network, for each work.

The present invention also readily enables controlling reproduction of information and greatly simplifies updating of text, and the dynamic encryption of text provides copyright and proprietary information owners sufficient confidence in the present network to allow such information to be transmitted on the network.

Brief Description of the Drawings

These and other objects and advantages of the present invention, together with further features and advantages thereof, will become apparent from the following detailed specification when read together with the accompanying drawings, in which:

Fig. 1 illustrates one embodiment of the present information distribution system architecture;

Fig. 2 illustrates information flow in the system architecture shown in Fig. 1;

Fig. 3 is a block diagram illustration of a point of purchase delivery system;

Fig. 4 is a more detailed block diagram illustration of the host fileserver shown in Fig. 3;

Fig. 5 is a perspective view of a Book Bank embodiment;

Fig. 5A is an alternative Book Bank embodiment;

Fig. 6 is a block diagram illustration of the Book Bank circuitry;

Fig. 7 is a block diagram illustration of an end user's storage media;

Fig. 8 illustrates the information flow for the point of purchase configuration;

Fig. 9 is a block diagram illustration of certain elements of a point of rental delivery system;

Fig. 10 is a block diagram illustration of certain elements of a Book Bank subsystem;

Fig. 11 is a block diagram illustration of certain elements of a promotional delivery system; and

Fig. 12 is a flow chart illustrating the encryption process implemented in accordance with the present invention; and

Fig. 13 is another Book Bank embodiment.

Detailed Description of the Drawings

The following sections provide a brief overview of the present system and a detailed description of the system architecture. Following the detailed architecture description is a detailed description of point of sale delivery configurations. A detailed description of the various levels of encryption which may be used in the present system is then provided.

A. Brief Overview

In accordance with one embodiment of the present invention, information is distributed from a central information bank to a user's personalized storage medium. Information to be so distributed by the present system is received from outside sources either electronically, over various communication networks (e.g., telephone lines, cable systems, cellular systems or other similar commercial communication networks) or from various storage mediums (e.g., magnetic or electronic disks, cartridges, or tape reels or compact disks, laser disks, tape cassettes, etc.), or in hard copy format. If information is received in a hard copy format, it is initially converted to a standard digital format (e.g., ASCII text, DOS text or other similar standard commercially available text format) by scanning or direct transcription. Then the information is digitized, formatted, compressed and initially encrypted to form an electronic master copy which is stored in the central information bank. The master copy is duplicated electronically and dispatched electronically through a communication network, such as a telephone or satellite network, to a point-of-sale delivery system. Book Banks form a part of such a delivery system, and the electronic copies are retained in the Book Banks for downloading into a user's personalized storage medium. Initially, a user selects the information to be downloaded and a tracking entry is made into a transactional database to record the transfer. Prior to and during downloading of the copy on the user's storage medium, the information is dynamically encrypted utilizing a varying level of encryption which is dependent upon a variety of variables, for example, an economic value of the information. A "dynamic" encryption process is utilized so that only the electronic reader associated with the user card used to access the information from the Book Bank and download the information to the user storage cartridge can be utilized to display the information in an understandable text format.

B. System Architecture

Figure 1 illustrates one embodiment of the present information distribution system. The system is shown, for illustration purposes only, as being implemented across the world. Referring to Figure 1, a central information bank 100 is the central "library," or storage location, for information. Peripheral information banks 102A-F, coupled to central information bank 100, are libraries, or storage locations, for community oriented information. For example, the information stored in central information bank 100 accessed most often from the San Francisco bay area peripheral information bank 102A may not be accessed often from the peripheral information bank for Rome, Italy 102E. In any event, central information bank 100 is coupled to each peripheral information bank 102A-F to enable sharing of information. As explained in more detail hereinafter with respect to peripheral information bank 102F, each peripheral information bank 102A-F is coupled to one or more point-of-sale sites.

A central transactional data base 104 coupled to the central information bank 100 and the peripheral information banks 102A-F, serves a central record keeping function for central information bank 100 and peripheral information banks 102A-F. Central information bank 100 and central transactional data base 104 preferably, are commercially available main frame computers, such as an IBM main frame computer. The particular main frame model selected depends on the amount of information to be centrally stored in the network, the extent of record keeping functions to be performed, and the speed at which transfer and processing of information is to occur. Importantly, the present invention is not limited to any one particular computer to serve as the central information bank and/or the central transactional data base.

As shown in Figure 1 is an exploded view of the various couplings between central information bank 100 and transactional data base 104, peripheral information bank 102F and various point-of-sale delivery sites, particularly, point of purchase sites 108A-C, point of rental sites 110A-D, promotional sites 112A-D,

and Book Bank subsystem sites 114A-C. Each point of purchase site 108 includes a point of purchase transactional database, represented by a box, and a user interface, represented by a circle. As explained above, the user interface is sometimes referred to herein as the "Book Bank." Specifically, point of purchase site 108A contains Book Bank 116A and transactional data base 118A, site 108B contains Book Bank 116B and transactional data base 118B, and site 108C contains Book Bank 116C and transactional data base 118C. Since the central information bank 100 and peripheral information bank 102F, and specifically peripheral information bank memory storage unit 106A, also could serve as Book Banks, such units are illustrated as circles. Further details regarding Book Banks and transactional data bases are provided below in Section C.

As illustrated in Figure 1, each point-of-sale delivery system, such as systems 112A, 108A-B, 110A, and 114A-B, may be networked directly to peripheral information bank 102F, or the point-of-sale delivery system, such as systems 108C, 110B-D, 112B, 112D and 114B-C, may be networked to the point of purchase site 108B, which is networked to the peripheral information bank 102F. Point-of-sale delivery system configurations are explained in more detail below in Section C. At the level illustrated in Figure 1, however, it is important to understand that the delivery systems may be integrated into various combinations, such as a promotional point of rental system as shown by 110B and 112B, or a promotional point of purchase system as shown by 108B and 112C or a combination of a promotional, point of purchase, and point of rental systems as shown by 108C, 110D and 112D.

Communication network links between the central information bank 100 central transactional data base 104, peripheral information banks 102A-F, and point of sale sites can be made utilizing one or a combination of many commercials available networks such as telephone, satellite or cable networks or any other medium suitable for transmitting information in digitized format. Many well-known protocols could be used in connection with the present system. For

example, if the Internet is used as the "backbone" network, the well-known TCP/IP protocol could be used.

Figure 2 illustrates the flow of information in accordance with the embodiment of the system architecture illustrated in Figure 1. For ease of illustration only, peripheral memory storage unit 106A is consolidated into central information bank 100, and peripheral transactional data base 106B is consolidated into central transactional data base 104. It should be understood, of course, that communication links between the peripheral information bank 102F and central information bank 100 and central transactional data base 104 are provided.

As illustrated by the inputs provided to block 202, a publisher will receive inventory reports from the central information bank 100 and sales data from central transactional data base 104. Based on this and other information the publisher can determine whether to place additional information on the network. For ease of reference such information is sometimes referred to herein as "information titles" as shown in block 204. If the information is not present in an electronic format, then the information is digitized 206, disposed in an electronic format 208 and then undergoes electronic authoring 210. The digitized information is then transmitted to a data converter 212 for converting the digitized information into a uniform format. For example, if the central information bank 104 and central transactional database 104 are DOS-based systems, the data converter will convert the information into a DOS format. If the information titles are in a digitized format, the information titles are transmitted directly to data converter 212 for direct conversion into the uniform format as illustrated by line 214.

Once the data is in a uniform, digitized format, it undergoes an initial encryption and compression to both reduce the amount of storage space required to store the data and to make the data ready for being transmitted with less risk of unauthorized use while being transmitted through a communications network. The compression is accomplished through the use of one of the commercially available

compression protocols. The initial encryption is performed using one of the standard available encryption protocols as discussed below in Section D.

Once in uniform, encrypted and digitized form, the information titles are stored in central information bank 100. An electronic index listing all titles available and accessible by author, title, subject or ISBN codes is prepared. As new information titles are added, the electronic index is updated to include the new titles. The information titles may then be downloaded to Book Bank 116A. The information titles and corresponding electronic index information may, in addition to or rather than being stored in central information bank 100, be disposed on laser disk masters as illustrated at block 216. Laser disk masters 216 can then be installed directly into Book Bank 116A.

Prior to downloading desired information titles, the user may access an electronic index which contains all the information titles available for downloading from Book Bank 116A. Through the electronic index, the user obtains the listing for available information titles by author's name, by specific title of the work, by ISBN code or by subject matter. Once compiled, a listing of the available information titles included in the index category selected and the other necessary information to allow the user to purchase or rent any information title contained in the index category listed is displayed on the video screen. Using the video listing, the user selects any title listed thereon and obtains a printout of the relevant information through the printer slot 342. Upon proper access by a user, the information titles may then be downloaded from Book Bank 116A onto a user's storage media 218.

After downloading of information and corresponding electronic index information form central information bank 100 or installation of laser masters 216 to Book Bank 116A, inventory reports are generated by Book Bank 116A and transmitted to central information bank 100. These inventory reports reflect the information titles presently stored in Book Bank 116A. These reports are then are sent to publisher 202. Also, a download completion report is sent from Book Bank

116A to transactional data base 118A, sometimes referred to herein as a host fileserver, which in turn generates a status sales report. The sales report is transmitted to central transactional data base 104. Transactional data base 104 sends the necessary action instruction back to host fileserver 118A and a transaction report to publisher 202 for uses such as accounting and auditing.

For publishers who wish to allow a user to be able to produce a hard printed copy of a portion of an information title for study purposes, when the information titles are being downloaded into the master file, special authorization codes are included in the data. The codes accompany the information title to the storage medium, or cartridge, of the user. The codes limit the particular amount of the information title which the user may produce in hard copy. With such coding in place, the user may print, from the cartridge, the allowable amount of text as a hard copy. The cartridge retains information relating to such printing and restricts further printing once the limits have been reached. The user determines, within the defined limits, or authorized purposes, the portion of the text to be produced as a hard copy by using the high lighting features of the reader programming to make a selection.

C. Point-of-Sale Delivery System Configurations

The point-of-sale delivery systems, as previously discussed, are classified by function. The functions include one or more of the following: (1) point of purchase delivery system, (2) point of rental delivery system, (3) book bank subsystem, and (4) promotional delivery system. The configurations for each of these functions are separately discussed in detail below.

1. Point of Purchase Delivery System

A point of purchase system is illustrated in block form in Figure 3. The point of purchase system is described herein for illustration purposes only as a system from which books can be purchased. As pointed out above, however, the

system is not limited to books and other media capable of being expressed in electronic form such as computer software, music and video could be purchased utilizing the present system.

The point of purchase system illustrated in Figure 3 includes a Book Bank 302 coupled to host fileserver 304. Server 304 is coupled to a customer service terminal 306 (of course, there could be more than one terminal) and a cashier's station 308A which is further interconnected to other cashier stations 308B-D. Server 304 also is coupled to an institution network 310 which in turn connects to institution terminals 312A-E. Service terminal 306, cashier stations 304A-D and institution network 310 are connected to server 304 via a computer communication link such as a commercially available computer networking system such as CompuServe or Internet. Book Bank 302 and server 304 are connected to central information bank 100 and central transactional database 104 as hereinbefore explained with reference to Figs. 1 and 2.

Cashier stations 308A-D are in serial, linear networking connections which allows the addition and removal of a number of cashier stations at any time. This configuration accommodates extra cashier stations required during rush seasons or rush hours and the desire to remove cashier stations for better utilization of space after the rush seasons. Customer service terminal 306 has local processing capability that provides customer services such as personal identification initiation, personal identification number changes, processing of complimentary books, book refunds, customer information entries and updates. The customer services terminal 306 can also provide the retail outlet with internal administration and the management functions, such as the book inventory cards management, the book list management, book requests, book reports, financial reports, and E-Mail and Bulletin Board management.

Referring now to Figure 4, point of purchase fileserver 304 is shown in more detail. Particularly, server 304 includes a central processing unit (CPU) 316, a primary power supply 318, an uninterrupted power supply 320 to assure

continuous operation during power failure, and a high density storage 322 that holds all the programs and the data bases required for server 304 operation.

Server 304 has four (4) interfaces, i.e., a network interface 324, a maintenance interface 326, a customer service station interface 328 and a cashier station interface 330. CPU 316 transmits instructions to Book Bank 302, creates transaction data bases and reports, and processes orders from cashier stations 308A-D and customer service terminals 306A-D.

From network interface 324, server 304 communicates with central transaction data base 104 for electronic filing of transaction reports and communicates with Book Bank 302 to give Book Bank 302 downloading instruction orders and to receive the status reports and the inventory reports from Book Bank 302. Server 304 also is coupled, through network interface 324 to a Book Bank subsystem to receive subsystem reports in order to give instructions and orders whenever necessary, as hereinafter discussed. External network systems such as institutional or corporate network systems with local merchants terminals, community bulletin board services and others can also be coupled to the network interface 324. The network interface 324 also allows two-way connecting with interbank networks such as Cirrus, Plus or other similar data transfer network. Coupling to merchants' terminals, promotional system provides local merchants and the local business direct access to update their promotions and coupons. Maintenance interface 326 enables remote or on-site diagnosis and repair of server 304.

Customer service station interface 328 provides for communication between server 304 and customer service terminals 306A-D to handle customer service transactions. Customer service terminals 306A-D are illustrated as being coupled through a data switch 332 to a printer 334. Cashier station interface 330 provides that cashier stations 308A-D can communicate with server 304.

Figure 5 illustrates one embodiment of Book Bank 302. Book Bank 302 includes a high resolution color graphic display 336 which is a touch screen device

used to display, for example, instructions, messages, and status reports to the user, indexing information and to receive the user's touch screen input selections. Book Bank 302 also has a keypad 338 that is for the user to input a personal identification as well as other inputs. A magnetic code or other generally accepted card reader 341, shown as an insertion slot, is provided for customers' transactions with a bank card, credit card or some other form of debit card. A bar code reader 340, shown as an insertion slot, is provided to allow users to insert cards containing ISBN codes for desired information titles for reading by the Book Bank. ISBN codes may also be manually inserted by typing the relevant numbered keys on the keypad 338. A printer slot 342 also is provided to enable the user to access the output of Book Bank 302, a printer (not shown in Fig. 5), as hereinafter described, to retrieve receipts and transactions reports and ISBN access vouchers. Book Bank 302 also includes a base member 344 with a cut-out portion 346 to enable a user to stand comfortably at keypad 338. Importantly, Book Bank 302 also includes a cartridge slot 348 for the user to input a reading cartridge, as explained in detail hereinafter, to obtain a copy of the information selected for downloading.

In another embodiment and as shown in Figure 5A, the physical embodiment of Book Bank 302 may be altered . More specifically and in one embodiment, Book Bank 302 is positioned on a desk 339 using a personal computer 349, for example those available from IBM Corporation. The user operates Book Bank 302 as described above with reference to Figure 5 except, the user may for example, sit in a chair (not shown). Such a configuration may be used, for example, in a corporate, dormitory, library, or other similar environment where the user may be accessing information for longer periods of time or in a professional type environment. In other embodiments, readers 340 and 341and cartridge slot 348 each may be located within computer 349 or in separate devices which are electrically coupled to computer 349.

Figure 6 is a block diagram description of Book Bank 302 circuitry. Particularly, Bonk Bank 302 includes a central processing unit (CPU) 350 which

is coupled to display 336, keypad 338, magnetic strip reader 341 and bar code reader 340. Although CPU 350 is illustrated as one unit, it is contemplated that CPU 350 could be a parallel processor or distributed processor arrangement. Selection of CPU 350 type depends, of course, on the amount of information to be processed, the desired speed of processing and costs. CPU 350 also is coupled to an automatic teller machine (ATM) module 352 to allow transactions with ATM cards. CPU 350 is coupled to a media driver 354 which enables users to insert personalized media for acknowledgment or other functions as hereinafter discussed. Book Bank 302 also includes a primary local storage device 356 provided for the storage of all information masters selected for loading into Book Bank 302 and related index information. A secondary storage device 358 is provided to hold other programs, instructions and transaction related information. A buffer memory 360 is utilized to speed up downloading in order to accommodate high volume users during the peak seasons. A printer 362 is provided to print coupons on demand, receipts and various reports for the users. A power supply 364 provides power to printer 362. CPU 350, secondary storage device 358 and local storage 356. An uninterrupted power supply 366 coupled to primary power supply 364 assures continuous operation even during power down time.

CPU 350 is coupled to a network interface 368 to provide communication to central information bank 100, host fileserver 304 or a Book Bank subsystem, as hereinafter discussed. CPU 350 also is coupled to a wireless communication port 370, which in turn is coupled to an antenna 372. Wireless communication port 370 enables compatibility with an alternative communication media in the event that such media is required.

Figure 7 illustrates, in block diagram form, the structure for a user's personalized media storage cartridge 374. As explained hereinafter, a user inserts cartridge 374 into cartridge slot 348 for downloading of the information selected from Book Bank 302. The downloaded information is stored, in an encrypted format, on cartridge 374 together with relevant basic index information copied from

the electronic index contained in Book Bank 202 at the time of initial downloading. Cartridge 374 is compatible with readers to enable the user to view information stored on the cartridge. Cartridge 374 includes reading software 376 which, as explained in more detail hereinafter, performs sequential encryption and decryption of information. Registry correspondence segment 378 also is provided. A book file registry 380 is created at the time of downloading information onto cartridge 374. Encrypted bookfiles 382 together with relevant electronic index information are stored on cartridge 374 as well as a non-erasable permanently marked serial number 384. Cartridge 374 also contains a commercial operating system environment 386 and free disk space 388.

Figure 8 illustrates the user process and component processing which occurs when a user utilizes the point-of-purchase system described above. Particularly, once the user enters the site 390, and if the user a first-time patron 392, the user will complete a user's application form 394. The user will then take the completed application and a picture I.D. to customer service station 306A, where the user will select and input a personal identification number (PIN) and a password 396. The customer service clerk will open an account for the customer 398. The user-selected password is automatically matched with a sequentially created customer account number within the central data banks. Using the keypad accompanying the cashier station, the clerk types in the name, address and social security number for the user. The written application will then be inserted into the printer slot accompanying the cashier station. While loading the customers' information, the central data bank reviews the information to determine if there are any prior problems with the customer or other discrepancies. When the verification process is completed, the designated customer account number for the user is printed on the users application. Then, the clerk places, into the card slot, a user identification card (for example, a plastic card having dimensions of the standard credit card and containing a magnetic strip on the back on which can be placed magnetic coded information). The card is then embossed with the user's name and account number

and the magnetic strip is encoded with the applicable user codes (account number, card number and designated password information). The written application is then transmitted to the central data storage center for retention. The user now is able to use the issued card to make purchases or to rent the use of information titles. The machine used to emboss and encode the cards is a standard commercially available machine of the type currently being used in connection with the issuance of a bank's card, credit cards or debit cards.

In addition to obtaining a personal identification card, a new enrollee purchases a reader/computer or other acceptable reading device (such as a special computerized interface, or an audio or video playback device). Each such device is assigned a unique serial number and a special code number. In one embodiment, the serial number is contained on a read only memory chip enclosed within the device. All of the many cartridges which accompany each such reading device is encoded in such a manner that information recorded on the cartridge can only be read by the related reading device. This is accomplished by a simple program contained within the permanent memory of the device. In one embodiment, if the special code on the device is not the same as the number which the cartridge is seeking, then the cartridge will cause to be displayed in the reading device the words "cartridge cannot be read by this device" and to not allow any further access to any information contained on the cartridge by the particular reading device in question. If the numbers match, further access will be allowed. At the time a reading device is purchased, the clerk enters the serial number for the device into the central data bank through the cashiers station. The central data bank contains a list of the serial numbers of all approved reading devices and the corresponding special code number. The personal identification card for the customer purchasing the reading device is placed by the clerk into an appropriate slot on the cashier's station and the magnetic strip on the identification card is encoded with the applicable serial number for the reading device being purchased. Thereafter, whenever the user desires to obtain additional cartridges for reading by his or her

reading device, the user needs only present his personal identification card and the cartridge to be properly coded to the clerk and by inserting the cartridge into the cartridge slot and the identification card into the card slot and pressing the designated button on the cashiers station, the new cartridge will be correctly encoded to be readable by the user's reading device.

The requirement that reading device codes and cartridge codes match, before access will be allowed, means that issued cartridges will not be readily readable by multiple reading devices. Multiple device reading will require special programming and the granting of special allowances, or approved purposes. In one embodiment, by purpose encoding the information, a publisher may expand or limit access to those users having a proper authorization or defined purpose. For example, a publisher may purpose encode a portion of a selected book to be readable by any user, i.e., any purpose encoded. This type of purpose encoding may be used on, for example, promotional books or some governmental publications. Other information may be purpose encoded to limit access to a single user to only view the information, i.e., classified material. The requirement reduces the possibility of unauthorized use of information titles.

In operation, the customer takes the customer identification card to the book display area for shopping 400. If the customer previously opened an account, the customer will not have to go through the above described process and can proceed directly to shopping area 400, where the customer will select a book inventory card matching his book selection. The book inventory card has the book ISB numbers, a bar code and information related to the particular information title, author, publisher, and edition date printed thereon. The customer brings the selected book inventory card to the cashier's station 308A. The cashier magnetically reads the codes on the customer's I.D. card and scans or manually enters the bar codes on book inventory cards 402. The customer then makes a proper payment, and the customer codes and information title bar codes are transmitted to host fileserver 304. Server 304 searches the existing customer account file to match the

identification (i.e., pin and password) and will generate a downloaded book list file based on the bar codes from the book inventory cards or as manually loaded. Server 304 downloads the file to Book Bank 302 which electronically generates a portfolio of information titles ready to be downloaded on demand. The user can then proceed to Book Bank 302 at any later time and insert the identification card into the slot 340 of Book Bank 302 and a coded point-of-purchase cartridge 374 into Book Bank cartridge slot 348 to identify himself with a personal identification number, as illustrated in step 404. The user also enters a password 406 into the key pad 338. When Book Bank CPU 350 matches the personal identification number with a downloaded list portfolio, Book Bank CPU 350 starts downloading the requested information from local storage 356, through buffer memory 360, to media driver 354 which copies the information onto cartridge 374. As part of the downloading process, the data is dynamically encrypted to make the data uniquely readable only by authorized reading devices. The dynamic encrypting is described below in Section D. After downloading, the user removes cartridge 374 and then inserts cartridge 374 into his personal reader/computer to access the information acquired. The reader/computers are configured for long-term reading applications. The reading application software is stored on cartridges with the ability to read the applicable software on the cartridges permanently stored within the memory of the reader/computers or other authorized reading device.

A user may select portions of selected information to combine and download. In one embodiment, the user may select at least one book and select at least one portion of each selected book. If more than one portion is selected, where each portion includes up to the entire selected book, the portions are combined and downloaded to user's cartridge 374. For example, for a specific college course, a student may be required to download specific chapters from ten different books. After selecting the ten books and ten specific chapters of the selected books, the selected information is combined, encrypted using the determined level of protection, and downloaded to the students cartridge 374. Similarly, a user may

select individual tracks from music information to combine and download the selected tracks to a single cartridge 374 for playback at a later time.

2. Point of Rental Delivery System

If a user is not interested in obtaining a permanent copy of a particular work but requires a copy for a period of time, e.g., a semester, the user may prefer to visit a point of rental site rather than a point of purchase site. A point of rental system is illustrated in Figure 9. The rental system is identical to the point of purchase system previously described herein (e.g., includes a host fileserver) except with respect to the differences pointed out below. In many instances, a single site or Book Bank may serve as a point of purchase system site and a point of rental system site and a point of delivery system for promotional or commercial information site or any combination thereof. As shown in Figure 9, the point of rental system includes Book Bank user terminal hubs 410A-B coupled to terminals 412A-E, and customer service station 414. User terminals 412A-E allow a customer to do an information title search and index search of the Book Bank memory and to transmit other information between Book Bank 302 and himself. Customer service station 414 combines the function of customer service as well as the cashier's station. For example, at customer service station 414, a credit customers' debit card can be credited and the ATM operation can be overridden, via ATM module 416, if necessary. Information can be printed out from customer service station 414 via printer 418.

Point of rental storage media 420 is used in the rental system. Point of rental media 420 is the same as point of purchase media 374 except that media 420 includes an automatic erasure mechanism that erases the information downloaded after the expiration of a preset time interval. More specifically, when information is downloaded from Book Bank 302 onto media 420, Book Bank 302 also downloads a "time stamp" equal to the time period for which the user has paid to retain a copy of the information. The time stamp could take the form of a value

loaded into a memory location on media 420, which value corresponds to the rental time period. During usage, the actual usage time elapsed is subtracted from the electronically stamped time period. Once the user has consumed all of the usage hours authorized, the information title will self-destruct, i.e., be deleted from media 420. This can be achieved simply by calling a stored program which erases the information associated with the memory location where the time value is stored. For example, when the value of the memory location where the authorized usage time is stored is zero, the stored erase program would be called upon to erase the information associated with the "zero" time usage authorization.

Another method for automatic erasure is for each rental or library cartridge to contain a real time clock and independent rechargeable power supply. When the cartridge is initially encoded for use, the real time clock mechanism is activated. As rented information titles are being downloaded, an expiration date is logged into the index information for each title. Any time after the real time clock on the cartridge reaches the designated expiration date, access to the relevant information title is denied. If use of the title is not extended, after the expiration of an additional number of days, use of the cartridge will cause a permanent erasure of the information title from the cartridge memory. With this method, if the real time clock falls to operate, the cartridge will become unreadable without repair. To repair a defective cartridge, the user need only bring the cartridge together with his personal identification card to the nearest service center where the real time clock will either be repaired or the relevant information titles will be loaded onto a replacement cartridge. The user will be credited for any lost time while the cartridge was unreadable. A service center could be located, for example, near each separate point of delivery site.

The automatic erasure program could be created as an operating system module or as a separate executable program designed to be "terminate and stay resident" (TAR). A module integral with the operating system is preferred since

such a structure ensures that if the operating system is viable, the automatic erasure module is viable.

If the user still needs more time with any particular information titles, the user may return to the point of rental site and "re-rent" the information. Alternatively, it is contemplated that the user could renew the rental via a modem coupled to the reader.

With respect to the user process for renting information, when a point of rental patron enters a point of rental site, the user will use a valid ATM card, bank card, credit card, or some other debit card and proceed directly to a user terminal 412A-E. Using such a terminal, the user can perform information title searching and download an order entry to Book Bank 302. When the download entry is complete, the user will go to Book Bank 302, insert a rental media, an identification card, and a credit card, bank card, or debit card into Book Bank 302 for the transaction approval. If the transaction is not cleared or if the ATM system is not working properly, the patron can proceed to the customer service center and have the attendant manually override the ATM process, if appropriate. If the user does not have a valid ATM credit card or debit card, the user will go to the customer service center, pay the service clerk to receive credit on the Book Bank debit card. Then the customer may proceed to the user terminal where the user downloads the order entry.

After the transaction approval is cleared, the patron inserts point of rental media 420 into Book Bank media driver 354, has his personal identification card scanned and enters a password. The information is dynamically encrypted and downloaded from Book Bank 302 to media 420 with an electronic stamp of the number of hours of usage authorized for each information title or an expiration date. After the downloading, the user will apply this media on the personal reader/ computer to access the information on the media.

Typical examples for the point of rental site are libraries (commercial, education or public access) and book rental shops. The information downloaded

by the user may be free of charge to the users such as in the case of a library, or may incur certain rental fees at a predetermined rate, such as in the case of a rental shop or library charging on a per page use basis. Any given point of rental site may operate as a traditional library in allowing free use to library members for a limited period of time or may operate as a rental shop where fees are collected from users in accordance with the period of use allowed.

3. Book Bank Subsystem

A Book Bank subsystem couples to a Book Bank and host fileserver as described in more detail below. The central element of the subsystem is a Book Bank which is a modified version of the point-of-purchase Book Bank 302. The subsystem is specifically configured for the collective use by members or the staff of a commercial or business entity or a corporation. It delivers and it recalls information titles among authorized users within the business or corporate entity, and provides the capability of limiting the number of copies of a given work that may be distributed to other authorized users. If all of the licensed copies of any information titles have been checked out by the staff of an organization, then no other users may access the same information title within that particular subsystem until one or more of the licensed copies of the particular information is uploaded or recalled to the subsystem or additional copies are purchased. In one embodiment, the information is purpose encoded so as to limit a purpose to which access by a user to the information is allowed. For example, a central corporate library may allow specific users, i.e., R&D personnel, to selected information, i.e., pending patent applications. All non-R&D personnel users without the proper purpose, or authorization code, are prevented from accessing the information.

Instead of purchasing the unlimited use of a limited number of copies, a commercial or business entity may lease the limited use of an unlimited number of copies or the use of a specified portion of a given information title. Under such circumstances, the commercial or business entity would be charged each time the

subsystem is accessed from a participating work station for the portion of a specified information titled accessed and for the period of time the access occurs. By restrictions encoded on the interface between a participating work station and the subsystem, while accessing information from the subsystem, the ability of the work station to perform certain operations would be restricted. The restricted operations would be those related to the duplication or transmission of data related to information titles being accessed through the subsystem.

More specifically, and referring to Figure 10, Book Bank subsystem 422 contains a high resolution color graphic display 424 coupled to CPU 426 to display the instructions or status of subsystem 422. Subsystem 422 also includes a keyboard 428 with limited access to the system for keying selections for operating certain given functions such as product display. Subsystem 422 has a media driver 430 for the downloading of information and a local storage 432 which holds a portfolio of information the business entity has ordered for use. A secondary storage 434 is also provided to hold all the software programs that control and perform the functions of subsystem 422. Subsystem 422 further includes a power supply 436 and an uninterrupted power supply 438 to assure continuous operation during power failure downtime. A printer 440 is provided to print various reports.

Book Bank subsystem 422 has a network interface 442 that connects subsystem 422 to Book Bank 302 and host fileserver 304. Network interface 442 also may couple to the corporate or business entity network system 444. With such a structure, the corporate entity may transmit or download its own corporate proprietary information through Book Bank subsystem 422.

Media port extension interface 446 provides access by an adequate number of media drivers to the desired corporate terminals for corporate network stations. Media driver 430 is connected to the terminals or stations by a proprietor driver card. The corporate administration can utilize the dynamic encryption and the dynamic downloading function of Book Bank subsystem 422 to incorporate and accommodate the corporate proprietary information. The corporate proprietary

information may be transmitted to Book Bank subsystem 422 using an encryption process and then downloaded selectively to the destination port and to the properly identified authorized personnel. Book Bank subsystem 422 is not only a customized corporate library of copyrighted and proprietary information, but also is a corporate document security device that encrypts and dispatches the corporate documents and the corporate confidential proprietary information in the corporate network system. As part of the network interface connection linking each participating work station to the subsystem and allowing access to encrypted information, a separate unit, e.g., a memory storage unit restricts certain operations which may be performed from the work station so long as the work station has access to encrypted information from the subsystem. The restrictions limit or prevent operations related to the duplication or transmission of data.

4. Promotional Delivery System

The promotional system is a point of delivery system for promotional and commercial information. It distributes promotional and commercial information in electronic format and users may either view the digitized promotional and commercial information at the site or download the information to their personalized media for later viewing. User's call access the promotional and commercial information including the dynamic viewing electronically of advertising available discounts, commercials, special promotional events, software demos and product catalogs. Users may even shop electronically by manipulating the promotional and commercial information and placing orders through E-Mail from a personal reader/computer or by ordering directly from an interactive promotional Book Bank. The promotional Book Bank has the same structure as Book Bank 302 for the point-of-purchase system.

A promotional system in accordance with the present invention is illustrated in block diagram form in Figure 11. As in the other point-of-sale systems, Book Bank 302 is networked to host fileserver 304. The promotional system further

includes a number of promotional units 448A-D which electronically display and promote products. Unit 448A is coupled directly to central transactional data base 104 and central information bank 100 while units 448B-D are coupled to host fileserver 304. Unit 448A receives information from merchant terminals 450-A-D and host fileserver 304, receives information via merchant terminals 450E-G. More specifically, host fileserver 304 receives advertising and special offer updates from the local businesses, national or regional advertisers, and corporate sponsors through merchants terminals (MT) 450E-G. The host fileserver 304 is also networked to a central transaction data base which, in turn, provides a report to the publishers, advertisers, accounting, auditing firms, merchandise vendors, and others.

The promotional Book Bank allows selective downloading of promotional and commercial information to the user's point of rental media (see discussion in Section B, System Architecture, for explanation of such downloading) for the user's private review and personal shopping at his convenience. The promotional and the commercial information downloaded will self-destruct (i.e., automatically erase) at the expiration of a pre-determined time interval as explained above with respect to point of rental delivery systems. The promotional Book Bank also provides a user interactive self-service vending feature. The user may order products or information electronically via the network. Some of the promotional functions are: coupons on demand, virtual shopping, catalog sales, demos, subscription orders, electronic applications of credit cards, calling cards, or other types of services. Some public domain information distributed such as community events, ticket sales, institutional events or even public bulletins could also be distributed with the promotional information as a free or low cost service to the community.

The promotional and the commercial information flow is very similar to the information flow within the point-of-sale delivery system. However, rather than a publisher or copyright information owners, the information sources are local

businesses, national or regional advertisers, and appropriate sponsors through advertising agents and other entities.

5. Information Tracking

In one embodiment, for each exchange, or download, of information, a tracking entry is transmitted, or stored, in an appropriate transactional database, for example central transactional database 104, to record movement, or transfer, of information from a first location to a second location. By reviewing these tracking entries, an information owner may monitor movement of the information and take appropriate action. For example, a tracking entry may be recorded in transactional database 104 each time any information is copied into, removed from, or copied from, Book Bank 302. Based upon the tracking entries, the information owner may charge the receiving or transferring party a fee as defined by the owner of the information. The fee charged may be based on a variety of factors including, but not limited to, an economic value of the information, use or purpose of the information, number of users, and the availability from other sources. The tracking entries may also include additional data so that the information owner may determine who transferred the information, the amount of information transferred, type of transfer, i.e., rental for specified period of time, and the time of transfer. Tracking entries, in one embodiment, are recorded for all transfers, i.e., Book Bank 302 to cartridge 374 and central information bank 100 to Book Bank 302. Utilizing these entries, an information owner may also determine the type of information that is being transferred, the number of transfers, and the identification of the information receivers. For example, the information may be used to determine if certain promotional materials are being received by targeted users, or to determine responses to different information pricing strategies.

D. Encryption

The above-described point-of-sale delivery systems have the capability of performing dynamic encryption of data as the data is downloaded onto a user's storage media. Dynamic encryption refers to the process in which the Book Bank works together with the storage media to perform a proprietary encryption of downloaded data. In one embodiment, different levels of encryption are utilized based on a series of factors or variables. These variables include, but are not limited to, an economic life, a market value, a general availability, a replacement cost, time sensitivity, and potential number of users, of the information. For example, today's TV listings may have a low level, or complexity, of encryption as a result of the low market value, low replacement cost, and general availability from many sources. Conversely, a multi-volume legal treatise may, for example, have a high, or complex, level of encryption as a result of the limited availability, replacement cost, and long economic life of the information. Based on the described factors, a source, i.e., publisher, of information may decide the appropriate level of encryption for each portion of information from the initial transmission to central information bank 100 to a user's cartridge 374. The level of information encryption, in any location, i.e., bank 100, may be higher or lower than another location, i.e., cartridge 374. More specifically, each time information is downloaded, or transmitted, the level of encryption may be independently altered, or determined. For example, the level of encryption at central information bank 100 may be different, i.e., higher or lower, than the level of encryption of a book downloaded to a user's cartridge 374.

In addition to dynamic encryption, other encryption may be performed as illustrated in Figure 12. Figure 12 illustrates a three level encryption process. For example, prior to transmitting information on the network, the data may be encrypted. This facilitates preventing unauthorized users from accessing the transmitted information on the network. In addition to the pre-transport encryption, the data, may be encrypted prior to being placed in a book bank. Publishers or other

owners of the information may have approval authority over this level of encryption to provide such information owners with satisfaction that the data is adequately protected.

Once the data is stored in the book bank, dynamic encryption techniques may be used when downloading the data onto storage media. The storage media (Figure 7) includes a proprietary environment for building, reading, viewing and processing. The media also has a commercial operating system environment for processing information files. An information file directory registry forms a part of the proprietary application, and a file directory pointer is contained in the operating system application.

The dynamic encryption process, in one form, uses the permanent serial numbers stored in the storage media, the user's personal identification number, a personal signature code number, and a password to further encrypt the data stored in the book bank as the data is downloaded to a user's storage media. The personalized variables and codings are combined with various individualized information file variables to form an individualized data structure for the data downloaded to the user's personalized media. As a result, those information files are individualized pertaining to the media, the version of software, the information file itself, and other variables.

The dynamic encryption assists in reducing the possibility of the unauthorized use of proprietary or other information by causing all information downloaded through the point-of-sale delivery system to be readable and accessible by a selected number of user readers/computers. Specifically, data storage medium accessible from one reader/computer will not be accessible using another reader/computer unless such access has been prearranged such as by providing the other reader/computer with an identical user identification number and password.

Examples of well-known encryption algorithms which may used in performing the above described three level encryption include the Z8068 Data Ciphering Processor (DCP). The DCP contains the structure to encrypt and decrypt

data using National Bureau of Standards encryption algorithms. It may be used in a variety of environments including in dedicated controllers, communication concentrators, terminals and peripheral task processors in general processor systems. DCP provides a high throughput rate using cipher feedback, electronic code book or cipher block chain operating modes. The provisions of separate ports for key input, clear data and enciphered data enhances security. The host system communicates with the DCP using commands entered in the master port or through auxiliary control lines. Once set up, data can flow through the DCP at high speeds because input, output and ciphering activities can be performed concurrently.

In alternative embodiments, encryption and decryption may be performed in dedicated hardware and/or software functions. For example, each reader and cartridge 374 may include a dedicated encryption integrated circuit (IC) and a dedicated decryption IC to maximize the transfer speed of the information. The level of encryption and decryption may altered by adding additional functions and by enabling or disabling the additional levels.

With respect to dynamic encryption, the following describes one of many methods of dynamic encryption which could be used. Particularly, each regularly used alpha or numeric symbol is assigned a corresponding number as illustrated in Table 1.

Table 1

symbol	A	B	C	D	E	F	G	H	I	J	K
code	1	2	3	4	5	6	7	8	9	10	11
symbol	L	M	N	O	P	Q	R	S	T	U	V
code	12	13	14	15	16	17	18	19	20	21	22
symbol	W	X	Y	Z	0	1	2	3	4	5	6
code	23	24	25	26	27	28	29	30	31	32	33

symbol	7	8	9	.	,	;	:	+	-	x	
code	34	35	36	37	38	39	40	41	42	43	44

The serial number stored on the cartridge would be used to determine how many slots the code should shift to the left at the start the encrypting. For example, if the serial number ended with six, before starting of encryption, the code would be shifted to the left by six places. Table 2 illustrates the code table after the shift.

Table 2

symbol	A	B	C	D	E	F	G	H	I	J	K
code	7	8	9	10	11	12	13	14	15	16	17
symbol	L	M	N	O	P	Q	R	S	T	U	V
code	18	19	20	21	22	23	24	25	26	27	28
symbol	W	X	Y	Z	0	1	2	3	4	5	6
code	29	30	31	32	33	34	35	36	37	38	39
symbol	7	8	9	.	,	;	:	+	-	x	
code	40	41	42	43	44	1	2	3	4	45	6

The selected user password then is used to determine after how many symbols the code should again shift to the left. As an example, if the password were ROSE, then using the codes from Table 2, the numeric statement for rose would be 24212511. When the corresponding numbers are added together until reaching, a number between 1 and 10, the number reached in our example is 9 [18.9]. So after every 9th letter, the codes would be shifted another 6 spaces to the

left. After the encrypting of 9 letters, the codes would be set as set forth in Table 3.

Table 3

symbol	A	B	C	D	E	F	G	H	I	J	K
code	13	14	15	16	17	18	19	20	21	22	23
symbol	L	M	N	O	P	Q	R	S	T	U	V
code	24	25	26	27	28	29	30	31	32	33	34
symbol	W	X	Y	Z	0	1	2	3	4	5	6
code	35	36	37	38	39	40	41	42	43	44	1
symbol	7	8	9	.	,	;	:	+	-	x	
code	2	3	4	5	6	7	8	9	10	11	12

Because the fact that the encrypting tables are constantly shifting, under this simple method, the phrase "My brown dog has fleas." would be encrypted as follows:

19 31 6 8 24 21 29 20 6
16 27 19 12 20 13 31 18 24
23 19 37 11

Decoding using only Table 1, the coded phrase would read as follows:

S 4 F H X U 2 T F
P 0 S L T M 4 R X
W S . K

Without knowing other information, it would be very difficult to find a pattern that would allow one to decode the symbols.

Knowing the placement of the codes relative to the symbols at the start of the encryption process and the number of symbols between shifts, decoding an

encrypted phrase is simply a reversal of the process applying each of the tables in reverse.

There are any number of similar methods of dynamic encryption which use a different manner of determining how encryption codes will vary as one proceeds through the data to be encrypted. The objective, of course, is to make decoding difficult by avoiding obvious patterns associated with conventional language and number usage.

E. Tamper Protection

In one embodiment of the present invention, access to the information is monitored, or recorded, to determine attempted unauthorized access to the information. If an unauthorized access is recorded, or stored, onto a user's media, for example cartridge 374, the next time that user attempts to download additional information to cartridge 374, an unauthorized access message may be transmitted to notify the appropriate party, for example the cashier. As a result of the unauthorized access message, the cashier may revoke user's cartridge 374, notify the proper authorities, or record an entry into the user's account for future action. More specifically and in one embodiment, the unauthorized access is determined by first reading, or recording, the specific identification data from the information requester, or receiver. If the data provided by the information receiver is determined to not match, i.e., is unequal, predefined values, the unauthorized access message is recorded and information exchange is prevented. The data determination may be completed using known comparison hardware and/or software functions.

Additionally, the unauthorized access message may be generated if a user having an incorrect purpose, or authorization code, attempts to access unauthorized information. For example, in a corporate environment, if a user attempted to access information for which the user did not have the proper authorization code, an unauthorized access message is generated and may be sent to, for example a system

administrator or a security official. Different level of unauthorized access messages may also be generated. For example, a high level message may be generated if a user attempts to decrypt the information stored in various locations within the system, for example Book Bank 302 using an unauthorized device. A lower level message may be generated if a remote user has attempted to access data that is one level above that user's authorized level.

F. Other Embodiments

In another embodiment, Book Bank 302 may be configured to capture and exchange real-time information. For example, as a professor presents material to students in a classroom, the professor's presentation may be captured and converted into copyrighted text and exchanged with remote users. This conversion may be completed using known voice to text conversion systems using a known computer system. The professor's presentation may be supplemented with previously prepared, or concurrently prepared, written text. The text may be digitized and properly integrated into the text using known methods. Remote users may receive information from the professor's lecture in real-time as the material is presented or may receive the information at a later time. Remote users receive only that information which the remote user is authorized to receive from Book Bank 302 as described above.

In yet another embodiment Book Bank 302 is configured to receive audio, video, and/or computer software code. For example and in one embodiment shown in Figure 13, Book Bank 302 is coupled to a Video Cassette Recorder (VCR) 600, a stereo system 610 including a cassette recorder/player 620, a Compact Disk (CD) player and/or recorder 630, a television 640, and a computer 650. As described above, authorized information is received from Book Bank 302, and in one embodiment is stored to a storage device, for example a memory device 660. The memory device 660 may be a plurality of memory cells, for example Read Access Ram (RAM), Read Only Memory (ROM), a rotating storage unit, i.e., a hard disk,

a magnetic storage media, i.e., magnetic tape, or other storage media, for example an optical storage media. After the remote user has selected the appropriate information to receive from Book Bank 302, the information is stored in device 660. Device 660 is configured to transfer the stored information to the selected playback device, i.e., Video Cassette Recorder (VCR) 600, stereo system 610, cassette recorder/player 620, CD player/recorder 630, television 640, or computer 650. For example, in one embodiment, the remote user downloads, or receives, the entire contents of a top ten music album. The contents of the album is stored in device 660. As described above, the information may be permanently stored or may be stored for a fixed period of time or number of uses. After downloading the information to device 660, the remote user may transfer the information to stereo system 610 for listening. In another embodiment, the information may be transferred to, or through, device 660 to one of the other components, i.e., cassette recorder 620, VCR 610, CD recorder 630, or computer 650. To limit unauthorized copying or playback, the information may be playback using only those components, i.e., cassette recorder 620, VCR 610, CD recorder 630, or computer 650, coupled to device 660. For example, the remote user may download a feature movie by saving the movie on a tape using VCR 610. The remote user may then playback the movie as authorized as long as the tape is playback in VCR 610 that is coupled to device 660. Similarly, the remote user may download a software program so that the information is stored in device 660 or in a storage media in computer 650. Depending upon the authorization code of the software, the program may be configured to execute only from computer 650 when computer 650 is coupled to device 660.

The above described system facilitates controlled and monitored exchange of information between many types of information owners, distributors, and users. By using the described system, a user may obtain many types of authorized information. The user may, as determined by the information owner, purchase, rent, or obtain without charge, the authorized information. The information, in one

embodiment, is encrypted using various levels, or complexities, of encryption to prevent unauthorized access. The level of encryption depends upon a variety of factors or variables, for example, the economic life of the information. For example, Book Bank 302 may include information representing a reference dictionary and a top ten music album. Information from the reference dictionary and the album may have the same or different levels of encryption. In addition, students from a determined class may access the reference dictionary information without charge as the result of the school purchasing an unlimited use copy of the information, however, those same students would be required to purchase any information downloaded from the album. Additionally, the type of access may differ for different portions of the information. For example, a first track of the album information may be coded so that anyone may download the information without charge, however, the remaining tracks of the album information may be coded to require payment to download.

While the present invention has been described with respect to specific embodiments, many modifications, variations, substitutions, and equivalents will be apparent to those skilled in the art. Accordingly, the invention is to be considered as limited only by the spirit and scope of the appended claims.

CLAIMS:

1. Apparatus for facilitating obtaining text of a book, comprising:

a storage device having stored therein text of a plurality of books;

a processor connected to said storage device, said storage device further having stored therein a program for controlling said processor, said processor operative with the program to:

receive a book selection request;

receive a user identification associated with the book selection request; and

output encrypted text of the selected book if the user identification and book selection are valid utilizing a determined level of book encryption.

2. Apparatus in accordance with Claim 1 wherein said processor is further operative with the program to purpose encode the text of the selected book so as to limit a purpose which access by the user to the text is authorized.

3. Apparatus in accordance with Claim 1 wherein said processor is further operative with the program to encrypt the text of the selected book utilizing an identifier associated with the user.

4. Apparatus in accordance with Claim 1 wherein said determined level of book encryption is based on at least an economic value of the selected book.

5. Apparatus in accordance with Claim 1 further comprising at least one local unit communicatively coupled to said processor, said local unit comprising a memory for storing, in electronic form, information transmitted to said unit from said processor, and a local unit processor for controlling transfer of information stored in said unit to electronic storage media of system users, said local unit configured to encrypt the information when the information is to be transferred to the electronic storage media, said local unit configured to encrypt the information utilizing a determined level of information encryption.

6. Apparatus in accordance with Claim 5 wherein the information stored on the user's storage media comprises a personal signature code number and serial number.

7. Apparatus in accordance with Claim 5 wherein said determined level of information encryption is based upon at least an economic value of the selected book.

8. Apparatus in accordance with Claim 7 wherein the level of information encryption is not equal to the level of the book encryption.

9. Apparatus in accordance with Claim 1 wherein said processor is further operative with the program to determine attempted unauthorized access to the output copies of the text of a book.

10. A method for operating a computer to obtain text of a book, comprising:

inputting into the computer a book selection request;

inputting into the computer a user identification associated with the book selection request; and

outputting encrypted text of the selected book if the user identification and book selection are valid utilizing a determined level of book encryption.

11. A method in accordance with Claim 10 wherein inputting the computer a book selection comprises the steps of:

selecting at least one of the books in the memory storage; and

selecting the text of at least a portion of each the selected books.

12. A method in accordance with Claim 11 wherein inputting the computer a book selection further comprises the steps of:

determining if more than one portion is selected; and

if more than one portion is selected, then combining the selected portions.

13. A method in accordance with Claim 11 further comprising the step of generating tracking information corresponding to the selected portions and the selected books.

14. A method in accordance with Claim 10 further comprising the step of purpose encoding the outputted text of the selected book so as to limit a purpose which access by the user to the text is authorized.

15. A method in accordance with Claim 10 further comprising the step of determining a level of book encryption.

16. A method in accordance with Claim 15 wherein determining a level of book encryption comprises the step of selecting at least one of a plurality of levels of encryption code.

17. A method for operating a processor communicatively coupled to a network to obtain text of a book, the network being coupled to a memory storage having stored therein text of a plurality of books, said method comprising:

determining a book selection request;

inputting into the processor the book selection request;

inputting into the processor a user identification associated with the book selection request; and

outputting encrypted text of the selected book if the user identification and book selection request are valid utilizing a determined level of encryption.

18. A method in accordance with Claim 17 wherein determining the book selection request comprises the steps of:

reviewing the books in the memory storage;

selecting the text of at least a portion of at least one of the books in the memory storage.

19. A method in accordance with Claim 17 wherein determining the book selection request comprises the steps of:

selecting at least one books in the memory storage;

selecting the text of at least a portion of each selected book in the memory storage;

combining the selected portions of the selected books.

20. A method in accordance with Claim 19 wherein selecting the text of at least a portion of each selected book in the memory storage comprises the step of selecting the entire book.

21. A method in accordance with Claim 19 wherein selecting the text of at least a portion of each selected book in the memory storage comprises the step of selecting at least one word of each selected book.

22. A method in accordance with Claim 19 wherein selecting the text of at least a portion of each selected book in the memory storage comprises the step of selecting at least one section of each selected book.

23. A method in accordance with Claim 19 further comprising the step of generating tracking information corresponding to the selected books in the memory storage.

24. A method in accordance with Claim 23 further comprising the step of generating tracking information corresponding to the selected portions of the selected books.

25. A method in accordance with Claim 17 further comprising the step of purpose encoding the outputted text of the selected book so as to define a purpose of use which access by the user to the text is authorized.

26. A method in accordance with Claim 17 further comprising the step of determining unauthorized access to the encrypted text.

27. Apparatus for facilitating obtaining text of a book, comprising:

a storage device having stored therein text of a plurality of books;

a processor connected to said storage device, said storage device further having stored therein a program for controlling said processor, said processor operative with the program to:

receive a book selection request;

receive a user identification associated with the book selection request; and

dynamically encrypting text of the selected book utilizing a determined level of encryption if the book selection request and user identification are valid, the dynamic encryption performed using a user identifier as the text is outputted.

28. Apparatus in accordance with Claim 27 wherein said apparatus is communicatively coupled to a communications network.

29. Apparatus in accordance with Claim 27 wherein said processor is further operative with the program to purpose encode the text of the selected book so as to limit a purpose which access by the user to the text is authorized.

30. Apparatus in accordance with Claim 27 further comprising at least one local unit communicatively coupled to said processor, said local unit comprising a memory for storing, in electronic form, information transmitted to said unit from said processor, and a local unit processor for controlling transfer of information stored in said unit to electronic storage media of system users, said local unit configured to encrypt the information when the information is to be transferred to the electronic storage media, said local unit configured to encrypt the information utilizing a determined level of encryption.

31. A method for operating a computer to obtain information, comprising:

inputting into the computer an information selection request;

inputting into the computer a user identification associated with the information selection request; and

outputting encrypted information of the selected information if the user identification and information selection are valid utilizing a determined level of information encryption.

32. A method in accordance with Claim 31 wherein inputting the computer an information selection comprises the step of selecting at least one portion of information in the memory storage.

33. A method in accordance with Claim 32 wherein inputting the computer a book selection further comprises the steps of:

determining if more than one portion is selected; and

if more than one portion is selected, then combining the selected portions.

34. A method in accordance with Claim 32 further comprising the step of generating tracking information corresponding to the selected portions of information.

35. A method in accordance with Claim 31 further comprising the step of purpose encoding the encrypted information so as to limit a purpose which access by the user to the information is authorized.

36. A method in accordance with Claim 31 wherein the information includes at least one of visual, music, software, and video information.

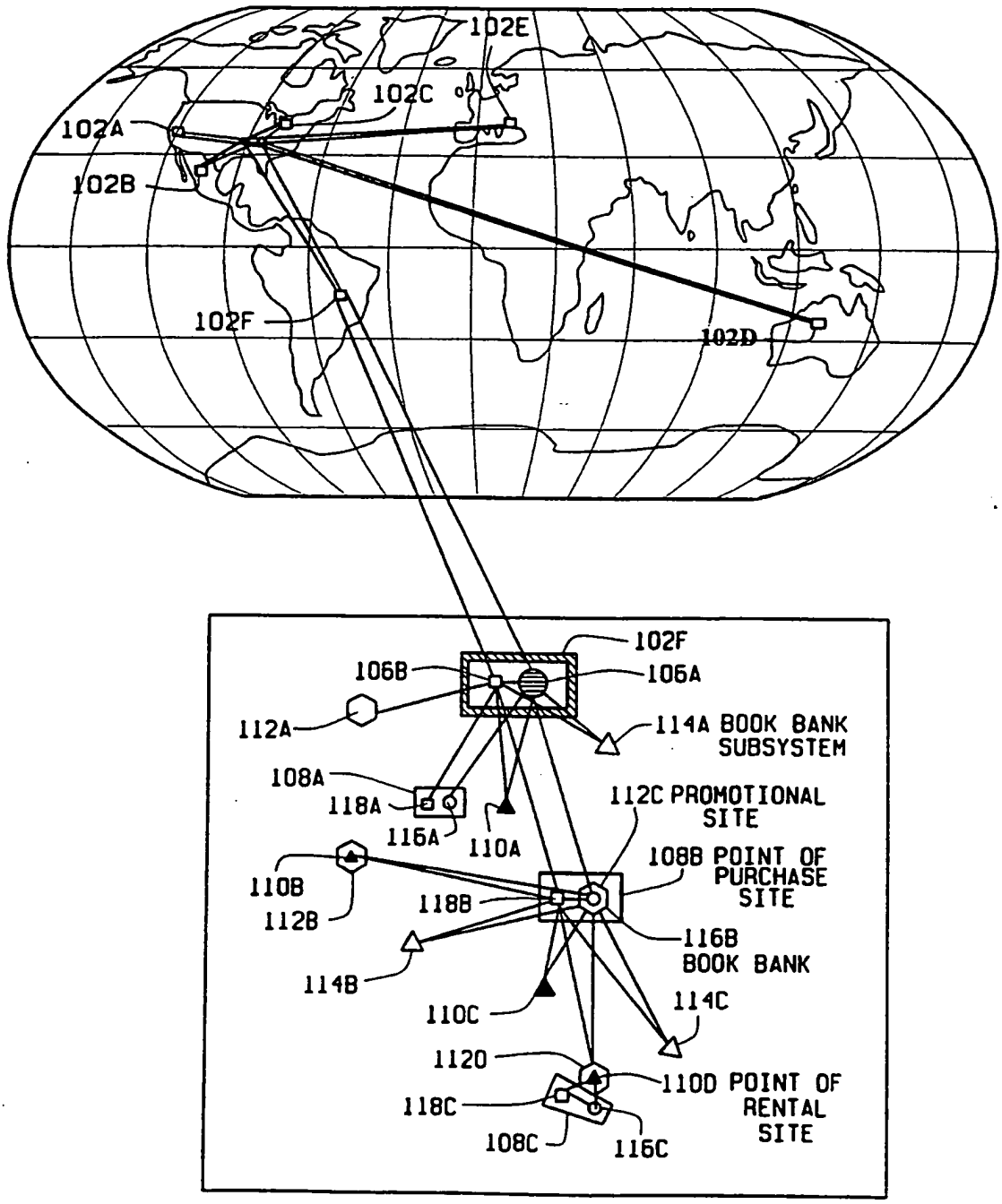

102E
102C
102A
102B
102F
102D
102F
106B
106A
112A
114A BOOK BANK SUBSYSTEM
108A
118A
116A
110A
112C PROMOTIONAL SITE
110B
112B
118B
108B POINT OF PURCHASE SITE
116B BOOK BANK
114B
110C
114C
1120
110D POINT OF RENTAL SITE
118C
108C
116C

FIG. 1

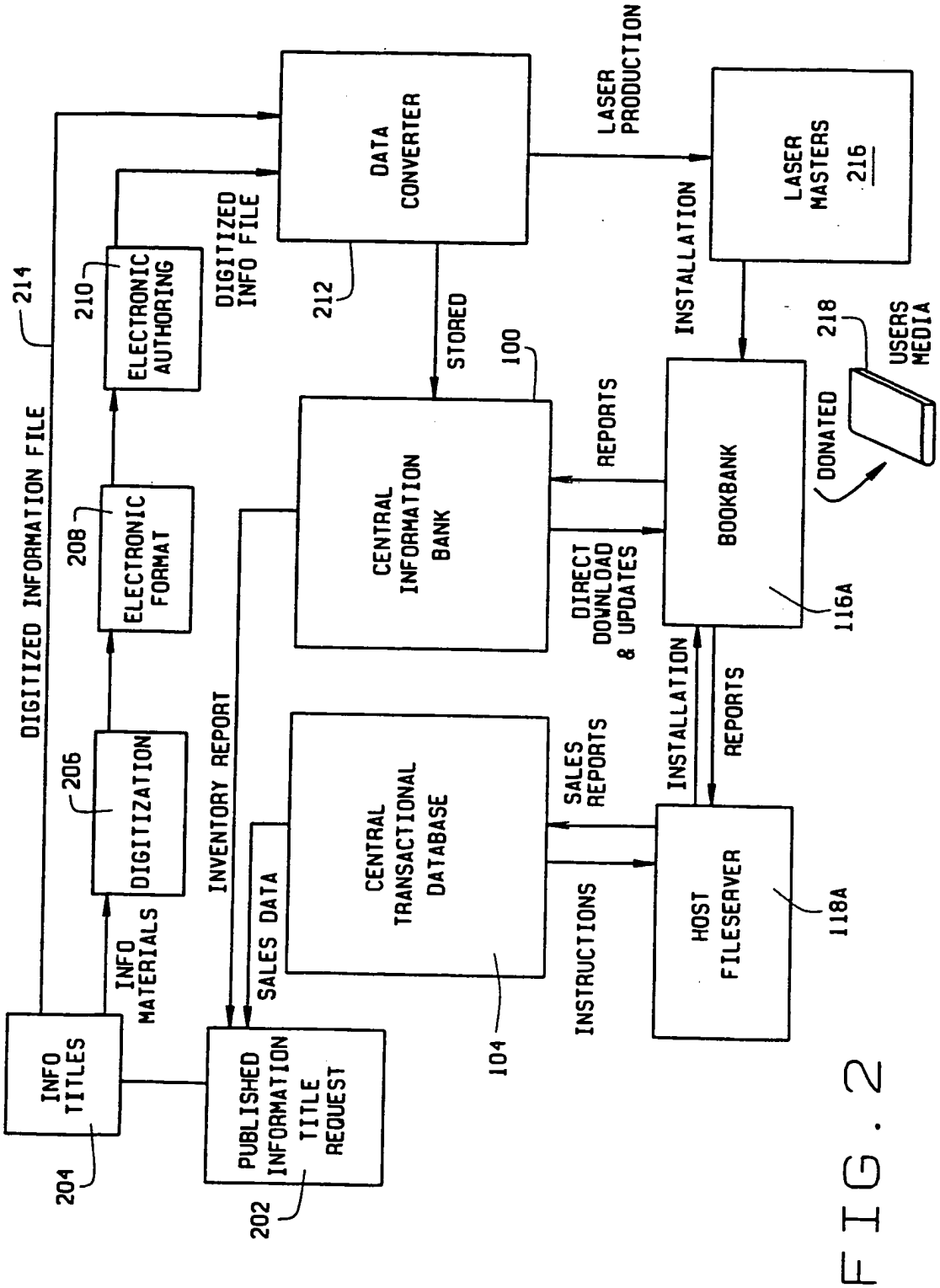

INFO TITLES
204
PUBLISHED INFORMATION TITLE REQUEST
202
INFO MATERIALS
DIGITIZATION
206
ELECTRONIC FORMAT
208
ELECTRONIC AUTHORING
210
DIGITIZED INFO FILE
212
DIGITIZED INFORMATION FILE
214
DATA CONVERTER
LASER PRODUCTION
LASER MASTERS
216
INSTALLATION
USERS MEDIA
218
DONATED
BOOKBANK
116A
REPORTS
STORED
100
CENTRAL INFORMATION BANK
DIRECT DOWNLOAD & UPDATES
INVENTORY REPORT
SALES DATA
CENTRAL TRANSACTIONAL DATABASE
104
SALES REPORTS
INSTRUCTIONS
INSTALLATION
REPORTS
HOST FILESERVER
118A

FIG. 2

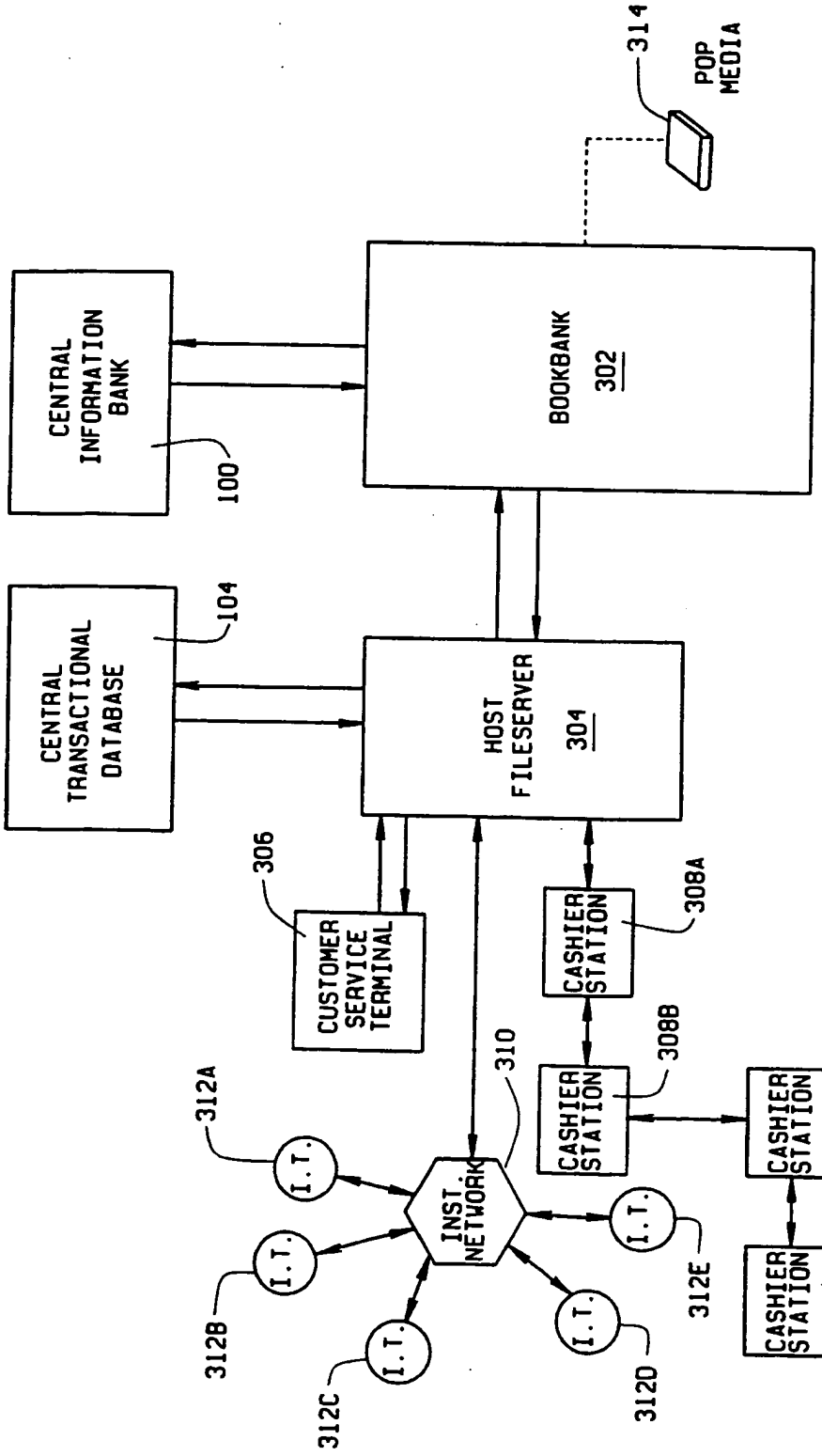

314
POP MEDIA
CENTRAL INFORMATION BANK
100
BOOKBANK
302
CENTRAL TRANSACTIONAL DATABASE
104
HOST FILESERVER
304
306
CUSTOMER SERVICE TERMINAL
CASHIER STATION
308A
CASHIER STATION
308B
CASHIER STATION
308C
CASHIER STATION
308D
310
INST. NETWORK
312A
312B
312C
312D
312E
I.T.

FIG. 3

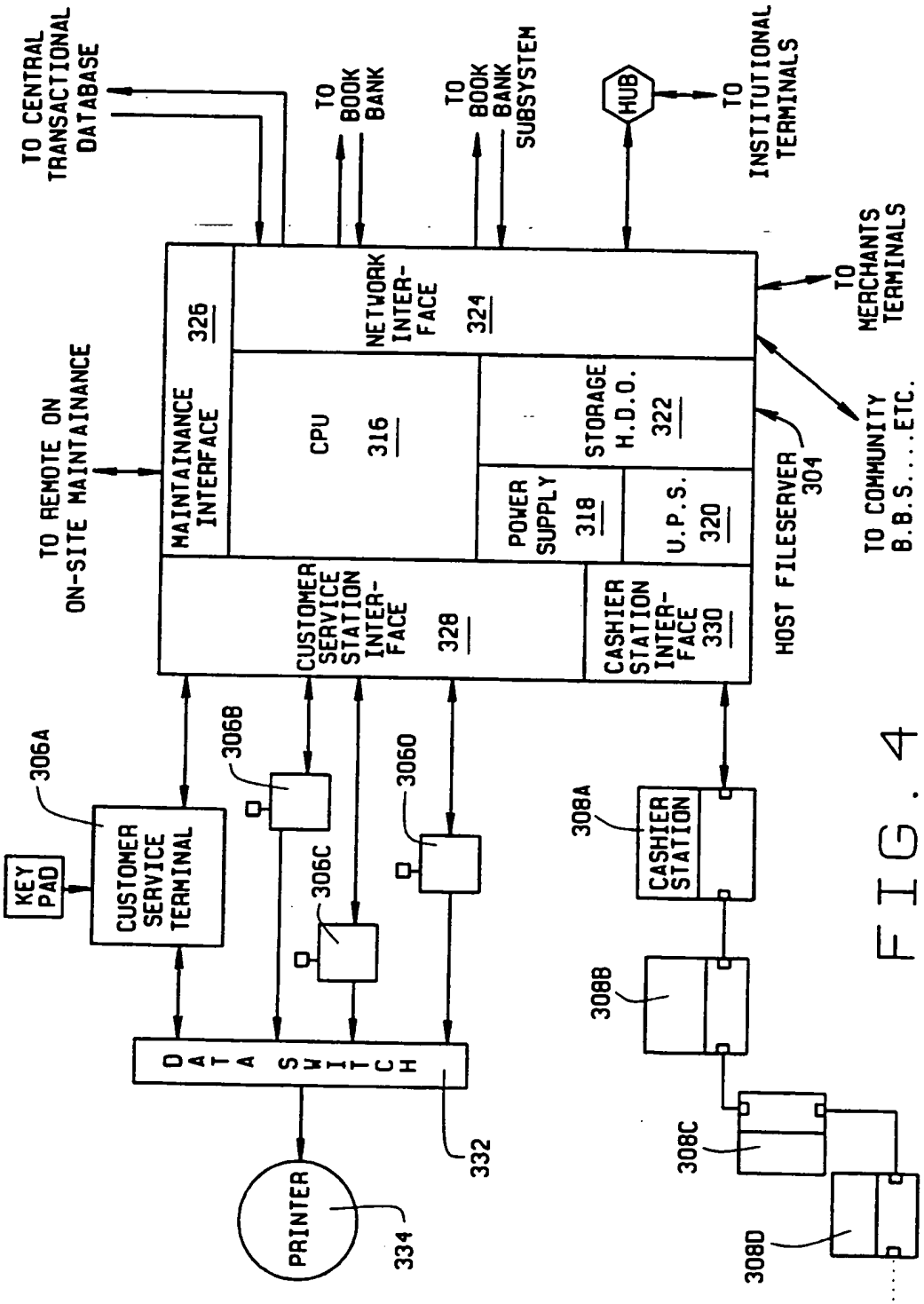

TO CENTRAL TRANSACTIONAL DATABASE
TO BOOK BANK
TO BOOK BANK SUBSYSTEM
HUB
TO INSTITUTIONAL TERMINALS
TO MERCHANTS TERMINALS
TO REMOTE ON ON-SITE MAINTAINANCE
MAINTAINANCE INTERFACE 326
NETWORK INTER-FACE 324
CPU 316
STORAGE H.D.O. 322
POWER SUPPLY 318
U.P.S. 320
HOST FILESERVER 304
TO COMMUNITY B.B.S.....ETC.
CUSTOMER SERVICE STATION INTER-FACE 328
CASHIER STATION INTER-FACE 330
306A
306B
306C
306D
KEY PAD
CUSTOMER SERVICE TERMINAL
308A
CASHIER STATION
308B
308C
308D
DATA SWITCH
332
PRINTER
334

FIG. 4

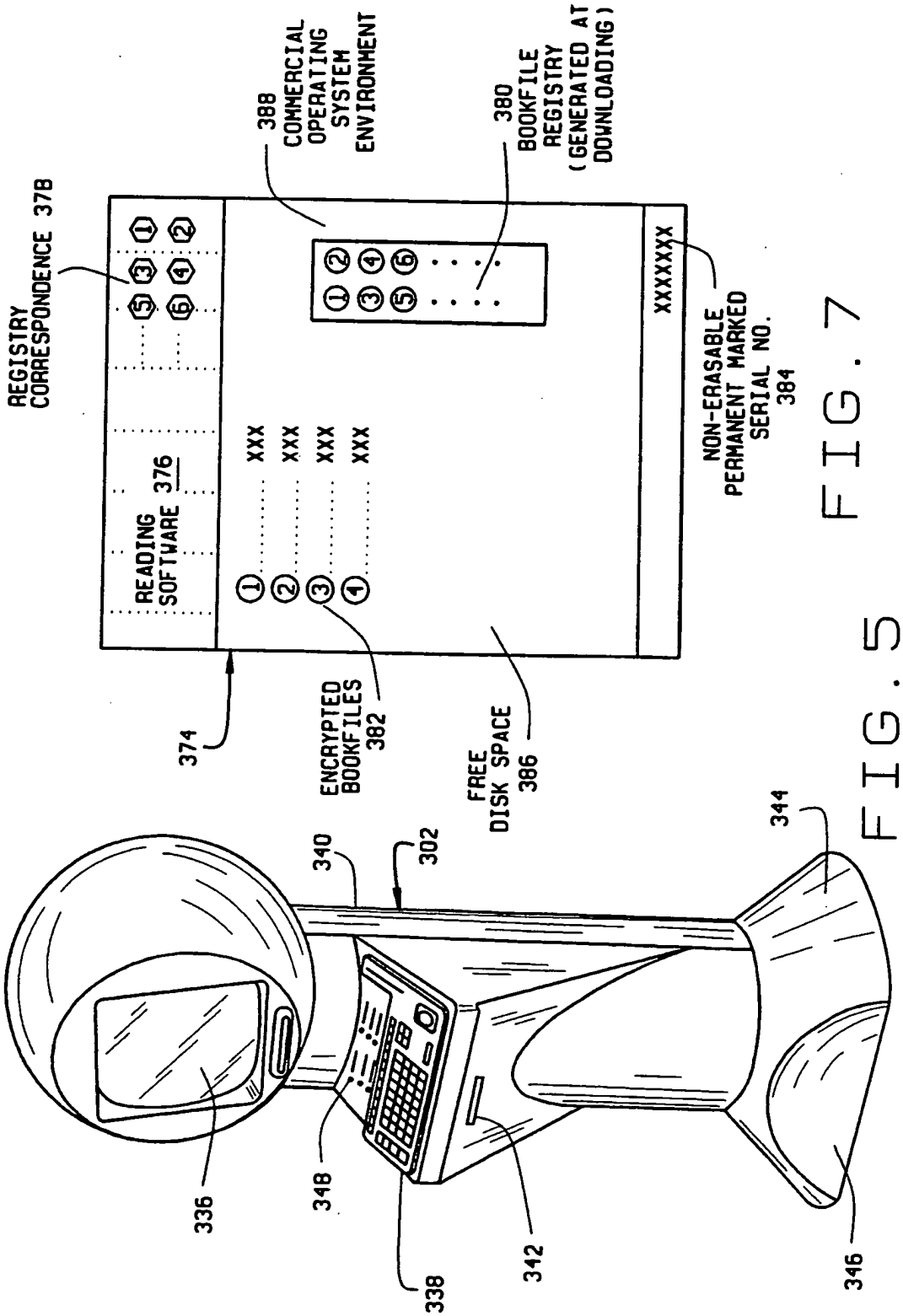

REGISTRY CORRESPONDENCE 378
READING SOFTWARE 376
388 COMMERCIAL OPERATING SYSTEM ENVIRONMENT
380 BOOKFILE REGISTRY (GENERATED AT DOWNLOADING)
XXXXXXX
NON-ERASABLE PERMANENT MARKED SERIAL NO. 384
XXX
XXX
XXX
XXX
374
ENCRYPTED BOOKFILES 382
FREE DISK SPACE 386
340
302
344
336
348
338
342
346

FIG. 5

FIG. 7

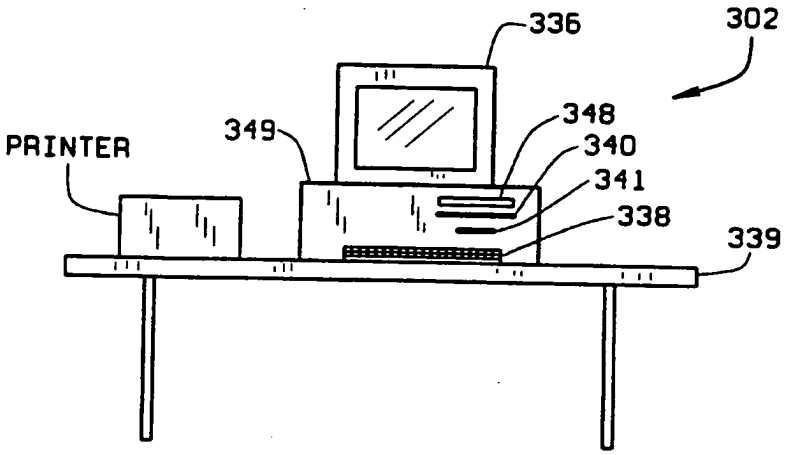

336
302
348
349
340
PRINTER
341
338
339

FIG. 5A

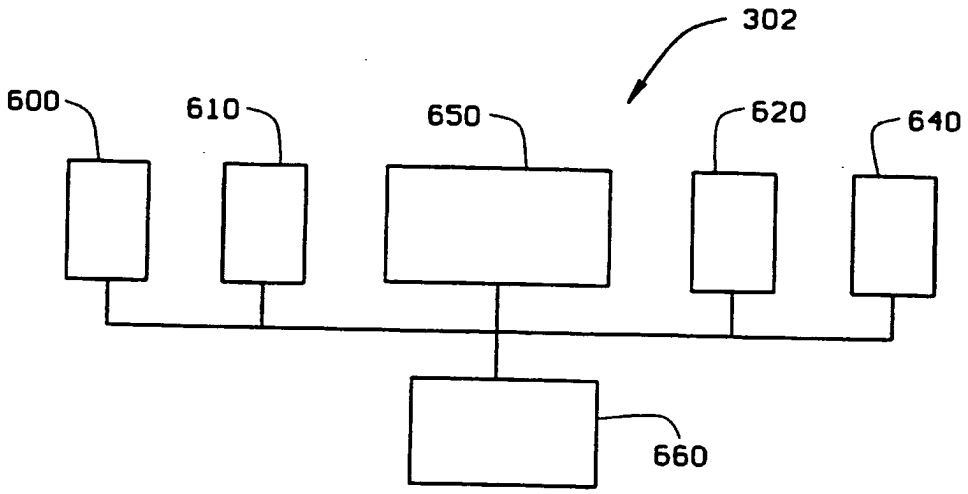

302
600
610
650
620
640
660

FIG. 13

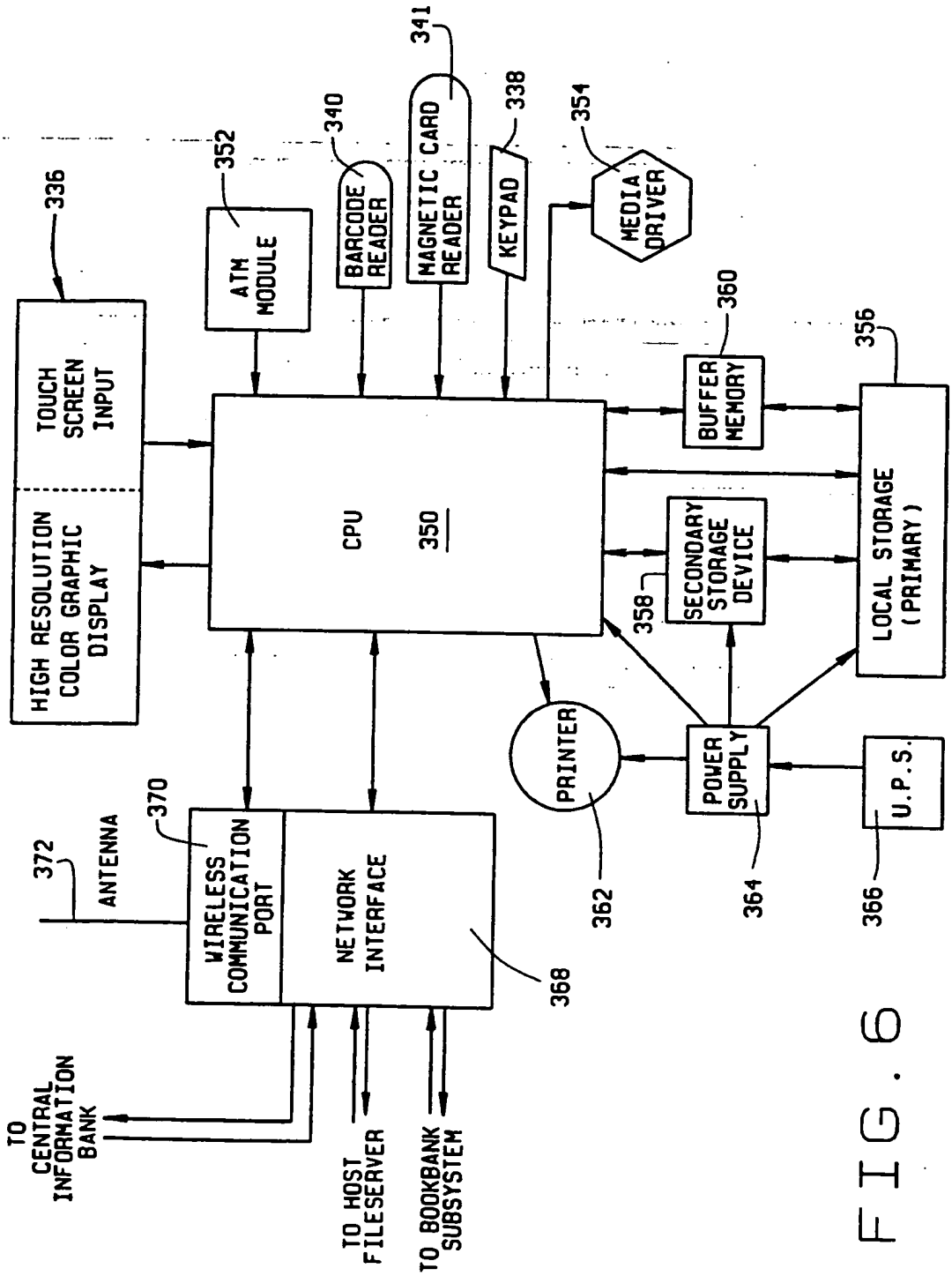

TOUCH SCREEN INPUT
HIGH RESOLUTION COLOR GRAPHIC DISPLAY
336
ATM MODULE
352
BARCODE READER
340
MAGNETIC CARD READER
341
KEYPAD
338
MEDIA DRIVER
354
CPU
350
BUFFER MEMORY
360
SECONDARY STORAGE DEVICE
358
LOCAL STORAGE (PRIMARY)
356
PRINTER
362
POWER SUPPLY
364
U.P.S.
366
ANTENNA
372
WIRELESS COMMUNICATION PORT
370
NETWORK INTERFACE
368
TO CENTRAL INFORMATION BANK
TO HOST FILESERVER
TO BOOKBANK SUBSYSTEM

FIG. 6

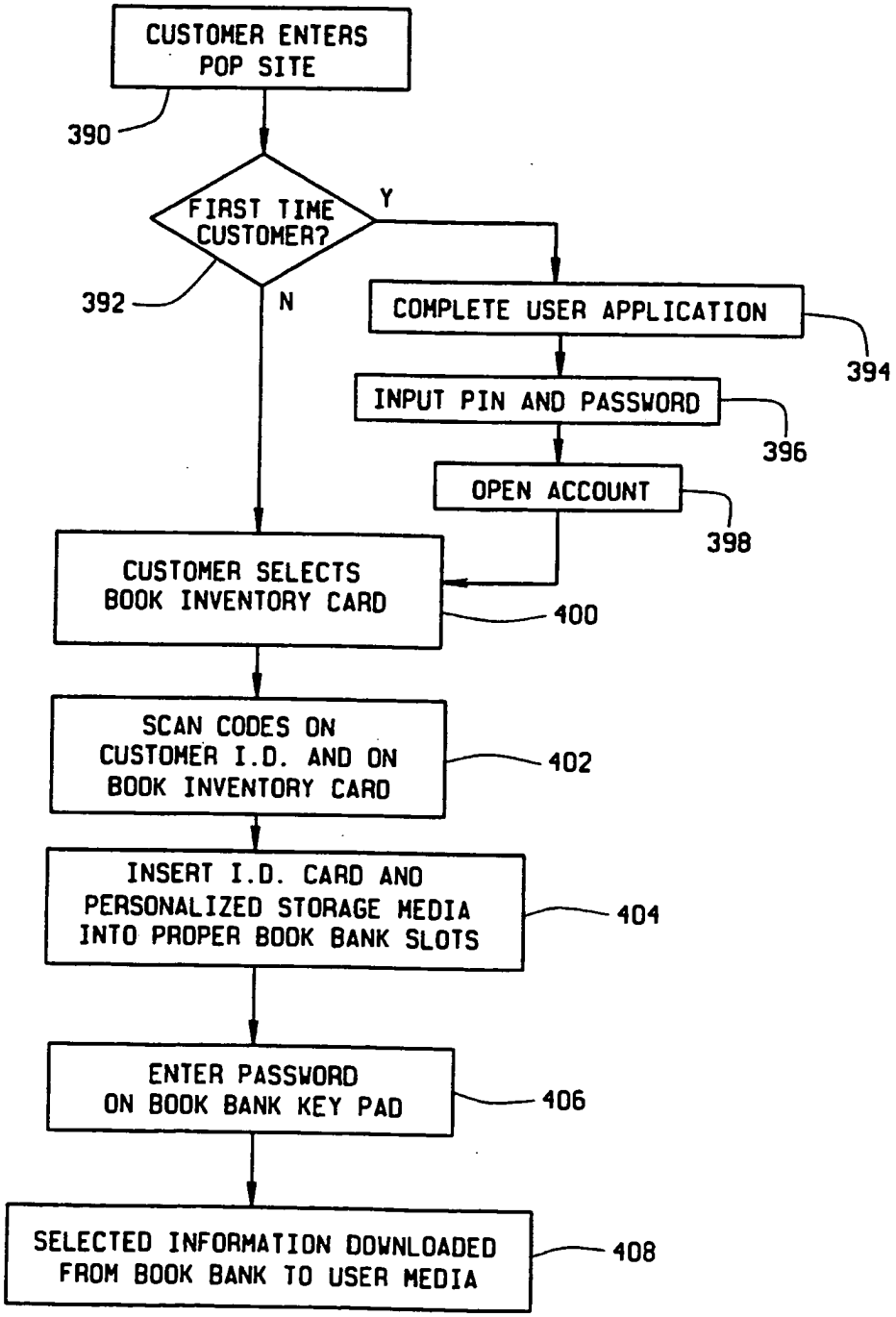

CUSTOMER ENTERS POP SITE
390
FIRST TIME CUSTOMER?
Y
N
392
COMPLETE USER APPLICATION
394
INPUT PIN AND PASSWORD
396
OPEN ACCOUNT
398
CUSTOMER SELECTS BOOK INVENTORY CARD
400
SCAN CODES ON CUSTOMER I.D. AND ON BOOK INVENTORY CARD
402
INSERT I.D. CARD AND PERSONALIZED STORAGE MEDIA INTO PROPER BOOK BANK SLOTS
404
ENTER PASSWORD ON BOOK BANK KEY PAD
406
SELECTED INFORMATION DOWNLOADED FROM BOOK BANK TO USER MEDIA
408

FIG. 8

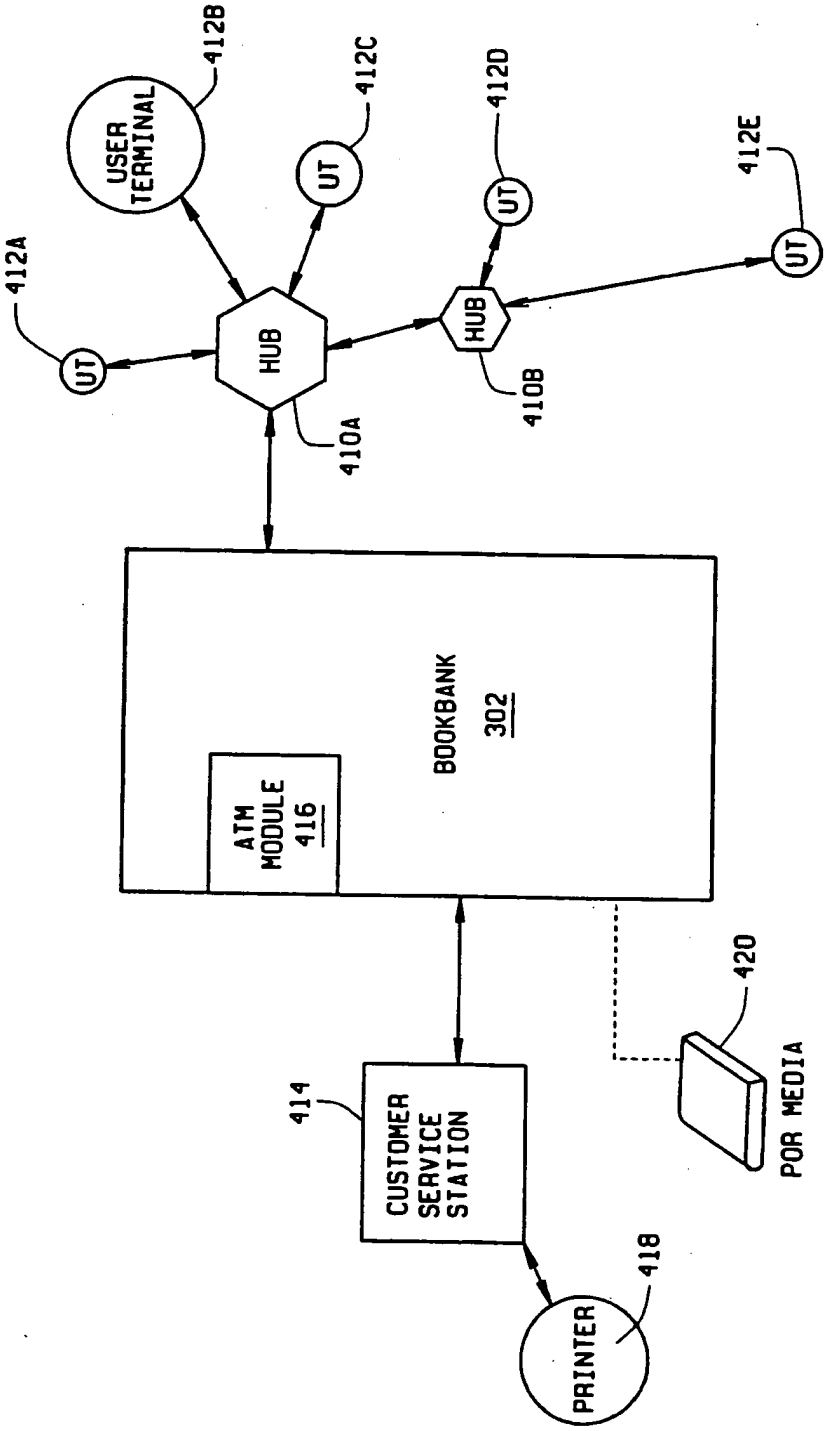

USER TERMINAL
412B
412C
412D
412E
412A
UT
HUB
410A
410B
BOOKBANK
302
ATM MODULE 416
CUSTOMER SERVICE STATION
414
PRINTER
418
420
POR MEDIA

FIG. 9

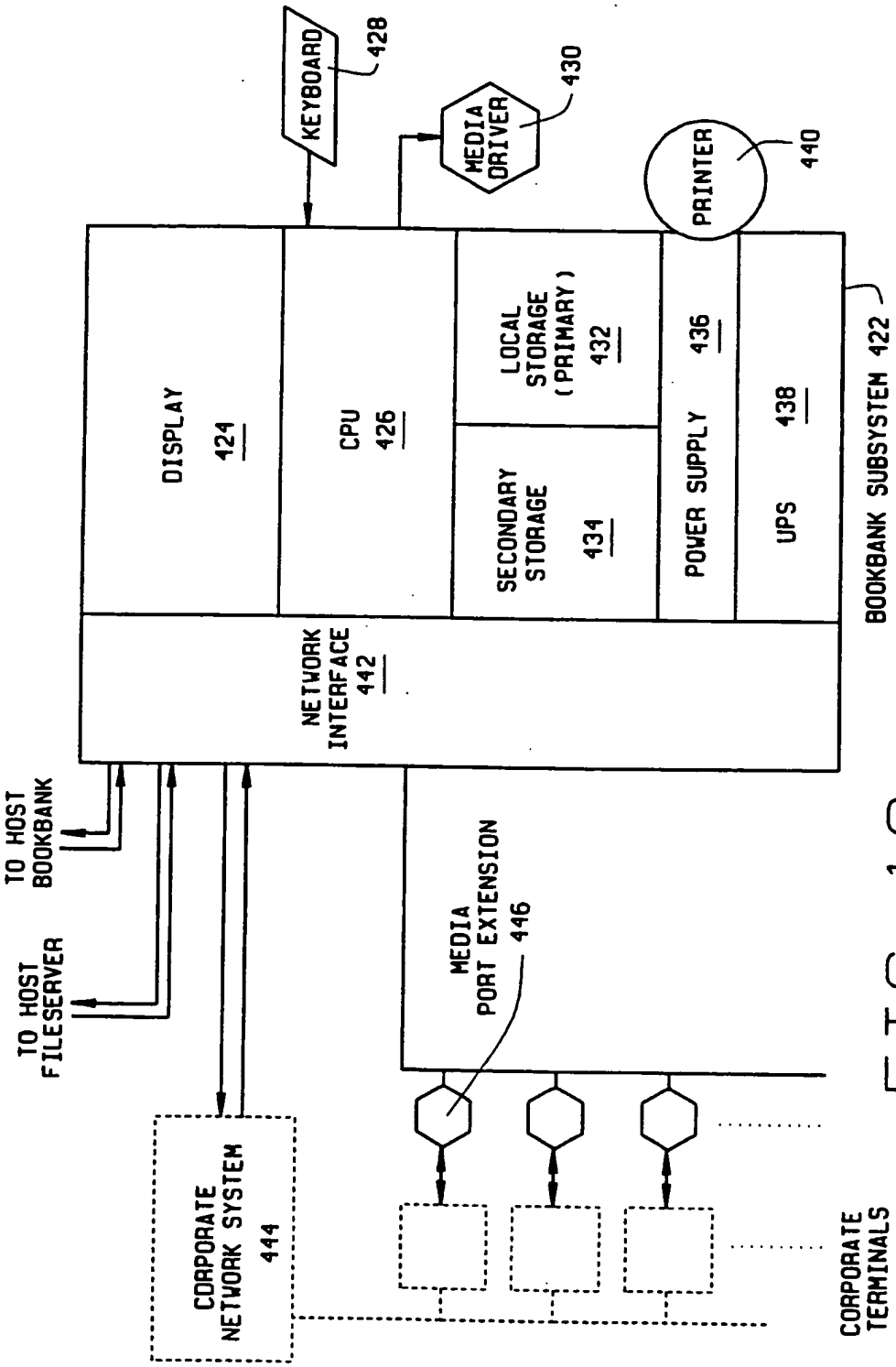

KEYBOARD
428
MEDIA DRIVER
430
PRINTER
440
DISPLAY
124
CPU
426
LOCAL STORAGE (PRIMARY)
432
SECONDARY STORAGE
434
POWER SUPPLY
436
UPS
438
BOOKBANK SUBSYSTEM 422
NETWORK INTERFACE
442
TO HOST BOOKBANK
TO HOST FILESERVER
MEDIA PORT EXTENSION
446
CORPORATE NETWORK SYSTEM
444
CORPORATE TERMINALS

FIG. 10

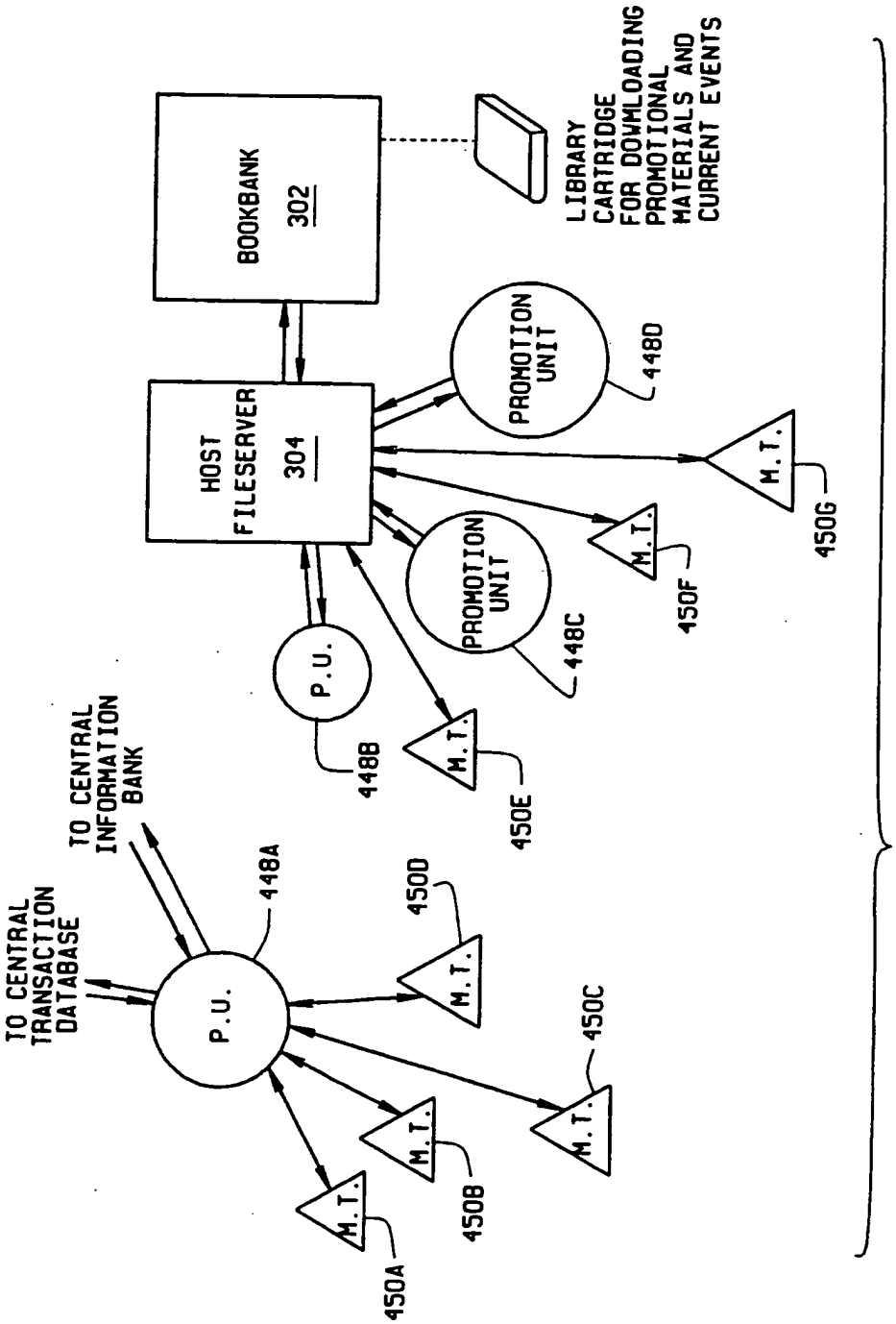

BOOKBANK
302
LIBRARY CARTRIDGE FOR DOWNLOADING PROMOTIONAL MATERIALS AND CURRENT EVENTS
HOST FILESERVER
304
PROMOTION UNIT
448D
PROMOTION UNIT
448C
P.U.
448B
P.U.
448A
TO CENTRAL INFORMATION BANK
TO CENTRAL TRANSACTION DATABASE
M.T.
450A
M.T.
450B
M.T.
450C
M.T.
450D
M.T.
450E
M.T.
450F
M.T.
450G

FIG. 11

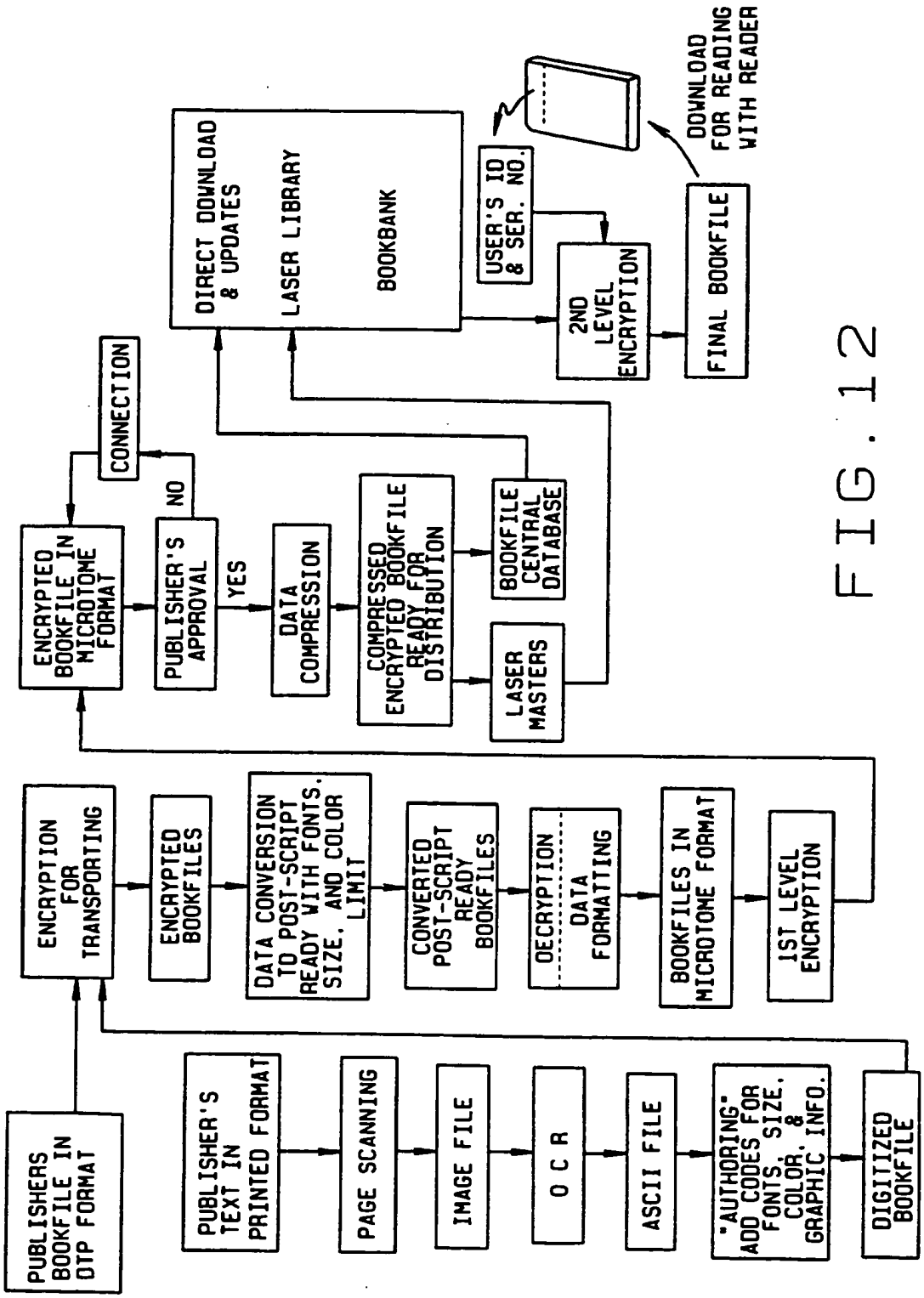

PUBLISHERS BOOKFILE IN DTP FORMAT
PUBLISHER'S TEXT IN PRINTED FORMAT
PAGE SCANNING
IMAGE FILE
O C R
ASCII FILE
"AUTHORING" ADD CODES FOR FONTS, SIZE, COLOR, & GRAPHIC INFO.
DIGITIZED BOOKFILE
ENCRYPTION FOR TRANSPORTING
ENCRYPTED BOOKFILES
DATA CONVERSION TO POST-SCRIPT READY WITH FONTS, SIZE, AND COLOR LIMIT
CONVERTED POST-SCRIPT READY BOOKFILES
DECRYPTION DATA FORMATTING
BOOKFILES IN MICROTOME FORMAT
1ST LEVEL ENCRYPTION
ENCRYPTED BOOKFILE IN MICROTOME FORMAT
PUBLISHER'S APPROVAL
NO
YES
CONNECTION
DATA COMPRESSION
COMPRESSED ENCRYPTED BOOKFILE READY FOR DISTRIBUTION
BOOKFILE CENTRAL DATABASE
LASER MASTERS
DIRECT DOWNLOAD & UPDATES
LASER LIBRARY
BOOKBANK
USER'S ID & SER. NO.
2ND LEVEL ENCRYPTION
FINAL BOOKFILE
DOWNLOAD FOR READING WITH READER

FIG. 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/22238

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :H 04 L 9/32
US CL :380/4,25,49

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. : 380/4,25,49

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

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

Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,708,709 A(ROSE) 13 January 1998, column 3, lines 11-47, column 5, lines 1-7, column 7, lines 16-31, column 8, lines 65-67, column 9, lines 1-8	1-3,5-6,8-14,16-36
Y	US 5,680,453 A (AKIYAMA ET AL.) 21 October 1997, abstract, column 3, lines 39- 48, column 5, lines 62-66, column 6,lines 58-67	1-3,5-6,8-14,16-36
A	US 5,532,920 A (HARTRICK ET AL.) 02 July 1996, column 4, lines 39-65, column 9,lines 6-15, column 16, lines 40-67	1-3,5-6,8-14,16-36
A	US 5,694,469 A (LE RUE) 02 December 1997, abstract, column 4, lines 20-33, column 3, lines 51-67, column 5, lines 48-53	1-3,5-6,8-14,16-36

☐ 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	Date of mailing of the international search report
05 MARCH 1999	14 APR 1999
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 GAIL HAYES James R. Matthews Telephone No. (703) 305-9711

Form PCT/ISA/210 (second sheet)(July 1992)★

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/22238

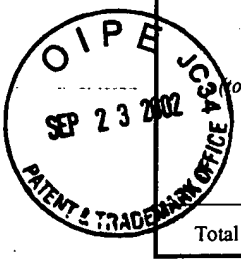
B. FIELDS SEARCHED
Electronic data bases consulted (Name of data base and where practicable terms used):

APS
search terms: download,transmit,request,send,receive,software,information,program,content,file,video,image
s,www,internet,world wide web,cipher,cypher,encrypt,encode,scramble,access code code,PIN,secret code,key

Form PCT/ISA/210 (extra sheet)(July 1992)*

Please type a plus sign (+) inside this box → [+]

PTO/SB/21 (08-00)
Approved for use through 10/31/2002. OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.



TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/162,701
Filing Date	June 6, 2002
First Named Inventor	Xin WANG et al.
Group Art Unit	
Examiner Name	
Total Number of Pages in This Submission	
Attorney Docket Number	111325-113

ENCLOSURES *(check all that apply)*

- [x] Fee Transmittal Form
  - [x] Fee Attached
- [ ] Amendment / Reply
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [x] Response to Missing Parts/ Incomplete Application
  - [x] Response to Missing Parts under 37 CFR 1.52 or 1.53
- [x] Assignment Papers *(for an Application)*
- [ ] Drawing(s)
- [x] Declaration and Power of Attorney
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
- [ ] After Allowance Communication to Group
- [ ] Other

Remarks

[x] The Commissioner is hereby authorized to charge any additional fees required or credit any overpayments to Deposit Account No. 19-2380 for the above identified docket number.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Nixon Peabody LLP 8180 Greensboro Drive Suite 800 McLean, VA 22102
Signature	
Date	September 23, 2002

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, Washington, DC 20231 on this date: _____

Type or printed name	
Signature	Date

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, Washington, DC 20231.

NVA240841.1

PTO/SB/17 (11-00)
Approved for use through 10/31/2002. OMB 0651-0032

FEE TRANSMITTAL FOR FY 2002

Patent fees are subject to annual revision.

TOTAL AMOUNT OF PAYMENT $982.00

Complete if Known	
Application Number	10/162,701
Filing Date	June 6, 2002
First Named Inventor	Xin WANG et al.
Examiner Name	
Group Art Unit	
Attorney Docket No.	111325-113

METHOD OF PAYMENT

1. ☒ The Commissioner is hereby authorized to charge indicated fees and credit any overpayments to:

Deposit Account Number: 19-2380 (111325-113)

Deposit Account Name: Nixon Peabody LLP

☒ Charge Any Additional Fee Required Under 37 CFR 1.16 and 1.17

☐ Applicant claims small entity status. See 37 CFR 1.27

2. ☒ **Payment Enclosed:**

☒ Check ☐ Credit Card ☐ Money Order ☐ Other

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
101	740	201	370	Utility filing fee	$740
106	330	206	165	Design filing fee	
107	510	207	255	Plant filing fee	
108	740	208	370	Reissue filing fee	
114	160	214	80	Provisional filing fee	

SUBTOTAL (1) $740.00

2. EXTRA CLAIM FEES

			Extra Claims		Fee from below		Fee Paid
Total Claims	24	-20** =	4	X	$18	=	$72.00
Independent Claims	3	-3** =	0	X	$84	=	
Multiple Dependent					$280	=	

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
103	18	203	9	Claims in excess of 20
102	84	202	42	Independent claims in excess of 3
104	280	204	140	Multiple dependent claim, if not paid
109	84	209	42	** Reissue independent claims over original patent
110	18	210	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) $72.00

**or number previously paid, if greater; For Reissues, see above

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
105	130	205	65	Surcharge – late filing fee or oath	$130
127	50	227	25	Surcharge – late provisional filing fee or cover sheet	
139	130	139	130	Non-English transaction	
147	2,520	147	2,520	For filing a request for *ex parte* reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	400	216	200	Extension for reply within second month	
117	920	217	460	Extension for reply within third month	
118	1,440	218	720	Extension for reply within fourth month	
128	1,960	228	980	Extension for reply within fifth month	
119	320	219	160	Notice of Appeal	
120	320	220	160	Filing a brief in support of an appeal	
121	280	221	140	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive – unavoidable	
141	1,280	241	640	Petition to revive – unintentional	
142	1,280	242	640	Utility issue fee (or reissue)	
143	460	243	230	Design issue fee	
144	620	244	310	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Processing fee under 37 CR 1.17(q)	
126	180	126	180	Submission of Information Disclosure Stmt	
581	40	581	40	Recording each patent assignment per property (times number of properties)	$40
146	740	246	370	Filing a submission after final rejection (37 CFR § 1.129(a))	
149	740	249	370	For each additional invention to be examined (37 CFR § 1.29(b))	
179	740	279	370	Request for Continued Examination (RCE)	
169	900	169	900	Request for expedited examination of a design application	

Other fee (specify) _____

* Reduced by Basic Filing Fee Paid

SUBTOTAL (3) $170.00

SUBMITTED BY

Complete *(if applicable)*

Name *(Print/Type)*	Daniel S. Song	Registration No. *(Attorney/Agent)*	43,143	Telephone	(703) 770-9300
Signature				Date	September 23, 2002

Please type a plus sign (+) inside this box → [+]

PTO/SB/21 (08-00)
Approved for use through 10/31/2002. OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/162,701
Filing Date	June 6, 2002
First Named Inventor	Xin WANG et al.
Group Art Unit	
Examiner Name	
Attorney Docket Number	111325-113

Total Number of Pages in This Submission

OIPE SEP 23 2002 JC34 PATENT & TRADEMARK OFFICE

ENCLOSURES *(check all that apply)*

- [x] Fee Transmittal Form
  - [x] Fee Attached
- [ ] Amendment / Reply
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [x] Response to Missing Parts/ Incomplete Application
  - [x] Response to Missing Parts under 37 CFR 1.52 or 1.53
- [x] Assignment Papers *(for an Application)*
- [ ] Drawing(s)
- [x] Declaration and Power of Attorney
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
- [ ] After Allowance Communication to Group
- [ ] Other

Remarks

[x] The Commissioner is hereby authorized to charge any additional fees required or credit any overpayments to Deposit Account No. 19-2380 for the above identified docket number.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Nixon Peabody LLP 8180 Greensboro Drive Suite 800 McLean, VA 22102
Signature	
Date	September 23, 2002

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, Washington, DC 20231 on this date:

Type or printed name	
Signature	Date

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, Washington, DC 20231.

NVA240841.1

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

APPLICATION NUMBER	FILING/RECEIPT DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NUMBER
10/162,701	06/06/2002	Xin Wang	111325-113

CONFIRMATION NO. 6475

22204
NIXON PEABODY, LLP
8180 GREENSBORO DRIVE
SUITE 800
MCLEAN, VA 22102

FORMALITIES LETTER

OC000000008490560

Date Mailed: 07/22/2002

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

Items Required To Avoid Abandonment:

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items and pay any fees required below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The statutory basic filing fee is missing.
  Applicant must submit $ **740** *to complete the basic filing fee for a non-small entity. If appropriate, applicant may make a written assertion of entitlement to small entity status and pay the small entity filing fee (37 CFR 1.27).*
- The oath or declaration is missing.
  A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.
- To avoid abandonment, a late filing fee or oath or declaration surcharge as set forth in 37 CFR 1.16(l) of $130 for a non-small entity, must be submitted with the missing items identified in this letter.

Items Required To Avoid Processing Delays:

The item(s) indicated below are also required and should be submitted with any reply to this notice to avoid further processing delays.

- Additional claim fees of **$72** as a non-small entity, including any required multiple dependent claim fee, are required. Applicant must submit the additional claim fees or cancel the additional claims for which fees are due.

09/24/2002 NMOHAMM1 00000110 10162701

01 FC:101 740.00 OP
02 FC:103 72.00 OP
03 FC:105 130.00 OP

SUMMARY OF FEES DUE:

Total additional fee(s) required for this application is **$942** for a Large Entity

- **$740** Statutory basic filing fee.
- **$130** Late oath or declaration Surcharge.
- Total additional claim fee(s) for this application is **$72**
  - **$72** for **4** total claims over 20.

A copy of this notice MUST be returned with the reply.

Customer Service Center
Initial Patent Examination Division (703) 308-1202
PART 2 - COPY TO BE RETURNED WITH RESPONSE

#20

Please type a plus sign (+) inside this box →

PTO/SB/01 (12-97)
Approved for use through 9/30/00. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

DECLARATION FOR UTILITY OR DESIGN PATENT APPLICATION (37 CFR 1.63)

[ ] Declaration Submitted With Initial Filing OR [X] Declaration Submitted after Initial Filing (surcharge (37 CFR 1.16(d)) required

Attorney Docket Number	111325-113
First Named Inventor	Xin Wang et al.
COMPLETE IF KNOWN	
Application Number	10/162,701
Filing Date	June 6, 2002
Group Art Unit	2122
Examiner Name	Not yet assigned

As a below named inventor, I hereby declare that:

My residence, post office address, and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

Method And Apparatus Managing The Transfer Of Rights

(Title of the Invention)

the specification of which

[ ] is attached hereto
OR
[X] was filed on (MM/DD/YYYY) 06/06/2002 As United States Application Number or PCT International Application Number 10/162,701
And was amended on (MM/DD/YYYY) _____ (If applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment specifically referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. 119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or of any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached YES	Certified Copy Attached No
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]

[ ] Additional foreign application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto:

I hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below.

Application Number(s)	Filing Date (MM/DD/YYYY)
60/331,624	11/20/2001
60/331,623	11/20/2001
60/331,621	11/20/2001

[X] Additional provisional application Numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.

Please type a plus sign (+) inside this box → [+]

PTO/SB/01 (12-97)
Approved for use through 9/30/00. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OBM control number.

DECLARATION – Utility or Design Patent Application

I hereby claim the benefit under 35 U.S.C. 120 of any United States application(s), or 365© of any PT international application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. 112, I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application.

U.S. Parent Application or PCT Parent Number	Parent Filing Date (MM/DD/YYYY)	Parent Patent Number *(if applicable)*

[ ] Additional U.S. or PCT international application are listed on a supplemental priority date sheet PTO/SB/02B attached hereto.

As a named inventor, I hereby appoint the following registered practitioner(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith: [X] Customer Number 22204

OR

[X] Registered practitioner(s) name/registration number listed below.

Name	Registration Number	Name	Registration Number
Stuart J. Friedman	24,312	Daniel S. Song	43,143
Charles M. Leedom, Jr.	26,477	Marc S. Kaufman	35,212
David S. Safran	27,997	Corinne R. Gorski	34,339
Thomas W. Cole	28,290	Jason H. Vick	45,285
Donald R. Studebaker	32,815	Luan C. Do	38,434
Jeffrey L. Costellia	35,483		
Tim L. Brackett. Jr.	36,092		

Direct all correspondence to: [X] Customer Number 22204

Name: Marc S. Kaufman

Firm: NIXON PEABODY LLP

Address: 8180 Greensboro Drive, Suite 800

City: McLean State: VA ZIP: 22102

Country: United States Telephone: (703) 770-9300 FAX: (703) 770-9400

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; ;and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Name of Sole or First Inventor: [ ] A petition has been filed for this unsigned inventor.

Given Name (first and middle [if any])	Family Name or Surname
XIN	**WANG**

Inventor's Signature: Date: 8/15/2002

Mailing Address (Street or P.O. Box): 3720 Emerald Street #V2

City: Torrance State: CA ZIP: 90503 Country: USA

Residence: City: State: Country:

Citizenship: USA

[X] Additional inventors are being named on the _____ Supplemental Additional Inventor(s) sheet(s) PTO/SB/02A attached hereto.

Docket No. 111325-113

Please type a plus sign (+) inside this box → [+]

PTO/SB/02A (10-00)
Approved for use through 10/31/2002. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OBM control number.

DECLARATION	ADDITIONAL INVENTOR(S) Supplemental Sheet Page ___ of ___

Name of Additional Joint Inventor, if any:	☐ A petition has been filed for this unsigned inventor		
Given Name: Thanh		Family Name Or Surname: TA	
Inventor's Signature:		Date: 08/15/2002	
Mailing Address (Street or P.O. Box): 18694 Stratton Lane			
City: Huntington Beach	State: CA	ZIP: 92648	Country: USA
Residence: City:	State:	Country:	
Citizenship: Australia			
Name of Additional Joint Inventor, if any:	☐ A petition has been filed for this unsigned inventor		
Given Name: Guillermo		Family Name Or Surname: LAO	
Inventor's Signature:		Date: 8/15/2002	
Mailing Address (Street or P.O. Box): 5531 Lorna Street			
City: Torrance	State: CA	ZIP: 90503	Country: USA
Residence: City:	State:	Country:	
Citizenship: USA			
Name of Additional Joint Inventor, if any:	☐ A petition has been filed for this unsigned inventor		
Given Name: Eddie J.		Family Name Or Surname: CHEN	
Inventor's Signature:		Date: 8/15/2002	
Mailing Address (Street or P.O. Box): 6796 Vallon Drive			
City: Ranchos Palos Verdes	State: CA	ZIP: 90275	Country: USA
Residence: City:	State:	Country:	
Citizenship: USA			

Burden Hour Statement: This form is estimated to take 21 minutes to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, Washington, DC 20231. [Page 3 of 4]

Please type a plus sign (+) inside this box → [+]

PTO/SB/02A (10-00)
Approved for use through 10/31/2002. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

DECLARATION – Supplemental Priority Data Sheet

Additional foreign applications:

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached? YES	NO
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐

Additional provisional applications:

Application Number	Filing Date (MM/DD/YYYY)
60/296,113	06/07/2001
60/296,117	06/07/2001
60/296,118	06/07/2001

Additional U.S. Applications:

U.S. Parent Application Number	PCT Parent Number	Parent Filing Date (MM/DD/YYYY)	Parent Patent Number *(if applicable)*

Burden Hour Statement: This form is estimated to take 21 minutes to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, Washington, DC 20231. [Page 4 of 4]

Application No. 10/162,701
Docket No. 111325/000113

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: Xin WANG	Examiner: Unassigned
Serial No. 10/162,701	Group Art Unit:
Filed: 06/06/2002	
For: METHOD AND APPARATUS MANAGING THE TRANSFER OF RIGHTS	

INFORMATION DISCLOSURE STATEMENT

Commissioner of Patents
Washington, D.C. 20231

Sir:

In accordance with the duty of disclosure as set forth in 37 C.F.R. § 1.56, Applicants hereby submit the following information in conformance with 37 C.F.R. §§ 1.97 and 1.98. Pursuant to 37 C.F.R. § 1.98, a copy of each of the documents cited is enclosed.

The documents are being submitted within three (3) months of the filing of this application or entry into the national stage of this application, or before the first Office Action on the merits, whichever is later, therefore no fee or certification is required under 37 C.F.R § 1.97(b).

It is requested that the accompanying information disclosure statement be considered and made of record in the above-captioned application. To assist the Examiner, the documents are listed on the attached form PTO-1449. It is respectfully requested that an Examiner initialed copy of this form be returned to the undersigned.

The Commissioner is hereby authorized to charge any fees connected with this filing which may be required now, or credit any overpayment to Deposit Account No. 19-2380.

Respectfully submitted,

By: Marc S. Kaufman
Registration No. 35,212

NIXON PEABODY LLP
8180 Greensboro Drive, Suite 800
McLean, Virginia 22102
Telephone: (703) 770-9300

NVA238413.1

Form PTO-1449 (Rev. 8-83)

U.S.Department of Commerce Patent and Trademark Office

INFORMATION DISCLOSURE STATEMENT

Atty Docket 111325-113	**Serial No.** 10/162,701
Applicants: Xin WANG	
Filing Date: June 06, 2002	**Group Art Unit:**

U.S. PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date (if appropriate)
	4,796,220	01/03/1989	Wolfe			

FOREIGN PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Country	Class	Subclass	Translation Yes	Translation No
	0 715 241	06/05/1996	JP			Full Eng	
	04-369068	12/21/1992	JP			Eng Abst	
	05-268415	10/15/1993	JP			Eng Abst	
	06-175794	06/24/1994	JP			Eng Abst	
	06-215010	08/05/1994	JP			Eng Abst	
	07-084852	03/31/1995	JP			Eng Abst	
	07-200317	08/04/1995	JP			Eng Abst	
	07-244639	09/19/1995	JP			Eng Abst	
	62-241061	10/21/1987	JP			Eng Abst	
	64-068835	03/14/1989	JP			Eng Abst	
	WO 94/01821	01/20/1994	PCT			Full Eng	
	WO 96/24092	08/08/1996	PCT			Full Eng	
	WO 97/48203	12/18/1997	PCT			Full Eng	
	WO 98/11690	03/19/1998	PCT			Full Eng	
	WO 98/42098	09/24/1998	PCT			Full Eng	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

Examiner Initial	
	Henry H. Perritt, Jr., "Technological Strategies for Protecting Intellectual Property in the Networked Multmedia Enviroment", April 2-3, 1993, Knowbots, Permissions Headers & Contract Law

Examiner	Date Considered

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this formwith next communication to applicant.

Requested Patent: JP4369068A

Title: USER RIGHT MANAGING SYSTEM FOR ON-LINE SYSTEM ;

Abstracted Patent: JP4369068 ;

Publication Date: 1992-12-21 ;

Inventor(s): NISHIWAKI SHINJI ;

Applicant(s): CHIYUUBU NIHON DENKI SOFUTOUEA KK ;

Application Number: JP19910145068 19910618 ;

Priority Number(s): ;

IPC Classification: G06F15/00 ;

Equivalents: ;

ABSTRACT:

PURPOSE:To easily execute the batch grasp of user right and to reduce the maintenance manhour with respect to user right managing information by providing a user right managing table to a host computer.

CONSTITUTION:This user right managing system is constituted of a terminal equipment 1 including a CRT device 11 and a using information input means 12, the host computer 2 and the user right managing table for collectively managing respective user using right. The host computer 2 is provided with a user right registering means 21 for registering a new user in the table 3 from the terminal equipment 1, a user right changing means 22 for changing user right and a user right judging means 23 for judging the validity of a using request inputted from the input means 12 while referring the table 3. The table 3 consists of a user-sorted management table 31 for registering the password and right class (sort code of using right) of each user and a function-sorted management table 32 for registering a usable function code in each right class.

(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（A）　(11)特許出願公開番号

特開平4－369068

(43)公開日　平成４年(1992)12月21日

(51)Int.Cl.[5]	識別記号	庁内整理番号	Ｆ１	技術表示箇所
Ｇ０６Ｆ　15/00	３３０　Ａ	7323－5L		

審査請求　未請求　請求項の数１(全　５　頁)

(21)出願番号　特願平3－145068

(22)出願日　平成３年(1991)６月18日

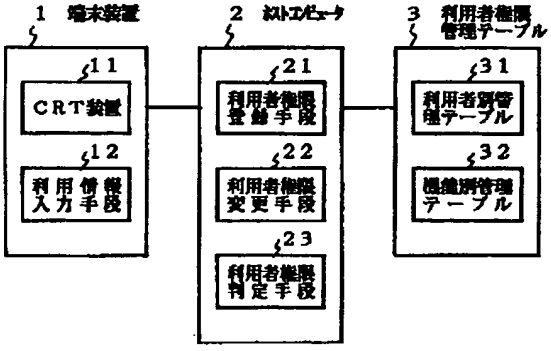
(71)出願人　000213301
中部日本電気ソフトウエア株式会社
愛知県名古屋市中区新栄２丁目28番22号
(72)発明者　西脇　信二
愛知県名古屋市中区新栄二丁目28番22号中部日本電気ソフトウエア株式会社内
(74)代理人　弁理士　内原　晋

(54)【発明の名称】　オンラインシステムの利用者権限管理方式

(57)【要約】

【構成】　ＣＲＴ装置１１，利用情報入力手段１２を含む端末装置１と、ホストコンピュータ２と、各利用者の利用権限を一括管理する利用者権限管理テーブル３で構成される。ホストコンピュータ２は、端末装置１から利用者権限管理テーブル３に新規利用者を登録する利用者権限登録手段２１と、利用者権限を変更する利用者権限変更手段２２と、利用者権限管理テーブル３を参照して利用情報入力手段１２からの利用要求の正当性を判定する利用者権限判定手段２３とを備えている。利用者権限管理テーブル３は、利用者別のパスワードと権限クラス（利用権限の種別コード）を登録した利用者別管理テーブル３１と、各権限クラスの利用可能な機能コードを登録した機能別権限管理テーブル３２とから成る。

【効果】　ホストコンピュータの利用者権限管理テーブルにより、利用者権限の一括把握が容易で、利用者権限管理情報のメンテナンス工数が低減される。



1

【特許請求の範囲】

【請求項1】　利用者が端末装置からホストコンピュータの各種オンラインサービス機能を利用する際の利用権限を管理するオンラインシステムの利用者権限管理方式において、利用者が入力した利用要求をホストコンピュータに伝達する利用情報入力手段を端末装置に備え、各利用者のオンラインサービス機能の利用権限の管理情報を記録した利用者権限管理テーブルと、端末装置から前記利用者権限管理テーブルに新規利用者を登録する利用者権限登録手段と、端末装置から前記利用者権限管理テーブルの管理情報を変更する利用者権限変更手段と、前記利用者権限管理テーブルを参照して前記利用情報入力手段からの利用要求の正当性を判定する利用者権限判定手段とをホストコンピュータに備えたことを特徴とするオンラインシステムの利用者権限管理方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明はオンラインシステムの利用者権限管理方式に関し、特に利用者が端末装置からホストコンピュータの各種オンラインサービス機能を利用する際の利用権限を管理するオンラインシステムの利用者権限管理方式に関する。

【0002】

【従来の技術】従来のオンラインシステムの利用者権限管理方式は、ホストコンピュータを利用して各種サービスを受ける端末装置内に、その端末装置を利用する利用者のパスワードやオンラインサービス機能別の利用権限に関する管理情報を持たせる方式であり、その管理情報の登録や変更を行う際には、各端末装置が配置されている支店や部署などの管理者の申請により、オンラインシステム管理部門が登録や変更を行った新たなフロッピーを作成送付し、端末装置側でこれを利用するなどの処置がとられていた。

【0003】

【発明が解決しようとする課題】上述した従来のオンラインシステムの利用者権限管理方式は、端末装置が設置されている職場ごとに利用権限を管理し、その情報を各端末装置が保持しているため、システム全体を一元的に把握し管理することが難しく、全体の状況が不明確になりやすい欠点がある。更に、多数の支店を有する企業等においては、利用者の新規登録や利用者権限の変更などの度にシステム管理部門で作業が発生し、かなりの工数を必要とするなどの問題がある。

【0004】本発明の目的は、上述の欠点に鑑み、オンラインシステムの利用者権限の管理を一元化し、利用者と利用権限の関係の全体像が把握しやすく、新規登録や変更による余分な工数が発生しないオンラインシステムの利用者権限管理方式を提供することにある。

【0005】

【課題を解決するための手段】本発明のオンラインシステムの利用者権限管理方式は、利用者が端末装置からホストコンピュータの各種オンラインサービス機能を利用する際の利用権限を管理するオンラインシステムの利用者権限管理方式において、利用者が入力した利用要求をホストコンピュータに伝達する利用情報入力手段を端末装置に備え、各利用者のオンラインサービス機能の利用権限の管理情報を記録した利用者権限管理テーブルと、端末装置から前記利用者権限管理テーブルに新規利用者を登録する利用者権限登録手段と、端末装置から前記利用者権限管理テーブルの管理情報を変更する利用者権限変更手段と、前記利用者権限管理テーブルを参照して前記利用情報入力手段からの利用要求の正当性を判定する利用者権限判定手段とをホストコンピュータに備えて構成されている。

【0006】

【実施例】次に、本発明の実施例について図面を参照して説明する。

【0007】図1は本発明のオンラインシステムの利用者権限管理方式の一実施例の機能構成を示すブロック図である。

【0008】本実施例のオンラインシステムの利用者権限管理方式は、ＣＲＴ装置１１と利用情報入力手段１２とを含む端末装置１と、ホストコンピュータ２と、ホストコンピュータ２に接続され各利用者の利用権限を一括管理する利用者権限管理テーブル３とから構成されている。ホストコンピュータ２には、端末装置１から利用者権限管理テーブル３に新規利用者を登録する利用者権限登録手段２１と、端末装置１から利用者権限管理テーブル３を更新する利用者権限変更手段２２と、利用者権限管理テーブル３を参照して利用情報入力手段１２からの利用要求の正当性を判定する利用者権限判定手段２３とを備えている。又、利用者権限管理テーブル３は、利用者別のパスワードと権限クラス（利用権限の種別を表すコード）とを登録した利用者別管理テーブル３１と、各権限クラスについて利用可能な機能コードを登録した機能別権限管理テーブル３２とで構成されている。

【0009】図2は利用情報入力手段１２による入力画面の説明図である。利用者はＣＲＴ装置１１の画面上に表示される図2の入力画面により、利用者を識別する利用者コードと、利用者コードと1対1に対応して定められているパスワードと、利用しようとするオンラインサービス機能を識別するために定められている機能コードと、処理に必要な固有処理項目とを入力する。

【0010】図3は利用者権限管理テーブル３の構成を示す説明図である。図3（a）に示すように、利用者別管理テーブル３１は、利用者コード，パスワード，権限クラスと、利用者権限管理テーブル３にエントリを登録できるか否かを識別する登録許可コードと、利用者権限管理テーブル３のエントリを変更できるか否かを識別する変更許可コードとから構成されている。一方、機能別

管理テーブル３２は、図３（ｂ）に示すように、オンラインサービスの機能コードと、これらの各機能を実行可能な権限クラスとの関係を示す機能許可コードテーブル部とから構成されている。

【００１１】次に、このように構成された本実施例の動作について、図２及び図３を参照して具体的なデータ例を用いて説明する（データは〔　〕で囲んで示す）。

【００１２】図２に示すように、入力画面上に利用者コード〔1〕，パスワード〔１１〕，機能コード〔A〕及び固有処理項目から成る利用要求が入力され、利用情報入力手段１２によりホストコンピュータ２に送信される。ホストコンピュータ２は、この利用要求を利用者権限判定手段２３に渡す。利用者権限判定手段２３は、利用者権限管理テーブル３を参照してこの利用要求が正当な利用者からのものであるか否かを判定する。

【００１３】その判定方法は、入力された利用者コード〔1〕で利用者別管理テーブル３１を検索し、テーブル上のパスワードが入力されたパスワード〔１１〕と一致するかを判定し、一致しなければ不当入力として処理を中断する。一致した場合は権限クラスを取得する。図３（ａ）の例では、入力されたパスワード〔１１〕が利用者別管理テーブル３１上のパスワード〔１１〕と一致するので、その権限クラス〔１０〕を取得する。

【００１４】次に、入力された機能コード〔A〕で機能別管理テーブル３２を検索し、機能許可コードテーブル部の権限クラス〔１０〕と等しいテーブル番号の許可コードが“0”ならば、利用者の権限は不当として処理を中断し、“1”ならば利用者の権限を正当と判定してオンライン処理を継続する。図３（ｂ）の例では、機能別管理テーブル３２のテーブル番号〔１０〕の許可コードは“1”であるから、利用要求は正当とみなされオンライン処理が継続される。

【００１５】次に、利用者権限管理テーブル３にエントリを登録する場合につき、その機能コードを〔B〕と仮定し説明する。利用者コード〔2〕，パスワード〔２２〕，機能コード〔B〕及び固有処理項目（テーブルへの登録内容）が入力され、ホストコンピュータ２へ送信される。ホストコンピュータ２は、この利用要求を利用者権限判定手段２３に渡す。利用者権限判定手段２３は、利用者権限管理テーブル３を参照して上述と同様な判定を行い、正当と判断した場合に利用者権限登録手段２１に利用要求を渡す。利用者権限登録手段２１は更に利用者別管理テーブル３１上の登録許可コードを判断し、登録許可コードが“0”ならば利用者に登録機能の利用権限が無いと判定し処理を中断する。登録許可コードが“1”ならば利用者は正当な権限を有すると判定し、利用者権限管理テーブル３へエントリを登録する。図３（ａ）の例では、利用者別管理テーブル３１上の利用者コード〔2〕の登録許可コードは“1”であるから、利用要求は正当とみなされ、固有処理項目の内容に

従って利用者権限管理テーブル３にエントリを登録する。

【００１６】次に、利用者権限管理テーブル３のエントリを変更する場合について、その機能コードを〔C〕と仮定して説明する。利用情報入力手段１２から利用者コード〔3〕，パスワード〔３３〕，機能コード〔C〕及び固有処理項目（エントリの変更内容）がホストコンピュータ２に送信されると、ホストコンピュータ２はこの利用要求を利用者権限判定手段２３へ渡す。利用者権限判定手段２１は、利用者権限管理テーブル３を参照して上述と同様な判定を行い、正当と判断した場合に利用者権限変更手段２２に利用要求を渡す。利用者権限変更手段２２は更に利用者別管理テーブル３１上の変更許可コードを判断し、変更許可コードが“0”ならば利用者に変更機能の利用権限は無いと判定し処理を中断し、変更許可コードが“1”ならば利用者は正当な権限を有すると判定し、利用者権限管理テーブル３のエントリを変更する。図３（ａ）の例では、利用者別管理テーブル３１上の利用者コード〔3〕の変更許可コードは“1”であるから、利用要求は正当とみなされ利用者権限管理テーブル３のエントリを変更する。

【００１７】上述したように、利用者権限管理テーブル３の登録および変更は、利用者権限判定手段２３による利用権限の判定に加え、それぞれ利用者権限登録手段２１及び利用者権限変更手段２２により利用権限の再チェックを経た後に実行される。なお、利用者別管理テーブル３１上の登録許可コード及び変更許可コードは、ホストコンピュータにおいてのみ変更可能であり、利用者権限登録手段２１及び利用者権限変更手段２２を用いて端末装置から変更することはできない。

【００１８】

【発明の効果】以上説明したように、本発明のオンラインシステムの利用者権限管理方式は、利用者権限の管理情報を利用者権限管理テーブルとしてホストコンピュータで一括して持つことにより、利用者とオンラインサービス機能との関連を一括して把握することが容易となり、加えて利用者権限管理情報のメンテナンス工数を省くことができる効果がある。

【図面の簡単な説明】

【図１】本発明の一実施例の構成を示すブロック図である。

【図２】本実施例の入力画面の説明図である。

【図３】本実施例の利用者権限管理テーブルの構成を示す説明図である。

【符号の説明】

１　端末装置
２　ホストコンピュータ
３　利用者権限管理テーブル
１１　ＣＲＴ装置
１２　利用情報入力手段

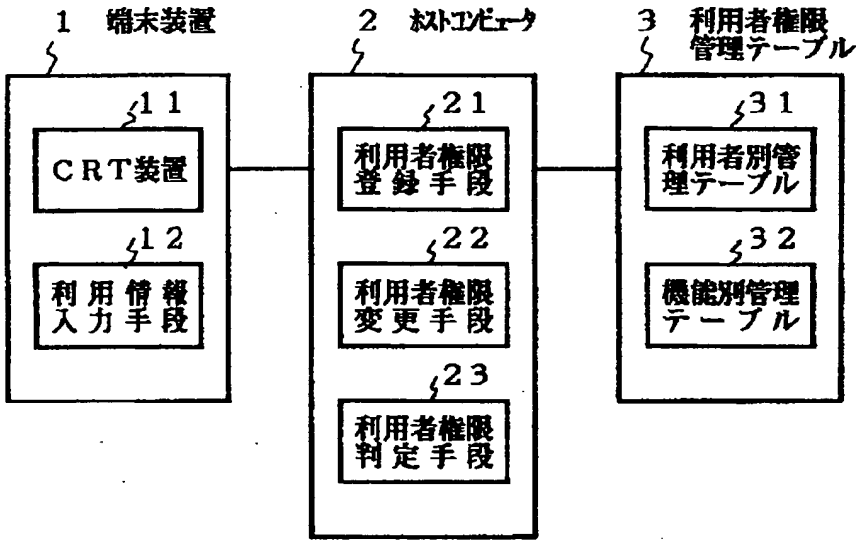
5

21　利用者権限登録手段
22　利用者権限変更手段
23　利用者権限判定手段

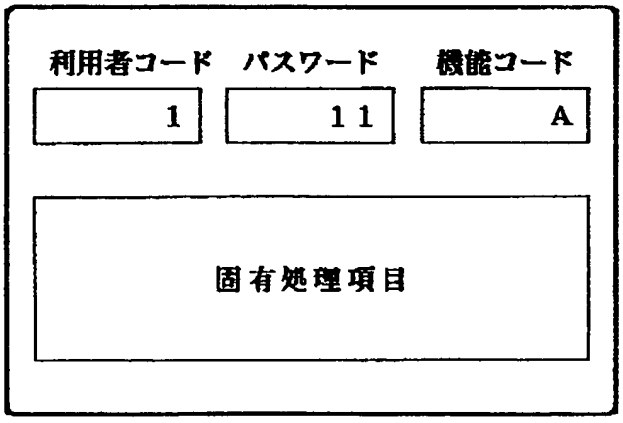
6

31　利用者別管理テーブル
32　機能別管理テーブル

【図1】



【図2】



【図3】

31　利用者別管理テーブル

利用者コード	パスワード	登録許可コード	変更許可コード	権限クラス
1	11	0	0	10
2	22	1	0	20
3	33	0	1	24

(a)

32　機能別管理テーブル

機能コード	機能許可コードテーブル部																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
A	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

(b)

Requested Patent: JP5268415A

Title: PICTURE READER ;

Abstracted Patent: JP5268415 ;

Publication Date: 1993-10-15 ;

Inventor(s): TAGUCHI KAZUE; others: 04 ;

Applicant(s): RICOH CO LTD ;

Application Number: JP19920006527 19920117 ;

Priority Number(s): ;

IPC Classification:

H04N1/04; B41J29/40; B42D9/04; G03B27/50; G03B27/62; G03G15/00; G03G15/04; G06F15/64; H04N1/00; H04N1/10 ;

Equivalents: ;

ABSTRACT:

PURPOSE:To cope with the problem of copy right by detecting an identification code for copy right management with a detecting means and automatically blocking the read of an original added with a code.

CONSTITUTION:A page turnover read unit 1 is provided so as to applies exposure scanning to a book original 92 placed opening on an original platen 18 to read a book original picture. Moreover, a bar code scanner 341a reading a bar code pattern 341c of a back cover of the book original 92 is provided to read an identification code for copy right management and the code is inputted to a one-chip microcomputer 330. The one-chip microcomputer 330 discriminates whether or not the copy of the bar code pattern is permitted and the copying is attained only when the copy of the pattern is permitted.

(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（A）　(11)特許出願公開番号

特開平5－268415

(43)公開日　平成５年(1993)10月15日

(51)Int.Cl.[5]	識別記号	庁内整理番号	Ｆ１	技術表示箇所
Ｈ０４Ｎ　1/04		Ｚ　7251－5C		
Ｂ４１Ｊ　29/40		Ｚ　8804－2C		
Ｂ４２Ｄ　9/04		Ｃ　8604－2C		
		Ｚ　8604－2C		
Ｇ０３Ｂ　27/50		Ａ　9017－2K		

審査請求　未請求　請求項の数８(全 62 頁)　最終頁に続く

(21)出願番号　特願平4－6527

(22)出願日　平成４年(1992)１月17日

(71)出願人　000006747
株式会社リコー
東京都大田区中馬込１丁目３番６号

(72)発明者　田口　和重
東京都大田区中馬込１丁目３番６号・株式会社リコー内

(72)発明者　藤岡　哲弥
東京都大田区中馬込１丁目３番６号・株式会社リコー内

(72)発明者　高橋　浩
東京都大田区中馬込１丁目３番６号・株式会社リコー内

(74)代理人　弁理士　樺山　亨　(外１名)

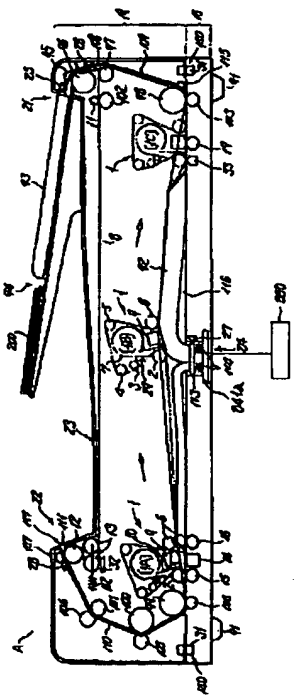
最終頁に続く

(54)【発明の名称】　画像読み取り装置

(57)【要約】

【目的】コピーされるブック原稿が著作権利用となるか否かを複写システム側でのブック原稿の認識およびカウント動作によって管理することのできる原稿読み取り装置を提供する。

【構成】原稿台（１８）上に見開かれて載置されたブック原稿（９２）を露光走査して該ブック原稿画像の読み取りを行なう読み取り手段（ページめくり読取ユニット１）と、原稿台に載置されたブック原稿に付された識別符号（バーコードパターン３４１Ｃ）を検知する検知手段（バーコードスキャナ３４１ａ）と、前記検知手段によって、前記識別符号が検知された場合にのみ前記読み取り手段による読み取り動作を可能とするように前記読み取り手段を制御する制御手段（ワンチップマイコン３３０）と、を有する。



1

【特許請求の範囲】

【請求項１】原稿台と、

前記原稿台上に見開かれて載置されたブック原稿の原稿面上を露光走査することによって前記ブック原稿画像の読み取りを行なう読み取り手段と、

原稿台に載置されたブック原稿の表紙又は背表紙の所定位置に付された識別符号を検知する検知手段と、

前記検知手段によって、前記識別符号が検知された場合にのみ前記読み取り手段による読み取り動作を可能とするように前記読み取り手段を制御する制御手段と、

を有することを特徴とする画像読み取り装置。

【請求項２】原稿台上に載置されたブック原稿のめくり・読み取りを行なうブック原稿読み取りモードと、

シート原稿トレイ上に載置されたシート原稿を１枚毎に原稿台上に給送して、前記シート原稿画像の読み取りを行なうＡＤＦモードと、が選択可能な画像読み取り装置であって、

原稿台に載置されたブック原稿の表紙又は背表紙の所定位置に付された識別符号を検知する検知手段と、

ブック原稿読み取りモードが選択されている場合には、前記検知手段によって前記識別符号が検知されたときにのみ前記読み取り手段による読み取り動作を可能とするように前記読み取り手段を制御する制御手段と、

を有することを特徴とする画像読み取り装置。

【請求項３】原稿台と、

前記原稿台上に載置されたブック原稿のめくり・読み取りを行なうブック原稿めくり・読み取りモードと、

前記原稿台上に載置された任意の原稿上の原稿画像の読み取りを行なう圧板モードと、が選択可能な画像読み取り装置であって、

原稿台に載置されたブック原稿の表紙又は背表紙の所定位置に付された識別符号を検知する検知手段と、

ブック原稿読み取りモードが選択されている場合には、前記検知手段によって前記識別符号が検知されたときにのみ前記読み取り手段による読み取り動作を可能とするように前記読み取り手段を制御する制御手段と、

を有することを特徴とする画像読み取り装置。

【請求項４】原稿台と、

前記原稿台上に見開かれて載置されたブック原稿の原稿面を露光走査することによって前記ブック原稿画像の読み取りを行なう読み取り手段と、

装置外部から読み取り手段による読み取り動作を可能にするための識別信号を入力する入力手段と、

前記識別信号の入力を検知する検知手段と、

前記検知手段によって前記識別信号の入力が検知された場合にのみ前記読み取り手段による読み取り動作を可能とするように前記読み取り手段を制御する制御手段と、

を有することを特徴とする画像読み取り装置。

【請求項５】原稿台上に載置されたブック原稿のめくり・読み取りを行なうブック原稿読み取りモードと、

2

シート原稿トレイ上に載置されたシート原稿を１枚毎に原稿台上に給送して、前記原稿画像の読み取りを行なうＡＤＦモードと、が選択可能な画像読み取り装置であって、

装置外部から読み取り手段による読み取り動作を可能とするための識別信号を入力する入力手段と、

前記識別信号の入力を検知する検知手段と、

ブック原稿読み取りモードが選択されている場合には前記検知手段によって前記識別信号の入力が検知されたときにのみ前記読み取り手段による読み取り動作を可能とするように前記読み取り手段を制御する制御手段と、

を有することを特徴とする画像読み取り装置。

【請求項６】原稿台と、

前記原稿台上に載置されたブック原稿のめくり・読み取りを行なうブック原稿読み取りモードと、

前記原稿台上に載置された任意の原稿上の原稿画像の読み取りを行なう圧板モードと、が選択可能であって、

装置外部から読み取り手段による読み取り動作を可能とするための識別信号を入力する入力手段と、

前記識別信号の入力を検知する検知手段と、

ブック原稿読み取りモードが選択されている場合には、前記検知手段によって前記識別信号の入力が検知されたとにきのみ前記読み取り手段による読み取り動作を可能とするように前記読み取り手段を制御する制御手段と、

を有することを特徴とする画像読み取り装置。

【請求項７】原稿台と、

前記原稿台上に載置された原稿の原稿面上を露光走査することによって原稿画像の読み取りを行なう読み取り手段と、

前記読み取り手段によって読み取られた画像情報を出力する出力手段と、を有する画像読み取り装置であって、

前記読み取り手段による原稿面上の露光走査によって原稿面上の所定位置に付された識別符号の検出動作を行なうと共に、前記検出動作によって前記識別符号が検知された場合にのみ前記出力手段による出力動作を可能とするように前記出力手段を制御する、

ことを特徴とする画像読み取り装置。

【請求項８】請求項７において、読み取り手段によって読み取られた画像情報を出力する際、所定位置に識別符号が付されたブック原稿から読み取った情報である場合は、その旨を識別するための付加情報を出力手段による出力情報に付加する、

ことを特徴とする画像読み取り装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、複写機及びファクシミリ等のブック原稿の画像読み取り装置に関する。

【０００２】

【従来の技術】ブック原稿の読み取り装置および方法としては、（１）原稿支持台上に上向きに見開いて載置し

たブック原稿の原稿面を情報のコンタクトガラスに当接させ、このコンタクトガラスを挟んで対向配置された走査光学系の走査によって原稿画像を露光走査する装置、（2）原稿載置面上に上向きに見開いて載置されたブック原稿の原稿面上を密着型読み取りセンサにより露光走査して原稿画像を読み取る装置、等が知られている。

【0003】一方、複写機及びファクシミリ等における原稿読み取り装置として、シート原稿を原稿読み取り位置に自動的に搬送して原稿情報を読み取り、読み取りの終了した原稿を上記原稿読み取り位置から自動的に排出する自動原稿給送装置（ADF）が知られている。

【0004】このように、原稿がシート原稿の場合には、ADFを搭載することにより、その原稿情報の読み取りを自動的に行うことができるが、原稿がブック原稿の場合には、自動ページめくり機構の実現が事実状困難なため、現段階では、ブック原稿のページめくりを手動的にしか行うことができない状況にある。

【0005】従来、こうした現状に鑑み、手間の掛るブック原稿の読み取りを自動化するための方法や手段の提案が種々なされている。

【0006】このブック原稿の自動読み取りを実現させるために不可欠となるブック原稿の自動ページめくり装置および方法としては、（3）下向きに見開いて載置したブック原稿を移動させながら、その原稿ページを吸引して分離する装置、（4）上向きに見開いて載置したブック原稿の原稿ページを吸引して分離する装置、（5）上向きに見開いて載置したブック原稿の原稿ページをローラ，アーム等でめくる装置、等が知られている。

【0007】しかしながら、上記の従来技術は、アイデアのみの提案が多く、およそ実現し得るレベルには到達しておらず、以下のような問題点がある。

【0008】すなわち、（1）原稿支持台上に上向きに見開いて載置したブック原稿の原稿面を情報のコンタクトガラスに当接させ、このコンタクトガラスを挟んで対向配置された走査光学系の走査によって原稿画像を露光走査する装置では、ブック原稿の原稿ページをめくる位置から露光走査を行なう位置までブック原稿を移動させる必要があり、原稿露光走査効率が低下するとともに、装置の大型化が避けがたくなる不具合があった。

【0009】また、（2）原稿載置面上に上向きに見開いて載置されたブック原稿の原稿面上を密着型読み取りセンサにより露光走査して原稿画像を読み取る装置では、装置の大型化という点は解消し得るものの、ブック原稿の原稿面を押圧する手段を備えていないため、載置された原稿面が浮き上がり易く、原稿面の読み取り走査が不安定になる不具合を有していた。

【0010】一方、（3）下向きに見開いて載置したブック原稿を移動させながら、その原稿ページを吸引して分離する装置では、ブック原稿の自重の影響によりページめくりの信頼性が損なわれるばかりでなく、そのページめくり時に原稿面が擦れて原稿が破損する虞れがあり、さらに、その構成上装置の大型化を余儀なくされる不具合があった。

【0011】また、（4）上向きに見開いて載置したブック原稿の原稿ページを吸引して分離する装置、並びに、（5）上向きに見開いて載置したブック原稿の原稿ページをローラ，アーム等でめくる装置等では、ブック原稿の上側空間を移動する従来のページ分離機構が複雑且つ大型なため、装置の大型化が避けがたくなる不具合があった。

【0012】こうした現状に鑑み、本出願人は、例えば、特願平2－193589号明細書等に開示したように、原稿台の原稿載置面に沿って張架されたページ押えベルトの一部に上記原稿載置面から離間する迂回部を形成させながら、上記原稿載置面とページ押えベルトとの間に見開かれて載置されたブック原稿の原稿面に対して、ページ収納手段，ページ吸着手段，ページ分離手段および読み取り手段等が配設されたページめくり読み取りユニットを相対移動させることによって、上記ブック原稿のページめくりおよび原稿読み取り走査を行なうブック原稿のページめくり読み取り装置（以下、MFDSという）を提案した。

【0013】この提案によるMFDSによれば、上記明細書等に記述したように、複写作業等に多大な労力を要していたブック原稿のページめくり操作および原稿読み取り走査を完全に自動化させることができ、複写等の生産性を著しく向上させる多機能原稿読み取りシステムを実現することができる。

【0014】

【発明が解決しようとする課題】前記MFDSは、ブック原稿の原稿情報の読み取り走査および原稿ページめくり動作を自動で行なう機能を持っているので、このMFDSをプリンタと接続することによって、書籍や雑誌等のコピーを人手を掛けずに高速で行なうことができる。

【0015】しかしながら、周知のように、一般の書籍などを著作者の許可を得ないで複写することは、純粋な私的利用を除いて原則として著作権法違反となる。

【0016】そこで、本発明は、前記MFDSを含むブック原稿画像の読み取り手段を有する画像読み取り装置において、許可された特定のブック原稿から、或いは、特定ユーザーによる画像読み取り（複写）のみを可能とすることにより、著作権問題への対処を図り得る画像読み取り装置を提供することを目的とする。

【0017】

【課題を解決するための手段】本発明は、上述の課題を解決するために、次の（1）乃至（8）の構成とした。

【0018】（1）．原稿台と、前記原稿台上に見開かれて載置されたブック原稿の原稿面上を露光走査することによって前記ブック原稿画像の読み取りを行なう読み取り手段と、原稿台に載置されたブック原稿の表紙又は

5

背表紙の所定位置に付された識別符号を検知する検知手段と、前記検知手段によって、前記識別符号が検知された場合にのみ前記読み取り手段による読み取り動作を可能とするように前記読み取り手段を制御する制御手段と、を有することとした。

【0019】（2）．原稿台上に載置されたブック原稿のめくり・読み取りを行なうブック原稿めくり・読み取りモードと、シート原稿トレイ上に載置されたシート原稿を1枚毎に原稿台上に給送して、前記シート原稿画像の読み取りを行なうADFモードと、が選択可能な画像読み取り装置であって、原稿台に載置されたブック原稿の表紙又は背表紙の所定位置に付された識別符号を検知する検知手段と、ブック原稿めくり・読み取りモードが選択されている場合には、前記検知手段によって前記識別符号が検知されたときにのみ前記読み取り手段による読み取り動作を可能とするように前記読み取り手段を制御する制御手段と、を有することとした。

【0020】（3）．原稿台と、前記原稿台上に載置されたブック原稿のめくり・読み取りを行なうブック原稿めくり・読み取りモードと、前記原稿台上に載置された任意の原稿上の原稿画像の読み取りを行なう圧板モードと、が選択可能な画像読み取り装置であって、原稿台に載置されたブック原稿の表紙又は背表紙の所定位置に付された識別符号を検知する検知手段と、ブック原稿めくり・読み取りモードが選択されている場合には、前記検知手段によって前記識別符号が検知されたときにのみ前記読み取り手段による読み取り動作を可能とするように前記読み取り手段を制御する制御手段と、を有することとした。

【0021】（4）．原稿台と前記原稿台上に見開かれて載置されたブック原稿の原稿面を露光走査することによって前記ブック原稿画像の読み取りを行なう読み取り手段と、装置外部から読み取り手段による読み取り動作を可能にするための識別信号を入力する入力手段と、前記識別信号の入力を検知する検知手段と、前記検知手段によって前記識別信号の入力が検知された場合にのみ前記読み取り手段による読み取り動作を可能とするように前記読み取り手段を制御する制御手段と、を有することとした。

【0022】（5）．原稿台上に載置されたブック原稿のめくり・読み取りを行なうブック原稿めくり・読み取りモードと、シート原稿トレイ上に載置されたシート原稿を1枚毎に原稿台上に給送して、前記原稿画像の読み取りを行なうADFモードと、が選択可能な画像読み取り装置であって、装置外部から読み取り手段による読み取り動作を可能とするための識別信号を入力する入力手段と、前記識別信号の入力を検知する検知手段と、ブック原稿めくり・読み取りモードが選択されている場合には前記検知手段によって前記識別信号の入力が検知されたときにのみ前記読み取り手段による読み取り動作を可

6

能とするように前記読み取り手段を制御する制御手段と、を有することとした。

【0023】（6）．原稿台と、前記原稿台上に載置されたブック原稿のめくり・読み取りを行なうブック原稿めくり・読み取りモードと、前記原稿台上に載置された任意の原稿上の原稿画像の読み取りを行なう圧板モードと、が選択可能であって、装置外部から読み取り手段による読み取り動作を可能とするための識別信号を入力する入力手段と、前記識別信号の入力を検知する検知手段と、ブック原稿めくり・読み取りモードが選択されている場合には、前記検知手段によって前記識別信号の入力が検知されたときにのみ前記読み取り手段による読み取り動作を可能とするように前記読み取り手段を制御する制御手段と、を有することとした。

【0024】（7）．原稿台と、前記原稿台上に載置された原稿の原稿面上を露光走査することによって原稿画像の読み取りを行なう読み取り手段と、前記読み取り手段によって読み取られた画像情報を出力する出力手段と、を有する画像読み取り装置であって、前記読み取り手段による原稿面上の露光走査によって原稿面上の所定位置に付された識別符号の検出動作を行なうと共に、前記検出動作によって前記識別符号が検知された場合にのみ前記出力手段による出力動作を可能とするように前記出力手段を制御することとした。

【0025】（8）．（7）において、読み取り手段によって読み取られた画像情報を出力する際、所定位置に識別符号が付されたブック原稿から読み取った情報である場合は、その旨を識別するための付加情報を出力手段による出力情報に付加することとした。

【0026】

【作用】著作権管理用としての識別符号の付された原稿の読み取りは自動的に阻止され

【0027】る。

【実施例】以下、本発明の実施例を図に基づいて詳細に説明する。本発明の実施例の説明に先立って、この発明が実施される前記のMFDS（ブック原稿のページめくり読み取り装置）について、その構成および動作を説明する。

【0028】先ず、図1を参照しながら、本発明による原稿読み取り装置におけるめくり搬送ベルトの周辺の構成について説明する。図1は、本発明を実施した、マルチ・ファンクション・ドキュメント・スキャナA（以下、"MFDS"とする）の概略断面図である。図1において、めくり搬送ベルト8は、駆動ローラ12、テンションローラ13、第1ベルト支持ローラ97、第2ベルト支持ローラ98、第3ベルト支持ローラ99、第4ベルト支持ローラ100、及び、第5ベルト支持ローラ101により支持されている。

【0029】このとき、めくり搬送ベルト8は、ページめくり読取ユニット1のめくりローラ2、第1バイアス

7

ローラ3、第1ローラ4、第2ローラ5、及び、押えローラ6を介して、このページめくり読取ユニット1を取り巻くように支持されている。

【0030】テンションローラ13は、ベルトテンションスプリング112によって、図1において左方向に引っ張られ、めくり搬送ベルト8に適度なテンションを与え、ブック原稿92の原稿面を押圧している。ベルトテンションセンサ32は、テンションローラ13の移動量からめくり搬送ベルト8のテンションを検出している。

【0031】一方、MFDSの搬送部19の上部には、シート原稿200をセットするためのシート原稿トレイ94と、シート原稿200のサイド方向のセット位置を調整するシート原稿サイドガイド93と、排紙されたシート原稿200が載置される排紙トレイ23とが、それぞれ配設されている。

【0032】また、MFDSの給紙部21（図1の右上方）には、シート原稿200のセットの有無を検知するシート原稿センサ25と、シート原稿200を一枚ずつ分離給紙する給紙ローラ96及び給紙分離パッド95と、シート原稿200の給紙タイミングをはかる給紙センサ26とが配設されており、さらに、シート原稿200の搬送路を構成する第1搬送ガイド108及び第2搬送ガイド109が配設されている。

【0033】さらに、ベルト支持ローラ97の僅かに左側のめくり搬送ベルト8の外側には、シート原稿200の搬送時の帯電用としての第2バイアスローラ11があり、その内側には、支持用のバイアス対向ローラ102がある。

【0034】また、第2搬送ガイド109の下端には、搬送ガイド爪115があり、シート原稿200の搬送を補助している。さらに、第4ベルト支持ローラ100には第6対向ローラ105、第5ベルト支持ローラ101には第7対向ローラ106、駆動ローラ12には排紙ローラ107が、めくり搬送ベルト8を挟んでめくり搬送ベルト8の外側にそれぞれ配設されている。

【0035】また、第3ベルト支持ローラ99から駆動ローラ12までのめくり搬送ベルト8の外側には、シート原稿200の搬送路を構成する第3搬送ガイド110が配設されている。ここで、第2ベルト支持ローラ98から第3ベルト支持ローラ99までの間は、原稿台18の上面の原稿載置面116がシート原稿200の搬送路として使用される。

【0036】この原稿載置面116は、プレスキャンしてブック原稿92のサイズを検出する際に、このブック原稿92の先端の検知をし易くするために、黒色に形成されている。一方、図1において左上方のMFDSの排紙部22には、排紙ローラ107の手前に排紙センサ28があって、シート原稿200の排紙時におけるジャムの有無を検知している。

【0037】また、この排紙部22の排紙口117の下

8

側には、排紙分離爪111が形成されており、シート原稿200のスムーズな排紙を補助している。MFDSの原稿台18には、第2ベルト支持ローラ98の下側に第4対向ローラ103が、第3ベルト支持ローラ99の下側に第5対向ローラ104がそれぞれ配設され、ページめくり読取ユニット1がホームポジション位置〔1A〕にあるときのめくりローラ2の下側に第1対向ローラ15が、押えローラ6の下側に第2対向ローラ16が、第1読み取りセンサユニット9の下側に第2読み取りセンサユニット14が、それぞれ配設されている。

【0038】また、ページめくり読取ユニット1がエンドポジション位置（シート原稿200の読み込みモード時におけるページめくり読取ユニット1の停止位置）〔1C〕にある時の、第1読み取りセンサユニット9の下側には第3対向ローラ17が、めくりローラ2の下側にはブックサイズ上限センサ33が、それぞれ配設されている。

【0039】さらに、原稿載置面116の中央には、ブック原稿92の載置位置を決めるための中央基準位置決め部24が形成されている。この中央基準位置決め部24には、ブック原稿92が載置された状態でブック原稿92の背に当接する中央位置決め板113と、この中央位置決め板113に対して上昇する習性を与えるための中央位置決めスプリング114と、ブック原稿92の載置状態における中央位置決め板113の下方への変位量を検出するブック原稿センサ27とが、それぞれ配設されている。ここで、中央位置決め板113の一部は切り欠かれていて、この切欠部の下方には、図67にも示すように識別符号検知手段としてのバーコードスキャナ341aが配置され制御手段としてのワンチップマイコン330に接続されている。

【0040】このように構成された原稿台18の下部には、MFDSを略水平に支持するためのスタンド91が設けられている。一方、搬送部19の原稿台18と接する部位の両脇には、搬送部ロックセンサ31を内蔵した搬送部ロック装置140が配設されており、搬送部19と原稿台18との開閉状態を検知している。

【0041】以上述べたように、このMFDSは、シート原稿200の自動原稿給送・読取機能と、ブック原稿の読み取り及び自動ページめくり機能との両機能を合わせ持った原稿読み取り装置として構成される。次に、図2乃至図4を参照しながら、この原稿読み取り装置における駆動系の構成について説明する。

【0042】図2は、MFDSの駆動系の概略横断面図である。図3は、MFDSの駆動系の概略平面図である。図4は、ページめくり読取ユニット1の端部の斜視図である。図2乃至図4において、ページめくり読取ユニット1は、めくりユニット駆動板49のスキャニングパイプ51が、MFDSの手前側と奥側とに平行に横架された一対のスキャニングロッド50に嵌挿されること

9

によって、ＭＦＤＳの左右方向に沿って摺動自在に支持されている。

【００４３】各スキャニングロッド５０は、ＭＦＤＳの搬送部右側板５８と、搬送部左側板５９とに、それぞれの端部が固定されている。また、図３に示すように、搬送部前側板５６と搬送部後側板５７との間には、第１シャフト６４と第２シャフト６５とが、それぞれ回転自在に軸支されている。これらの第１シャフト６４と第２シャフト６５には、駆動プーリ６２と従動プーリ６３とが、その手前側と奥側とに、それぞれ１つずつ固定されていて、これらの駆動プーリ６２と従動プーリ６３を一対として、２本のめくりユニット駆動ベルト５２が、それぞれ懸架がされている。

【００４４】このページめくり読取ユニット１は、図３及び図４に示すように、めくりユニット駆動板４９の駆動ベルト固定部５３で、駆動ベルト固定板５４と駆動ベルト固定ビス５５によって、めくりユニット駆動ベルト５２と固定されており、このめくりユニット駆動ベルト５２の回動により駆動される。

【００４５】また、第１シャフト６４には、搬送部後側板５７を挟んで、その奥側に第１ギア６６が固定されており、この第１ギア６６がめくりユニット駆動モータ６０の出力軸に固定された第２ギア６７と噛み合って、めくりユニット駆動モータ６０の回転が伝達され、駆動プーリ６２、めくりユニット駆動ベルト５２、従動プーリ６３、及び、ページめくり読取ユニット１が、それぞれ駆動される。

【００４６】一方、図３に示すように、第４ベルト支持ローラ１００と同軸固定された第３シャフト１２１、駆動ローラ１２と同軸固定された第４シャフト１２２、給紙ローラ９６と同軸固定された第７シャフト１２５、第１ベルト支持ローラ９７と同軸固定された第８シャフト１２６は、搬送部前側板５６と搬送部後側板５７との間にそれぞれ回転自在に軸支されている。第４シャフト１２２には、搬送部後側板５７を挟んで、その奥側に第３ギア６８が固定されている。また、第４ギア６９と第５ギア７３とは、同軸固定され回転自在に軸支されている。

【００４７】さらに、めくり搬送ベルト駆動モータ６１の出力軸に固定された第６ギア７４は第５ギア７３と、また、第４ギア６９は第３ギア６８と、それぞれ噛み合っていて、めくり搬送ベルト駆動モータ６１の回転を順次伝達し、駆動ローラ１２を駆動してめくり搬送ベルト８を回動させる。

【００４８】また、図２及び図３に示すように、第８シャフト１２６には、搬送部後側板５７の奥側に第２給紙プーリ１３０が、第７シャフト１２５には、搬送部後側板５７の奥側に給紙クラッチ１２８を介して第１給紙プーリ１２９が、それぞれ固定されており、第２給紙プーリ１３０と第１給紙プーリ１２９とには、給紙駆動ベルト１２７が懸架されている。

10

【００４９】これにより、めくり搬送ベルト８の回動によって回転された第１ベルト支持ローラ９７の回転は、第８シャフト１２６、第２給紙プーリ１３０、給紙駆動ベルト１２７、第１給紙プーリ１２９に順次伝達され、給紙クラッチ１２８の入力側に伝達される。

【００５０】また、メイン制御ボード３１０（図３４参照）から送られる制御信号によって、給紙クラッチ１２８が作動し、第７シャフト１２５及び給紙ローラ９６が駆動される。一方、図１に示した、第１対向ローラ１５、第２対向ローラ１６、及び、第３対向ローラ１７は、図示しない駆動伝達機構によって、めくり搬送ベルト８と同期し、且つ、同じ周速でそれぞれ駆動される。

【００５１】次に、図１乃至図４を参照しながら、ページめくり読取ユニット１の構成について説明する。図１乃至図４において、第１めくりユニット側板４０と、第２めくりユニット側板４１は、めくり搬送ベルト８を挟む位置に互いに対向して配置されており、これらの第１めくりユニット側板４０と、第２めくりユニット側板４１との間には、めくりローラ２、第１バイアスローラ３、第１ローラ４、第２ローラ５、及び、押えローラ６がそれぞれ回転自在に支持されている。これらの各ローラの支持構造は、何れも同じに構成されている。

【００５２】そこで、第１ローラ４を例に取って説明すると、この第１ローラ４は、図５に示すように、第１めくりユニット側板４０と第２めくりユニット側板４１に両端部が固定された第５シャフト１２３が、中空軸からなるこの第１ローラ４の中空部に挿通され、この第５シャフト１２３に配設された一対の軸受１３１によって、この第１ローラ４の両端部が支持されることにより、第５シャフト１２３に対して回転自在に支持されている。

【００５３】ここで、これらのめくりローラ２、第１バイアスローラ３、第１ローラ４、第２ローラ５、及び、押えローラ６は、自らは回転せず、めくり搬送ベルト８の回動によってのみ回転される。また、第１めくりユニット側板４０と第２めくりユニット側板４１の外側には、図１に示す第１読み取りセンサユニット９の長手方向の延長位置に、回動支持ロッド４２（両側とも同じに構成されている）がそれぞれ回転自在に支持されている。

【００５４】そこで、この回動支持ロッド４２の片側のみの支持構造を説明すると、この回動支持ロッド４２は、図４に示すように、摺動パイプ４３、第２スプリング止め爪４８、上限検知部７６と一体に構成されている。図４において、傾き修正スプリング４４は、ねじりコイルスプリングで形成されており、その一端が回動支持ロッド４２に、他端が第１めくりユニット側板４０に、それぞれ固定されている。

【００５５】この傾き修正スプリング４４は、その自然状態、すなわち、傾き修正スプリング４４に外力が加わ

11

っていない状態において、摺動パイプ４３の軸心方向（図４の上下方向）と、第１読み取りセンサユニット９の読取光学系の光軸方向（第１読み取りセンサユニット９の第１めくりユニット側板４０及び第２めくりユニット側板４１に対する移動方向（詳しくは後述））とを一致させるように構成されている。

【００５６】これにより、第１めくりユニット側板４０及び第２めくりユニット側板４１が、各回動支持ロッド４２を中心としてそれぞれ一体的に回転した際、この傾き修正スプリング４４によって、これらの第１めくりユニット側板４０及び第２めくりユニット側板４１に対して常に初期状態の姿勢に戻そうとする回転力が作用し、これらの第１めくりユニット側板４０及び第２めくりユニット側板４１の傾きが適時修正される。

【００５７】また、摺動パイプ４３は、図４に示すように、摺動支持ロッド４６に対して滑らかに摺動するように構成されている。この摺動支持ロッド４６の上端は上部ロッド支持板７０に、また、この摺動支持ロッド４６の下端は下部ロッド支持板７１に、それぞれ固定されている。

【００５８】また、上部ロッド支持板７０には第１スプリング止め爪４７が、摺動パイプ４３の上部には第２スプリング止め爪４８が、それぞれ形成されており、これらの第１スプリング止め爪４７及び第２スプリング止め爪４８によって、摺動パイプ４３と上部ロッド支持板７０との間に軸装された、めくりユニット上下スプリング４５の両端がそれぞれ係止されている。

【００５９】ここで、摺動パイプ４３は、通常の状態では、下部ロッド支持板７１に当接しているが、ページめくり読取ユニット１が外力を受けることによって、めくりユニット上下スプリング４５の弾力に抗し、摺動支持ロッド４６に沿って、図４において上方に摺動する。このとき、摺動パイプ４３には、第２スプリング止め爪４８を常に下方に押圧する、めくりユニット上下スプリング４５の弾力によって、上述の通常の状態に戻ろうとする摺動力が作用している。

【００６０】また、この摺動パイプ４３の上方への摺動範囲は、めくりユニット駆動板４９に配置されたスキャンカットオフセンサ３４が、摺動パイプ４３に設けられた上限検知部７６を検知した状態で、摺動パイプ４３の上方への摺動位置が限界位置となるように設定されている。

【００６１】一方、めくりユニット駆動板４９は、図４に示すように、上部ロッド支持板７０、下部ロッド支持板７１、第１スプリング止め爪４７、スキャニングパイプ５１、駆動ベルト固定部５３、及び、ホーム検知フィラ７５と一体的に形成されている。

【００６２】また、このめくりユニット駆動板４９の駆動ベルト固定部５３は、前述したように、駆動ベルト固定板５４と駆動ベルト固定ビス５５によって、めくりユニット駆動ベルト５２に固定されている。さらに、このめくりユニット駆動板４９のスキャニングパイプ５１は、スキャニングロッド５０に対して滑らかに摺動するように嵌合されている。

12

【００６３】これにより、前述したように、めくりユニット駆動ベルト５２が駆動されて、ページめくり読取ユニット１が、そのホームポジション位置〔１Ａ〕に到達した時点で、図３に示すホームセンサ３０により、スキャニングパイプ５１のホーム検知フィラ７５が検知されるように構成されている。

【００６４】次に、図６乃至図８を参照しながら、この原稿読み取り装置における搬送部ロック装置の構成について説明する。図６乃至図８は、ＭＦＤＳの両側部に配設された搬送部ロック装置１４０（両側とも同じ構成）の一方の構成を示す概略図である。

【００６５】この搬送部ロック装置１４０は、ロック解除ソレノイド１３２、及び、ロック杆１３４等で構成されている。図６乃至図８において、ロック解除ソレノイド１３２は、ロック解除ソレノイドアーム１３３の一端と連結されている。

【００６６】ロック解除ソレノイドアーム１３３の他端は、ロック杆１３４の一端と回転自在に軸支されている。このロック杆１３４のロック解除ソレノイドアーム１３３に対向するがわの側部には、電磁ロック１４１が配置されている。また、ロック杆１３４の他端には、ロック解除ソレノイドアーム１３３が配置されているがわに向けて鍵爪状に折曲されたロック爪１３４ａが形成されている。

【００６７】ロック杆１３４は、回転部１３６で回転自在に支持されており、その回転部１３６の上部がわが、ロックスプリング１３５の一端に繋がれている。このロックスプリング１３５の他端は、搬送部１９の一部に繋がれていて、これにより、ロック杆１３４に対して、第６図において、時計回転方向への回動習性を与えている。

【００６８】このロック杆１３４の回動習性による回動は、その回転部１３６の左下側に配置されたロック爪ストッパ１３７によって、所定の角度に阻止されている。一方、原稿台１８側には、ロックピン１３９、及び、搬送部ロックセンサ３１が配置されたロック部１３８が形成されている。

【００６９】図６において、搬送部１９を押し下げながら閉じていくと、図７に示すように、ロックピン１３９にロック杆１３４のロック爪１３４ａの下端が当接して、ロック杆１３４がその回転部１３６を回転軸として、反時計回りに回動される。この状態から、搬送部１９をさらに閉じていくと、図８に示すように、ロック杆１３４のロック爪１３４ａがロックピン１３９に引っかかって、搬送部１９が原稿台１８に固定される。

【００７０】また、この搬送部１９のロック動作時に、

13

ロック杆１３４のロック爪１３４ａによって、搬送部ロックセンサ３１が作動される。この搬送部１９のロックの解除は、図１６に示す操作表示ボード３１３のオープンキー６２０を押すことによって実行される。

【００７１】すなわち、操作表示ボード３１３のオープンキー６２０を押下すると、ロック解除ソレノイド１３２が作動して、ロック杆１３４が回転部１３６を回転軸として反時計回りに回動し、ロックピン１３９からロック杆１３４のロック爪１３４ａが外れ、図示せぬ搬送部開閉スプリングによって、搬送部１９が上方へ開放される（図１０参照）。

【００７２】但し、このオープンキー６２０は、ブック原稿９２の一連のページめくり走査時、読み取り走査途中、及び、シート原稿２００の搬送中の場合には、作動しない（入力を受け付けない）ようにプログラムされている。

【００７３】また、このようにオープンキー６２０が入力を受け付けない状態では、電磁ロック１４１が作動され、この電磁ロック１４１によって、ロック爪１３４ａがロックピン１３９に係合されたままの状態で、ロック杆１３４の回動が拘束されるように構成されている。

【００７４】上述のように構成されたＭＦＤＳは、図９及び図１０に示すように、例えば、プリンタ３００の上部に搭載されて使用される。図１０は、上述したように、このＭＦＤＳの搬送部１９を開放した状態を示している。

【００７５】次に、図１１乃至図１３を参照しながら、この原稿読み取り装置におけるページめくり読取ユニット１の内部の第１読み取りセンサユニット９の構成について説明する。図１１は第１読み取りセンサユニット９の端部付近の斜視図、図１２は第１読み取りセンサユニット９の端部付近の側面図、図１３は第１読み取りセンサユニット９の端部の詳細断面図である。

【００７６】この第１読み取りセンサユニット９の両端部は、両方共同じに構成されているので、ここではその片方のみの構成を説明する。第１読み取りセンサユニット９は、図１１に示すように、コの字状に形成された読み取りセンサブラケット１４６によって、その上部が覆われており、かつ、この読み取りセンサブラケット１４６に対して、上下動自在に配設されている。　読み取りセンサブラケット１４６は、その両端が第１めくりユニット側板４０及び第２めくりユニット側板４１に固定されることによって、ページめくり読取ユニット１と一体に構成されている。

【００７７】この読み取りセンサブラケット１４６の端部の少し内側には、読み取りセンサスタッド１４８が下向きに固定されている。この読み取りセンサスタッド１４８の下端部は、図１３に示すように、第１読み取りセンサユニット９の端部に形成されているボス１４９に嵌合されている。これにより、第１読み取りセンサユニット９が、読み取りセンサスタッド１４８を介して、読み取りセンサブラケット１４６に対して上下動自在に支持される。

14

【００７８】ここで、読み取りセンサスタッド１４８とボス１４９とは、読み取りセンサスタッド１４８の下端に形成されたフランジ状の掛り部によって、それらの嵌合が外れないように構成されている。また、読み取りセンサブラケット１４６の読み取りセンサスタッド１４８の基部と、第１読み取りセンサユニット９のボス１４９の基部との間には、読み取りセンサスプリング１４７が軸装されており、この読み取りセンサスプリング１４７の伸長力によって、第１読み取りセンサユニット９に対して、下方への変位習性が付勢されている。

【００７９】これにより、この第１読み取りセンサユニット９は、常に、ページめくり読取ユニット１の最下部位に位置し、例えば、ブック原稿９２の表面の凹凸等によって外力を受けた際に、この外力に逆らうことなく、ブック原稿９２の表面の凹凸等に沿って滑らかに上下動される。また、第１読み取りセンサユニット９の端部には、図１１に示すように、読み取りセンサ解除ソレノイドアーム１５１を介して、読み取りセンサ解除ソレノイド１５０が取付けられている。

【００８０】この読み取りセンサ解除ソレノイド１５０は、図１２に示すように、第１めくりユニット側板４０に固定されており、第１読み取りセンサユニット９で原稿読み取り動作を行なわずにページめくり読取ユニット１が移動されるとき、例えば、ページめくり動作時、非読み取りページの空走査時、及び、シート原稿スキャンモードのリターン時等に、この読み取りセンサ解除ソレノイド１５０が作動される。

【００８１】この読み取りセンサ解除ソレノイド１５０が作動されると、読み取りセンサスプリング１４７の弾力に抗して、第１読み取りセンサユニット９が上方へ移動され、その原稿走査面が原稿の表面から退避（離隔）される。この第１読み取りセンサユニット９は、めくりローラ２の回転によって発せられるエンコーダ１５２の信号をその画像読み取りの基準信号としている。このめくりローラ２とエンコーダ１５２は、第１４図に示すように構成されている。

【００８２】第１４図は、めくりローラ２の奥側の側面図を示している。第１４図において、めくりローラ２の奥側の端部には、王冠状に形成されたフィラ１５３が配設されている。フィラ１５３は、等間隔の同じ幅のスリットを円周上に形成して構成されている。

【００８３】エンコーダ１５２は、このフィラ１５３を上下に挟むようにして、第２めくりユニット側板４１に固定されている。これにより、このエンコーダ１５２は、めくりローラ２の回転に応じ、フィラ１５３がエンコーダ１５２の検知光路を周期的に遮ることによって、第１読み取りセンサユニット９の画像読み取りの基準信

15

号を発生する。

【００８４】一方、原稿台１８に対するブック原稿９２の位置決めは、中央基準位置決め部２４によって行われる。この中央基準位置決め部２４の詳細断面図を第１５図に示す。この中央基準位置決め部２４は、ブック原稿９２の読み取り走査時、及び、ページめくり走査時の基準位置となっている。

【００８５】中央基準位置決め部２４は、原稿載置面１１６の中央部に形成された溝内に構成されている。この溝内には、中央位置決め板１１３が、原稿載置面１１６に対して昇降自在に嵌合されている。中央位置決め板１１３は、その下部に配設された中央位置決めスプリング１１４によって、常に、上昇する習性が与えられている。

【００８６】この習性による中央位置決め板１１３の上方への移動は、原稿載置面１１６の溝の縁部に形成されたストッパ１１８に、中央位置決め板１１３のストッパ爪１１９が当接することによって阻止され、平生、第１５図の破線で示す位置で停止されている。原稿載置面１１６へのブック原稿９２のセットは、溝内の中央位置決め板１１３の上にブック原稿９２の背（綴じ部）を載せることによって行われる。

【００８７】すなわち、溝内の中央位置決め板１１３の上にブック原稿９２の背（綴じ部）が載せられると、このブック原稿９２の自重によって、中央位置決め板１１３が下方に押し下げられる。これにより、原稿載置面１１６の溝内の側部に配置されたブック原稿センサ２７が、中央位置決め板１１３の移動を検知して、ブック原稿９２のセットが認知される。

【００８８】ところで、この原稿台１８の全面部位には、ＭＦＤＳの操作表示ボード３１３が配置されている（図９及び図１０参照）。この操作表示ボード３１３は、図１６に示すような、多数の入力キーが配置されている。以下、これらの入力キーの機能を順に説明する。

【００８９】スタートキー６００は、原稿の読み取り開始を指示するときに押される。エンターキー６０１は、テンキー入力や液晶表示パネル上での選択入力のときに、その入力を確定する際に押される。テンキー６０２は、原稿のプリント枚数、及び、ページめくり枚数等を設定するときに使用される。

【００９０】読み取り開始ページ選択キー６０３は、ブック原稿読み取りモード時おいて、ブック原稿９２の向かって「左」・「右」どちらのページから読み取りを開始するかを選択するキーであって、このキーを１回押下する毎に、ブック原稿９２の読み取り開始ページの左右が切り替わる。

【００９１】このキーの初期設定時における読み取り開始ページは、「左」ページに設定されており、このキーにより選択されたの読み取り開始ページがブック原稿９２の左右の何れであるかは、２つの読み取り開始ページ

16

表示ＬＥＤ６３１のどちらが点灯されているかによって表示される。

【００９２】読み取り総ページ設定キー６０６は、ブック原稿読み取りモードにおいて、そのページめくり枚数を入力する際の、読み取りたい総ページ枚数を設定するときに押下される。

【００９３】ブック原稿９２の読み取り総ページ枚数は、この読み取り総ページ設定キー６０６を押下して、テンキー６０２でそのページ数を入力した後、エンターキー６０１を押下することにより確定され、この確定された値が液晶表示パネル６３０に表示される。

【００９４】ブックサイズ選択キー６０７は、ブック原稿読み取りモードのときに、「自動ブックサイズ認識モード」、もしくは、「ブックサイズキー入力モード」の何れかを選択する際に押下される。また、このキーが１回押される毎に、ブックサイズ表示ＬＥＤ６３２の表示が、「自動」・「定形」・「不定形（mm入力）」の順に切り替えられ、このブックサイズ表示ＬＥＤ６３２により表示されたモードが選択される。

【００９５】このブックサイズ選択キー６０７の初期設定時の、ブックサイズ表示ＬＥＤ６３２の表示は、「自動」になっており、「自動ブックサイズ認識モード」が選択されている。ここで、ブックサイズキー入力モードには、「定形」と、「不定形（mm入力）」とがあり、ブックサイズが、Ａ５・Ｂ５・Ａ４の場合に限り、「定形」を選択することにより、定形ブックサイズ選択キー６１９によるブックサイズの入力が可能となる。

【００９６】この定形ブックサイズ選択キー６１９が１回押される毎に、定形ブックサイズ表示ＬＥＤ６３３の表示が、「Ａ５」・「Ｂ５」・「Ａ４」の順に切り替えられ、選択されたブックサイズが表示されて認識される。この定形ブックサイズ選択キー６１９の初期設定時の、定形ブックサイズ表示ＬＥＤ６３３の表示は、「Ａ４」となっている。但し、ここでいうブックサイズとは、ブック原稿９２の表紙の大きさを指している。

【００９７】ここで、ブック原稿９２が、上述した定形ブックサイズ以外の場合には、ブックサイズ選択キー６０７により、「不定形（mm入力）」を選択し、テンキー６０２で、セットされたブック原稿９２の縦サイズ、及び、横サイズ（mm単位）をそれぞれ入力した後、エンターキー６０１を押下して、そのブックサイズを確定する。

【００９８】このようにしてブック原稿９２のサイズが確定されると、その入力されたサイズ値が液晶表示パネル６３０に表示される。ブック綴じ部マスク領域設定キー６０８は、ブック原稿読み取りモードのときに、中央基準位置決め部２４のセンターからの非読み取り領域（マスク領域）を、「左（－）」・「右（＋）」の何れとするかを設定するときに押下される。

【００９９】すなわち、ブック原稿９２のブック綴じ部

にマスク領域を形成するときには、先ず、このブック綴じ部マスク領域設定キー６０８で、ブック原稿９２の左右何れのページにマスク領域を形成するかを設定し、次いで、この設定された「左マスク領域（－）」、もしくは、「右マスク領域（＋）」の長さ（mm単位）を、テンキー６０２により入力した後、エンターキー６０１でこの入力値を確定する。　このようにしてブック原稿９２のマスク領域が確定されると、その入力された値が液晶表示パネル６３０に表示される。このブック綴じ部マスク領域設定キー６０８の初期設定時におけるマスク領域の値は、「±１０mm」となっている。

【０１００】読み取り領域選択キー６０９は、ブック原稿読み取りモードのときに、ブック原稿９２の読み取り領域を、「片頁（左）」・「片頁（右）」・「両頁」のうちの何れとするかを選択する際に押下される。

【０１０１】この読み取り領域選択キー６０９が１回押される毎に、読み取り領域表示ＬＥＤ６３６の表示が、「片頁（左）」・「片頁（右）」・「両頁」の順に切り替えられ、選択された読み取り領域が表示されて認識される。

【０１０２】この読み取り領域選択キー６０９の初期設定時の、読み取り領域表示ＬＥＤ６３６の表示は、「両頁」となっている。ここで、「片頁（左）」が選択された場合には、ブック原稿９２の向かって左ページのみの読み取り走査が実行され、右ページの読み取り走査は行われない。また、ここで、「片頁（右）」が選択された場合には、ブック原稿９２の向かって右ページのみの読み取り走査が実行され、左ページの読み取り走査は行われない。

【０１０３】見開き連写キー６１０は、ブック原稿読み取りモードの「見開き２ページ連続読み取りモード」、及び、その「両面モード」のときに、読み取った原稿情報を等倍率でプリントアウトすることを指示する際に押下される。

【０１０４】見開き連写縮小キー６１１は、ブック原稿読み取りモードの「見開き２ページ連続読み取りモード」、及び、その「両面モード」のときに、読み取った原稿情報を縮小倍率でプリントアウトすることを指示する際に押下される。

【０１０５】この時の、原稿情報の縮小倍率の設定は、プリント変倍キー６１４を操作することによって行われる。また、この見開き連写縮小キー６１１の初期設定時における基準縮小倍率は、「原稿のサイズ×０．７１（Ａ３⇒Ａ４／Ｂ４⇒Ｂ５）」に設定されている。

【０１０６】両面モード選択キー６１２は、ブック原稿読み取りモードが、「見開き１ページ区切り読み取りモード」の「両面モード」のときに、何れの面を表にし、また、何れの面を裏にしてプリントするかを、「見開き両面モード」・「オリジナル両面モード」・「順次両面モード」の３つの両面モードの中から選択する際に押下される。

【０１０７】この両面モード選択キー６１２が１回押される毎に、両面モード表示ＬＥＤ６３４の表示が、「見開き両面モード」・「オリジナル両面モード」・「順次両面モード」の順に切り替えられ、選択された読み取り領域が表示されて認識される。

【０１０８】この両面モード選択キー６１２の初期設定時の、両面モード表示ＬＥＤ６３４の表示は、「オリジナル両面モード」となっている。ここで、「見開き両面モード」が選択された場合には、見開かれたブック原稿９２の左右両ページのうち、左ページを表にし、右ページを裏にして、両面プリントが実行される。

【０１０９】このとき、読み取り開始ページ選択キー６０３により、ブック原稿９２の読み取り開始ページが「右」に設定されている場合には、始めの１枚目のプリントは片面プリントとなる。

【０１１０】また、ここで、「オリジナル両面モード」が選択された場合には、見開かれたブック原稿９２の左右両ページのうち、右ページを表にし、ページめくり動作によりめくられた次のページの左ページを裏にして、両面プリントが実行される。すなわち、この「オリジナル両面モード」では、読み取られるブック原稿９２の装丁と全く同様にプリントされる。

【０１１１】このとき、読み取り開始ページ選択キー６０３により、ブック原稿９２の読み取り開始ページが「右」に設定されている場合には、「見開き両面モード」の場合と同様、始めの１枚目のプリントは片面プリントとなる。さらに、ここで、「順次両面モード」が選択された場合には、見開かれたブック原稿９２の左右両ページのうち、読み取り開始ページ選択キー６０３で設定されたページを表にし、ページめくり動作によりめくられた次のページを裏にして、以後、順次読み取った順番に両面プリントが実行される。

【０１１２】見開き連写高速プリント設定キー６１３は、ブック原稿読み取りモードの「見開き２ページ連続読み取りモード」の「片面モード」のときに、ブック原稿９２の綴じ部付近において、ページめくり読取ユニット１の操作を減速または停止させること無く、連続して読み取り走査を実行し、ブック原稿９２の左右両ページの連続プリントを指示する際に押下される。

【０１１３】プリント変倍キー６１４は、読み取った画像を変倍してプリントするときに、その変倍率を設定するためのキーである。このプリント変倍キー６１４を押下すると、液晶表示パネル６３０に、予め設定された変倍率が表示される。ここで、変倍率は、カーソル移動キー６１７で希望する変倍率にカーソルを合わせてから、エンターキー６０１を押下することにより確定される。

【０１１４】画像処理設定キー６１５は、読み取った画像を画像処理してプリントするときに、その画像処理モードを設定するためのキーである。この画像処理設定キ

19

ー６１５を押下すると、液晶表示パネル６３０に、予め設定された画像処理モードが表示される。

【０１１５】ここで、画像処理モードは、カーソル移動キー６１７で希望する画像処理モードにカーソルを合わせてから、エンターキー６０１を押下することにより確定される。モード設定選択キー６１６は、ＭＦＤＳの動作モードを設定するためのキーである。

【０１１６】このモード設定選択キー６１６を押下すると、液晶表示パネル６３０に、予め設定されたＭＦＤＳの動作モードが表示される。ここで、ＭＦＤＳの動作モードは、希望するＭＦＤＳの動作モードカーソルを合わせてから、エンターキー６０１を押下することにより確定される。

【０１１７】カーソル移動キー６１７は、液晶表示パネル６３０に表示された各選択エリアにカーソルを移動させるためのキーである。読み取りスキップページ設定キー６１８は、ブック原稿読み取りモードにおいて、読み取り走査を実行せずに読み飛ばすページを設定するためのキーである。　すなわち、ブック原稿９２の各ページのうち、原稿読み取り走査を実行せずに読み飛ばしたいページがある場合には、先ず、この読み取りスキップページ設定キー６１８を押下し、次いで、ブック原稿９２の読み飛ばしたいページ（スキップページ）が、その読み取り開始ページから何ページ目であるかをテンキー６０２で入力した後、エンターキー６０１を押下して、このスキップページを確定する。

【０１１８】このようにして入力されたスキップページは、液晶表示パネル６３０に表示される。定形ブックサイズ選択キー６１９は、ブック原稿９２のブックサイズを選択するためのキーである。

【０１１９】前述したように、ブックサイズキー入力モードには、「定形」と、「不定形（mm入力）」とがあり、ブックサイズが、Ａ５・Ｂ５・Ａ４の場合に限り、「定形」を選択することにより、この定形ブックサイズ選択キー６１９によるブックサイズの入力が可能となる。

【０１２０】この定形ブックサイズ選択キー６１９が１回押される毎に、定形ブックサイズ表示ＬＥＤ６３３の表示が、「Ａ５」・「Ｂ５」・「Ａ４」の順に切り替えられ、選択されたブックサイズが表示されて認識される。

【０１２１】この定形ブックサイズ選択キー６１９の初期設定時の、定形ブックサイズ表示ＬＥＤ６３３の表示は、「Ａ４」となっている。但し、ここでいうブックサイズとは、ブック原稿９２の表紙の大きさを指している。

【０１２２】オープンキー６２０は、ＭＦＤＳの搬送部１９を開放するときに押下される。シート原稿セット選択キー６２５は、シート原稿読み取りモードにおいて、「シート原稿スルーモード」の「片面原稿読み取りモード」のときに、原稿載置面１１６にセットされるシート原稿２００の原稿面を「上向き」・「下向き」の何れかに選択するためのキーである。

20

【０１２３】このシート原稿セット選択キー６２５が１回押される毎に、シート原稿セット表示ＬＥＤ６３５の表示が、「上向き」・「下向き」の順に切り替えられ、選択されたシート原稿セット面が表示されて認識される。このシート原稿セット選択キー６２５の初期設定時の、シート原稿セット表示ＬＥＤ６３５の表示は、「上向き」となっている。

【０１２４】次に、図９及び図１０に示したプリンタ３００について説明する。このプリンタ３００の概略断面図を図１７に示す。画像処理後の画像情報は、プリンタ３００の書き込み部において、レーザ光のラスター走査によって、光の点の集合の形で感光体ドラム１７０上に書き込まれる。

【０１２５】このときのレーザ光源には、半導体レーザが使用されている。このプリンタ３００の書き込み部の平面図を図１８に示す。図１８において、半導体レーザ１７１で発せられたレーザ光は、コリメートレンズ１７２で平行な光束に変えられ、アパーチャ１７３により、一定形状の光束に整形される。

【０１２６】この整形されたビームは、第１シリンダーレンズ１７４により、その副走査方向を圧縮された形でポリゴンミラー１７５に入射される。このポリゴンミラー１７５は、正確な多角形状をしており、ポリゴンモータ１７６により、一定方向に一定の速度で回転されている。

【０１２７】このポリゴンミラー１７５の回転速度は、感光体ドラム１７０の速度と、書き込み密度と、面数とによって決定される。ポリゴンミラー１７５に入射されたレーザ光は、その反射光がミラーの回転により偏向される。この偏向されたレーザ光は、各ｆθレンズ１７７ａ，１７７ｂ，１７７ｃに入射される。

【０１２８】これらのｆθレンズ１７７ａ，１７７ｂ，１７７ｃは、角速度が一定の走査光を感光体ドラム１７０上で等速走査するように変換する機能、感光体ドラム１７０上で最小光点となるようにこの走査光を結像させる機能、及び、その面倒れを補正する機能を有している。

【０１２９】各ｆθレンズ１７７ａ，１７７ｂ，１７７ｃを通過した光は、その画像領域外で、同期検知ミラー１７８により同期検知センサ１７９に導かれ、主走査方向の頭出し信号を出す同期信号が出力されてから、一定時間後に画像データが１ライン分だけ出力され、以下、これを繰り返すことにより、１つの画像を形成する。一方、感光体ドラム１７０の表面には、感光層が塗布されている。

【０１３０】ここで、半導体レーザ１７１の７８０ｎｍという波長に感度を有する感光体としては、有機感光体

21

（ＯＰＣ）、α－Ｓｉ、Ｓｅ－Ｔｅ等が知られているが、本実施例では、有機感光体を使用している。

【０１３１】また、一般に、レーザ書き込みの場合、画像部に光をあてるＮ／Ｐプロセスと、地肌部に光をあてるＰ／Ｐプロセスがあるが、本実施例では、そのレーザ書き込みプロセスとして、画像部に光をあてるＮ／Ｐプロセスを採用している。

【０１３２】図１７において、感光体ドラム１７０の表面は、感光体ドラム１７０がわにグリッドを持つスコロトロン方式の帯電チャージャ１８０により、均一に負帯電させる。

【０１３３】次いで、この負帯電された感光体ドラム１７０の画像部に、レーザ光が照射されて、この画像部電位が落されると、この感光体ドラム１７０の表面に、地肌部電位が－７５０～－８００Ｖ、画像部電位が－５０Ｖ程度の静電潜像が形成される。

【０１３４】この静電潜像は、現像器１８１の現像ローラに、－５００～－６００Ｖのバイアス電圧を与えて、負帯電されたトナーによって、顕像化される。この現像器１８１によって顕像化された画像は、感光体ドラム１７０の回転にシンクロして給送された転写紙の紙面上に、この転写紙の裏面側から正電位のチャージをかける転写チャージャ１８２の転写作用によって転写される。

【０１３５】この画像の転写された転写紙は、転写チャージャ１８２と一体に保持された分離チャージャ１８３により交流除電されることによって、感光体ドラム１７０の表面から分離される。

【０１３６】このとき、転写紙に転写されずに、感光体ドラム１７０上に残留されたトナーは、クリーニングブレード１８４により感光体ドラム１７０の表面から掻き落され、このクリーニングブレード１８４の周囲に配設されたタンク１８５内に回収される。また、感光体ドラム１７０の表面に残留された電位のパターンは、除電ランプ１８６により光が照射されることによって消去される。

【０１３７】現像器１８１のすぐ下流側には、フォトセンサ１８７が設けられている。このフォトセンサ１８７は、受光素子と発光素子とで構成されており、感光体ドラム１７０の表面の反射濃度を計測し、この反射濃度（現像後のトナー濃度）が予め設定された基準値以下になったときに、現像器１８１内に新たなトナーを補給するためのトナー補給信号を出力する。

【０１３８】すなわち、このフォトセンサ１８７は、例えば、このフォトセンサ１８７の読み取り位置に対応した位置に、その光書き込み部で一定パターン（純黒または網点のパターン）を書き込み、このパターンを現像した後のパターン部の光反射率と、このパターン部位外の感光体ドラム１７０の光反射率との比から現像された画像の濃淡を判断し、この画像の濃度がその基準値よりも低いときにトナー補給信号を出力するように構成される。

22

【０１３９】ここで、新たな補給トナーが不足している場合には、トナー補給信号を出力しても、その現像濃度が高くならない点を利用して、このフォトセンサ１８７をトナーの残量不足を検知するセンサとして兼用させるように構成してもよい。

【０１４０】一方、本実施例のプリンタ３００は、複数のカセット１８８ａ，１８８ｂを備えており、且つ、画像が一度転写された転写紙を再給紙ループ１８９を通して両面給紙し得るように構成されている。

【０１４１】すなわち、図１７において、所定のカセットが選択された後、プリンタ３００のスタートボタンが押されると、各カセット１８８ａ，１８８ｂの各給紙コロ１９０ａ，１９０ｂのうちの選択されたがわの給紙コロが回転されて、そのカセット内の転写紙がレジストローラ１９１のニップに突き当たるまで給送される。

【０１４２】このレジストローラ１９１は、感光体ドラム１７０に形成された画像の位置と転写紙の位置とがシンクロするタイミングをとって、回転が開始され、感光体ドラム１７０の表面に向けて、転写紙を給送する。

【０１４３】これにより、この転写紙は、前述したように、画像が転写され、さらに、感光体ドラム１７０の表面から分離された後、分離搬送部１９２に吸引搬送され、ヒートローラ１９３、及び、加圧ローラ１９４からなる定着ローラによって、その紙面上に転写されたトナーが定着される。

【０１４４】このトナーの定着された転写紙は、通常のプリント時には、切換爪１９５が図１９（ａ）に示すような位置に臨み、この切換爪１９５により、プリンタ３００の排紙口を通して排紙トレイ１６９上に排出される。

【０１４５】ここで、プリンタ３００のプリントモードが「両面モード」の場合には、切換爪１９５が図１９（ｂ）に示す位置に切り換えられ、プリンタ３００の左側部に形成された両面搬送路に向けて転写紙が搬送される。

【０１４６】この転写紙は、反転ガイド爪１９６を通過して、一旦、反転ガイドトレー１９７上へ導かれた後、反転ガイド爪１９６が切り換えられ、且つ、反転ガイドローラ１９８が逆回転（反転）されることによって、再給紙ループ１８９を通して、再び、レジストローラ１９１のニップに当接されるまで給送される。

【０１４７】このようにして、再給送された転写紙は、前述の通常のプリント時と同様にして、感光体ドラム１７０上に形成されたトナー画像が転写・定着された後、図１９（ａ）で示す初期状態に切り換えられた、切換爪１９５を経て排紙トレイ１６９上に排出される。

【０１４８】ここで、片面プリントモード時の排出時における転写紙の画像定着面を排紙トレイ１６９の積載面に対向させて転写紙の排出を行う、「裏面排紙モード」

が設定されている場合には、上述の「両面モード」時と同様に、切換爪１９５が図１９（ｂ）に示す位置に切り換えられ、転写紙がそのまま排紙されること無く、一旦、両面搬送路に向けて搬送される。

【０１４９】そして、この転写紙の後端が、切換爪１９５を通過すると、その直後に、切換爪１９５が図１９（ｃ）に示す位置に切り換えられるとともに、スイッチバックローラ１９９の回転方向が反転されて、この両面搬送路に導かれた転写紙が、スイッチバック搬送され、プリンタ３００の排紙口を通して排紙トレイ１６９上に排出される。

【０１５０】この「裏面排紙モード」では、片面プリントされた転写紙が、その画像定着面を排紙トレイ１６９の積載面に対向させて排紙トレイ１６９上に排出されるので、原稿の読み取りページ順どおりに、転写紙のプリントページ順序の揃った転写紙の排出が行われる。

【０１５１】ところで、ページめくり読取ユニット１は、図４を参照して説明したように、第１めくりユニット側板４０と第２めくりユニット側板４１との間に、めくりローラ２、第１ローラ４、第２ローラ５、押えローラ６、及び、第１バイアスローラ３が、それぞれ回転自在に配設されている。

【０１５２】これらの、めくりローラ２、第１ローラ４、第２ローラ５、押えローラ６、及び、第１バイアスローラ３には、図２０に示すように、めくり搬送ベルト８が掛け渡されている。

【０１５３】第１バイアスローラ３には、第１高圧電源３２０が接続されていて、２本のポートより、吸着用，除電用の各周波数の交流電圧が、それぞれ与えられるように構成されている。　また、このページめくり読取ユニット１の内部の、めくりローラ２と第１読み取りセンサユニット９との間には、めくりガイド１０が、さらに、このめくりガイド１０の上方には、めくり搬送ベルト８の内周面に沿うようにしてページ収納部７が、それぞれ配設されている。

【０１５４】このページ収納部７には、ページめくり時のエラーを検知するための、フォトセンサなどからなるページめくりセンサ２９が配置されている。このページめくりセンサ２９は、めくりガイド１０に配置してもよい。

【０１５５】また、めくりローラ２と押えローラ６との間には、図１１を参照して説明したように、第１読み取りセンサユニット９が配置されている。この第１読み取りセンサユニット９は、ページめくり読取ユニット１に対して、約３ｍｍ程度上下移動可能に取付けられており、原稿読み取り時には、読み取りセンサスプリング１４７により下方に押下されて、ブック原稿９２、またはシート原稿２００の原稿面に密着されるように構成されている。

【０１５６】また、この第１読み取りセンサユニット９

は、図２１に示すように、原稿の照明系としてのＬＥＤ３１６、原稿像の結像系としてのＲＭＬＡ８１（ルーフミラーレンズアレイ）、及び、結像された原稿の光像を電気信号に変換する光電変換系としてのＳｉ等倍センサ３１５を備えている。

【０１５７】この第１読み取りセンサユニット９による原稿読み取りは、次のようにように行われる。図２１において、先ず、ＬＥＤ３１６から発せられた光が、バーレンズ８３により、原稿面上に集光されて、原稿が照明される。

【０１５８】次に、この原稿面からの反射光が、光路分離ミラー８４で反射されて、ＬＡ８５（レンズアレイ）、及び、ＲＭＡ８６（ルーフミラーアレイ）を通り、再び、光路分離ミラー８４により反射される。　この光路分離ミラー８４により反射された原稿の画像光は、Ｓｉ等倍センサ３１５の受光面上に結像され、このＳｉ等倍センサ３１５によって、これに結像された画像情報が、電気信号に変換されて読み取られる。

【０１５９】次に、上述のように構成された本願実施例における基本的なページめくり動作について説明する。先ず、本実施例における原稿の読み取りは、次のような手順で行われる。

【０１６０】原稿がブック原稿９２の場合には、操作表示ボード３１３のオープンキー６２０を押して、図１０に示すように搬送部１９を上方に開き、原稿台１８の中央基準位置決め部２４に、ブック原稿９２の綴じ部（背）をセットし、このブック原稿９２の読み取り開始ページを上向きに開いた状態で、図９に示すように搬送部１９を閉じる。

【０１６１】この状態で、操作表示ボード３１３の各キーを操作して、このブック原稿９２の読み取り条件を設定した後、スタートキー６００を押してＭＦＤＳをスタートさせる。これにより、図１に示すように、めくりユニット駆動ベルト５２が、めくりユニット駆動モータ６０（図３４参照）により駆動されて、ページめくり読取ユニット１が、左端のホームポジション位置〔１Ａ〕から、右方向に移動を開始し、このページめくり読取ユニット１の移動により、第１読み取りセンサユニット９が、ブック原稿９２の原稿情報を読み込んでいく。

【０１６２】このとき、めくり搬送ベルト８は、その回転が停止されており、見開かれたブック原稿９２の原稿面をその上から押え付けている。また、ページめくり読取ユニット１は、前述したように、その回動支持ロッド４２を支点として回動され、且つ、これと一体化された摺動パイプ４３が摺動支持ロッド４６に沿って上下動されることによって、ブック原稿９２の原稿面に沿って、この原稿面に第１読み取りセンサユニット９を密着させながら移動される。

【０１６３】さらに、このページめくり読取ユニット１が、ブック原稿９２の略中央のブック原稿読み取り途中

位置〔1B〕に到達すると、第1バイアスローラ3に、図20に示した第1高圧電源320から、吸着用の交流電圧が印加されて、めくり搬送ベルト8上にストライプ形状の電荷パターンが形成される。

【0164】このように、ページめくり読取ユニット1は、めくり搬送ベルト8上にストライプ形状の電荷パターンを形成しながら、図1の右端のエンドポジション位置〔1C〕まで移動し、その第1読み取りセンサユニット9によって、ブック原稿92の原稿情報を読み取る。

【0165】上述のようにして、ブック原稿92の原稿情報の読み取りが完了すると、ページめくり読取ユニット1は、図22に示すように、そのエンドポジション位置〔1C〕から、ホームポジション位置〔1A〕に向けて、復帰移動される。

【0166】このとき、図23に示すように、めくり搬送ベルト8上には、その読み取り動作時に形成された電荷パターンによって、不平等な電界が発生しており、この静電界により、ブック原稿92の右ページが、めくり搬送ベルト8に、静電的に吸着されるようになっている。

【0167】従って、この状態で、ページめくり読取ユニット1が、そのエンドポジション位置〔1C〕に向けて移動され、このページめくり読取ユニット1が、図22に示す、めくり開始位置〔1D〕に到達すると、図24に示すように、このめくり搬送ベルト8と一緒に、ブック原稿92の右ページ1枚分の原稿の端部が、ページめくり読取ユニット1の中に巻き込まれる。

【0168】このように、ページめくり読取ユニット1内に巻き込まれた原稿は、このページめくり読取ユニット1の移動に伴って、そのめくりローラ2の曲率と原稿の腰の強さとによる曲率分離により、その先端が、めくり搬送ベルト8から徐々に分離される。

【0169】このようにして、めくり搬送ベルト8から徐々に分離された原稿は、図20に示しためくりガイド10に沿って移動され、この原稿の移動方向の下流側に配置されているページ収納部7内に導かれる。

【0170】このページ収納部7は、図20に示すように、円筒状に形成されており、その内周面に沿うように、ページめくり読取ユニット1内に巻き込まれた原稿を巻き取ることによって、極めて僅かなスペースに、読み取りを終えた1ページ分の原稿を収納することができる。

【0171】従って、このページめくり読取ユニット1によれば、図22に示すように、このページめくり読取ユニット1が、そのめくり開始位置〔1D〕から巻き取り完了位置〔1E〕に向けて復帰移動されることにより、その読み取りを終えた1ページ分の原稿と、次の読み取り動作により読み取りが行われる次頁の原稿とを、極めてスムーズに分離させることができる。

【0172】このようにしてページ収納部7に収納された原稿は、ページめくり読取ユニット1が、その巻き取り完了位置〔1E〕から、ブック原稿92の中央を越えて、ブック原稿92の左ページがわのページ排出位置〔1F〕に向けて、さらに復帰移動されることによって、このページめくり読取ユニット1と、これに収納された原稿との相対的な移動により、この原稿がページめくり読取ユニット1のページ収納部7から排出され始める。

【0173】そして、このページめくり読取ユニット1が、そのホームポジション位置〔1A〕に復帰移動されて、その移動が完了することにより、このページめくり読取ユニット1のページ収納部7に収納されていた原稿の排出が完了して、読み取りを終えた原稿の1ページ分のめくり動作が完了される。

【0174】一方、この原稿のめくり動作中においては、ページめくり読取ユニット1のページ収納部7への原稿の巻き取りが開始された直後から、この原稿の巻き取りが完了する間にかけて、第1バイアスローラ3に、図20に示した第1高圧電源320から除電用の交流電圧が印加されることによって、めくり搬送ベルト8上に形成されていた電荷パターンが除電されるようにプログラムされている。

【0175】従って、このページめくり読取ユニット1の復帰移動時においては、読み取りを終えた原稿と、この原稿をめくるめくり搬送ベルト8との間に、静電的な吸着力が発生しないので、この原稿の巻き取り動作、及び、排出動作を極めてスムーズに行うことができる。

【0176】本願実施例では、図20、図23及び図24に示したように、無端ベルトからなる、めくり搬送ベルト8に、高圧電源320から吸着用の交流電圧を印加して、めくり搬送ベルト8の表面に、交番的なストライプ形状、もしくは、市松模様などの電荷パターンを形成することによって、このめくり搬送ベルト8に不平等電界を発生させて、原稿の保持搬送、及び、ページめくりを行っている。

【0177】このページめくり方式によれば、原稿の保持搬送、及び、ページめくり動作を極めてスムーズに行うことがでる。

【0178】以下、このページめくり方式における基本的な構成、及び、静電吸着原理について説明する。このページめくり方式に使用されるめくり搬送ベルト8としては、無端ベルト状に形成された誘電体の裏面に、導電処理を施してなる2層構造のベルトを使用した。

【0179】このめくり搬送ベルト8に対して不平等電界を発生させる手段としては、例えば、このめくり搬送ベルト8の表面に周面を接触させて回転自在に支持された第1バイアスローラ3に、第1高圧電源320により吸着用の交流電圧を印加すればよい。

【0180】図20及び図23に示すように、めくり搬送ベルト8の導電層8bをアース面として、第1バイア

スローラ３に交番的な電界を印加させながら、このめくり搬送ベルト８と第１バイアスローラ３とを相対移動させることによって、めくり搬送ベルト８の誘電体８ａの表面に、ストライプ形状の電荷パターンが形成される。

【０１８１】これによって、めくり搬送ベルト８の誘電体８ａの表面近傍に、不平等電界が発生する。この不平等な電界中に、読み取り原稿となる用紙等の誘電体を近づけると、その内部が分極をおこし、この電界が不平等なために、この用紙にめくり搬送ベルト８がわへの吸引力が働く。

【０１８２】本実施例の具体的な構成としては、めくり搬送ベルト８として、厚さ７５μmのＰＥＴフィルム（誘電体８ａ）に、厚さ１０μmのアルミ蒸着層（導電層８ｂ）が形成された無端ベルトを使用し、これに形成される電荷パターンのピッチを２．４mmとした。すなわち、ブック原稿９２の読み取り速度を１２０mm／ｓ、交流周波数を５０Ｈｚ、印加電圧を±２ｋＶｐ－ｐとした。また、図２５に印加電圧を±２ｋＶｐ－ｐに一定としたときの搬送力のピッチ特性の実験値を、図２６に印加電圧を±２ｋＶｐ－ｐに一定としたときの吸着力のピッチ特性の実験値を、図２７に電荷パターンのピッチを２．４mmに一定としたときの搬送力の印加電圧特性の実験値を、第２８図に電荷パターンのピッチを２．４mmに一定としたときの吸着力の印加電圧特性の実験値を示す。

【０１８３】これらの実験値から明らかなように、本実施例に使用される電荷パターンのピッチ、及び、印加電圧は、上述した値に限定されるものではなく、例えば、電荷パターンのピッチとしては、０．５mm～１０mmの範囲であれば良く、また、印加電圧としては、±１ｋＶｐ－ｐ以上であれば良い。

【０１８４】また、本実施例では、除電用の高周波交流電圧として、２ｋＨｚの周波数と、±２ｋＶｐ－ｐの印加電圧を使用しているが、これらの各値についても、その除電効果が得られる値であれば、どのような値であっても良い。

【０１８５】ところで、上記実施例では、第１バイアスローラ３に、第１高圧電源３２０により、交流電圧を印加することによって、搬送ベルト８上を正・負に帯電させ、この搬送ベルト８の表面近傍に不平等な電界を発生させて、原稿の保持搬送およびページめくりを行なうように構成されている。

【０１８６】ここで、搬送ベルト８に印加される電圧は、交番電圧であれば、原稿を吸引することが可能であり、交流電圧のみに限定する必然性は無い。すなわち、交流電圧は、周期的にその方向を変える電流であって、１周期にわたって平均値が“０”であることを言い、本発明では、このような電圧に限定されない。

【０１８７】但し、搬送ベルト８に印加される電圧を交流電圧にすることによって、搬送ベルト８の原稿吸着力が均一化されるという一応の効果は得られる。従って、搬送ベルト８の表面近傍に不平等電界を発生させるための手段としては、上記実施例以外の構成としてもよい。

【０１８８】例えば、搬送ベルト８の表面近傍に不平等電界を発生させるための手段として、図２９および図３０に示すように、金属等の導電性電極ローラからなるバイアスローラ３－Ａの外周面に、市松模様の凹凸を形成し、このバイアスローラ３－Ａに、第１高圧電源３２０により、直流電圧を印加するように構成してもよい。

【０１８９】この手段によれば、搬送ベルトの表面に対して、バイアスローラ３－Ａの外周面をなす凸部３ａのみが接触し、凹部３ｂが非接触となるので、搬送ベルト８の表面に、バイアスローラ３－Ａの周面の凹凸に対応した市松模様の高密度な電荷パターンが形成される。

【０１９０】これにより、誘電体である搬送ベルト８上の帯電領域（図２９の＋部分）と非帯電領域との境界部には、他の部分よりも強い電場が存在し、不平等電界が形成され、上記の市松模様の電荷パターンのそれぞれの端縁部に発生した強い電場によって、ブック原稿のページが搬送ベルト８に静電的に強く吸着される。

【０１９１】この第２の実施例のように、電荷パターンを市松模様とした場合には、その模様の単位を略正方形にすることによって、図２４に示した第１の実施例のようなストライプ模様の電荷パターンに比べ、その単位面積中に存在する端縁部の長さが略２倍になるので、原稿ページの保持吸着力が倍増され、ページめくり動作の信頼性が向上される。

【０１９２】一方、上記第２の実施例では、バイアスローラ３－Ａの周面に凹凸を形成して市松模様を構成したが、例えば、樹脂やセラミック等からなる絶縁ローラの表面に、プリントや蒸着等により市松模様の導電層パターンを形成したり、市松模様のシート状の電極を接着し、この導電層パターンや電極を電気的に結合されるように構成してもよい。

【０１９３】また、搬送ベルト８の表面近傍に不平等電界を発生させるための第３の実施例としては、図３１に示すように、全体が金属等の導電性材料によって形成された電荷付与部材としてのローラ状電極３－Ｂの外周面に、フランジ状に突出する多数の電極部３ｃと、周溝状の非電極部３ｄとを、ローラ状電極３－Ｂの軸方向に沿って交互に配列し、このローラ状電極３－Ｂに、第１高圧電源３２０により、直流電圧を印加するように構成してもよい。

【０１９４】この第３の実施例によれば、前記第１の実施例と同様に、搬送ベルト８の表面に、帯電領域と帯電領域とが直線状に交互に配列されたストライプ状の電荷パターンが形成される。

【０１９５】一方、図３２に示す第４の実施例は、電極部３ｃと非電極部３ｄとを交互に配列してなる電荷印加手段を、前記のローラ状電極３－Ｂに替えて、櫛歯状の

29

電極板３－Ｃで構成したもので、前記第３の実施例のものと同様な作用が得られる。

【０１９６】これらの第３，第４の各実施例では、ローラ状電極３－Ｂ、および、電極板３－Ｃが、何れも導電性の材料で構成されるため、母材の一部を切り欠くことによって、その非電極部３ｄを形成したが、例えば、樹脂やセラミック等からなる絶縁ローラの表面に、電極部３ｃとしての多数の導電部を電気的に接続して配列することにより、電荷印加手段を形成してもよい。

【０１９７】上述のように、これらの第３，第４の各実施例では、搬送ベルト８に対して、ローラ状電極３－Ｂ、および、電極板３－Ｃを対向させて配置するだけの構造であるので、回転駆動するための機構が不要になり、そのコストを低減、および、小型化を図ることができる。

【０１９８】ところで、このようにして形成されるストライプ状の電荷パターンは、電極部３ｃの縁端から電荷が回り込むため、その帯電領域の幅が非帯電領域の幅よりも広くなる傾向があり、その境界の電場が弱められる虞れが高い。

【０１９９】そこで、これらの実施例では、搬送ベルト８上に形成される電荷パターンの帯電領域の幅と非帯電領域の幅とが略均等となるように、ローラ状電極３－Ｂ、および、電極板３－Ｃの電極部３ｃの幅が、非電極部３ｄの幅よりも予め大きく形成されている。

【０２００】従って、これらの第３，第４の実施例によれば、その電荷パターンの帯電領域と非帯電領域との境界が明確となり、この境界部の電場の強度を大きくできるので、その電極部３ｃと非電極部３ｄとの配列ピッチを密にしてその端縁効果を増大させることにより、ブック原稿のページの搬送ベルト８に対する吸着力を高めることができる。

【０２０１】また、さらに他の、搬送ベルト８の表面近傍に不平等電界を発生させるための手段としては、図３３に示すように、直流電源からなる高圧電源３２０の印加電圧をスイッチ３２０ａでオン／オフを繰り返しながら、バイアスローラ３－Ｄに直流電圧を印加するように構成してもよい。

【０２０２】上述のように、不平等電界を発生させるための手段は、交流電圧に限定されるものでなく、矩形状、三角状、鋸形状等の交番電圧、あるいは、交流交番電圧に直流成分を重畳し、正・負のどちらかに偏った電圧でもよい。

【０２０３】このように、本実施例におけるページめくり方式によれば、めくり搬送ベルト８に吸着させる用紙（ブック原稿９２）には、何等細工を施す必要が無いので、この用紙どうしが静電気により互いに引き合うことが無く、用紙端部の乱れ（不揃い）の発生による、ブック原稿９２のページめくりミスの発生を無くすことができる。

30

【０２０４】また、このページめくり方式によれば、その吸着力の発生している箇所が、めくり搬送ベルト８の表面の近傍であるので、このめくり搬送ベルト８の表面に接触している用紙、すなわち、ブック原稿９２のページめくりが実行されるがわの最上位の原稿に対しては、充分に大きな搬送力、及び、吸着力が作用するが、この原稿の下位に位置する、２枚目以下の原稿に対しては、これらの搬送力、及び、吸着力が、ほとんど作用することが無い。

【０２０５】従って、このページめくり方式によれば、１枚だけのページめくりを確実に実行することができるので、この方式は、ブック原稿９２のページめくり方式として最適な方式となる。

【０２０６】次に、上述のように構成されたＭＦＤＳの動作について説明する。図３４はＭＦＤＳの電装ブロック図、図３５、図３６、図３７はブック原稿読み取り時の動作モード遷移図、図３５、図３８はシート原稿読み取り時の遷移図、及び、図３９はＭＦＤＳの動作モードを示すフローチャートである。

【０２０７】先ず、図３４に基づいて、ＭＦＤＳの制御手段について説明する。図３４において、メイン制御ボード３１０は、各ボード間のコマンド、及び、データを制御し、各負荷のＯＮ／ＯＦＦタイミングや、各センサ入力による異常処理、及び、モード切り換え等を行いＭＦＤＳ全体のコントロールをしている。

【０２０８】また、このメイン制御ボード３１０は、接続機器との通信を行うことにより、通信プロトコルを設定し、各接続機器に対して個別に対応できるように構成されている。

【０２０９】例えば、プリンタ３００がその出力装置として接続されている場合には、その画素密度、処理速度、両面プリントの可否、及び、裏面排紙の可否等をチェックし、その対応モード選択域を決定できるように構成されている。

【０２１０】さらに、このメイン制御ボード３１０は、各モードに対応して、外部機器へのインターフェースを２系統備えている。本実施例のメイン制御ボード３１０では、各ボード間のコマンドをシリアル通信で行い、そのデータ・制御線から分離して、そのデータ出力中にもコマンド送受信を可能としている。

【０２１１】ここで、その汎用性を高める場合には、そのインターフェースとして、ＧＰＩＢ、セントロニクス、ＳＣＳＩ等の、どちらか一方、もしくは、両方を設定することにより、汎用プリンタや、パーソナルコンピュータを介して、そのディスプレイ表示や、光ディスク装置、ＨＤＤ、及び、ＦＤＤ等の記憶装置に対して、特別なインターフェースを使用すること無く、ストアすることができる。

【０２１２】一方、図３４において、めくり搬送ベルト駆動制御ボード３１１は、めくり搬送ベルト駆動モータ

６１の制御を行っている。また、めくりユニット駆動制御ボード３１２は、めくりユニット駆動モータ６０の制御を行っている。

【０２１３】ここで、めくり搬送ベルト駆動モータ６１は、このめくり搬送ベルト駆動モータ６１に一体的に取付けられたエンコーダの発するエンコーダパルスのフィードバックによって、めくり搬送ベルト８の速度を検出し、その速度位置制御及び正逆転動作を行っている。

【０２１４】これに対し、めくりユニット駆動モータ６０は、めくりローラ２に取付けられているエンコーダ１５２の発するエンコーダパルスのフィードバックによって、めくりローラ２の位置を検出し、その速度位置制御及び正逆転動作を行っている。

【０２１５】また、これらのめくり搬送ベルト駆動制御ボード３１１、及び、めくりユニット駆動制御ボード３１２は、メイン制御ボード３１０とそれぞれ接続されており、このメイン制御ボード３１０との間で、それぞれシリアル通信によりコマンドの送受信を行っている。

【０２１６】操作表示ボード３１３は、プリント置数、変倍率、ページめくり枚数、及び、各モード等を設定する各キー入力と、これらのキー入力に対する表示や、エラー表示、めくり状態表示、及び、各モードの原稿セット方法等の表示を行う。

【０２１７】ここで、モード表示においては、接続機器（プリンタ３００）の能力により可能なモードしか表示しないか、あるいは、不可能なモードが選択された場合にエラー表示を行う。

【０２１８】例えば、両面プリントのできないプリンタが接続されているにも拘らず、両面モード選択キー６１２のキー入力がなされた場合に、”接続のプリンタは両面不可です”等のエラー表示が行われる。

【０２１９】また、原稿の読み取りは、この操作表示ボード３１３上のスタートキー６００のキー入力により開始される。さらに、この操作表示ボード３１３と、メイン制御ボード３１０とは、シリアル通信によりコマンド、あるいは、データの送受信を行っている。

【０２２０】第１画像処理ボード３１４は、第１読み取りセンサユニット９に内蔵されているＳｉ等倍センサ３１５（以下これを第１ＣＣＤ３１５とする）のドライブクロックを発生する機能と、同じく第１読み取りセンサユニット９に内蔵されている第１ＬＥＤ３１６のＯＮ／ＯＦＦタイミングをとる機能と、第１ＣＣＤ３１５の出力を増幅し、この出力をＡ／Ｄ変換して画像処理を行う機能を有している。

【０２２１】また、この第１画像処理ボード３１４では、シェーディング補正、ＭＴＦ補正、主走査変倍、文字処理、写真処理、及び、ネガ・ポジ反転等の画像処理が行われる。この第１画像処理ボード３１４は、メイン制御ボード３１０に接続されており、このメイン制御ボード３１０との間で、データやコマンドを送受信している。

【０２２２】第２画像処理ボード３１７は、上述の第１画像処理ボード３１４と同様に、第２読み取りセンサユニット１４（第１読み取りセンサユニット９と同様に構成されている）に内蔵されている第２ＣＣＤ３１８のドライブクロックを発生する機能と、同じく第２読み取りセンサユニット１４に内蔵されている第２ＬＥＤ３１９のＯＮ／ＯＦＦタイミングをとる機能と、第２ＣＣＤ３１８の出力を増幅し、この出力をＡ／Ｄ変換して画像処理を行う機能を有している。

【０２２３】また、この第２画像処理ボード３１７では、第１画像処理ボード３１４と同様に、シェーディング補正、ＭＴＦ補正、主走査変倍、文字処理、写真処理、及び、ネガ・ポジ反転等の画像処理が行われる。

【０２２４】さらに、この第２画像処理ボード３１７は、メイン制御ボード３１０に接続されており、このメイン制御ボード３１０との間で、データやコマンドを送受信している。

【０２２５】第１高圧電源３２０は、前述したように、第１パイアスローラ３に高圧交流電圧を印加する電源であって、原稿吸着用と、ベルト除電用の２通りの周波数を発生できるように構成されており、各周波数の切り換えは、メイン制御ボード３１０の２本の出力ポートからの切り換え信号により行われる。

【０２２６】また、第２高圧電源３２１は、第１高圧電源３２０と同様に、第２パイアスローラ１１に高圧交流電圧を印加する電源であって、原稿吸着用と、ベルト除電用の２通りの周波数を発生できるように構成されており、各周波数の切り換えは、メイン制御ボード３１０の２本の出力ポートからの切り換え信号により行われる。

【０２２７】給紙クラッチ１２８は、メイン制御ボード３１０からの制御信号に基づいて、シート原稿２００の給紙開始タイミングを制御している。また、メイン制御ボード３１０の各入力ポートに接続されている各種センサ２５，２６，２７，２８，２９，３０，３１，３２，３３，３４は、前記の機構説明において述べた通り、モード切り換え、タイミング検知、及び、異常検知等を行っており、それぞれ、その検知信号をメイン制御ボード３１０に与えている。

【０２２８】次に、ＭＦＤＳの動作モードについて説明する。ＭＦＤＳの動作モードは、大きく２つのモードに分けられている。このＭＦＤＳの第１の動作モードは、図３５に示すような、ブック原稿の自動ページめくり読み取り動作を行うブック原稿読み取りモードであり、第２の動作モードは、同図３５に示すような、シート原稿の自動給紙読み取り動作を行うシート原稿読み取りモードである。

【０２２９】これらのブック原稿読み取りモード、及び、シート原稿読み取りモードは、それぞれさらに細分化されたモードを持っている。すなわち、ブック原稿読

33

み取りモードには、図３７に示すように、ブック原稿９２のブックサイズを自動的に認識する自動ブックサイズ認識モードと、操作表示ボード３１３のキー入力でブックサイズを指定するブックサイズキー入力モードとがある。

【０２３０】これらの、自動ブックサイズ認識モード、及び、ブックサイズキー入力モードでは、何れも、原稿面を上向きに見開いた状態でブック原稿９２をセットする見開き読み取りモードにより、ブック原稿９２に対する自動ページめくり読み取り動作が行われる。

【０２３１】さらに、このブック原稿読み取りモードは、その読み取り方式として、出力装置（特にプリンタ）と関連して、見開かれたブック原稿９２の左右２ページ分の原稿画像を連続して読み取り、この２ページ分の画像を１枚の転写紙上にプリントする見開き２ページ連続読み取りモードと、両面画像形成機能を有するプリンターを使用して、上述の見開き２ページ連続読み取りモードにより左右２ページ分の原稿画像がプリントされた１枚の転写紙の裏面に、見開かれたブック原稿９２の次の左右２ページ分の読み取り画像を連続して形成する、この見開き２ページ連続読み取りモードの両面モードと、見開かれたブック原稿９２の左右両ページの原稿画像を１頁ずつ区切って読み取り動作を行う見開き１ページ区ぎり読み取りモードと、この見開き１ページ区ぎり読み取りモードによりプリントされた転写紙の裏面に、上記両面画像形成機能を有するプリンタにより、見開かれたブック原稿９２の次の左右両ページの原稿画像を１頁ずつ区切って形成する、この見開き１ページ区切り読み取りモードの両面モードとを有している。

【０２３２】一方、シート原稿読み取りモードには、図３８に示すように、第１読み取りセンサユニット９、もしくは、第２読み取りセンサユニット１４の位置を固定した状態で、シート原稿２００を自動で給送・排出移動させながら、シートスルー方式により原稿画像の読み取りを行うシート原稿スルーモードと、原稿載置面１１６上にシート原稿２００をセット（定置）した状態で、ページめくり読取ユニット１の第１読み取りセンサユニット９を繰返し往復移動（スキャン）させて、シート原稿２００の読み取り動作を行うシート原稿スキャンモードと、自動原稿給送機能（ＡＤＦ）で原稿をセットできない（あるいはセットしない）場合に、手動で原稿をセットするシート原稿手動開閉モードとがある。

【０２３３】また、シート原稿スルーモードには、シート原稿２００の片面のみの画像を読み取る片面読み取りモードと、第１読み取りセンサユニット９及び第２読み取りセンサユニット１４で、シート原稿２００の両面の画像を同時に読み取る両面読み取りモードとがある。

【０２３４】さらに、この両面読み取りモードは、第１読み取りセンサユニット９及び第２読み取りセンサユニット１４を、互いに向き合った同一位置に配置して、原

34

稿画像の読み取りを行う同一位置読み取りモードと、第１読み取りセンサユニット９及び第２読み取りセンサユニット１４を、互いにずらした別位置に配置して、原稿画像の読み取りを行う別位置読み取りモードとを有している。　以上、ＭＦＤＳの動作モードについて説明してきたが、次に、図３９を参照しながら、上述した個々のモードの切り換えについて説明する。図３９において、ＭＦＤＳのメイン電源をＯＮすると、図３４で示したメイン制御ボード３１０、めくり搬送ベルト駆動制御ボード３１１、めくりユニット駆動制御ボード３１２、操作表示ボード３１３、第１画像処理ボード３１４、第２画像処理ボード３１７が、それぞれリセットされて初期設定が行われる。

【０２３５】その後、プリンタ３００等の接続機器の接続をチェックし、この接続機器に対応可能なモードを表示する一方、プリント置数、変倍率、ページめくり枚数、及び、各モード等を設定する各キーの入力を受け付ける。また、この間に、ブック原稿９２またはシート原稿２００のセットが行われる。ここで、シート原稿２００が、シート原稿トレイ９４にセットされた場合には、シート原稿センサ２５がＯＮされる。また、ここで、操作表示ボード３１３のオープンキー６２０によりＭＦＤＳの搬送部１９が開放され、その原稿載置面１１６の中央基準位置決め部２４に、ブック原稿９２が、見開かれた状態でセットされた場合には、ブック原稿センサ２７がＯＮされる。

【０２３６】これらのシート原稿センサ２５とブック原稿センサ２７のＯＮ／ＯＦＦの状態、及び、操作表示ボード３１３のテンキー６０２により入力されたシート原稿２００のプリント枚数を示すプリント置数に応じて、原稿画像の読み取りモードが、図３９に示すように切り替わる。

【０２３７】すなわち、ここで、ブック原稿センサ２７がＯＦＦ、シート原稿センサ２５がＯＮで、且つ、プリント置数が「１」の場合には、シート原稿スルーモードへ遷移する。また、ここで、ブック原稿センサ２７がＯＦＦ、シート原稿センサ２５がＯＮで、且つ、プリント置数が「２」以上の場合には、シート原稿スキャンモードへ遷移する。

【０２３８】さらに、ブック原稿センサ２７、及び、シート原稿センサ２５が、両方共ＯＦＦの場合には、シート原稿手動開閉モードへ遷移する。また、ブック原稿センサ２７がＯＮで、シート原稿センサ２５がＯＦＦの場合には、ブック原稿読み取りモードに遷移する。

【０２３９】さらに、ここで、ブック原稿センサ２７、及び、シート原稿センサ２５が、両方共ＯＮの場合には、異常処理１（警告ブザーＯＮ、及び、エラー表示）を行って、ユーザーに注意を促した後、ブック原稿読み取りモードに遷移する。

【０２４０】このようにして、それぞれ選択されたモー

ドサブルーチンへ入った後、操作表示ボード３１３のスタートキー６００が押されていなければ、画像読み取り動作を実行せずに、キー入力を受け付けるステップにリターンされる。

【０２４１】次に、上述した、各モードの動作について、説明する。先ず、図４０を参照して、ブック原稿読み取りモードについて説明する。ＭＦＤＳのモードが、ブック原稿読み取りモードに入ると、スキャンカットオフセンサ３４のＯＮ／ＯＦＦのチェックが行われる。

【０２４２】このスキャンカットオフセンサ３４は、図４に示したように、ページめくり読取ユニット１が、その上限位置まで上昇したときに、その上限検知部７６を検知してＯＮされる。ページめくり読取ユニット１は、前述したように、原稿載置面１１６上に載置されたブック原稿９２の厚さに応じて上下動される。このページめくり読取ユニット１の上昇、すなわち、ブック原稿９２の厚さが厚くなるに従って、これを駆動するめくり搬送ベルト８のテンションが高くなる。

【０２４３】従って、このめくり搬送ベルト８のテンションが高くなりすぎると、つまり、ブック原稿９２が厚すぎてページめくり読取ユニット１が上昇しすぎると、このテンションがブレーキとなって、ページめくり読取ユニット１のスキャンができなくなる。

【０２４４】スキャンカットオフセンサ３４は、このような、ブック原稿９２が厚すぎて、ページめくり読取ユニット１がスキャンできないレベルを検知している。このスキャンカットオフセンサ３４がＯＮの場合には、異常処理２（警告ブザーのＯＮと、”ブック原稿が厚すぎます”のエラー表示）を行って、ユーザーが無理に搬送部１９を閉じて、この搬送部１９を破損することがないようにしている。

【０２４５】ここで、ブック原稿９２の厚さが適応レベル以下の場合には、ユーザーにより搬送部１９が閉じられることによって、搬送部ロックセンサ３１がＯＮされる。このとき、搬送部１９が開いたまま、すなわち、搬送部ロックセンサ３１がＯＦＦのままであれば、異常処理３（警告ブザーのＯＮと、”搬送部を閉じて下さい”の表示）が行われる。

【０２４６】そして、この搬送部１９が閉じられると、ブックサイズ上限センサ３３のＯＮ／ＯＦＦのチェックが行われる。ここで、ブックサイズ上限センサ３３が、ブック原稿９２のブックサイズを検知してＯＮされている場合、すなわち、原稿載置面１１６にセットされているブック原稿９２のブックサイズが、ページめくり読取ユニット１の読み取り領域を越えた読み取り不可能なブックサイズの場合には、異常処理４（警告ブザーのＯＮと、”ブック原稿のサイズが大きすぎます”の表示）が行われる。

【０２４７】その後、プリント置数、変倍率、めくり枚数のキー入力の有無のチェックが行われた後、自動ブックサイズ認識モードの場合には、プレスキャンフラグがセットされ、自動ブックサイズ認識モードでない場合には、ブックサイズ入力モードとなり、操作表示ボード３１３により所定のキー入力を行ってブックサイズを設定することにより、ブック原稿９２の読み取り領域が決定される。

【０２４８】そして、スタートキー６００が押されることにより、搬送部ロック装置１４０が作動して、搬送部１９が原稿台１８にロックされる。これにより、ページめくり読取ユニット１によるページめくり動作中に、ユーザーが間違って搬送部１９を開放して、ブック原稿９２を破損するような事態が回避される。

【０２４９】次いで、この搬送部ロック装置１４０の作動後、前述したプレスキャンフラグがセットされている場合には、ページめくり読取ユニット１のプレスキャン動作が実行される。ここで、プレスキャンフラグがセットされていない場合には、このプレスキャン動作がスキップされる。

【０２５０】このプレスキャン動作後は、接続機器（プリンタ３００）の準備が整うまで待機される。そして、この接続機器（プリンタ３００）の準備が整った段階で、接続機器（プリンタ３００）から出力される読み取り開始信号を受けると、ページめくり読取ユニット１が駆動されて、前述したように、ブック原稿９２の原稿画像の読み取り動作が開始される。

【０２５１】このページめくり読取ユニット１による読み取り動作は、前に設定されているプリント置数に応じた回数だけ、繰り返して行われる。そして、この所定回数の読み取り動作が完了すると、次ページの原稿画像の読み取り動作を行うべく、前述したように、ページめくり読取ユニット１によって、ページめくり動作が実行される。

【０２５２】このようにして、ブック原稿９２の原稿画像の読み取り動作、及び、ページめくり動作は、それぞれ、予め設定された置数に応じて、その最終ページがめくられるまで、繰り返し実行される。そして、予め設定された最終めくりページのページめくり動作が完了し、且つ、最終読み取りページに対する所定回数の読み取り動作が完了すると、この最終読み取りページのページめくり動作を行わずに、ページめくり読取ユニット１が、そのホームポジション位置〔１Ａ〕に復帰されて、このブック原稿読み取りモードルーチンがリターンされる。

【０２５３】ブック原稿読み取りモードの基本動作は、以上の図４０に示すフローチャートの通りであるが、このモード中の前述した、見開き２ページ連続読み取りモードと、その両面モード、及び、見開き１頁区切り読み取りモードと、その両面モードは、図４１のキー入力セット時に受け付けた入力モードとなる。

【０２５４】これらの各入力モードについては、スタートキー６００が押されてからの経過を示すタイミングチ

ャートを参照して説明する。先ず、図４１に示すタイミングチャートを使用して、見開き２ページ連続読み取りモードについて説明する。

【０２５５】図４１において、操作表示ボード３１３のスタートキー６００が押されて、読み取りスタートスイッチＳＷがＯＮされると、第１読み取りセンサユニット９内のシャッタ（図示せず）が閉じる。このシャッタの内面は、白色基準板になっていて、これにより、第１読み取りセンサユニット９のシェーディング補正が行われる。

【０２５６】このシェーディング補正は、ページめくり読取ユニット１の立上り時に毎回行われ、ページめくり読取ユニット１の移動速度が一定速度になる前に完了される（この部分のタイミングチャートは図示せず）。このページめくり読取ユニット１のスタートは、メイン制御ボード３１０から、めくりユニット駆動制御ボード３１２へ送られる正転スタート信号によって行われる。

【０２５７】この正転スタート信号により、ページめくり読取ユニット１が、プレスキャンを開始する。ここで、ページめくり読取ユニット１の移動速度が一定速度Ｖｆに立ち上がった後は、前述のシャッタが既に開かれており、第１ＣＣＤ３１５により、ブック原稿９２の原稿画像の読み取りが開始されている。

【０２５８】このページめくり読取ユニット１のプレスキャン時の読み取りにより、見開かれたブック原稿９２の端部を画像処理でエッジ検出し、このときのめくりローラ２に取付けられたエンコーダ１５２の出力をカウントすることによって、原稿載置面１１６にセットされたブック原稿９２のブックサイズを認識して、ページめくり読取ユニット１の原稿画像の読み取り領域、及び、ページめくり領域を決定する。

【０２５９】このように、このＭＦＤＳは、ブック原稿９２が原稿載置面１１６のセンターを基準としてセットされるように構成されているので、このセットされたブック原稿９２の左端部のみを検出することによって、そのページめくり読取ユニット１の原稿画像の読み取り領域、及び、ページめくり領域を算出することができる。

【０２６０】従って、このＭＦＤＳでは、ブック原稿９２のブックサイズ検知に際して、ページめくり読取ユニット１を全面プレスキャンさせる必要がなく、ページめくり読取ユニット１をショートプレスキャンさせるだけで、原稿載置面１１６上に載置されたブック原稿９２のブックサイズを検知することができる。

【０２６１】これにより、このページめくり読取ユニット１のプレスキャン時に、ブック原稿９２の読み取り開始ページを誤ってページめくりすることがなくなる。

【０２６２】このブック原稿９２のブックサイズデータは、メイン制御ボード３１０から、外部接続機器としてのプリンタ３００に送信される。

【０２６３】一方、このブックサイズ検知が完了し、そのブックサイズデータがプリンタ３００に送信されている間に、メイン制御ボード３１０から、めくりユニット駆動制御ボード３１２に対して、めくりユニット駆動モータ６０を逆転させる逆転信号が与えられ、ページめくり読取ユニット１が速度Ｖｒの速さで、そのホームポジション位置〔１Ａ〕に向けて復帰移動される。

【０２６４】なお、このＭＦＤＳの原稿台１８の原稿載置面１１６を、有彩色、例えば、黄色などのような、ブック原稿としてあまり使用されていないような有彩色で着色しておくことにより、この原稿載置面１１６と、ブック原稿９２の原稿画像の読み取り領域との、領域識別精度を高めることができるので、上述のプレスキャン時におけるブックサイズ検知をより正確に行うことができ、そのブックサイズデータの信頼性が向上される。

【０２６５】また、ここで、原稿載置面１１６を、例えば、灰色などの中間色で着色しておけば、第１読み取りセンサユニット９の第１ＣＣＤ３１５として、カラーセンサを使用せずに、プレスキャン時の領域識別が可能となる。上述のプレスキャンが終了して、ページめくり読取ユニット１がホームポジション位置〔１Ａ〕に退避した後、プリンタ３００の準備ができて、プリンタ３００からメイン制御ボード３１０に対し、データの転送を要求する転送要求信号が与えられると、これにより、メイン制御ボード３１０から、めくりユニット駆動制御ボード３１２に対して、めくりユニット駆動モータ６０を正転させる正転信号が与えられ、ページめくり読取ユニット１の正転動作が開始される。

【０２６６】このページめくり読取ユニット１は、前述したように、めくりローラ２のエンコーダ１５２の出力がフィードバックされることによって、ブック原稿９２の原稿面に沿って、速度Ｖｆの速さに定速度制御されて、そのエンドポジション位置〔１Ｃ〕に向けて移動される。

【０２６７】このとき、第１読み取りセンサユニット９のＬＥＤ３１６は、既に点灯されており、この第１読み取りセンサユニット９の第１ＣＣＤ３１５により、ブック原稿９２の原稿情報の読み取りが開始されている。

【０２６８】ここで、図１に示すように、原稿載置面１１６上に見開かれてセットされたブック原稿９２の綴じ部（センター部）付近は、原稿面の反りがきついため、この綴じ部付近の原稿情報を正確に読み取ることが困難となる。また、このブック原稿９２の綴じ部付近には、通常、文字や画像等の原稿情報が形成されることがない。

【０２６９】そこで、このＭＦＤＳでは、図４１に示すように、このブック原稿９２の綴じ部付近において、第１ＣＣＤ３１５の読み取りＳＦＧＡＴＥがＯＦＦされ、この綴じ部付近の原稿情報の読み取りがマスクされるようになっている。

【０２７０】この原稿情報の初期設定時におけるマスク

39

領域は、前述したように、中央基準位置決め部２４のセンターより＋１０mm、－１０mmとなるように設定されているが、このマスク領域は、操作表示ボード３１３のブック綴じ部マスク領域設定キー６０８のキー入力によって、その設定値を変更し得るようになっており、特殊な装丁のブック原稿にも対応できるようになっている。

【０２７１】ところで、第１読み取りセンサユニット９によるブック原稿９２の原稿情報の読み取りは、ページめくり読取ユニット１がそのエンドポジション位置〔１Ｃ〕に至るまで行われるが、このページめくり読取ユニット１が、ブック原稿９２のセンター付近に到達した時点で、第１高圧電源３２０が周波数ｆ１でＯＮされる。これにより、めくり搬送ベルト８のブック原稿９２の右側のページに当接する部分に、前述した電荷パターンが形成される。

【０２７２】このめくり搬送ベルト８は、上述のように、ページめくり読取ユニット１が駆動されているときには駆動されず、ブック原稿９２を押える働きをしている。このようにして、ページめくり読取ユニット１が、そのエンドポジション位置〔１Ｃ〕まで駆動されて、ブック原稿９２の右端部の読み取り領域までの原稿情報の読み取りが終了すると、第１高圧電源３２０の出力がＯＦＦされるとともに、メイン制御ボード３１０からめくりユニット駆動制御ボード３１２に対して、めくり信号が与えられ、ページめくり読取ユニット１によるブック原稿９２のページめくり動作が開始される。

【０２７３】このとき、この第１読み取りセンサユニット９の読み取った原稿面が、予め設定されたブック原稿９２の最終読み取りページに相当している場合には、メイン制御ボード３１０からめくり信号が出力されず、このページめくり読取ユニット１は、このブック原稿９２の最終読み取りページの読み取りを終えた後、上述のページめくり動作を行わずに復帰移動されて、そのホームポジション位置〔１Ａ〕に退避される。

【０２７４】ブック原稿９２のページめくり動作時において、ページめくりセンサ２９がＯＮされるまでの間は、ページめくり読取ユニット１の逆転動作がスロースタートされ、めくり搬送ベルト８上に静電的に吸着されたブック原稿９２の読み取り終了ページ（図１における右ページ）が、そのページ収納部７内に導かれる。

【０２７５】そして、ページめくり読取ユニット１が、そのホームポジション位置１－Ａに向けて、さらに復帰移動され、このブック原稿９２の読み取り終了ページのセンター部分までが、ページ収納部７内にめくり込まれる。

【０２７６】また、このページめくり読取ユニット１のページめくり動作時においては、めくり搬送ベルト８の原稿吸着領域の電荷パターンを除電するために、第１高圧電源３２０が周波数ｆ２でＯＮされる。

40

【０２７７】このようにして、ページめくり読取ユニット１のページ収納部７内にめくり込まれたブック原稿９２の読み取り終了ページは、ページめくり読取ユニット１が、ブック原稿９２のセンター部を越してから、そのエンドポジション位置〔１Ｃ〕に至る間に、ページ収納部７内から排出されて、そのページめくり動作が完了する。

【０２７８】このページ排出時におけるページめくり読取ユニット１のリターン速度Ｖｒmは、その原稿読み取り時における速度Ｖｆよりも大きくなるように、すなわち、Ｖｒm＞Ｖｆとなるように設定されており、このページめくり動作の高速化が図られている。

【０２７９】この１連のページめくり動作の状態は、ページめくりセンサ２９により検知されている。すなわち、このＭＦＤＳは、ページめくり読取ユニット１の各位置におけるページめくりセンサ２９のＯＮ／ＯＦＦがチェックされることにより異常の有無が検出され、異常がある場合に、異常処理動作が実行される。

【０２８０】この異常処理動作としては、警告ブザーが作動されるとともに、例えば、ページ収納部７へ原稿ページをめくり込めない場合には、”ページめくり不能”、ページ収納部７にめくり込まれた原稿ページを排出できない場合には、”ページ重ね不能”等の表示が、操作表示ボード３１３に表示され、さらに、このページめくり動作が停止される。

【０２８１】また、この異常が発生したページめくり状態をユーザーが確認できるようにするために、このページめくり状態の異常発生箇所が、操作表示ボード３１３に表示される。

【０２８２】以下、ブック原稿９２の２ページ目以降の原稿画像の読み取り、及び、ページめくり動作は、プリンタ３００からの転送要求信号によって、順次スタートされ、上述と同様の読み取り動作、及び、ページめくり動作が、予め設定された最終読み取りページに至るまで、繰り返し実行される。

【０２８３】ここで、図４１のタイミングチャートでは、原稿のプリント置数を「１」とした場合、すなわち、ブック原稿９２の各読み取りページを、それぞれ１回読み取る毎に、この読み取りを終えたページのめくり動作を実行する場合について示したが、この原稿のプリント置数が「２」以上に設定されている場合には、第１読み取りセンサユニット９による第１回目の原稿読み取り動作が完了した時点で、メイン制御ボード３１０からめくりユニット駆動制御ボード３１２に対して、めくり信号の代わりに逆転信号が送信され、ページめくり読取ユニット１が速度Ｖｒの速さで、そのホームポジション位置〔１Ａ〕に向けてリターンされ、このリターン動作が終えると同時に、このページめくり読取ユニット１の第２回目の原稿読み取り動作がスタートされる。

【０２８４】そして、このようなページめくり読取ユニ

41

ット1の原稿読み取り動作が、予め設定された原稿のプリント置数の回数分だけ、繰り返し実行され、この動作の回数とプリント置数とが一致した時点で、初めて、メイン制御ボード310からめくりユニット駆動制御ボード312に対して、めくり信号が送信され、上述したページめくり動作が実行される。

【0285】以下、ブック原稿92の2ページ目以降の原稿画像の読み取り、及び、ページめくり動作は、プリンタ300からの転送要求信号によって、順次スタートされ、上述と同様のプリント置数に応じた回数の読み取り動作、及び、ページめくり動作が、予め設定された最終読み取りページに至るまで、繰り返し実行される。

【0286】また、ここで、プリント変倍キー614により、読み取った原稿画像が変倍されるように設定されている場合には、第1読み取りセンサユニット9のスキャン速度Vfが、その変倍率に応じた速度に可変されることによって、この原稿画像の副走査方向の変倍がなされるとともに、第1画像処理ボード314によって、この原稿画像の主走査方向の変倍処理がなされる。

【0287】一方、上述のような、見開き2ページ連続読み取りモードの両面モードにおいては、プリンタ300が原稿画像の両面プリントを行なうために、このプリンタ300からのメイン制御ボード310への転送要求信号の送信タイミングが多少遅れる点は異なるが、それ以外の動作に関しては、前述した見開き2ページ連続読み取りモードと同様の動作が実行される。

【0288】次に、図42を参照して、見開き1ページ区切り読み取りモードについて説明する。この見開き1ページ区切り読み取りモードは、見開かれたブック原稿92の左ページと、右ページとを別々に読み取って、画像形成を行なうモードで、これらの各画像を別々の転写紙にプリントするモードと、1枚の転写紙の表裏にプリントするモード（この見開き1ページ区切り読み取りモードの両面モード）とがある。

【0289】また、このモードにおいては、ブック原稿92の左ページから読み取る動作と、右ページから読み取る動作との、何れか一方の動作を、操作表示ボード313の読み取り開始ページ選択キー603のキー入力により選択設定することができる。　図42は、ブック原稿92の向かって左ページから読み取りを開始する場合のタイミングチャートを示している。

【0290】この見開き1ページ区切り読み取りモードの動作は、前述した見開き2ページ連続読み取りモードの両面モードの動作と略同じであり、この見開き2ページ連続読み取りモードの両面モードの動作と異なる点としては、プリンタ300からメイン制御ボード310への転送要求信号が、1ページ毎に送信される点であって、これにより、ページめくり読取ユニット1が、ブック原稿92の左ページから右ページに向けて等速で移動（スキャン）される。

42

【0291】図43は、この見開き1ページ区切り読み取りモードにおいて、ブック原稿92の右ページから読み取りを開始するように、セットされている場合のタイミングチャートを示している。

【0292】このタイミングチャートから明らかなように、ブック原稿92の右ページから読み取りが開始される場合には、このブック原稿92の読み取りが行なわれない左ページに対するページめくり読取ユニット1の移動速度が、右ページに対する第1読み取りセンサユニット9のスキャン速度Vfよりも速い速度となるようにプログラムされていて、原稿読み取り時間の短縮化、すなわち、読み取り機能の性能アップが図られている。

【0293】また、この見開き1ページ区切り読み取りモードにおいて、読み取りスキップページ設定キー618により、スキャンをせずに読み飛ばす（スキップする）ページが設定されている場合には、この読み取りスキップページに対するページめくり読取ユニット1が、読み取りを行なう時の第1読み取りセンサユニット9のスキャン速度Vfよりも速い速度で移動されて、直ちに、この読み取りスキップページのめくり動作が実行されるようにプログラムされている（タイミングチャートは図示せず）。

【0294】ところで、一般的には、図42及び図43に示したように、プリンタ300からメイン制御ボード310に対して、通常状態で連続して転送要求信号を要求できるプリンタは少ない。

【0295】そこで、このような一般的なプリンタを使用して、この見開き1ページ区切り読み取りモードを実施する場合には、このプリンタ側において、2枚の転写紙を重連するように給紙させることによって実現させることができる。

【0296】また、ここで、転写紙と感光体ドラムとのレジストの関係上、これらの2枚の転写紙間に、どうしても、ある程度の距離を設けなければならないような場合には、ブック原稿92の左ページ側の画像データはそのままプリンタに流し、ブック原稿92の右ページ側の画像データは、ディレーメモリを通して、これら転写紙間の距離に相当する時間だけ、プリンタへの転送タイミングを遅延させることにより実現させることができる。

【0297】この後者の場合の一例のブロック図を図44に、また、そのタイミングチャートを図45に示す。この例では、メイン制御ボード310上の内部の各ゲートA，Bにより、切り換え器を通してプリンタへそのまま出力される画像データに対して、ディレーメモリを通してプリンタへ出力される画像データが、時間Tdだけ遅延されるように構成されている。

【0298】次に、この見開き1ページ区切り読み取りモードにおける両面モードについて説明する。先ず、見開かれたブック原稿92の左右2ページの画像データを1枚の転写紙の表裏にプリントする場合について説明す

る。

【０２９９】この場合には、プリンタ３００側において、転写紙を反転搬送する必要があるため、その左ページの画像データのプリントを終えてから、右ページの画像データのプリントを開始するまでに、多少の時間がかかる。

【０３００】このため、このモードでは、ページめくり読取ユニット１がブック原稿９２の左ページの読み取りを終えて、その綴じ部付近に到達した時点で、このページめくり読取ユニット１の駆動が一旦停止される。

【０３０１】そして、プリンタ３００側での左ページ画像のプリント、及び、この転写紙の反転搬送が完了し、プリンタ３００からメイン制御ボード３１０に転送要求信号が出されると、停止されていたページめくり読取ユニット１が再び所定のスキャン速度Ｖｆで駆動されて、ブック原稿９２の右ページの読み取りが行なわれ、この右ページの画像データがプリンタ３００に転送されて、表面に左ページ画像のプリントされた転写紙の裏面に、この右ページ画像がプリントされて、両面プリントが行なわれる。この時の、タイミングチャートを図４６に示す。

【０３０２】次に、この見開き１ページ区切り読み取りモードにおいて、プリントされた転写紙の表裏と、ブック原稿９２の原稿面の表裏との関係を、一致させて画像形成する場合の両面モードについて説明する。このモードにおいて、ブック原稿９２の左ページから読み取りを開始する場合には、前述したように、プリンタ３００に対して、２枚連続給紙を行なうように指示し、左ページ画像のプリントされた転写紙はそのまま排紙させる一方、右ページ画像のプリントされた転写紙は反転搬送させて、その裏面にブック原稿９２の次ページの左ページ画像をプリントするための準備を行なっておく。

【０３０３】この間に、ＭＦＤＳ側においては、読み取りを終えた原稿ページのページめくり動作が行なわれる。このＭＦＤＳ側でのページめくり動作が完了されると、次ページに対する読み取り動作が実行され、前に、右ページ画像がプリントされて反転搬送された転写紙の裏面に、読み取られた左ページ画像がプリントされて排出される。

【０３０４】また、この時読み取られた右ページ画像は、新たに給紙された転写紙にプリントされる。この転写紙は、先の転写紙と同様に、反転搬送されて、その裏面にブック原稿９２の次ページの左ページ画像をプリントするための準備が行なわれる。このような１連の動作が繰り返し実行されることにより、ブック原稿９２と全く同様なページ構成の原稿画像がプリントされた転写紙が得られる。以上が、ブック原稿読み取りモードにおける動作の一例である。

【０３０５】ここで、見開き読み取りモードにおけるブック原稿のセットについて説明する。　図３４の接続機器チェックにおいて、プリンタ３００が裏面排紙可能な場合には、次のような表示が行なわれる。

【０３０６】すなわち、”ブック原稿が横書き（左開き）ならば、読み取りたい先頭ページを見開いて正面上向きにセットして下さい。”　”ブック原稿が縦書き（右開き）ならば、読み取りたい先頭ページを見開いて天地逆上向きにセットして下さい。”のように、ブック原稿の横書き、縦書きに対する、それぞれのセット方法が指示される。

【０３０７】これに対し、図３４の接続機器チェックにおいて、プリンタ３００が表面排紙のみ可能な場合には、次のような表示が行なわれる。

【０３０８】すなわち、”ブック原稿が横書き（左開き）ならば、読み取りたい最終ページを見開いて天地逆上向きにセットして下さい。”　”ブック原稿が縦書き（右開き）ならば、読み取りたい最終ページを見開いて正面上向きにセットして下さい。”のように、ブック原稿の横書き、縦書きに対する、それぞれのセット方法が指示される。

【０３０９】このＭＦＤＳでは、上述のような指示に従って、ブック原稿９２をセットすることにより、これに接続されるプリンタ３００の機能に関わらず、排出される転写紙のページ揃えを行なうことができる。また、このＭＦＤＳにおけるページめくり枚数の入力方法には、操作表示ボード３１３のキー入力による２通りの入力方法があり、これらの入力方法の内の何れか１つを、ユーザーが好みによって選択できるようになっている。　この入力方法の一つは、読み取り総ページ設定キー６０６等を使用して、読み取りたい総ページ数を設定する方法であり、他の入力方法は、読み取り開始ページ設定キー６０４、読み取り最終ページ設定キー６０５等を使用して、読み取りたい先頭ページと、最終ページを入力する方法である。

【０３１０】このＭＦＤＳにおけるページめくり枚数の入力は、上記の何れの方法で行なうにせよ、ページめくり読取ユニット１のページめくり回数を正確に算出できればよい。以下、上述の各方法におけるページめくり回数の算出方法について説明する。　先ず、読み取りたい総ページ数を入力した場合について説明する。

【０３１１】ここで、読み取りたい総ページ数をＸとし、めくり回数をＭとすれば、左ページから読み取る場合には、

（Ｘ－２）／２＝Ｍ＋余り・・・①

右ページから読み取る場合には、

（Ｘ－１）／２＝Ｍ＋余り・・・②

となり、これらの式よりＭの値を算出する事により、そのめくり回数が求められる。

【０３１２】次に、読み取りたい先頭ページと、最終ページを入力した場合について説明する。ここで、読み取りたい先頭ページをＹ、最終ページをＺととし、読み取

45

りたい総ページ数をXとすれば、

X＝Z－Y＋1・・・③

となり、この③式を、前記の②式、③式に代入することにより、めくり回数Mを算出する事ができる。

【0313】ところで、このMFDSは、前述したように、1台の装置で、ブック原稿と、シート原稿とを選択的に読み取って、その画像形成を行うことができるように構成されている。ブック原稿に関する画像読み取りモードは、上述した通りであり、以下、このMFDSにおけるシート原稿の画像読み取りモードについて説明する。

【0314】このシート原稿の読み取りモードには、図38に示したように、シート原稿スルーモードと、シート原稿スキャンモードと、シート原稿手動開閉モードとがある。

【0315】先ず、シート原稿スルーモードについて説明する。このシート原稿スルーモードには、片面原稿読み取りモードと、両面原稿読み取りモードとがあり、この両面原稿読み取りモードに、同一位置読み取りモードと、別位置読み取りモードとがある。

【0316】これらのモードの選択は、図47に示すように行われる。先ず、片面原稿読み取りモードでは、図48、及び、図49に示すように、シート原稿トレイ94に、シート原稿200が下向きにセットされ、この状態で、スタートキー600が押されると、先ず、めくり搬送ベルト駆動モータ61の電源がONされ、駆動ローラ12が回転されて、めくり搬送ベルト8が回動される。

【0317】このとき、各センサと、各入力データに異常がなければ、給紙クラッチ128がONされ、第1ベルト支持ローラ97に取付けられた第2給紙プーリ130から給紙駆動ベルト127を介して、給紙ローラ96が回転され、シート原稿200が給紙分離パッド95に向けて搬送される。

【0318】この給紙分離パッド95によって、最下位の1枚だけのシート原稿200が他のシート原稿から分離され、第1搬送ガイド108及び第2搬送ガイド109に案内されながら、めくり搬送ベルト8に接する位置まで搬送される。

【0319】ここで、この第2搬送ガイド109には、給紙センサ26が取付けられており、この給紙センサ26によって、シート原稿200の後端が検知されされた後、給紙クラッチ128がOFFされるように、シート原稿200の搬送タイミングが設定されている。

【0320】一方この間に、このめくり搬送ベルト8の回転とともに、第2バイアスローラ11に交流電源から交流の高電圧が印加されて、めくり搬送ベルト8にストライプ状の電荷パターンが形成される。これにより、このめくり搬送ベルト8によって、給紙されたシート原稿200が吸着されて搬送される。このときのめくり搬送

46

ベルト8の線速は、等倍時では360mm／sに、変倍時ではその設定倍率に応じて変化されるようになっている。

【0321】一方、このモードでは、ページめくり読取ユニット1が、そのエンドポジション位置〔1C〕に位置しており、その第1読み取りセンサユニット9によって、めくり搬送ベルト8により搬送されたシート原稿200の原稿情報が順次読み取られる（画素密度は400dpi）。

【0322】そして、この読み取りを終えたシート原稿200は、第3搬送ガイド110及び第4対向ローラ103，第5対向ローラ104，第6対向ローラ105，第7対向ローラ106により、挟持搬送されて、排紙口117から排紙トレイ23上に排紙される。

【0323】この排紙口117の近傍の第3搬送ガイド110には、排紙センサ28が取付けられていて、この排紙されるシート原稿200の排紙ジャムが検知されている。　そして、1枚目のシート原稿200の読み取り動作が完了されると、次に、所定の給紙タイミングで、給紙クラッチ128が再びONされ、2枚目のシート原稿200が給紙されて、上述と同様な読み取り動作が実行される。

【0324】このようにして、シート原稿トレイ94にセットされたシート原稿200が順次給送されて読み取られ、最後の（最上位の）シート原稿200が給紙されて、シート原稿センサ25がOFFすると、第2バイアスローラ11の電源が高周波交流電圧に切り替えられ、めくり搬送ベルト8の電荷パターンが除電されて、この最終のシート原稿200が排紙された後、MFDSの全ての動作が停止される。以上の読み取り動作は、このMFDSに裏面排紙機能を有するプリンタが接続されている場合の動作であり、これにより、シート原稿200及びプリントされた転写紙のページ揃え排紙が実現される。

【0325】ここで、このMFDSに接続されているプリンタが表面排紙機能しか有していない場合には、シート原稿トレイ94上に、シート原稿200を上向きにセットして、第2読み取りセンサユニット14によって、このシート原稿200の読み取りを行うことにより、上述した裏面排紙機能を有するプリンタの場合と同様に、シート原稿200及びプリントされた転写紙のページ揃え排紙が実現される。また、この表面排紙機能を持ったプリンタが接続されている場合におけるその他の動作は、裏面排紙機能を有するプリンタが接続されている場合と同様である。

【0326】次に、このシート原稿スルーモードにおける両面原稿読み取りモードについて説明する。この両面原稿読み取りモードでは、横書きのシート原稿が、その原稿先端から下向きにして、シート原稿トレイ94上にセットされる。

47

【0327】これは、第1読み取りセンサユニット9及び第2読み取りセンサユニット14がメモリーを持っていないため、その主走査方向でのミラー反転しか用いることができないことによる。

【0328】先ず、この両面原稿読み取りモードにおける同一位置読み取りモードの基本動作は、図50、及び、図51に示すように、片面原稿読み取りモードの場合と同様であるが、このモードでは、ページめくり読取ユニット1が、そのホームポジション位置〔1A〕に位置されて、原稿台18がわの第2読み取りセンサユニット14の位置と同一位置で、シート原稿200の表裏両面の原稿情報の読み取りが、これらの第1読み取りセンサユニット9、及び、第2読み取りセンサユニット14によって、同時に平行して実行される。

【0329】一方、この両面原稿読み取りモードにおける別位置読み取りモードの基本動作は、図52、及び、図53に示すように、片面原稿読み取りモードの場合と同様であるが、このモードでは、ページめくり読取ユニット1が、そのエンドポジション位置〔1C〕に位置されている。

【0330】ここで明らかなように、この別位置読み取りモードでは、第1読み取りセンサユニット9と、第2読み取りセンサユニット14との読み取り位置の間隔が、最大サイズの原稿の長さよりも大きくなるため、これらの第1読み取りセンサユニット9及び第2読み取りセンサユニット14によ読み取られたシート原稿200の表裏両面の原稿情報が、時系列的に出力される。

【0331】次に、シート原稿読み取りモードにおけるシートスキャンモードについて説明する。このシートスキャンモードでは、図54、及び、図55に示すように、シート原稿トレイ94に、シート原稿200が上向きにセットされ、この状態で、スタートキー600が押されると、先ず、めくり搬送ベルト駆動モータ61の電源がONされ、駆動ローラ12が回転されて、めくり搬送ベルト8が回動される。

【0332】このとき、各センサと、各入力データに異常がなければ、給紙クラッチ128がONされ、第1ベルト支持ローラ97に取付けられた第2給紙プーリ130から給紙駆動ベルト127を介して、給紙ローラ96が回転され、シート原稿200が給紙分離パッド95に向けて搬送される。

【0333】この給紙分離パッド95によって、最下位の1枚だけのシート原稿200が他のシート原稿から分離され、第1搬送ガイド108及び第2搬送ガイド109に案内されながら、めくり搬送ベルト8に接する位置まで搬送される。

【0334】ここで、この第2搬送ガイド109には、給紙センサ26が取付けられており、この給紙センサ26によって、シート原稿200の後端が検知されされた後、給紙クラッチ128がOFFされるように、シート

48

原稿200の搬送タイミングが設定されている。

【0335】一方この間に、このめくり搬送ベルト8の回転とともに、第2バイアスローラ11に交流電源から交流の高電圧が印加されて、めくり搬送ベルト8にストライプ状の電荷パターンが形成される。

【0336】これにより、このめくり搬送ベルト8によって、給紙されたシート原稿200が吸着されて搬送される。このときのめくり搬送ベルト8の線速は、360mm／sに設定されており、シート原稿200の先端がホームポジション位置〔1A〕に到達したときに、めくり搬送ベルト8の回動が停止されるようになっている。

【0337】一方、このモードでは、ページめくり読取ユニット1が、そのホームポジション位置〔1A〕に位置しており、めくり搬送ベルト8の回動が停止された後に、ページめくり読取ユニット1がめくりユニット駆動モータ60によって、そのエンドポジション位置〔1C〕に向けて移動されながら、その第1読み取りセンサユニット9によって、めくり搬送ベルト8により搬送されたシート原稿200の原稿情報が順次読み取られる（画素密度は400dpi）。

【0338】この原稿情報の読み取り時には、第1バイアスローラ3に、第1高圧電源320により高周波交流電圧が印加され、めくり搬送ベルト8に形成されていた電荷パターンが除電される。これにより、ページめくり読取ユニット1の復帰移動時におけるシート原稿200のページめくり読取ユニット1内への進入が防止されている。このようにしてシート原稿200の読み取りを終えたページめくり読取ユニット1は、そのホームポジション位置〔1A〕に向けてリターンされる。

【0339】また、このページめくり読取ユニット1のリターン時には、第1読み取りセンサユニット9がシート原稿200の原稿面から上方に退避されるとともに、第1バイアスローラ3に第1高圧電源320から高圧交流電圧が印加されて、めくり搬送ベルト8にストライプ状の電荷パターンが形成されて、このシート原稿200がめくり搬送ベルト8に吸着されて固定される。

【0340】そして、上述のような動作がその設定回数だけ繰り返された後、めくり搬送ベルト駆動モータ61がONされて、めくり搬送ベルト8が回動され、この読み取りを終えたシート原稿200が、排紙口117から排紙トレイ23上に排紙される。

【0341】この排紙口117の近傍の第3搬送ガイド110には、排紙センサ28が取付けられていて、この排紙されるシート原稿200の排紙ジャムが検知されている。そして、1枚目のシート原稿200の読み取り動作が完了されると、次に、所定の給紙タイミングで、給紙クラッチ128が再びONされ、2枚目のシート原稿200が給紙されて、上述と同様な読み取り動作が実行される。

【0342】このようにして、シート原稿トレイ94に

セットされたシート原稿２００が順次給送されて読み取られ、最後の（最上位の）シート原稿２００が給紙されて、シート原稿センサ２５がＯＦＦすると、第２パイアスローラ１１の電源が第２高圧電源３２１の高周波交流電圧に切り替えられ、めくり搬送ベルト８の電荷パターンが除電されて、この最終のシート原稿２００が排紙された後、ＭＦＤＳの全ての動作が停止される。

【０３４３】次に、このシート原稿スルーモードにおけるシート原稿手動開閉モードについて説明する。このシート原稿手動開閉モードの動作は、図５６、及び、図５７に示すように、上述したシート原稿スキャンモードの動作における、シート搬送手段、及び、シート吸着用の各電源をそれぞれＯＦＦの状態にし、オペレータが手操作により、シート原稿の入れ替えが行われる。

【０３４４】以下、このＭＦＤＳにおけるページめくり読取ユニット１の操作制御について説明する。図５８に、このＭＦＤＳにおけるページめくり読取ユニット１の操作制御回路を示す。

【０３４５】この操作制御回路は、ページめくり読取ユニット１の往復駆動制御、及び、その速度制御を行っており、めくりユニット駆動制御ボード３１２に組み込まれている。図５８において、マイクロコンピュータ５２０（以下、単にマイコンという）は、このＭＦＤＳのモード制御、及び、シーケンス制御も行っている（詳細は図示せず）。

【０３４６】このようなマイコン５２０としては、例えば、μＰＤ７１０５４Ｇによるプログラマブルインターバルタイマ５２１（以下、単にタイマという）が接続されている。このタイマ５２１は、マイコン５２０の制御により、めくりユニット駆動モータ６０（直流モータ）の速度制御を行うためのパルス幅変調ＰＷＭ出力を送出するためのものである。

【０３４７】このＰＷＭ制御の周期は、５０（μsec）であり、これを４００ビットの分解能で制御する。このタイマ５２１には、８ＭＨｚの発振器５２２が接続され、クロック信号が与えられるように構成されている。また、めくりユニット駆動モータ６０は、マイコン５２０に対し、駆動用トランジスタＱ１～Ｑ４を介して接続されている。

【０３４８】すなわち、トランジスタＱ１，Ｑ４がＯＮで、トランジスタＱ２，Ｑ３がＯＦＦの状態で、めくりユニット駆動モータ６０には、時計方向（ＣＷ）に回転する電流が供給され、トランジスタＱ２，Ｑ３がＯＮで、トランジスタＱ１，Ｑ４がＯＦＦの状態で、めくりユニット駆動モータ６０には、反時計方向（ＣＣＷ）に回転する電流が供給される。

【０３４９】ここで、めくりユニット駆動モータ６０が、時計方向（ＣＷ）に回転すると、ページめくり読取ユニット１は往動され、めくりユニット駆動モータ６０が、反時計方向（ＣＣＷ）に回転すると、ページめくり

読取ユニット１は復動されるように設定されている。

【０３５０】このめくりユニット駆動モータ６０の回転方向は、マイコン５２０のポートＰＦ６，ＰＦ７からそれぞれ出力されるＣＷ信号、及び、ＣＣＷ信号により制御される。また、めくりローラ２には、その回転に従ってパルスを発生させるエンコーダ１５２が直結されている。

【０３５１】ここで、このエンコーダ１５２は、めくりユニット駆動モータ６０の回転量、及び、回転方向に応じて、位相の異なる２つのパルス信号を発生する。１つは、Ａ相エンコーダパルスＥＮＣＡであり、他の１つは、Ｂ相エンコーダパルスＥＮＣＢである。Ａ相エンコーダパルスＥＮＣＡは、分周マルチプレクサ５２４を介して、マイコン５２０のカウンタインプット端子Ｃ１に入力されている。

【０３５２】これにより、マイコン５２０は、Ａ相エンコーダパルスＥＮＣＡのパルス間隔が、マイコン５２０の内部のカウンタ（マイコン５２０の発振器５２５の発振周波数１０ＭＨｚにより規制される）によって計測される。

【０３５３】また、このカウンタインプット端子Ｃ１への入力信号は、割込み入力となっており、割込みプログラムの処理中に、Ａ相エンコーダパルスＥＮＣＡのパルス間隔の測定データの値を読み、このデータに基づいて、めくりユニット駆動モータ６０の回転数の算出、比例・積分制御演算によるモータ制御量の算出、並びに、出力（タイマ５２１へのデータロード）等が行われる。

【０３５４】具体的には、Ａ相エンコーダパルスＥＮＣＡの出力を目標速度に応じて分周マルチプレクサ５２４により、１，２，４，８分周することにより、カウンタインプット端子Ｃ１に割込み入力信号が与えられている。

【０３５５】ここで、１分周時には、第１読み取りセンサユニット９が、エンコーダ１５２の１パルスによって、０．１１６ｍｍ移動することにより、その速度が割込み間隔によりマイコン５２０の内部で演算される。そして、この算出された速度データに基づいて比例・積分演算処理により、出力タイマ値が決定される。

【０３５６】また、Ａ相エンコーダパルスＥＮＣＡ、及び、Ｂ相エンコーダパルスＥＮＣＢは、フリップフロップ５２６を介して、マイコン５２０の入力端子ＰＣ３に入力され、両者間の位相差検知に供されて、その位相差によりめくりユニット駆動モータ６０の回転方向が決定される。

【０３５７】つまり、Ａ相エンコーダパルスＥＮＣＡの立上り時における、Ｂ相エンコーダパルスＥＮＣＢの状態がマイコン５２０のポートに入力されることによって、めくりユニット駆動モータ６０の回転方向が判断される。

【０３５８】次に、めくりユニット駆動モータ６０の速

51

度制御について説明する。このめくりユニット駆動モータ６０の速度制御はＰＷＭ制御によって行われる。先ず、ページめくり読取ユニット１のスキャナ走査時、すなわち、めくりユニット駆動モータ６０の時計方向への回転時には、トランジスタＱ１をＯＮさせる一方、タイマ５２１からのＰＷＭ出力により、ゲート５２７を介して、トランジスタＱ４をＯＮ／ＯＦＦさせ、めくりユニット駆動モータ６０の両端子間に電位差を生じさせて、ＰＷＭ信号のデューティ比に応じた速度でこのめくりユニット駆動モータ６０を回転させる。

【０３５９】一方、ページめくり読取ユニット１のリターン時には、上述の場合と逆に、トランジスタＱ３をＯＮさせるとともに、タイマ５２１からのＰＷＭ出力により、ゲート回路５２８を介して、トランジスタＱ２をＯＮ／ＯＦＦさせ、めくりユニット駆動モータ６０の両端子間に逆向きの電位差を生じさせて、ＰＷＭ信号のデューティ比に応じた速度でこのめくりユニット駆動モータ６０を回転させる。

【０３６０】以下、上述のように構成されたＭＦＤＳでの著作権管理支援システムについて説明する。先ず、図５９により、ＭＦＤＳのメイン制御ボード３１０の内部の構成、および機能について説明する。

【０３６１】図５９において、ワンチップマイクロコンピュータ３３０（以下、単にマイコンという）は、内部ＲＡＭ、内部ＲＯＭ、Ｉ／Ｏ、タイマ、外部・内部割込み、シリアルインターフェース等を含んだ構成となっている。また、アドレスバス、データバスにより、外部ＲＯＭ３３１、外部ＲＡＭ３３２をはじめとする外部デバイスをアクセスすることができるようになっている。

【０３６２】外部ＲＯＭ３３１は、マイコン内部ＲＯＭと同様に動作プログラムが組み込まれている。外部ＲＡＭ３３２は、外部デバイスへの設定データ等をストアすることができ、バッテリーバックアップ用電池３３４により、装置の電源オフ時でもメモリできるようにバックアップされている。

【０３６３】シリアルインターフェース３３７，３３８，３３９は、外部ユニット、外部機器等とシリアル通信で接続されていて、ワンチップマイコンとコマンドおよびデータの送受信が可能となっている。ＩＤカード読み取り装置３４０は、個人や団体に発行されたカードに組み込まれている磁気データ等を読み取り、コード化してシリアルインターフェース３３８を介してマイコンに入力される。このデータにより、誰がブック原稿をコピーしたかを判断することができる。

【０３６４】バーコード読み取り装置３４１は、書籍や雑誌等のブック原稿に貼付または印刷されたバーコードパターンによって、ブック原稿の種類を識別するためのブック原稿識別装置である。

【０３６５】本実施例では、このブック原稿識別装置としてバーコード読み取り装置３４１を示したが、ブック

52

原稿に貼付した磁気パターンを識別する装置であってもよい。

【０３６６】このバーコード読み取り装置３４１は、バーコードスキャナとバーコードデータとからなっている。本例では、バーコードスキャナにＣＣＤを使用したタイプで説明するが、このバーコードスキャナとしては、反射光センサを使用したペンタイプや、レーザスキャンタイプのスキャナであってもよい。

【０３６７】このＣＣＤスキャナにより読み取られたバーコードデータは、そのバーコード信号がＡＳＣＩＩにコードに変換され、シリアル通信でメイン制御ボード３１０へ送信されるようになっている。操作表示ボード３１３は、モードの設定や表示、操作手順表示、エラー表示等を行なう。

【０３６８】メモリコントローラ３４２は、ワンチップマイコン３３０からのコマンドより画像処理ボード３１４からの画像データをプリンタへ直接出力するか、画像メモリボード３４３へ出力するかを切り換えたり、画像処理ボード３１４またはプリンタ３００からの主走査、副走査の同期信号およびゲート信号、クロック信号の切り換えを行なう。

【０３６９】また、メモリコントローラ３４２は、画像メモリボード３４３からの出力をコントロールできるように、主走査、副走査の同期信号（ＬＳＹＮＣ）およびゲート信号（ＦＧＡＴＥ）、クロック信号を発生することができる。さらに、メモリコントローラ３４２は、キャラクタージェネレータ３４４をコントロールする。

【０３７０】キャラクタージェネレータ３４４は、文字フォントや図形パターンのデータを内部ＲＯＭに持っていて出力可能になっている。このキャラクタージェネレータ３４４の出力タイミングは、メモリーコントローラ３４２によって指示され、その出力は、ＯＲ回路３４５を経てプリンタ３００に出力される。

【０３７１】ＯＲ回路３４５は、メモリコントローラ３４２から出力される画像データとキャラクタジェネレータ３４４から出力されるパターンデータとを合成することができる。画像メモリボード３４３は、原稿最大読み取りサイズ分のメモリを標準で２ページ分持っている。

【０３７２】また、この画像メモリボード３４３は、１ドット多値化出力の１ドット当りのビット数増分用およびプリンタジャム時のリカバリー用にメモリを増設できるように構成されている。

【０３７３】次に、本発明に関する動作モードについて説明する。前述したように、ＭＦＤＳの動作モードは、大きく２つのモードにわかれており、一方は、ブック原稿のページめくり読み取り動作を行なうブック原稿読み取りモードであり、他方は、シート原稿に対する読み取りを行なうシート原稿読み取りモード（ＡＤＦモードと圧板モード）である（図３８）。

【０３７４】本発明は、ブック原稿読み取りモードのう

53

ちで、著作権利用料の管理ができるようにＩＤカード認識モードとバーコード認識モードとを設定できるようになっている（図３６）。

【０３７５】ここで、各モードの設定のオン／オフは、使用者が簡単に変更できないようにするために、操作表示ボード３１３からの暗号キー入力によって設定するか、はじめから各モードが組み込まれたプログラムＲＯＭを使用して行なうようにしてもよい。その他の細分化されたモードは、前記の図３７を参照して述べた通りである。

【０３７６】次に、操作部からのキー入力による個々のモードの切り換えと、センサ入力による自動モード切り換えについて説明する。図６０において、ＭＦＤＳのメイン電源をオンすると、図３４で示したメイン制御ボード３１０、操作表示ボード３１３、めくりユニット駆動制御ボード３１２、めくり搬送ベルト駆動制御ボード３１１、各画像処理ボード３１４，３１７がリセットされ、初期設定が行なわれる。

【０３７７】この初期設定では、図３５に示す自動選択モードが設定される。この自動選択モードでは、各センサ入力によってモードを自動的に選択する。その後、プリンタ３００等の出力装置の接続をチェックし、プリンタ等の接続器機に対応可能なモードを表示する一方、置数枚、変倍、ページめくり枚数、モードのキー入力を受け付ける。

【０３７８】ここで、自動選択モードをオフすると、操作表示ボード３１３からキー入力でモード選択を設定できる。すなわち、操作表示ボード３１３のモード選択キーによりモードが指定されると、この操作表示ボード３１３からのシリアル通信により、モード選択コードが送信され、これによりモード選択フラグがセットされる。

【０３７９】そして、このモード選択フラグチェックにより各モードへの分岐（サブルーチンコール）が行なわれる。一方、自動選択モードのオン時には、ブック原稿またはシート原稿のセットにより、各センサ入力および設定条件によって、次に示すようにモードが分岐される。

【０３８０】シート原稿がシート原稿トレイ９４に置かれると、シート原稿センサ２５がオンされる。また、搬送部１９を開いて原稿台１８の中央位置基準決め部２４に、ブック原稿が見開いた状態でセットされると、ブック原稿センサ２７がオンされる。この２つのセンサ２５，２７により、図６０に示すようにモードが切り替わる。

【０３８１】すなわち、ブック原稿センサ２７がオフ、シート原稿センサ２５がオンで、置数枚が１枚の場合はシート原稿スルーモード（ＡＤＦモード）へ、置数枚が２枚以上の場合はシート原稿スキャンモード（ＡＤＦモード）へ、各センサ２５，２７が共にオフの場合はシート原稿手動開閉モード（圧板モード）へ、それぞれ切り

54

替わる。

【０３８２】また、ブック原稿センサ２７がオンで、シート原稿センサ２５がオフの場合はブック原稿読み取りモードへ、各センサ２５，２７が共にオンの場合は以上処理１を実行して、警告ブザーとエラー表示を行なってユーザに注意を促した後、ブック原稿読み取りモードへ遷移する。

【０３８３】ここで、ブック原稿読み取りモードは、著作権利用管理ができるモードであるのに対し、シート原稿読み取りモード（ＡＤＦモードと圧板モード）は、通常複写動作ができるようになっている。

【０３８４】すなわち、ブック原稿モードと圧板モードのみ（ＡＤＦモード無）の場合には、中央位置決め部２４のブック原稿センサ２７の状態を見てモードの判断がなされ、（１）ブック原稿センサ２７がオンでブック原稿モードの場合には、ブック原稿の画像の読み取りを行なう画像読み取りセンサユニット９の画像読み取り走査回数、または、この読み取りセンサユニット９によって読み取られた画像情報を出力するメモリコントローラ３４２の上記画像情報出力回数が、カウント手段によりカウントが実行され、（２）ブック原稿センサ２７がオフで圧板モードの場合には、上記のカウント手段による画像読み取りセンサユニット９の画像読み取り走査回数、または、メモリコントローラ３４２の上記画像情報出力回数のカウントが不実行となる。

【０３８５】また、それぞれモードサブルーチンへ入った後は、読み取りスタートキーが押されていなければ、処理を実行せずにリターンするようになっている。

【０３８６】それぞれのモード動作は、前述した通りである。

【０３８７】次に、本発明に関する、ブック原稿読み取りモードについて、図６１乃至図６３のフローチャートを用いて説明する。図６１乃至図６３において、ブック原稿読み取りモードに入ると、ベルトテンションセンサ３４のオン／オフのチェックを行なう。

【０３８８】これは、載置されるブック原稿が厚すぎて、めくり読み取りユニット１が原稿面をスキャンできないレベルを検知している。このベルトテンションセンサ３４がオンすると、異常処理２が実行され、ユーザが無理に搬送部１９を閉じて装置を破損することのないように、警告ブザーがオンするとともに、”本が厚すぎます”のエラー表示が行なわれる。

【０３８９】ここで、ブック原稿の厚さが適応レベル以下なら、ユーザにより搬送部１９が閉じられて、搬送部ロックセンサ３１がオンする。このとき、搬送部１９が開いたままであれば、異常処理３が実行され、警告ブザーがオンするとともに、”搬送部を閉じてください”のエラー表示が行なわれる。

【０３９０】このようにして搬送部１９が閉じられると、ブックサイズ上限検知センサ３３のオン／オフのチ

55

ェックを行ない、このセンサがオンの場合には、本のサイズが読み取り領域以上となるため、異常処理４が実行され、警告ブザーがオンするとともに、”本のサイズが多きすぎます”のエラー表示が行なわれる。

【０３９１】そして、ここで、ブックサイズ上限検知センサ３３がオフの場合は、置数枚、変倍、めくり枚数がセット済みであることを判断し、さらに、ＩＤカード認識モードであることを判断して、サブルーチンがコールされる。

【０３９２】ＩＤカード認識モードは、図６４に示すフローチャートに示すように動作される。このＩＤカード認識モードが実行されると、図６４に示すように、先ず、ＩＤカード読み取り装置３４０が接続されているか否かが、シリアルインターフェース３３８（図５９）を介して、ブレーク信号通のやり取りでチェックされる。

【０３９３】ここで、ＩＤカード読み取り装置３４０が接続されていない場合には、シリアルインターフェース３３７を介して、操作表示ボード３１３へ接続方法の指示表示を行なうようにコマンドを送信し、ＮＧフラグをセットして、ブック原稿読み取りモードへリターンする。

【０３９４】一方、ＩＤカード読み取り装置３４０が接続されている場合には、操作表示ボード３１３へＩＤカード読み取り装置３４０の操作方法の表示を行なうようにコマンドを送信する。

【０３９５】次に、ＩＤカードより個別コードが読み取られたかどうかのシリアル通信上でのコマンドのやり取りを行なう。その結果、個別コードの読み取りがまだ行なわれていない場合には、ＮＧフラグをセットしてリターンする。ここで、個別コードの読み取りが行なわれていると判定された場合には、そのＩＤコードがブック原稿の複写を許可しているか否かの判断を行なう。そして、ＩＤコードがブック原稿の複写を許可していない場合には、”ブック原稿の複写が許可されていません”等のエラー表示を行なうように、操作表示ボード３１３へコマンドを送信する。

【０３９６】一方、ＩＤコードがブック原稿複写を許可済の場合には、このＩＤコード別に設定されている外部ＲＡＭ３３２上のカウンタのアドレスをセレクトして、分岐もとルーチンへリターンする。この外部ＲＡＭ３３２上のカウンタは、ＩＤコード別にブック原稿の複写枚数をカウントする機能を有している。

【０３９７】上述したＩＤカード認識モードのリターン終了後は、ＮＧフラグのセット・リセットを判断する。ここで、ＮＧフラグがセットされている場合には、ＮＧフラグをリセット後、ブック原稿読み取りモードをリターンして終了する。

【０３９８】一方、ＮＧフラグがセットされていなければ、バーコード認識モードかを判断して、そのサブルーチンをコールする。このとき、バーコード認識モードでなければ、次のモードへ進む。

56

【０３９９】〔請求項１に対応する説明〕図６５に、上記のバーコード認識モードのフローチャートを示す。

【０４００】このサブルーチンでは、先ず、バーコード読み取り装置３４１の接続チェックを行なう。

【０４０１】ここで、バーコード読み取り装置３４１が接続されていない場合には、操作表示ボード３１３へエラー表示をするコマンドを送信し、ＮＧフラグをセットしてリターンする。

【０４０２】一方、ここで、異常がなければ、バーコードスキャナ内部のＬＥＤを点灯させ、同じくバーコードスキャナ内部のＣＣＤにより、ブック原稿に貼付または印刷されたバーコードの読み取りを行なう。

【０４０３】このバーコードスキャナの読み取り位置は、ブック原稿セット基準位置近辺に設定することにより、ブック原稿のサイズに左右されずに、図６６に示すようなブック原稿の背表紙部のバーコードパターン３４１ｃを読み取ることができる。そこで、本発明の実施例では、図６７および図６８に示すように、ブック原稿９２の中央綴じ部（背表紙部）を支持する原稿台１８の中央基準位置決め部２４のブック原稿セット基準位置側に、バーコード読み取り装置３４１のバーコードスキャナ３４１ａが取付けられている。

【０４０４】この実施例以外のバーコードスキャナ３４１ａの取り付け位置としては、読み取りめくり可能な最小ブック原稿サイズ以内の、ブック原稿の表紙または裏表紙の原稿セット基準位置の近傍でもよい。なお、図６８に示すページめくり読み取りユニット１は、めくりローラ２によってめくられた原稿ページを収納するためのページ収納部７と、原稿ページを世に取るための読み取りセンサユニット９とをブック原稿９２の載置面に沿うように、原稿読み取り走査方向に並列に配置した例を示している。

【０４０５】バーコードスキャナ３４１ａによって読み取られたバーコードパターン３４１ｃのデータは、バーコード読み取り装置３４１内のバーコードデコーダ３４１ｂに送られ、ＡＳＣＩＩコードに変換されて、メイン制御ボード３１０のシリアルインターフェース３３９を介してワンチップマイコン３３０に送られる。

【０４０６】ここで、バーコードパターン３４１ｃが読み取れない場合には、”バーコードパターンが読み取れません”のエラー表示をし、ＮＧフラグをセットしてリターンする。また、バーコードパターン３４１ｃが読み取れた場合には、このバーコードパターンが複写許可されているかどうかを判断する。そして、複写許可されていない場合は、操作表示ボード３１３へ”このブック原稿（バーコードパターンは複写（コピー）許可されていません”のエラー表示を行なう。

【０４０７】一方、複写許可されている場合には、読み取りコード別に設定されている外部ＲＡＭ３３２上のカ

ウンタのアドレス値をセレクトしてリターンする。この外部ＲＡＭ３３２上のカウンタは、バーコードパターン別に定められた、すなわち、例えば、出版社別、著作社別、あるいは、価格等によってウエイト付けされたカウンタである。

【０４０８】ブック原稿読み取りモードへリターン後は、ＮＧフラグを判断し、ＮＧフラグがセットされていれば、このＮＧフラグをリセットしてリターンする。また、ＮＧフラグがリセットならば、次の自動ブックサイズ認識モードに移行する。

【０４０９】自動ブックサイズ認識モードの場合には、プレスキャンフラグをセットし、そうでなければ、ブック原稿サイズ入力モードとなり、操作表示ボード３１３よりキー入力を行なって、ブック原稿のサイズを設定し、その読み取り領域を決定する。

【０４１０】ここで、読み取りスタートキーが押されている場合には、前述した搬送部ロック機構が作動する。これは、ブック原稿のページめくり動作中に、ユーザが間違って搬送部１９を開いて原稿ページを破損しないようにするためである。そして、この搬送部ロック機構の作動後に、プレスキャンフラグがセットされていれば、プレスキャン動作を行ない、プレスキャンフラグがセットされていなければ、スキップする。

【０４１１】このプレスキャン動作後、またはスキップ後は、接続機器（プリンタ３００）の準備ができるまで、待機される。そして、準備ができると、ブック原稿の読み取り動作が開始される。この読み取り動作の終了後、ＩＤコード・バーコード別カウント動作処理サブルーチンがコールされる。

【０４１２】このＩＤコード・バーコード別カウント動作処理サブルーチンでは、図６９に示すように、先ず、中央綴じユニットセンサ３４６の状態を判断する。図７０に示すように、中央綴じユニット３４７は、著作権の無いステープルされた原稿等を複写する場合に使用される。

【０４１３】すなわち、この中央綴じユニット３４７は、ステープルされた原稿をクリップ（挟持）した状態で、原稿台１８の中央基準位置決め部２４内にセットできるように、装置本体に対して着脱自在に構成されている。そして、この中央綴じユニット３４７が原稿台１８の中央基準位置決め部２４内にセットされると、その端部に形成された検知片３４７ａによって、中央綴じユニットセンサ３４６がオンされ、原稿台１８上にセットされた原稿がブック原稿でないことが判断される。

【０４１４】この中央綴じユニットセンサ３４６がオンの状態では、前記のカウント動作を必要としないので、直ちにリターンされる。ここで、中央綴じユニットセンサ３４６がオフの時には、先程サブルーチンコールされたＩＤカード認識処理（図６４）、バーコード認識処理（図６５）でコード別にセレクトしたカウンタ値をロードしてカウントアップし、さらに、もとのアドレス値にストアする。このときに使用される外部ＲＡＭ３３２は、バッテリーバックアップ用電池３３４によって、電源オフ時にもバックアップされるようになっている。

【０４１５】また、このときのカウントアップは、ブック原稿の読み取りページ数と対応するようにカウントする。すなわち、ブック原稿を２ページずつ読み取れば、２カウントアップずつカウントされる。

【０４１６】なお、上記の中央綴じユニットセンサ３４６には、光透過型センサを使用しているが、例えば、中央綴じユニット３４７の裏側に独自のバーコードパターンを貼付もしくは印刷し、この不正防止とコストダウンを兼ねたバーコードパターンをバーコードスキャナ３４１ａで読み取るようにして、ブック原稿と区別してもよい。

【０４１７】このようにして、上記のカウント値により、ブック原稿別に何ページの複写が実行されたかを判断することができる。そして、上記の読み取り動作を予め設定された置数枚分繰り返して、ページめくり動作を実行し、最終ページのめくり読み取り動作が完了するまでこの動作を繰り返す。これにより、ブック原稿の最終ページのページめくり読み取り動作が実行された後、ブック原稿読み取りモードをリターンして動作を終了する。

【０４１８】一方、前述したＩＤコード別、および、バーコード別にカウントしたデータは、操作部よりキー入力することで、随時、表示部へ出力することが可能となっている。また、このＭＦＤＳにプリンタ３００が接続してある場合には、操作表示ボード３１３からのキー入力により、メイン制御ボード３１０のワンチップマイコン３３０が、外部ＲＡＭ３３２にストアしているカウンタ値をコード別に呼び出して、メモリコントローラ３４２により発生される同期信号で制御されたキャラクタジェネレータ３４４によって、一覧表やグラフ等の画像出力とともに、カウントデータを出力させることが可能となる。

【０４１９】これにより、例えば、ＩＤコードカウンタ値に基づいて、個人、あるいは、団体別に著作権料を正確に徴収することができる。また、操作表示ボード３１３よりキー入力で、予め、１ページ複写時の代金を設定しておくことにより、上記のカウンタ値によって、複写終了時の合計金額を計算することも可能である。

【０４２０】さらに、バーコードパターン３４１ｃにより、ブック原稿別にカウントした値により、原稿１ページ当りの代金を予め入力しておけば、原稿の著作者や出版社別に、その支払金額を期間別に一覧表としてプリンタ３００へ出力することができる。

【０４２１】これらの出力は操作表示ボードのキー入力により随時出力できるが、メイン制御ボード内のマイコンのタイマ機能により定期的に出力をさせてもよい。

59

週、月極めにより自動出力させてキー入力の手間を省く事ができる。

【0422】ところで、このMFDSに接続されたプリンタ300側で、コピー用紙がジャムした場合には、このプリンタ300側からMFDSのメイン制御ボード310へ、シリアル通信によりジャム発生コードが送信される。

【0423】これによって、MFDS側は、プリンタジャムリカバリー動作を行なう。

【0424】このリカバリー動作の一部に、先のジャム発生コードの送信タイミングにより、図63、および図69で示したように、読み取り動作が終了して、既に、IDコード・バーコードカウント動作済みであるならば、マイナスカウントを行なう。 また、ここで、IDコード・バーコードカウント動作済みでなければ、このときのカウント値をそのまま保持する。

【0425】一方、本実施例では、ブック原稿読み取りモードルーチン内に、IDコード・バーコード別カウント動作処理サブルーチンを組み込んだ例を示したが、プリンタ300から送信されてくる排出OKコード（転写紙がプリンタ300から排出されたときにMFDSへ出力されるコード）を受信したときに、先のサブルーチンコールを実行するように組み込んでもよい。

【0426】このように、プリンタ300から送信されてくる排出OKコード（転写紙がプリンタ300から排出されたときにMFDSへ出力されるコード）を受信したときに、先のサブルーチンコールを実行するように組み込むことによって、ジャム時のマイナスカウントを行なう必要がなくなる。

【0427】ところで、上記実施例は、画像出力がリアルタイム処理される場合の例であるが、例えば、プリンタ300の処理速度が速い場合や、リピート複写の高速化および原稿保護のために、一旦、メモリに画像データを蓄えた後、このメモリから画像データを適時出力するように構成されている場合の実施例を以下に示す。

【0428】上記のように、例えば、図57において、画像メモリボード343にメモリコントローラ342を介してブック原稿の読み取り画像を一旦記憶し、この記憶された画像データをプリンタ300へ逐次出力する場合、MFDSのワンチップマイコン330がメモリコントローラ342に対してFGATE出力を要求する毎に、IDコード・バーコード別カウント動作サブルーチンを実行するように組み込むことにより、メモリリテンション時にも、先のカウント動作を行なうことができる。

【0429】〔請求項7に対応する説明〕また、上記実施例では、ブック原稿の背表紙部のバーコードパターン341cを読み取ることによって、複写の許可・不許可を判断したが、例えば、ブック原稿のページに貼られたり、押されたりしている、印紙、印鑑、シール等を読み

60

取り、そのパターンを認識することによって、そのブック原稿の複写の許可・不許可を判断することもできる。

【0430】このように、ブック原稿の印紙、印鑑、シール等を読み取り、そのパターンを認識することによって、そのブック原稿の複写の許可・不許可を判断する手段としては、画像メモリボード343に、ブック原稿の印紙、印鑑、シール等の読み取りデータをストアし、メモリ上の所定位置（所定アドレス）に複写許可マークがあるか無い可をパターンマッチングにより判断して、メモリからの複写許可あるいはカウント動作のするしないを決定する。

【0431】ここで、メモリボード343を使用しない場合には、プレスキャンにおいて、上記の判断を行なって、ブック原稿の読み取り動作の許可・不許可を決定するようにしてもよい。

【0432】〔請求項8に対応する説明〕ところで、このMFDSでは、読み取った画像情報をプリンタ300へ出力する場合、キャラクタジェネレータ344により、文字やマーク（読み取ったIDコードやバーコードパターン）等の付加情報をOR回路345によって、ブック原稿の読み取り画像データとともに、合成して出力することができる。これにより、著作権のあるブック原稿から複写したか否かが明らかとなり、誤った複写による著作権の更なる侵害を防止することができる。

【0433】例えば、著作権のあるブック原稿をじかに複写した場合には、このシステムによりその著作権料の徴収が可能となるが、このブック原稿のコピーを原本として複写した場合には、著作権料を算出するためのカウントができなくなる。

【0434】そこで、著作権のあるオリジナル原稿の複写時には、そのコピー画像として、読み取り画像データと複写禁止マークやコードとを予め合成して出力することによって、このようなブック原稿のコピーを原本とする複写時にも著作権料の徴収が可能となる。

【0435】また、本実施例では、IDカード認識手段により、ブック原稿の読み取りの許可・不許可を決定しているが、例えば、操作部からのパスワードや、暗証番号などのキー入力によって、読み取り動作の許可・不許可を決定してもよい。

【0436】さらに、ブック原稿の読み取りの許可・不許可や著作権料の徴収に際して、このMFDSにプリペイドカードの読み取り装置を接続することにより、ブック原稿の読み取り処理時に、著作権料を前払したプリペイドカードのポイントをダウンさせることによって、著作権料を徴収するようにしてもよい。

【0437】

【発明の効果】本発明によれば、許可された特定のブック原稿から、或いは、特定ユーザーによる画像読み取り（複写）のみを可能とすることにより、著作権問題への対処を図り得る画像読み取り装置を提供することができ

61

る。

【図面の簡単な説明】

【図１】本発明の実施に適するマルチ・ファンクション・ドキュメント・スキャナ（ＭＦＤＳ）の概略断面図である。

【図２】前記ＭＦＤＳの駆動系の概略横断面図である。

【図３】前記ＭＦＤＳの駆動系の概略平面図である。

【図４】前記ＭＦＤＳにおけるページめくり読取ユニットの端部の斜視図である。

【図５】前記ページめくり読取ユニットを構成するローラの支持構造を示すローラ端部の断面図である。

【図６】前記ＭＦＤＳにおける搬送部ロック装置のロック解除態様を示す側面図である。

【図７】前記搬送部ロック装置のロック開始作動態様を示す側面図である。

【図８】前記搬送部ロック装置のロック完了態様を示す側面図である。

【図９】前記ＭＦＤＳが搭載されたプリンタの外観を示す斜視図である。

【図１０】前記ＭＦＤＳが搭載されたプリンタの搬送部開放時の外観を示す斜視図である。

【図１１】前記ページめくり読取ユニット内に配設された第１読み取りセンサユニットの端部付近の斜視図である。

【図１２】前記第１読み取りセンサユニットの端部付近の側面図である。

【図１３】前記第１読み取りセンサユニットの端部の支持構造を示す部分拡大断面図である。

【図１４】前記ページめくり読取ユニットのめくりローラ２の奥側の側面図である。

【図１５】前記ＭＦＤＳにおける原稿載置面の中央基準位置決め部の構造を示す断面図である。

【図１６】前記ＭＦＤＳにおける操作表示ボードの平面図である。

【図１７】前記プリンタの概略断面図である。

【図１８】前記プリンタの書込部の平面図である。

【図１９】前記プリンタの転写紙搬送経路を切り換える切換爪の作動態様図である。

【図２０】前記ページめくり読取ユニットの概略断面図である。

【図２１】前記第１読み取りセンサユニットの構成を示す断面図である。

【図２２】前記ＭＦＤＳの作動態様を示す概略断面図である。

【図２３】前記ＭＦＤＳにおけるめくり搬送ベルトの説明図である。

【図２４】前記めくり搬送ベルトのページめくり動作を示す部分斜視図である。

【図２５】前記めくり搬送ベルトの搬送力のピッチ特性を示す線図である。

62

【図２６】前記めくり搬送ベルトの吸着力のピッチ特性を示す線図である。

【図２７】前記めくり搬送ベルトの搬送力の印加電圧特性を示す線図である。

【図２８】前記めくり搬送ベルトの吸着力の印加電圧特性を示す線図である。

【図２９】前記めくり搬送ベルトに不平等電荷を付与する他の手段の部分斜視図である。

【図３０】前記めくり搬送ベルトに不平等電荷を付与するさらに他の手段の部分斜視図である。

【図３１】前記めくり搬送ベルトに不平等電荷を付与するさらに他の手段の部分斜視図である。

【図３２】前記めくり搬送ベルトに不平等電荷を付与するさらに他の手段の部分斜視図である。

【図３３】前記めくり搬送ベルトに不平等電荷を付与するさらに他の手段の部分斜視図である。

【図３４】前記ＭＦＤＳの電装ブロック図である。

【図３５】前記ＭＦＤＳの著作権管理支援システムの動作モードの遷移図である。

【図３６】前記ＭＦＤＳの著作権管理支援システムの動作モードの遷移図である。

【図３７】前記ＭＦＤＳの著作権管理支援システムの動作モードの遷移図である。

【図３８】前記ＭＦＤＳの著作権管理支援システムの動作モードの遷移図である。

【図３９】前記各モードの切り換え動作を示すフローチャートである。

【図４０】前記ブック原稿読み取りモードの動作を示すフローチャートである。

【図４１】ブック原稿読み取りモードにおける見開き２ページ連続読み取りモードの動作を示すタイミングチャートである。

【図４２】ブック原稿読み取りモードにおける見開き１ページ区切り読み取りモードの動作を示すタイミングチャートである。

【図４３】前記見開き１ページ区切り読み取りモードにおいてブック原稿の右ページから読み取るようにセットした場合の動作を示すタイミングチャートである。

【図４４】前記見開き１ページ区切り読み取りモードにおいて読み取られたデータの転送タイミングを遅延させる回路のブロック図である。

【図４５】前記見開き１ページ区切り読み取りモードにおいて読み取られたデータの転送タイミングを遅延させる回路の動作を示すタイミングチャートである。

【図４６】前記見開き１ページ区切り読み取りモードにおける両面モードの動作を示すタイミングチャートである。

【図４７】前記シート原稿読み取りモードにおけるシート原稿スルーモードの切り換え動作を示すフローチャートである。

63

【図４８】前記シート原稿スルーモードにおける片面読み取りモードの動作を示すフローチャートである。

【図４９】前記片面読み取りモードの動作を示すタイミングチャートである。

【図５０】前記シート原稿スルーモードにおける同一位置読み取りモードの動作を示すフローチャートである。

【図５１】前記同一位置読み取りモードの動作を示すタイミングチャートである。

【図５２】前記シート原稿スルーモードにおける別位置読み取りモードの動作を示すフローチャートである。

【図５３】前記別位置読み取りモードの動作を示すタイミングチャートである。

【図５４】前記シート原稿読み取りモードにおけるシート原稿スキャンモードの動作を示すフローチャートである。

【図５５】前記シート原稿スキャンモードの動作を示すタイミングチャートである。

【図５６】前記シート原稿読み取りモードにおけるシート原稿手動開閉モードの動作を示すフローチャートである。

【図５７】前記シート原稿手動開閉モードの動作を示すタイミングチャートである。

【図５８】前記ページめくり読取ユニットの走査制御回路図である。

【図５９】前記ＭＦＤＳの著作権管理支援システムに関するブロック図である。

【図６０】前記著作権管理支援システムの動作モードの設定動作を示すフローチャートである。

【図６１】前記著作権管理支援システムのブック原稿読み取りモードのフローチャートである。

64

【図６２】前記著作権管理支援システムのブック原稿読み取りモードのフローチャートである。

【図６３】前記著作権管理支援システムのブック原稿読み取りモードのフローチャートである。

【図６４】前記著作権管理支援システムのＩＤカード認識モードのフローチャートである。

【図６５】前記著作権管理支援システムのバーコード認識モードのフローチャートである。

【図６６】前記ＭＦＤＳによって読み取られるブック原稿の斜視図である。

【図６７】本発明にかかる画像読み取り装置におけるバーコード読み取り装置を説明した概略斜視図である。

【図６８】本発明にかかる他の画像読み取り装置におけるバーコード読み取り装置を説明するための概略断面図である。

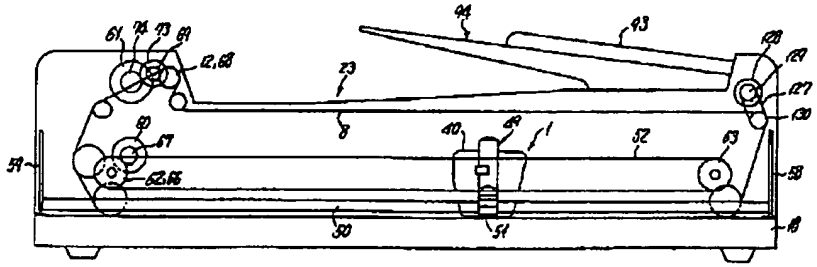
【図６９】前記著作権管理支援システムのＩＤコード・バーコード別カウント動作処理モードのフローチャートでである。

【図７０】本発明にかかる画像読み取り装置にセットされる中央綴じユニットを説明するための概略斜視図である。

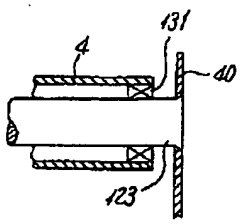
【符号の説明】

１	（読取手段としての）ページめくり読取ユニット
１ａ	（識別符号検知手段としての）バーコードスキャナ
１８	原稿台
９２	ブック原稿
３３０	（制御手段としての）ワンチップマイコン
３４１Ｃ	（識別符号としての）バーコードパターン

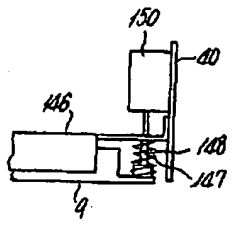
【図２】



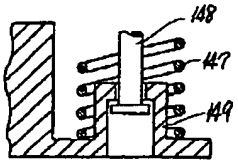
【図５】



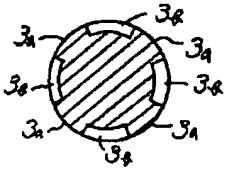
【図１２】



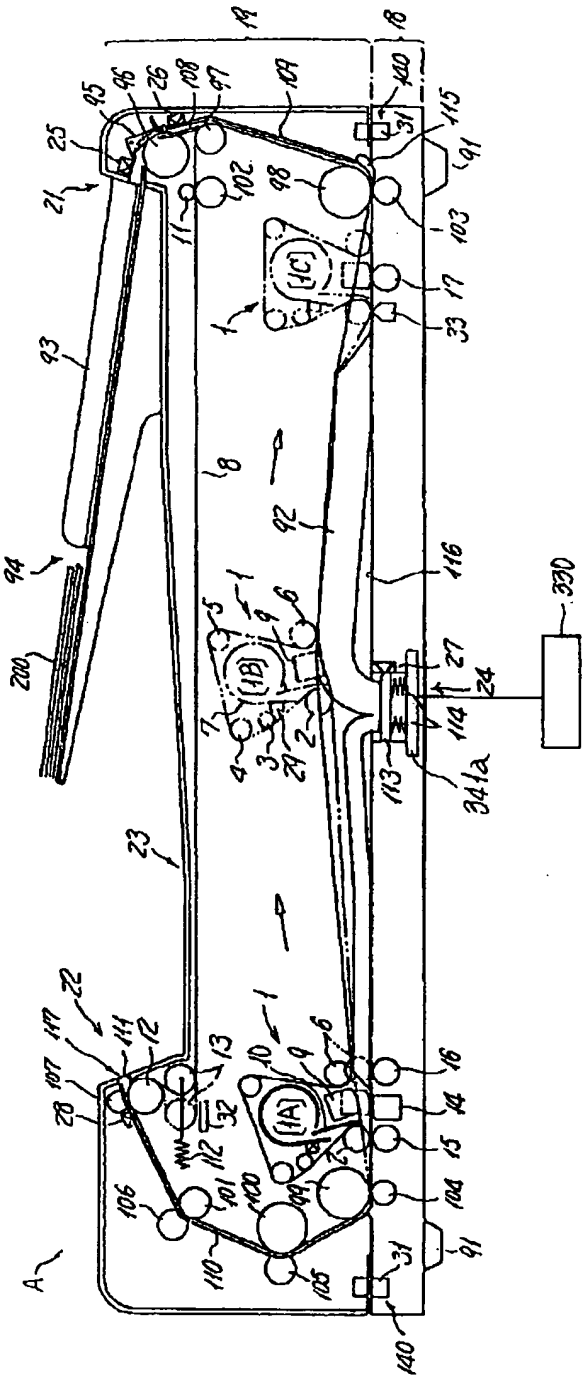
【図１３】



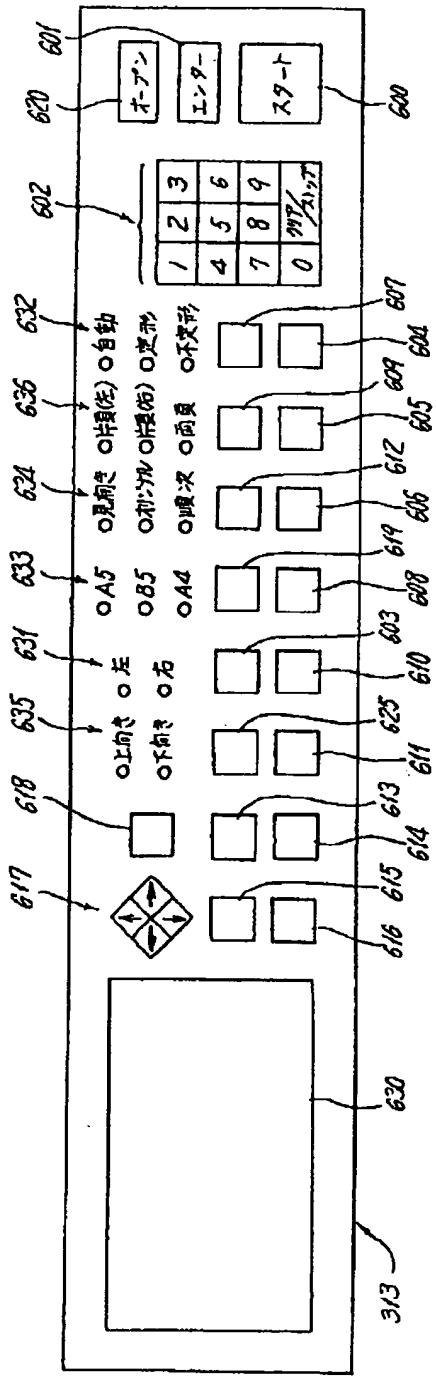
【図３０】



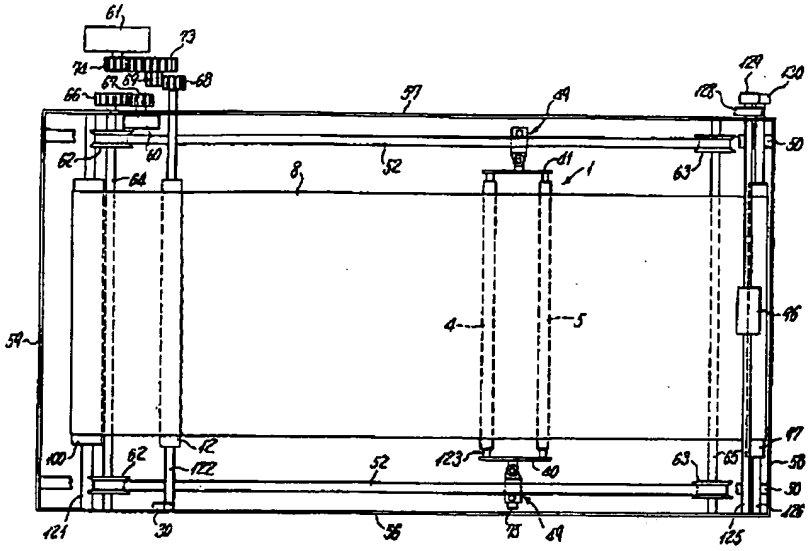
【図1】



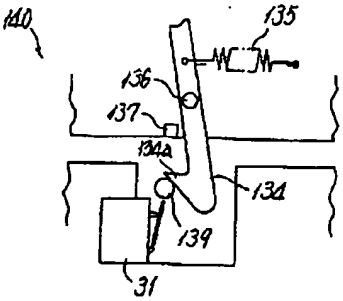
【図16】



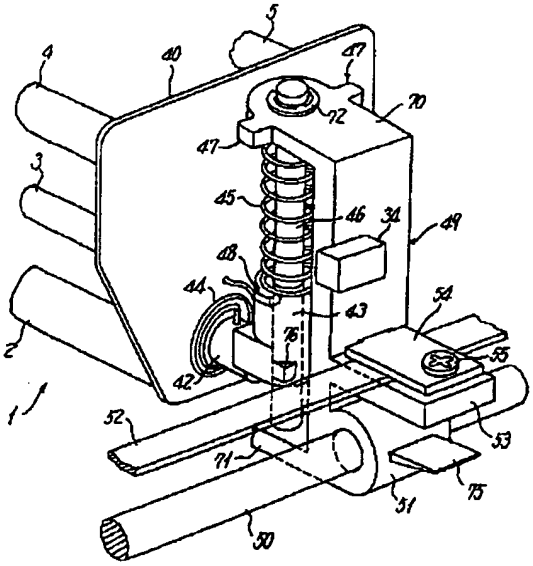
【図3】



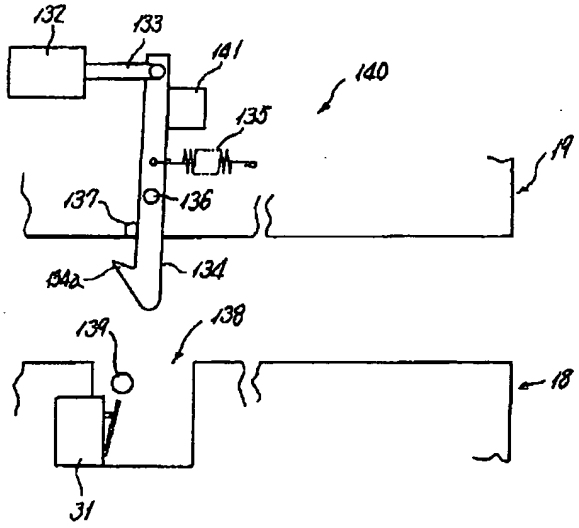
【図7】



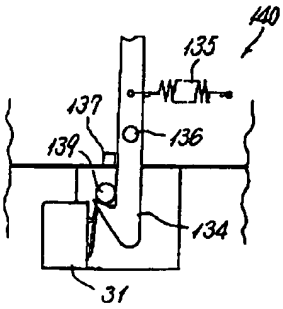
【図4】



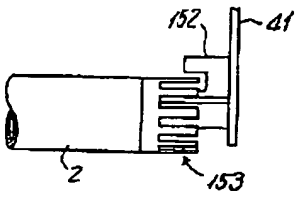
【図6】



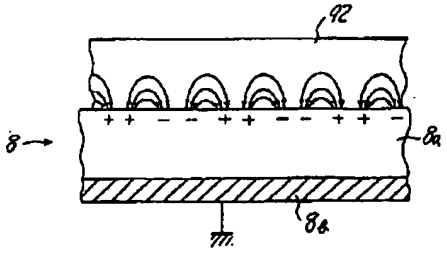
【図8】



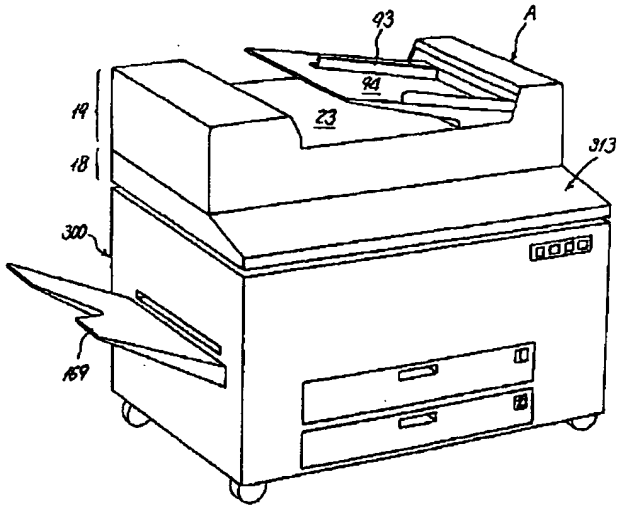
【図14】



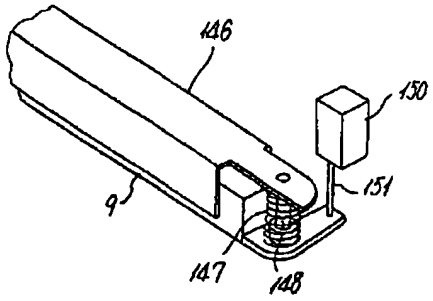
【図23】



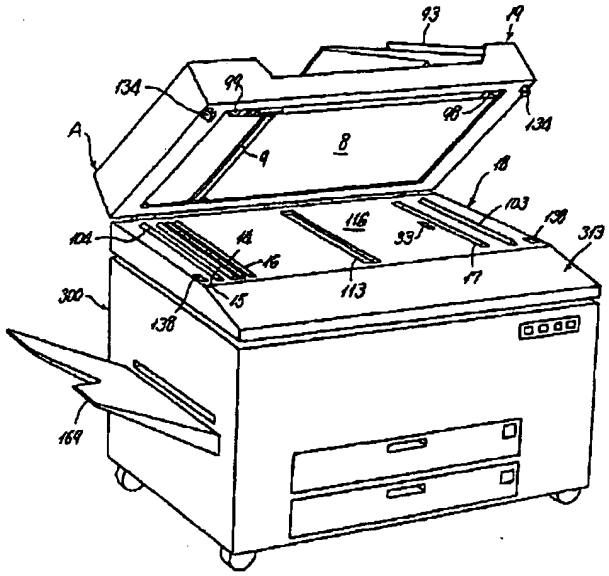
【図9】



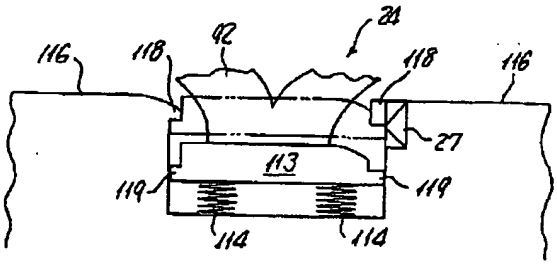
【図11】



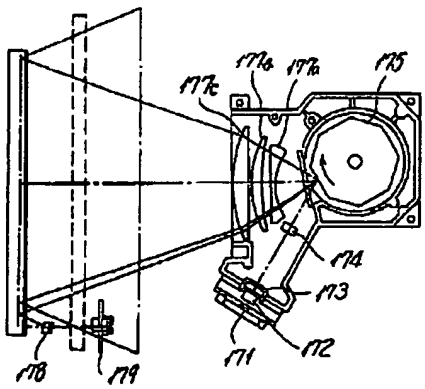
【図10】



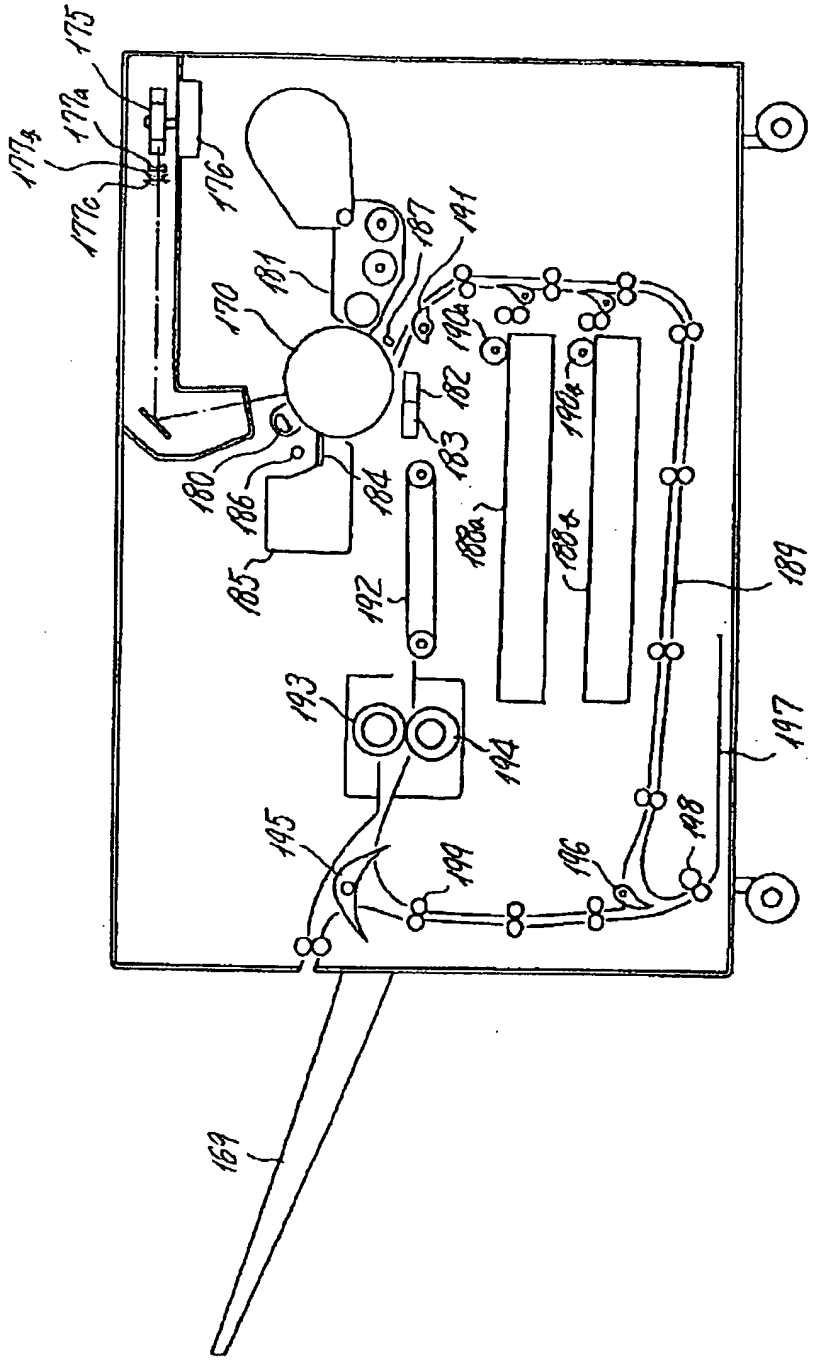
【図15】



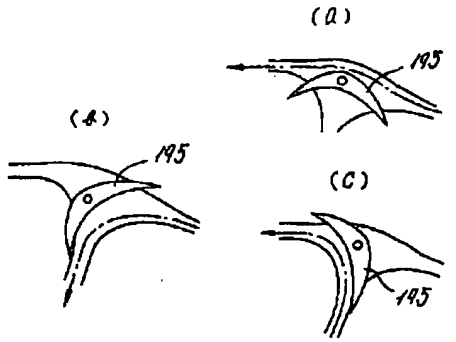
【図18】



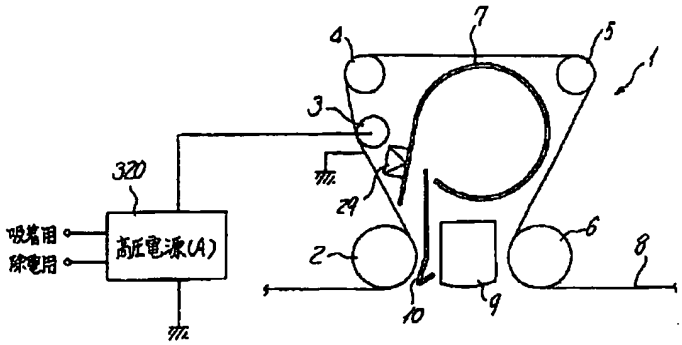
【図17】



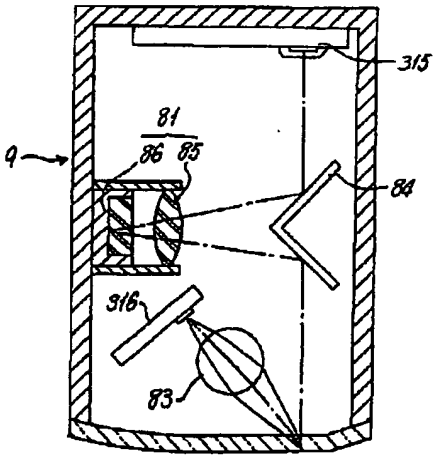
【図19】



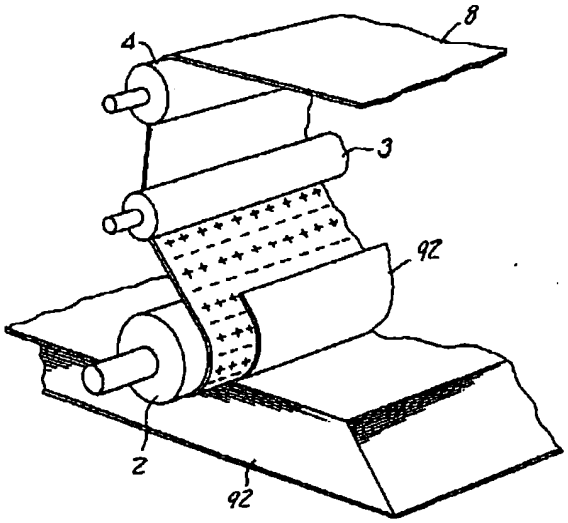
【図20】



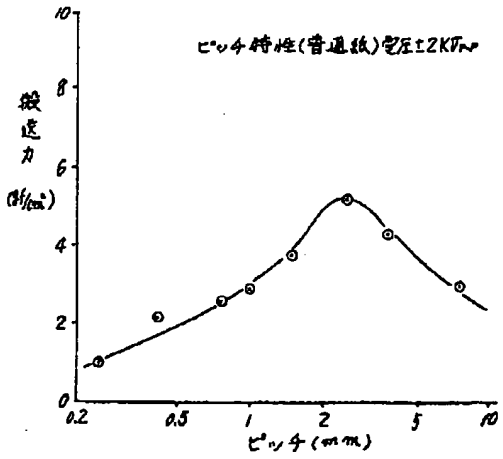
【図21】



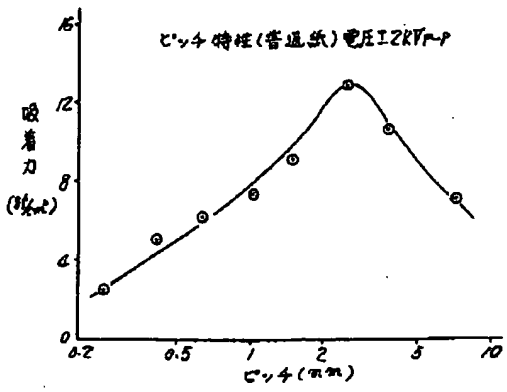
【図24】



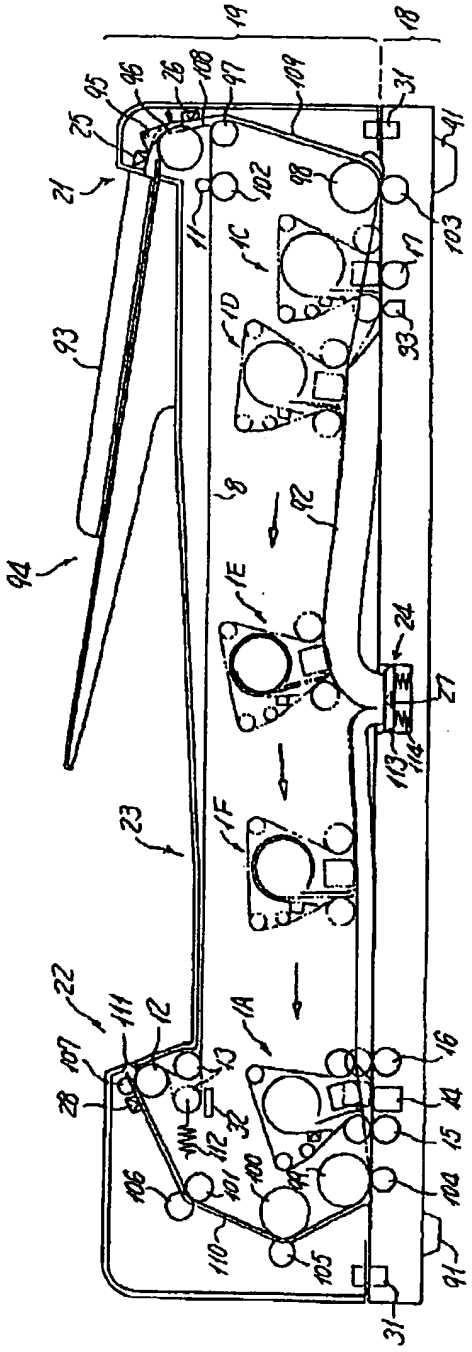
【図25】



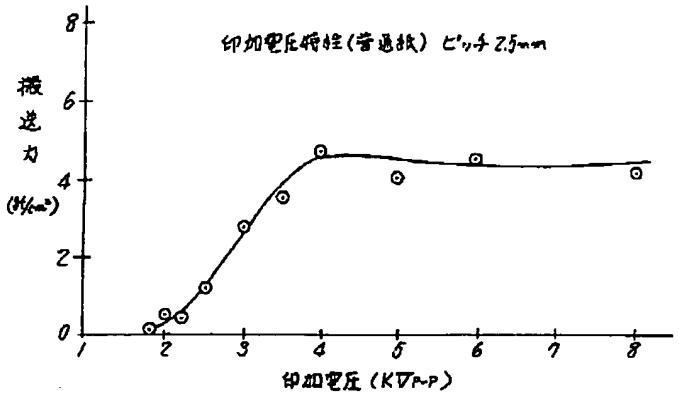
【図26】



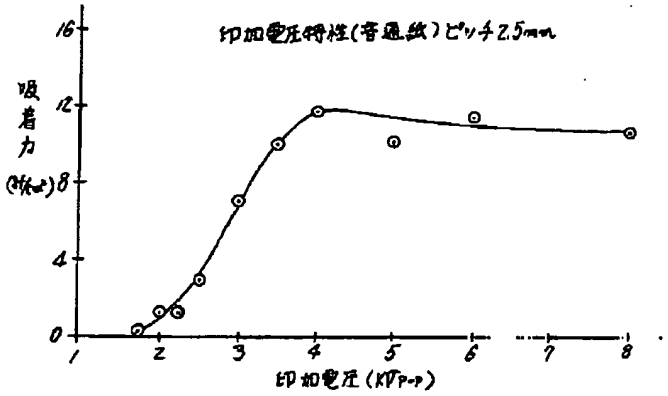
【図22】



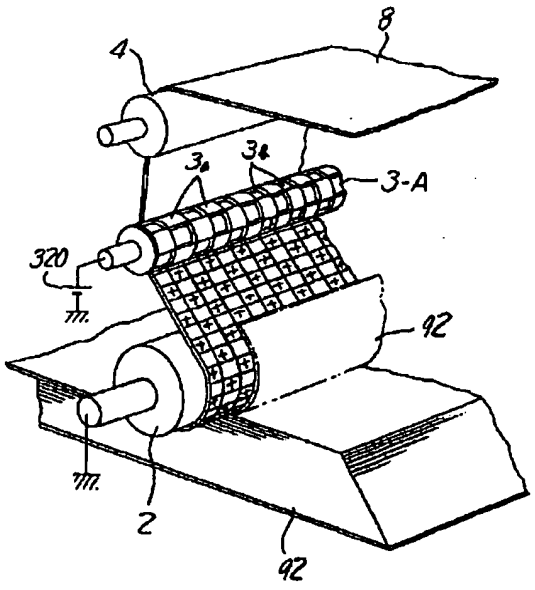
【図２７】



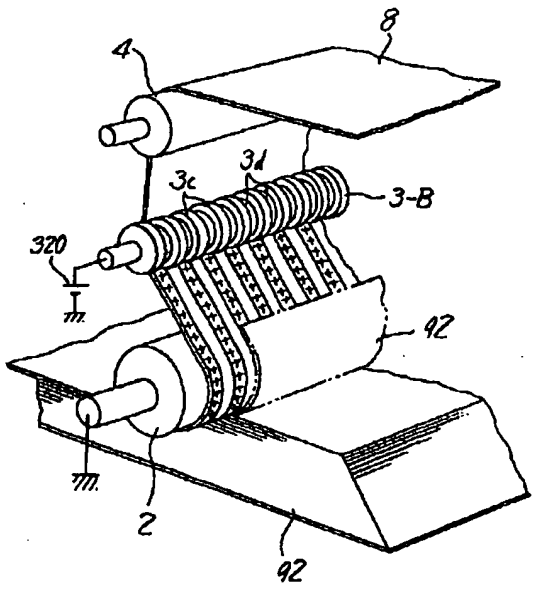
【図２８】



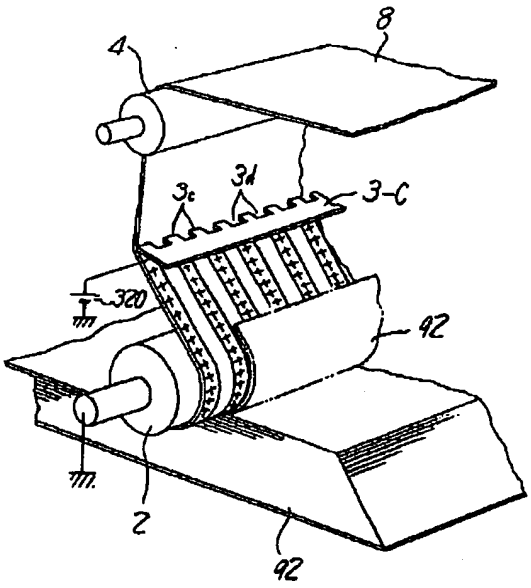
【図２９】



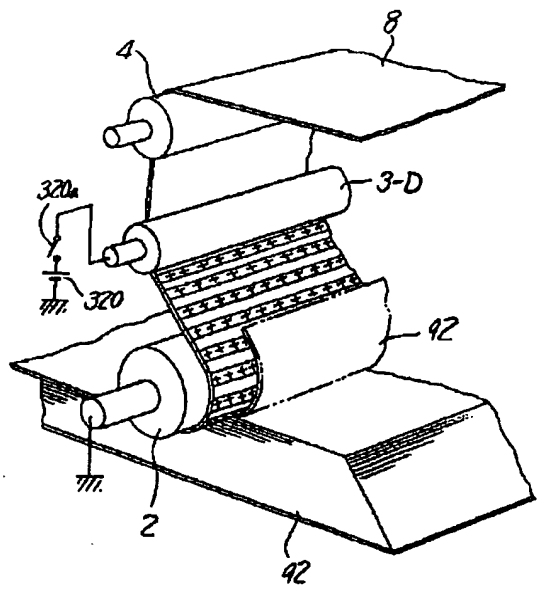
【図３１】



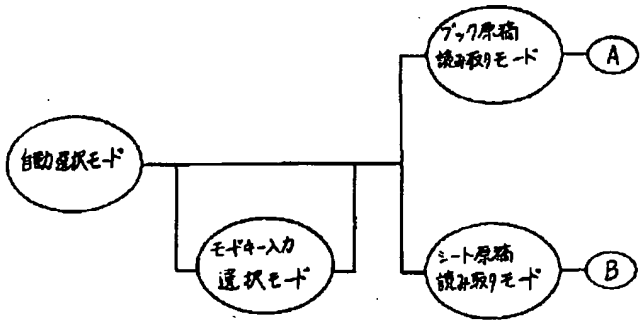
【図３２】



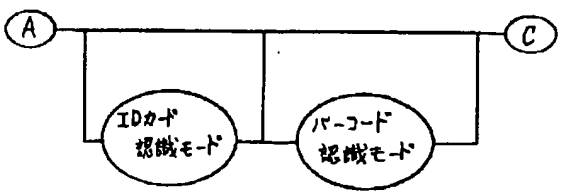
【図３３】



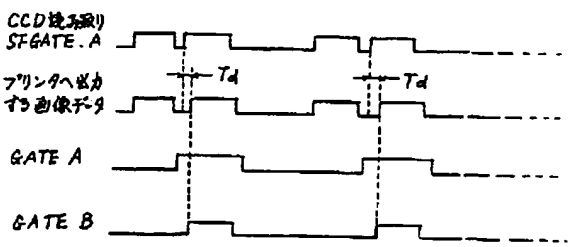
【図３５】



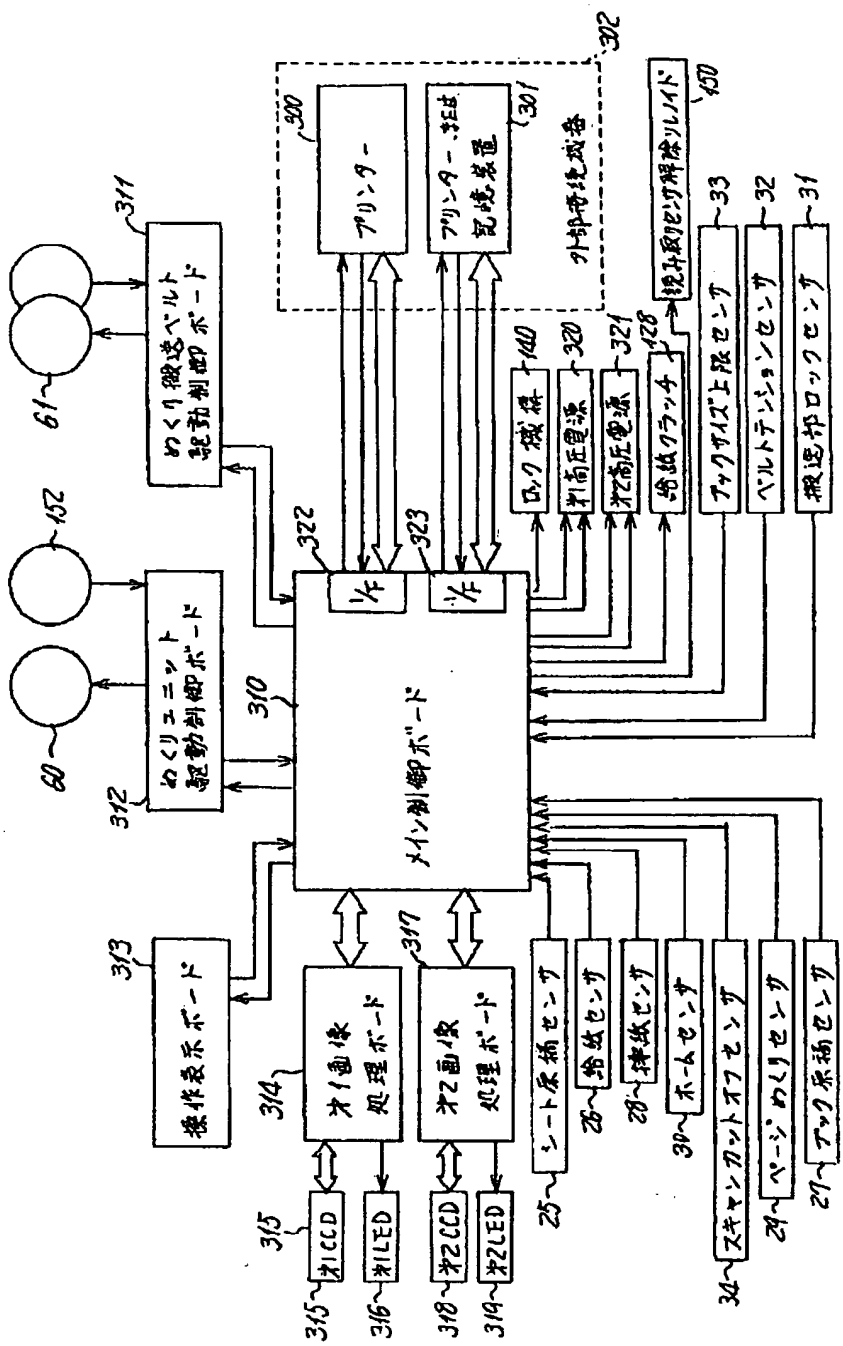
【図３６】



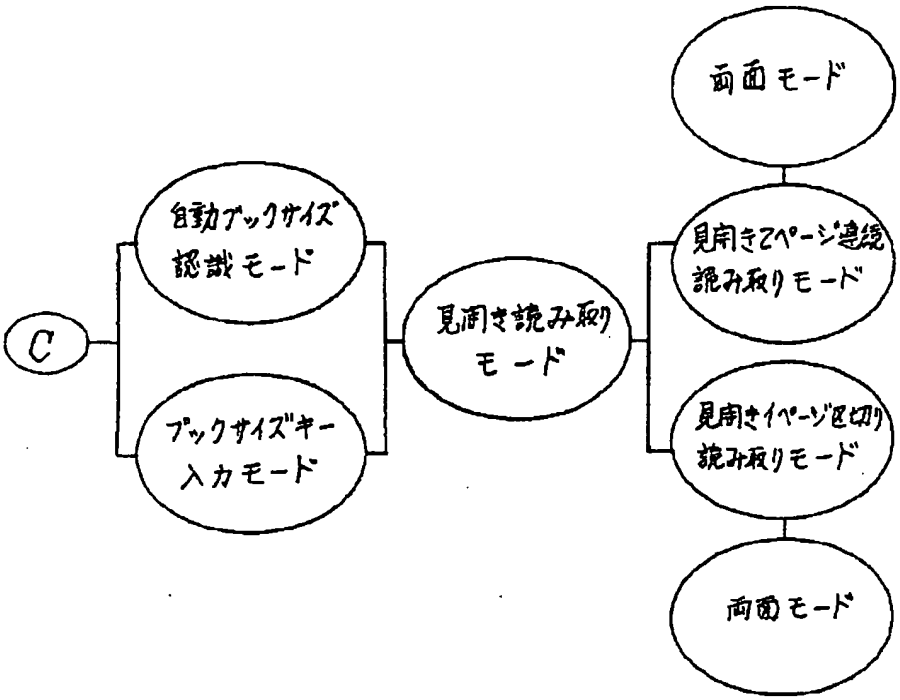
【図４５】



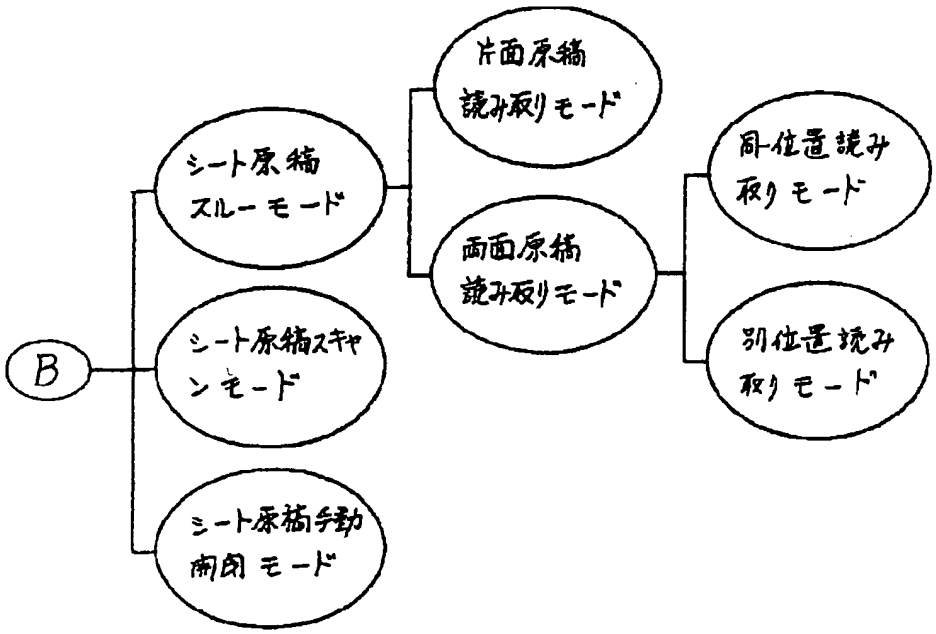
【図34】



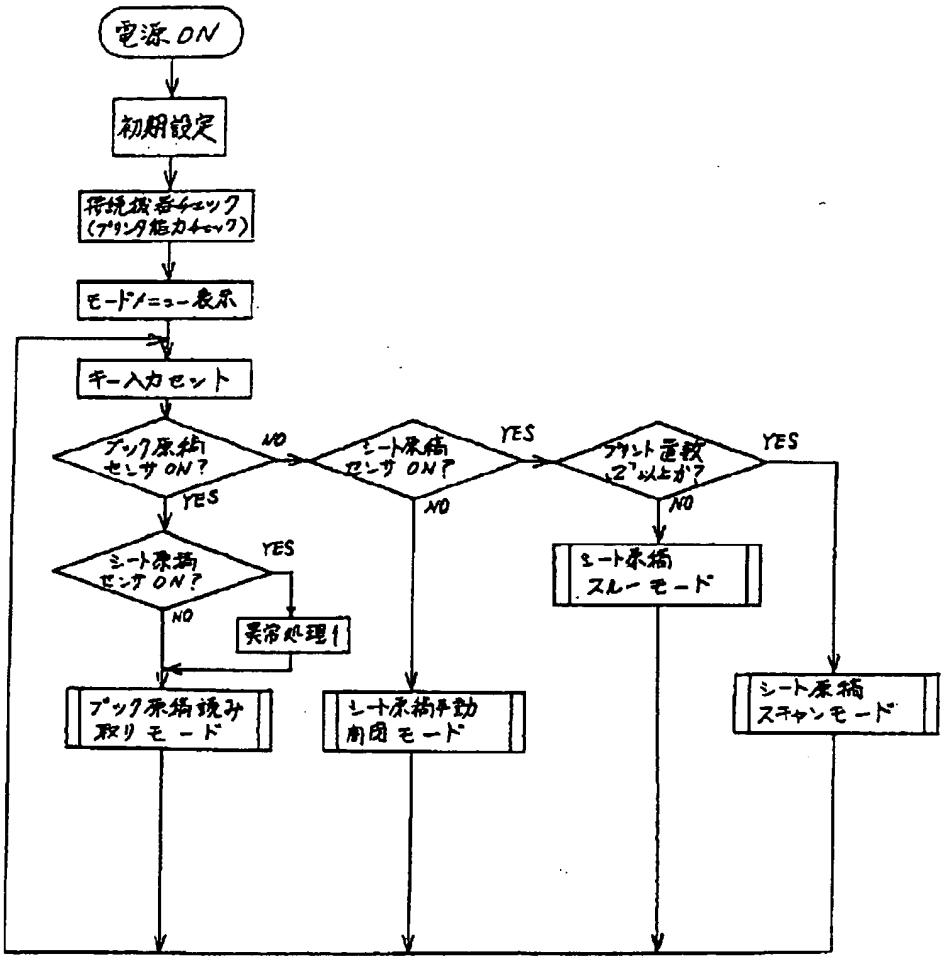
【図３７】



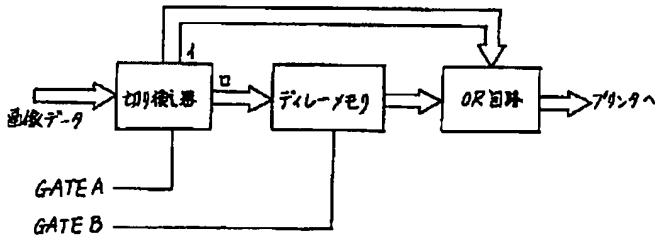
【図３８】



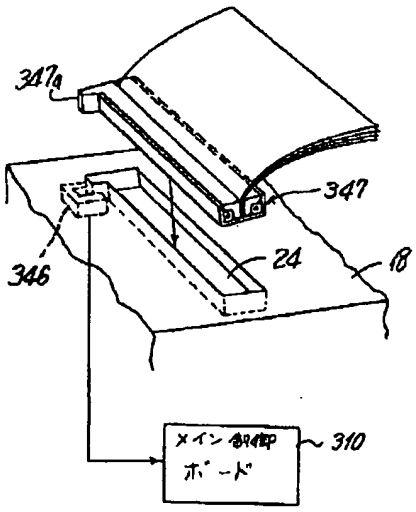
【図39】



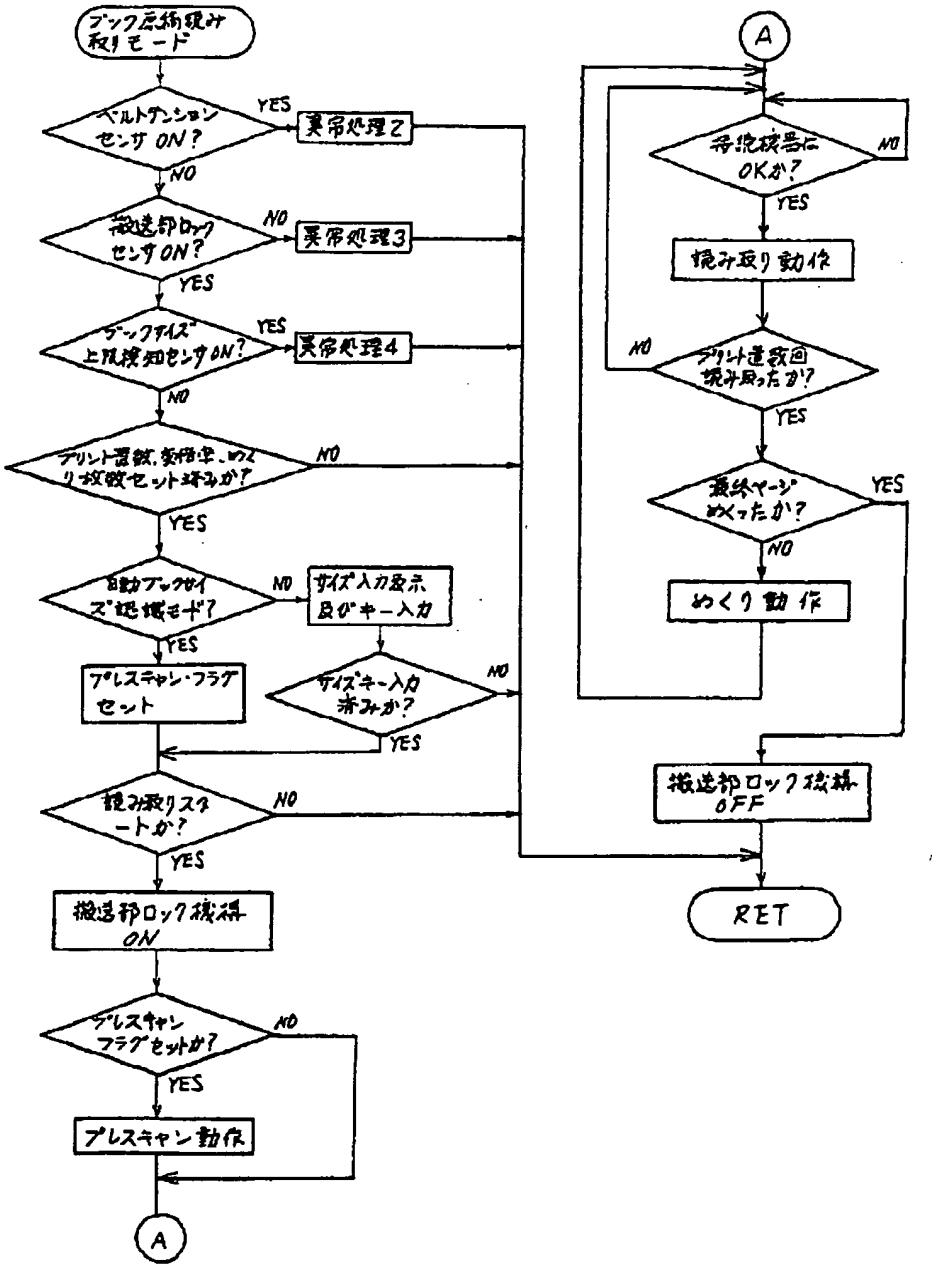
【図44】



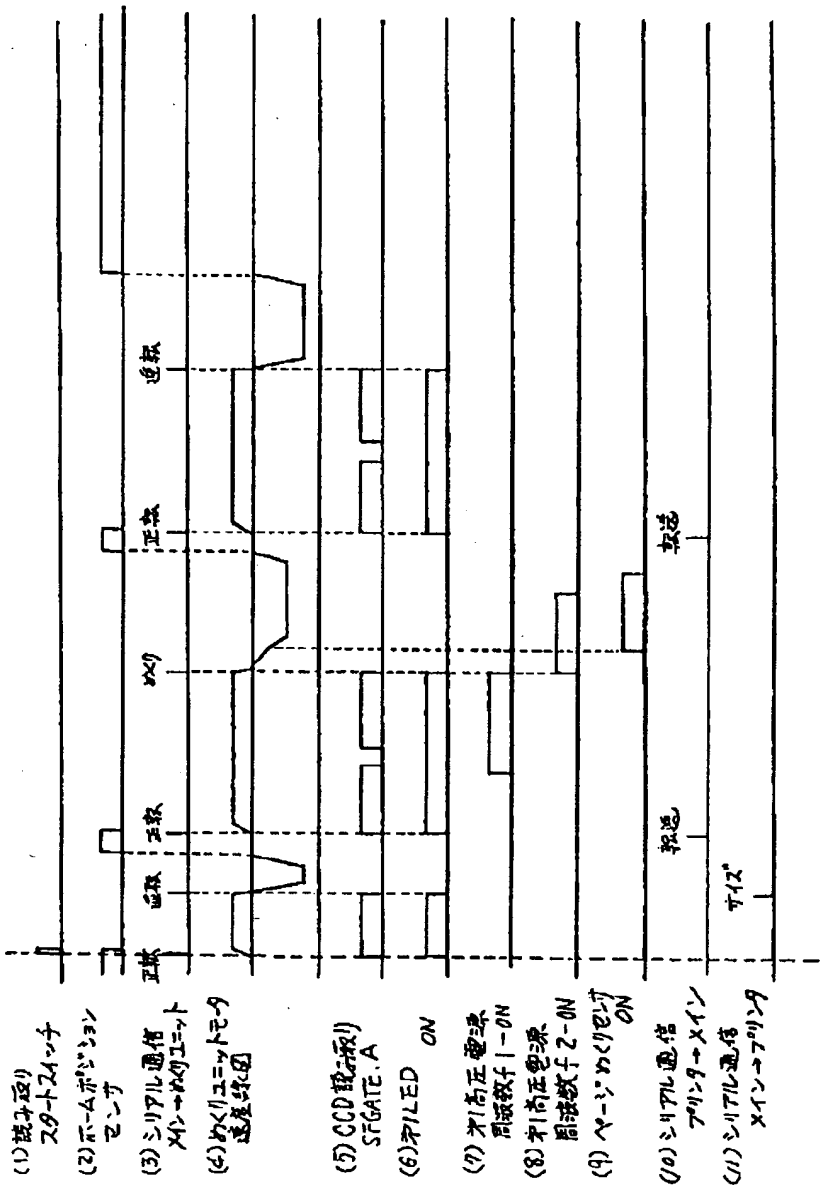
【図70】



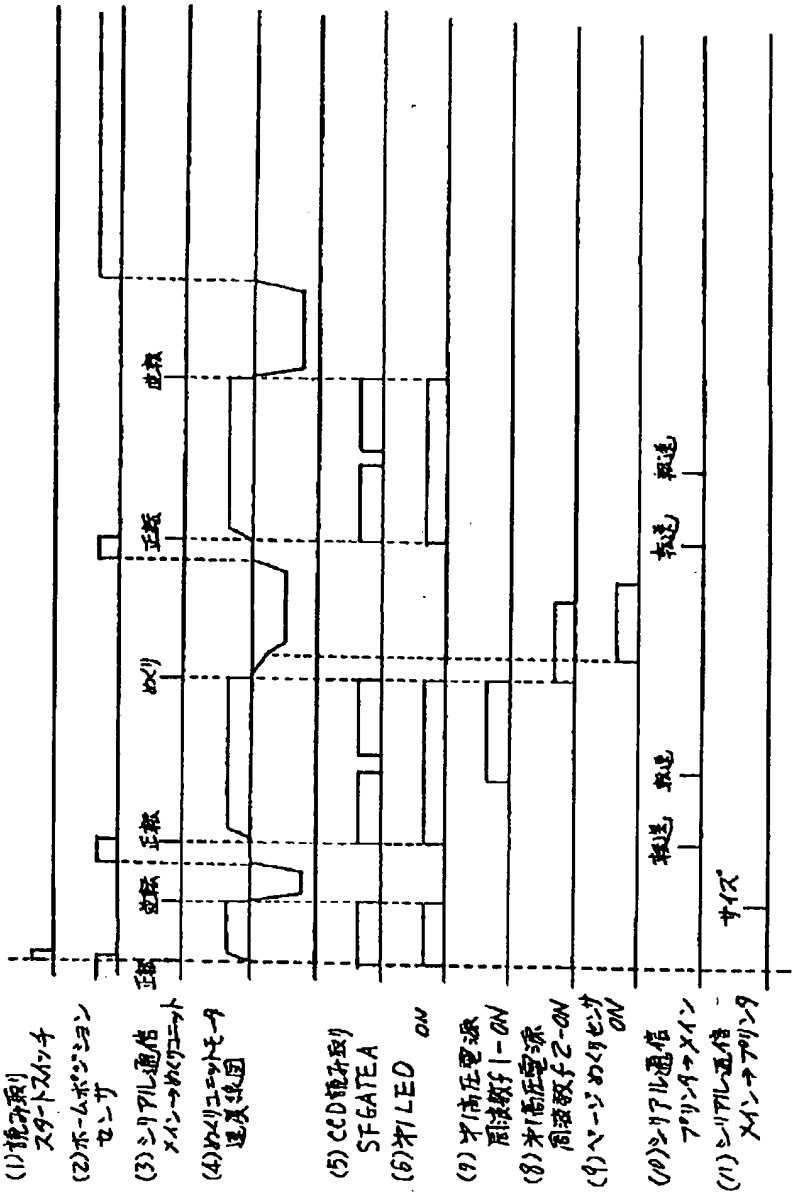
【図40】



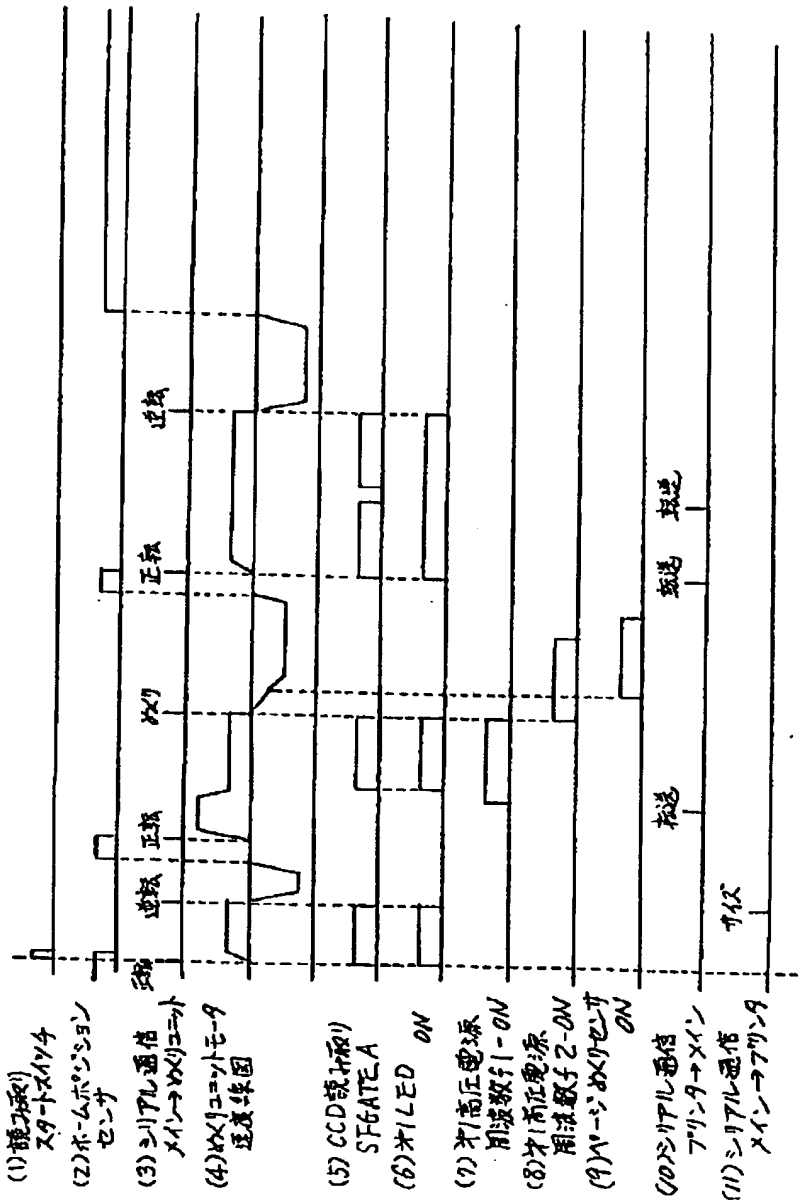
【図41】



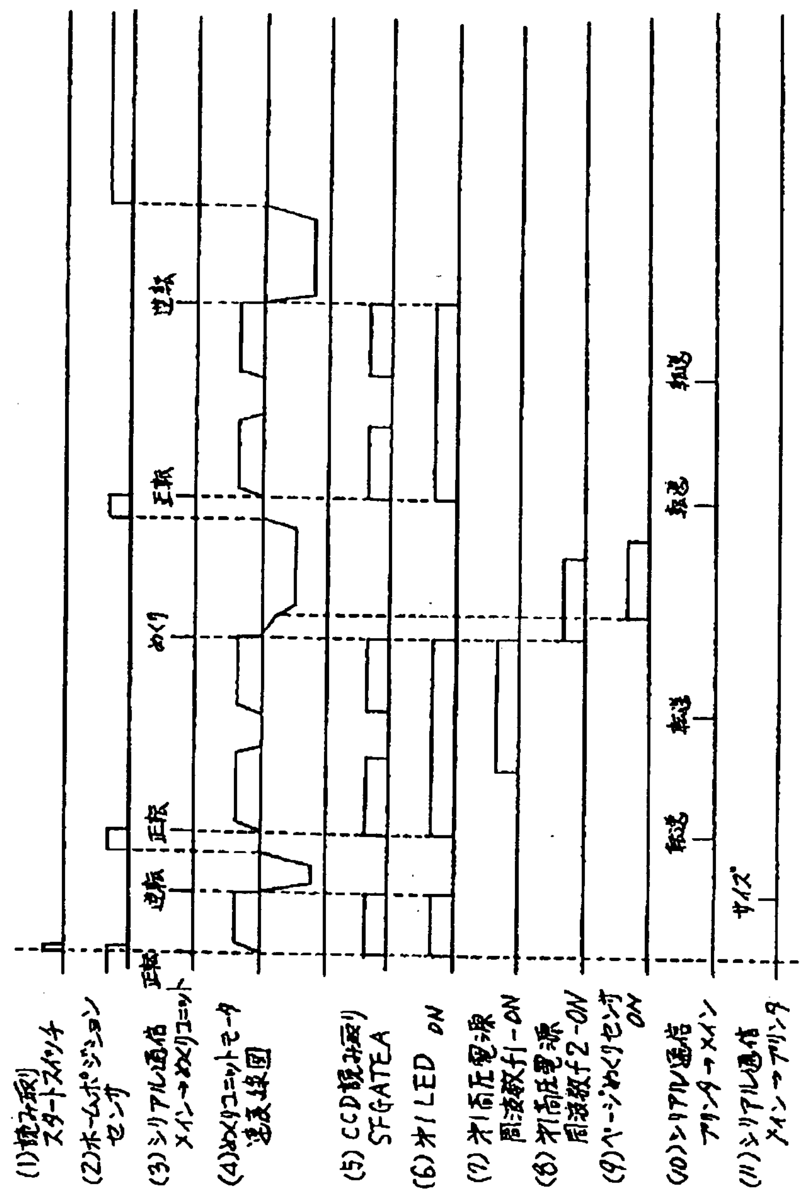
【図42】



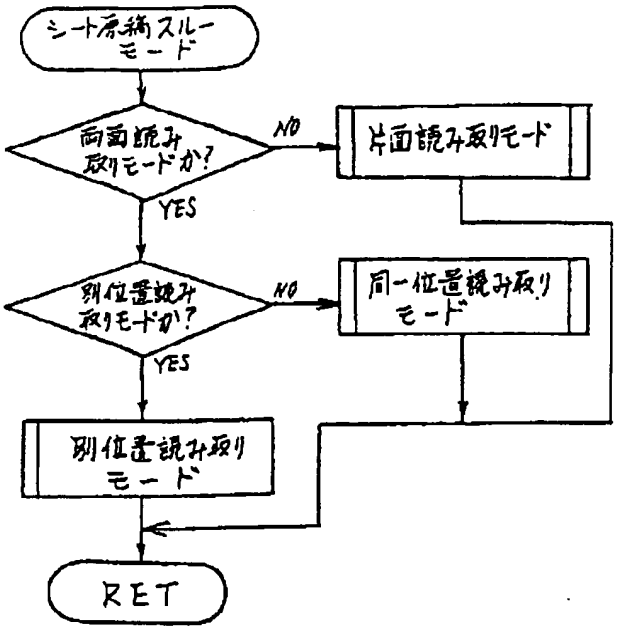
【図４３】



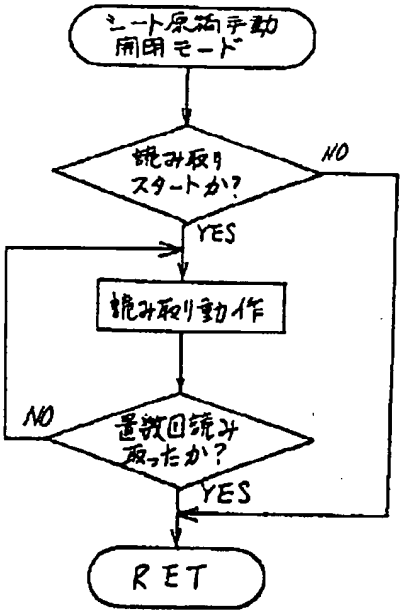
【図46】



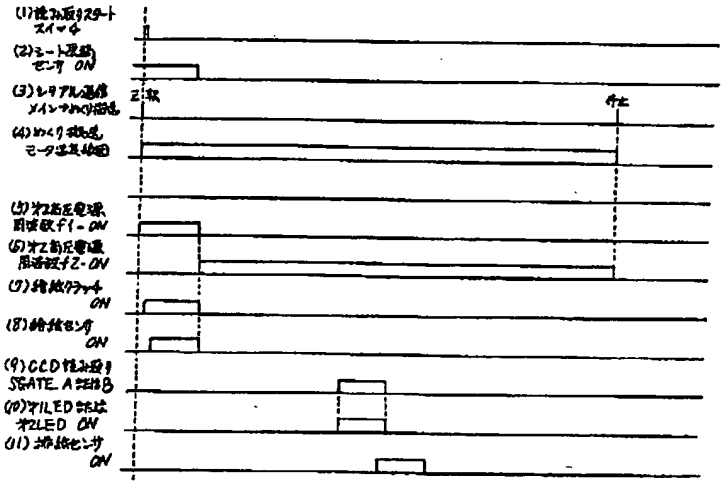
【図４７】



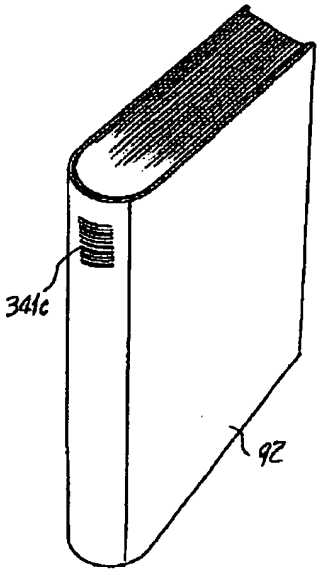
【図５６】



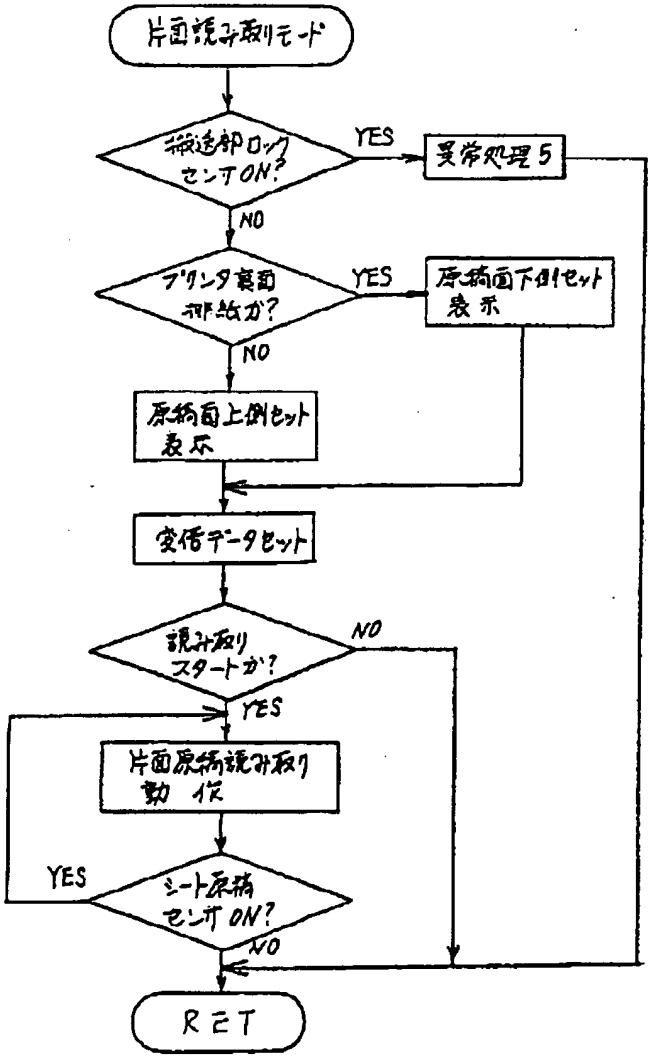
【図４９】



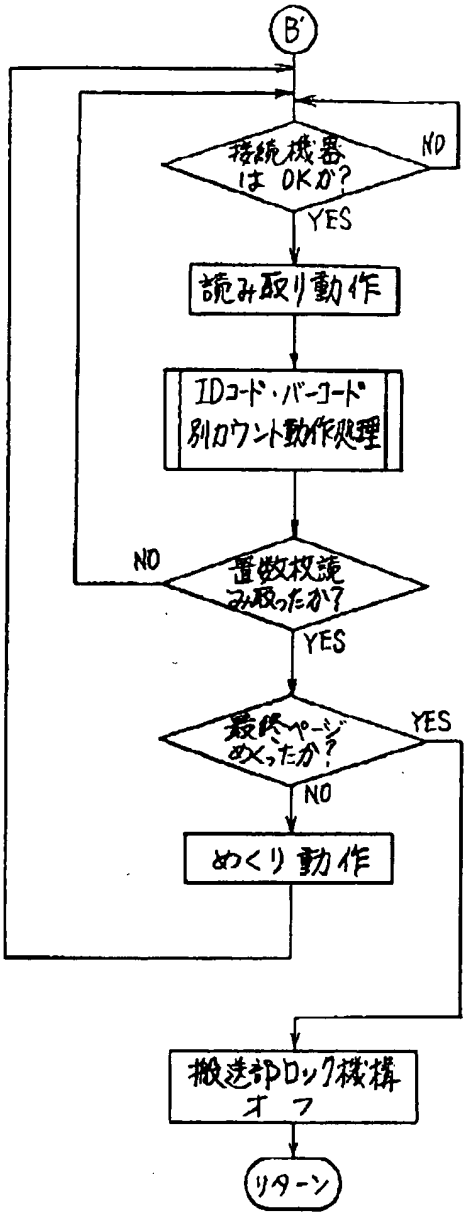
【図６６】



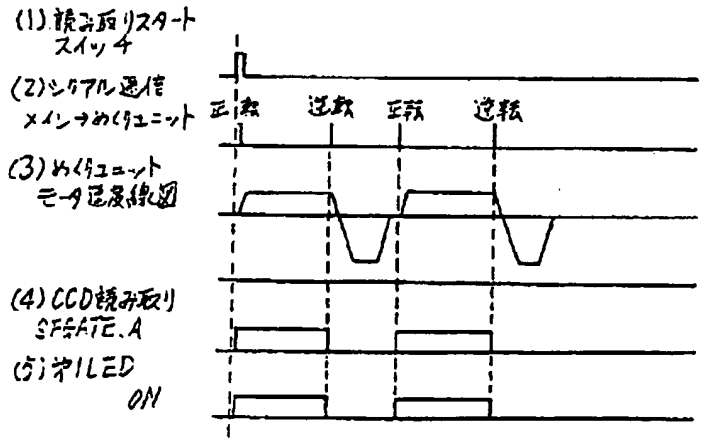
【図48】



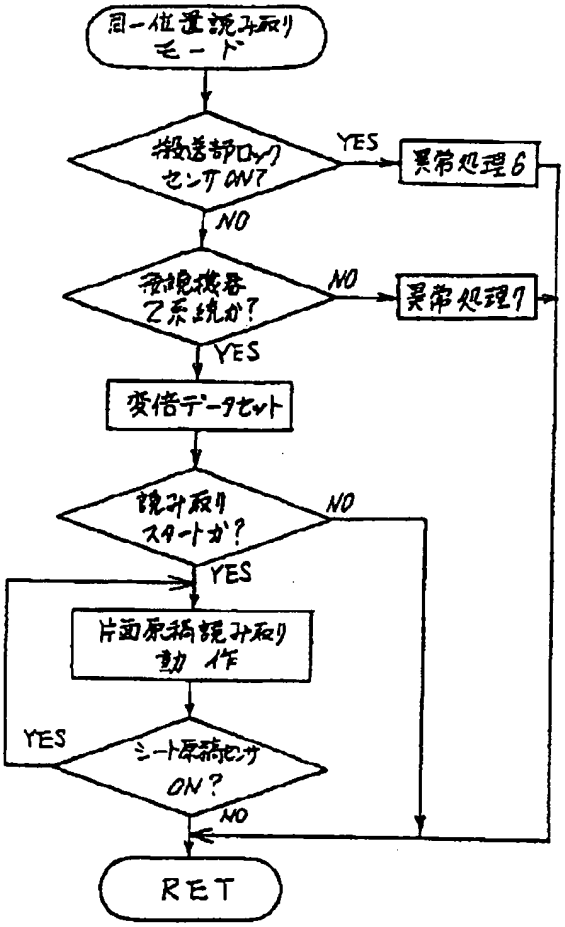
【図63】



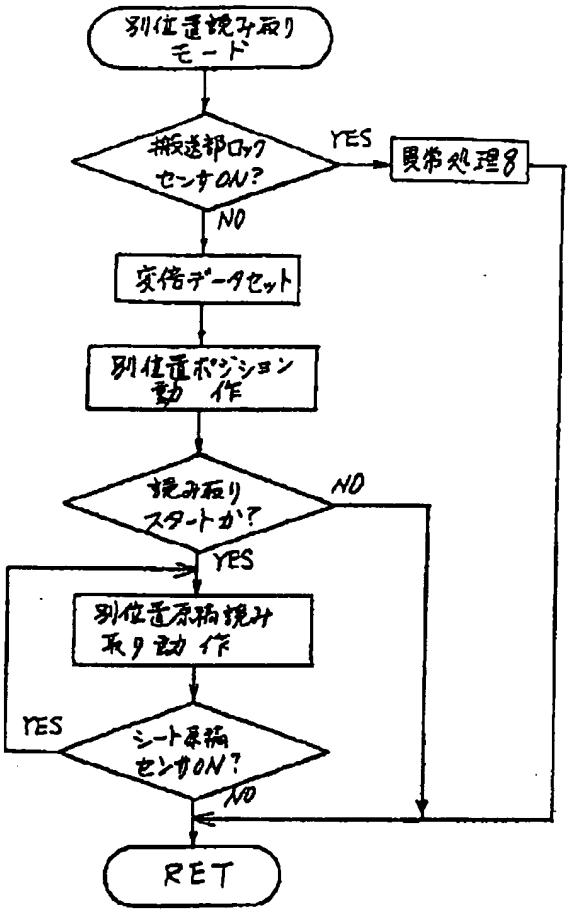
【図57】



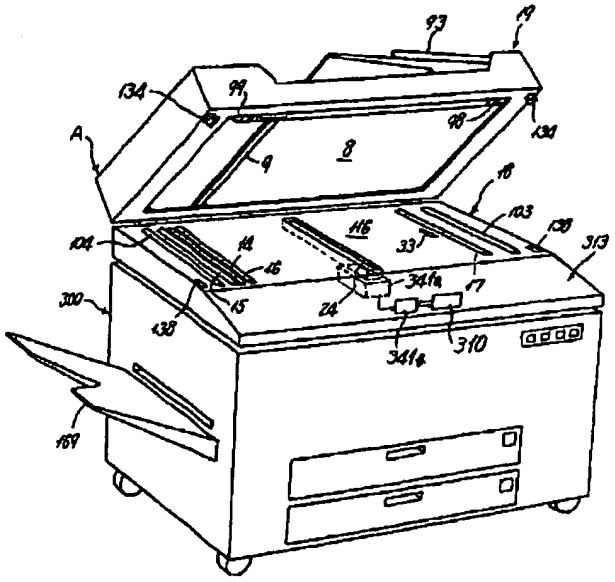
【図５０】



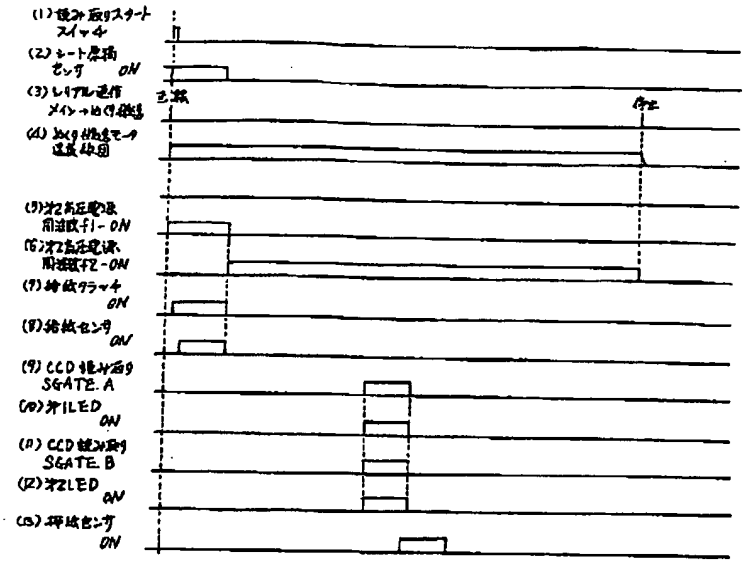
【図５２】



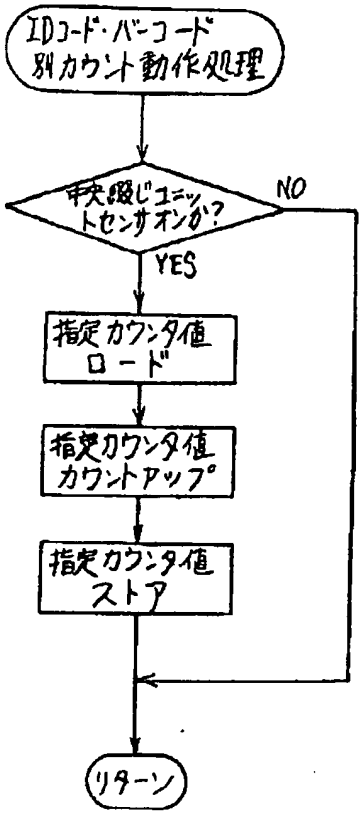
【図６７】



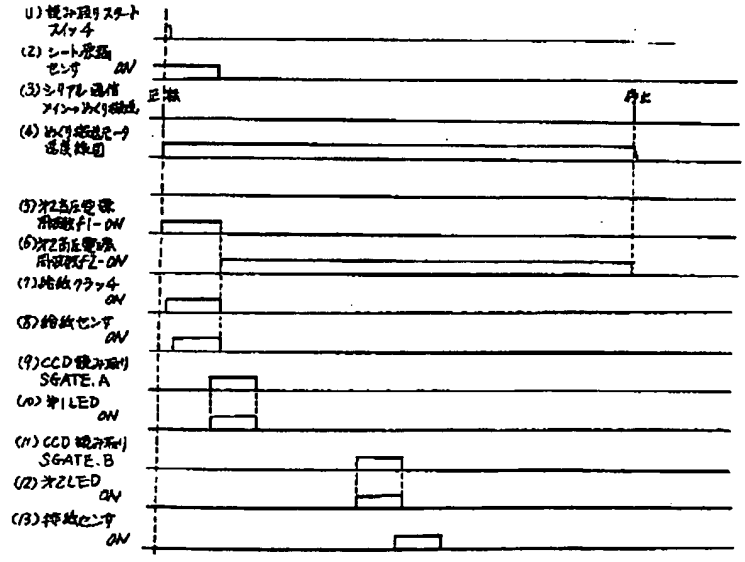
【図５１】



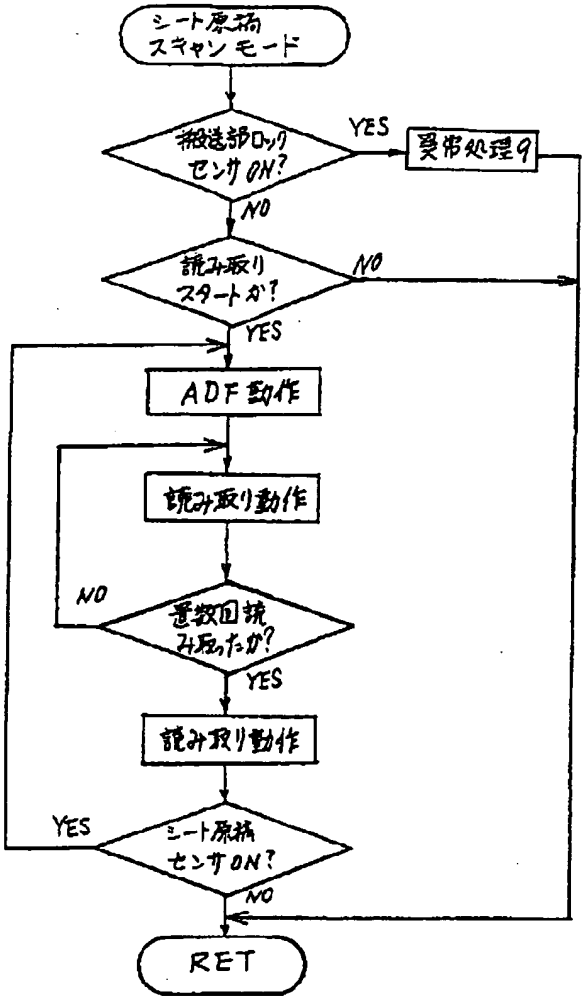
【図６９】



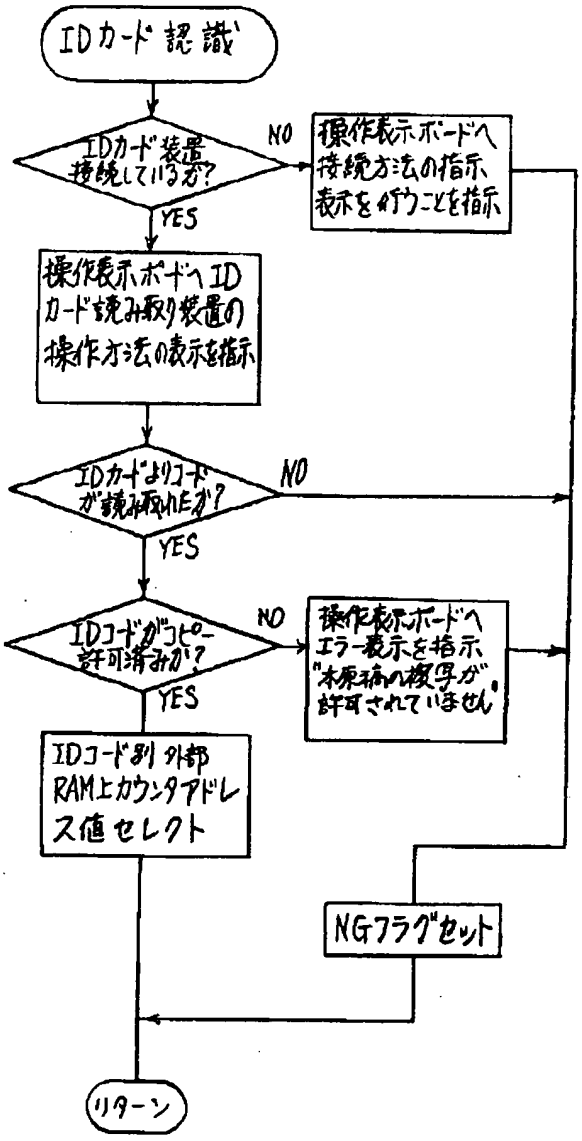
【図５３】



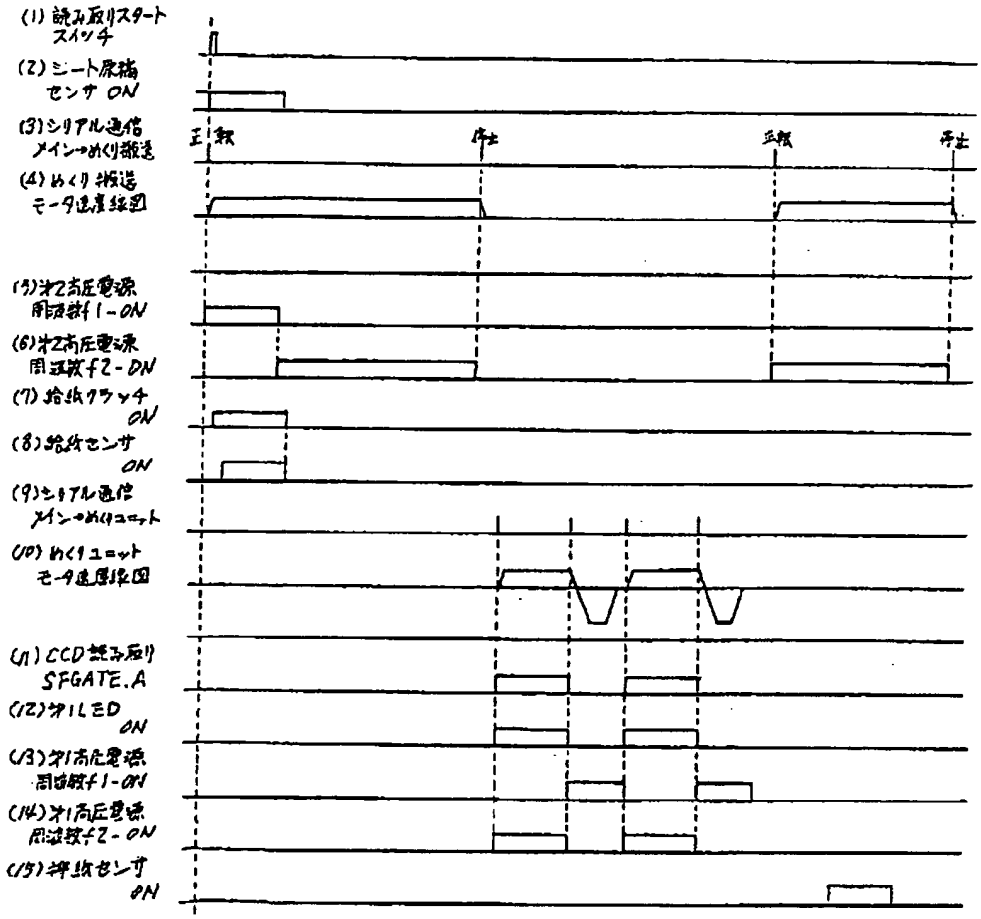
【図54】



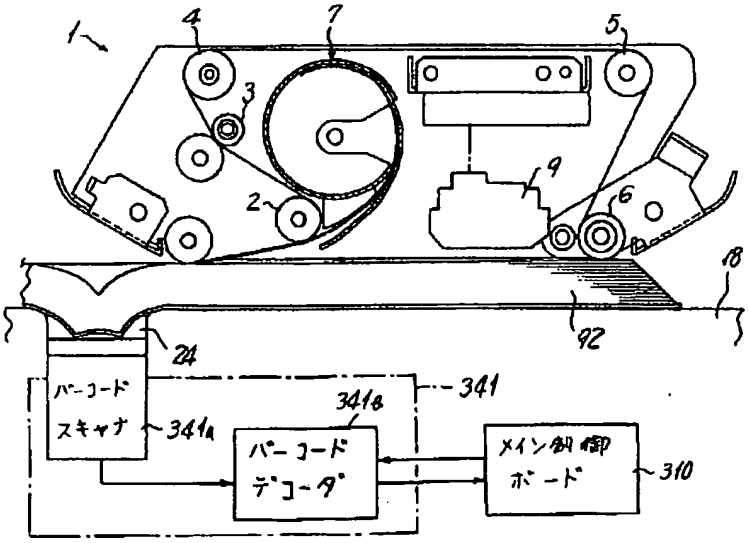
【図64】



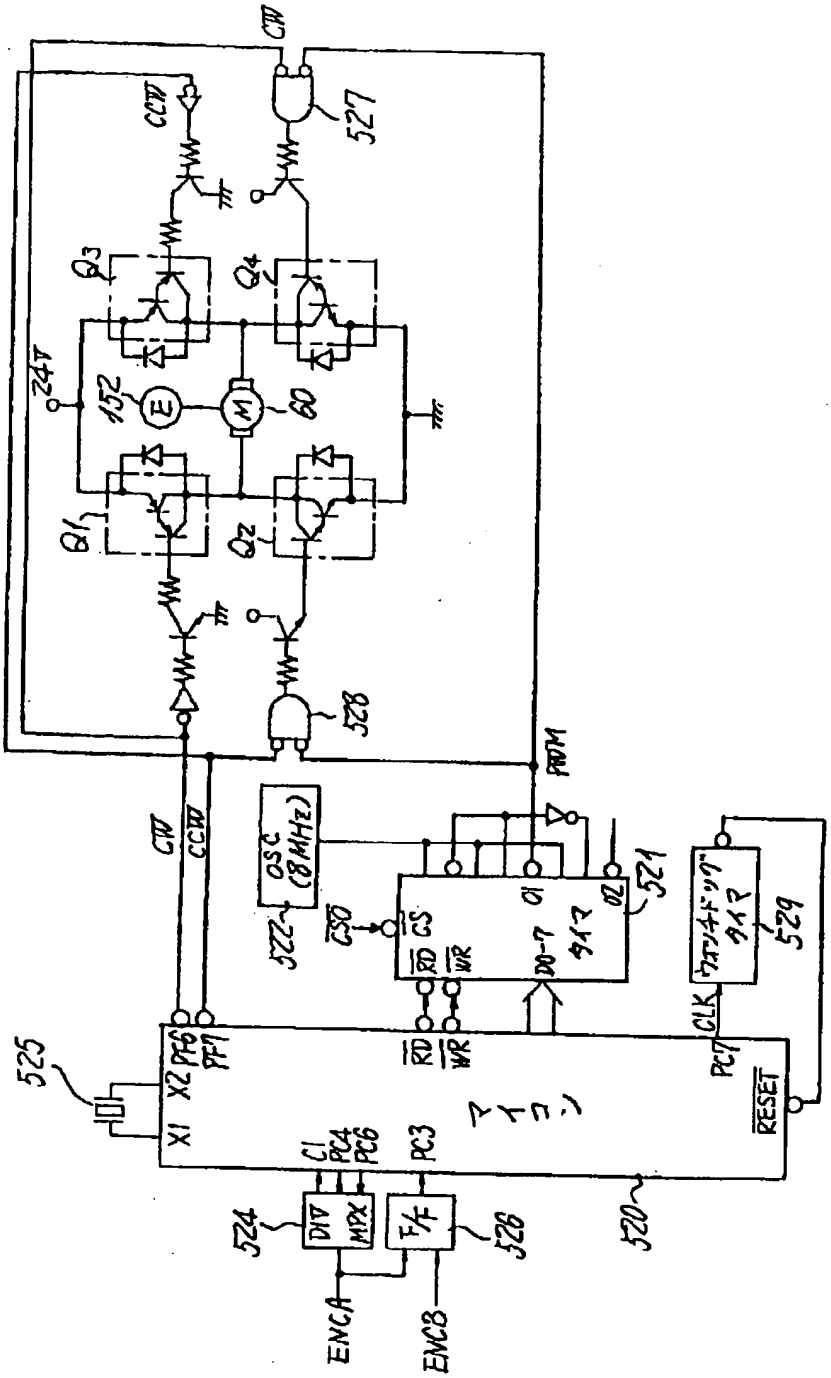
【図５５】



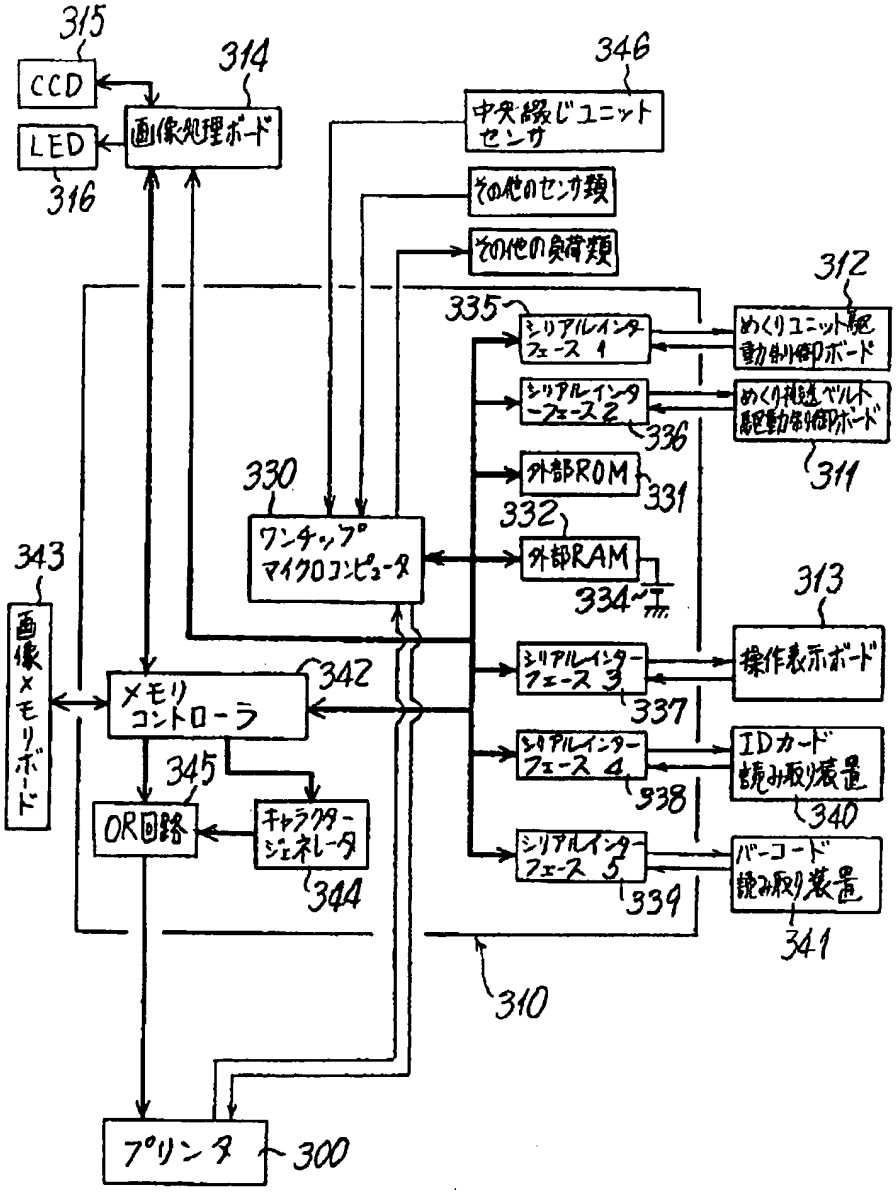
【図６８】



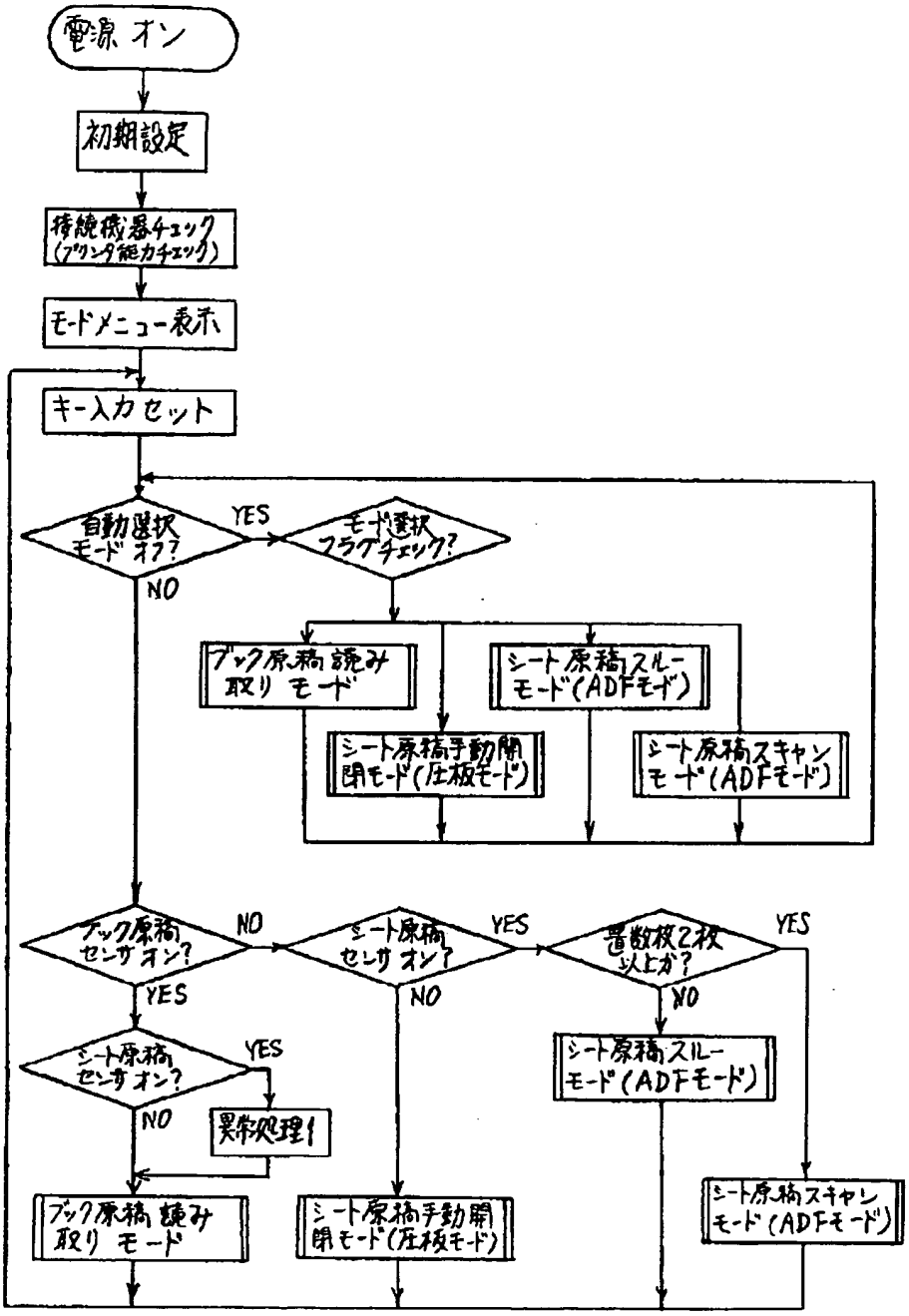
【図58】



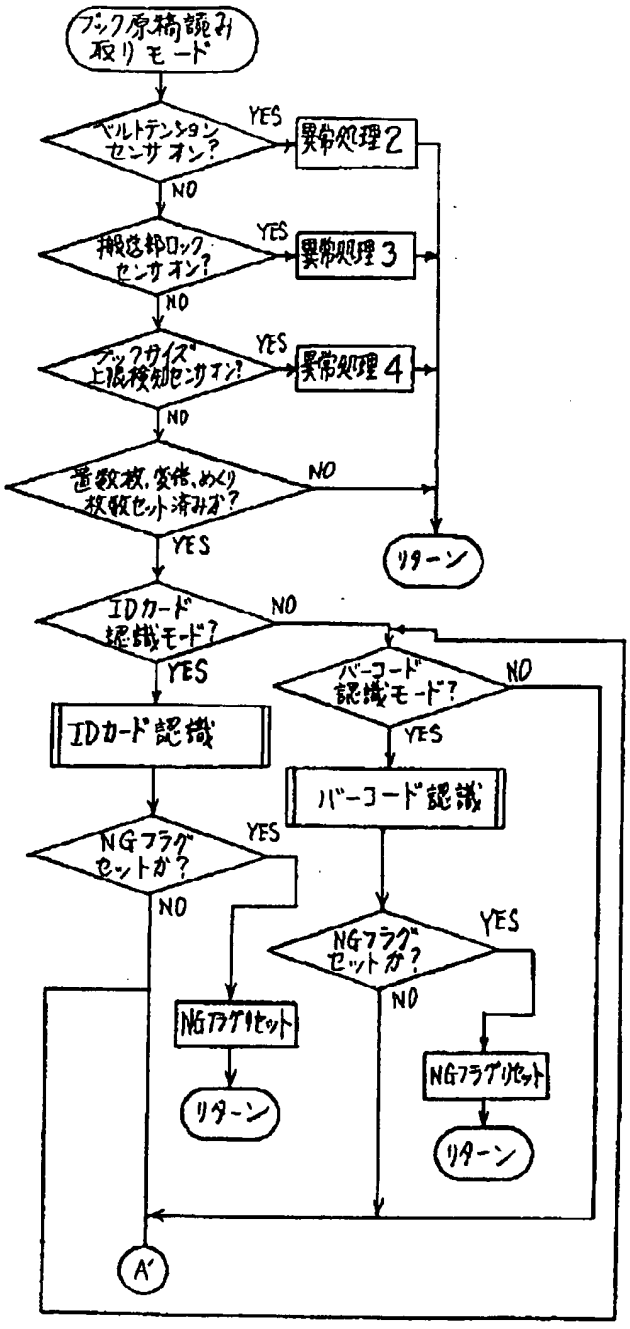
【図59】



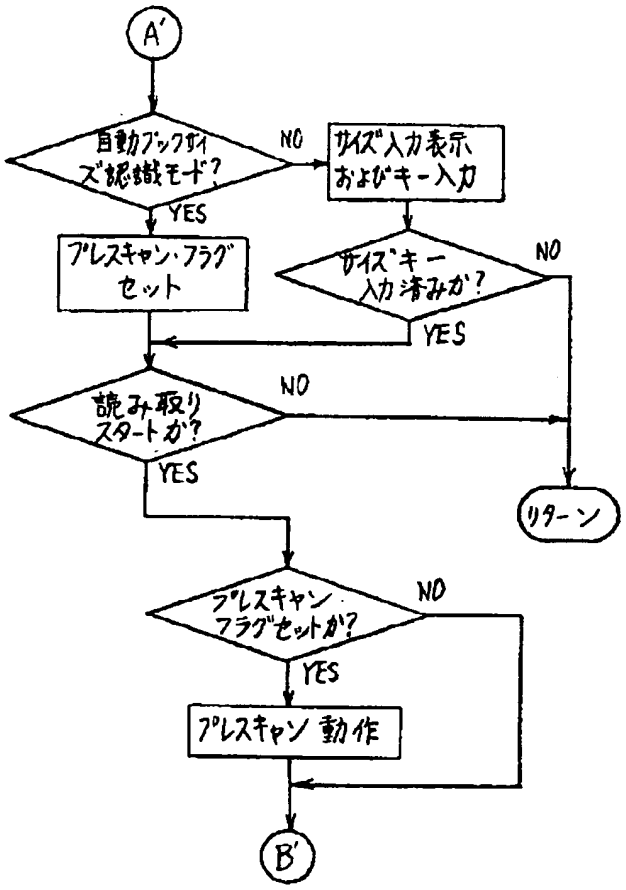
【図60】



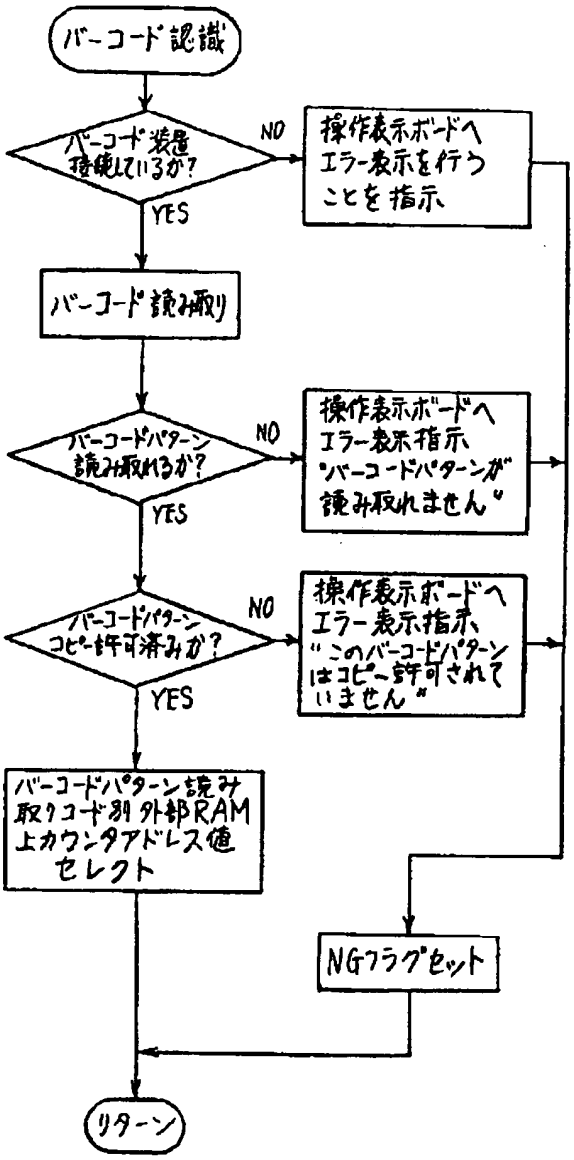
【図61】



【図６２】



【図65】



フロントページの続き

(51) Int. Cl.[5]		識別記号	庁内整理番号	FI	技術表示箇所
G03B	27/62		8106－2K		
G03G	15/00	102			
		107	8530－2H		
	15/04	119			
G06F	15/64	325 B	8840－5L		
H04N	1/00	108 M	7046－5C		
	1/10		7205－5C		

(72)発明者　坂内　和典
東京都大田区中馬込1丁目3番6号・株式会社リコー内

(72)発明者　椎名　将
東京都大田区中馬込1丁目3番6号・株式会社リコー内

Requested Patent: JP6175794A

Title: PRINT PROCESSING SYSTEM ;

Abstracted Patent: JP6175794 ;

Publication Date: 1994-06-24 ;

Inventor(s): TANAKA KOICHIRO ;

Applicant(s): FUJI XEROX CO LTD ;

Application Number: JP19920329533 19921209 ;

Priority Number(s): ;

IPC Classification: G06F3/12; B41J29/38 ;

Equivalents: ;

ABSTRACT:

PURPOSE:To prevent a third person from freely using and printing a print resource stored in a printer.

CONSTITUTION:When print is requested, a requester access right-acquiring part 22 acquires the requester access right from an access right management part 23, and a print resource use instruction in which this requester access right is built is issued. The requester access right built in this instruction is collated with a user access right added to a print resource 39 with access right by a printer 12. If it is discriminated as the result that the requester does not have the just right, the use of the print resource 39 is rejected, and the result is reported to a host computer 11. Deletion and update of the print resource are controlled in the same manner.

(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（A）　(11)特許出願公開番号

特開平6－175794

(43)公開日　平成６年(1994)６月24日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　3/12	A			
	B			
Ｂ４１Ｊ　29/38	Z	9113－2C		

審査請求　未請求　請求項の数４(全 29 頁)

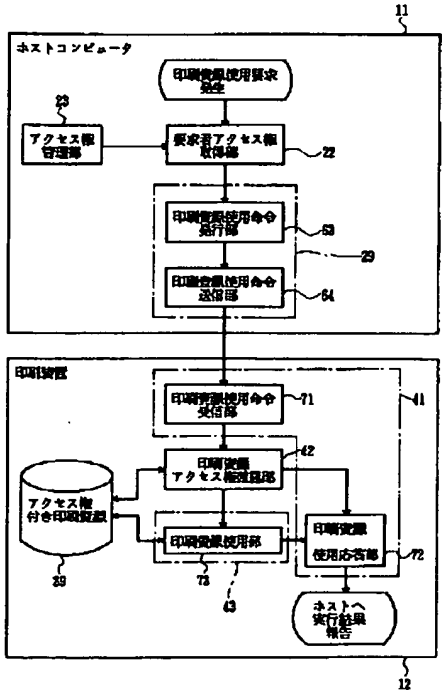
(21)出願番号　特願平4－329533

(22)出願日　平成４年(1992)12月９日

(71)出願人　000005496
富士ゼロックス株式会社
東京都港区赤坂三丁目３番５号
(72)発明者　田中　浩一郎
埼玉県岩槻市府内３丁目７番１号　富士ゼロックス株式会社内
(74)代理人　弁理士　山内　梅雄

(54)【発明の名称】　印刷処理システム

(57)【要約】
【目的】　印刷装置に格納された印刷資源を第三者が勝手に使用して印刷できないようにする。
【構成】　印刷の要求があると、要求者アクセス権取得部２２がアクセス権管理部２３から要求者アクセス権を取得し、これを組み込んで印刷資源使用命令が発行される。この命令に組み込まれた要求者アクセス権は印刷装置１２でアクセス権付き印刷資源３９に付加されていた利用者アクセス権と照合される。この結果、要求者が正当な権限を有しないことが判別されたらその印刷資源の使用が拒否され、その結果がホストコンピュータ１１に報告される。印刷資源の使用の他、削除および更新についても同様の規制を行うことができる。



1

【特許請求の範囲】

【請求項1】　文字フォント等の印刷に必要とされる印刷資源を格納した送信側印刷資源格納手段と、印刷資源ごとにそれらを処理できる利用者を利用者アクセス権として規定した利用者管理テーブルと、印刷資源の処理要求者ごとに印刷資源のアクセス範囲を要求者アクセス権として規定した要求者管理テーブルと、前記送信側印刷資源格納手段に格納された印刷資源の送信が要求されたとき印刷資源ごとに対応する前記利用者アクセス権を付加して転送する印刷資源転送手段と、印刷資源の処理要求があったときその要求者アクセス権を組み込んだ印刷資源処理命令を発行する印刷資源使用命令発行手段とを備えたホストコンピュータと、

このホストコンピュータと回線によって接続され前記印刷資源が送られてきたときこれを格納する受信側印刷資源格納手段と、前記印刷資源処理命令を受信する印刷資源処理命令受信手段と、前記印刷資源処理命令が受信されたとき対象となる印刷資源の利用者アクセス権と受信した要求者アクセス権を比較する比較手段と、比較結果が一致しないときその印刷資源の処理を拒否する使用拒否手段とを備えた印刷装置とを具備することを特徴とする印刷処理システム。

【請求項2】　文字フォント等の印刷に必要とされる印刷資源を格納した送信側印刷資源格納手段と、印刷資源ごとにそれらを印刷時に使用できる利用者を利用者アクセス権として規定した利用者管理テーブルと、印刷資源の使用要求者ごとに印刷資源のアクセス範囲を要求者アクセス権として規定した要求者管理テーブルと、前記送信側印刷資源格納手段に格納された印刷資源の送信が要求されたとき印刷資源ごとに対応する前記利用者アクセス権を付加して転送する印刷資源転送手段と、印刷資源の使用要求があったときその要求者アクセス権を組み込んだ印刷資源使用命令を発行する印刷資源使用命令発行手段とを備えたホストコンピュータと、

このホストコンピュータと回線によって接続され前記印刷資源が送られてきたときこれを格納する受信側印刷資源格納手段と、前記印刷資源使用命令を受信する印刷資源使用命令受信手段と、前記印刷資源使用命令が受信されたとき対象となる印刷資源の利用者アクセス権と受信した要求者アクセス権を比較する比較手段と、比較結果が一致しないときその印刷資源の使用を拒否する使用拒否手段とを備えた印刷装置とを具備することを特徴とする印刷処理システム。

【請求項3】　文字フォント等の印刷に必要とされる印刷資源を格納した送信側印刷資源格納手段と、印刷資源ごとにそれらを削除できる利用者を利用者アクセス権として規定した利用者管理テーブルと、印刷資源の削除要求者ごとに印刷資源のアクセス範囲を要求者アクセス権として規定した要求者管理テーブルと、前記送信側印刷資源格納手段に格納された印刷資源の送信が要求されたとき印刷資源ごとに対応する前記利用者アクセス権を付加して転送する印刷資源転送手段と、印刷資源の削除要求があったときその要求者アクセス権を組み込んだ印刷資源削除命令を発行する印刷資源使用命令発行手段とを備えたホストコンピュータと、

このホストコンピュータと回線によって接続され前記印刷資源が送られてきたときこれを格納する受信側印刷資源格納手段と、前記印刷資源削除命令を受信する印刷資源削除命令受信手段と、前記印刷資源削除命令が受信されたとき対象となる印刷資源の利用者アクセス権と受信した要求者アクセス権を比較する比較手段と、比較結果が一致しないときその印刷資源の削除を拒否する使用拒否手段とを備えた印刷装置とを具備することを特徴とする印刷処理システム。

【請求項4】　文字フォント等の印刷に必要とされる印刷資源を格納した送信側印刷資源格納手段と、印刷資源ごとにそれらを更新できる利用者を利用者アクセス権として規定した利用者管理テーブルと、印刷資源の更新要求者ごとに印刷資源のアクセス範囲を要求者アクセス権として規定した要求者管理テーブルと、前記送信側印刷資源格納手段に格納された印刷資源の送信が要求されたとき印刷資源ごとに対応する前記利用者アクセス権を付加して転送する印刷資源転送手段と、印刷資源の更新要求があったときその要求者アクセス権を組み込んだ印刷資源更新命令を発行する印刷資源使用命令発行手段とを備えたホストコンピュータと、

このホストコンピュータと回線によって接続され前記印刷資源が送られてきたときこれを格納する受信側印刷資源格納手段と、前記印刷資源更新命令を受信する印刷資源更新命令受信手段と、前記印刷資源更新命令が受信されたとき対象となる印刷資源の利用者アクセス権と受信した要求者アクセス権を比較する比較手段と、比較結果が一致しないときその印刷資源の更新を拒否する使用拒否手段とを備えた印刷装置とを具備することを特徴とする印刷処理システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は印刷資源を備えたホストコンピュータ等の印刷資源格納手段と、必要に応じてこの印刷資源格納手段から印刷資源の供給を受けて文字あるいはグラフィック等からなる印刷情報の印刷を行う印刷装置とを備えた印刷処理システムに係わり、詳細には印刷装置側で印刷資源の利用や削除あるいは更新等の各種処理の管理を行うことのできる印刷処理システムに関する。

【0002】

【従来の技術】印刷装置は、文書あるいはグラフィック描画情報等の印刷情報を印刷する際に、個別に指定された文字フォントや文字パターン等の印刷資源を用いながら印刷処理を行うようになっている。このような印刷処

理システムでは、印刷の開始するたびにそのときの印刷情報に使われるすべての印刷資源をホストコンピュータから印刷装置に転送するようになっているものがある。このようなシステムでは、印刷情報が切り替わるたびに、前に転送した印刷資源をすべて削除して新たに印刷資源を転送しているので、転送に手間と時間がかかるという問題がある。

【０００３】そこで、特開昭６３－１３０３６２号公報ではこれを改良し、印刷装置側に例えば磁気ディスクのような比較的大容量の記憶手段を用意させ、これに最低限必要な印刷資源を格納させるようになっている。そして、新たに印刷を行う際にはその印刷情報に必要な印刷資源のみをホストコンピュータ等の印刷資源格納手段から転送してもらうようになっている。

【０００４】

【発明が解決しようとする課題】ところが、この提案された印刷処理システムでは、ホストコンピュータ等の印刷資源格納手段に格納された印刷資源が印刷を要求する者によって結果的に自由に引き出され各個人等が管理する印刷装置側に渡ってしまうことになる。したがって、その印刷資源が機密性を有するようなものである場合には、機密が漏洩するといった問題を発生させた。

【０００５】また、印刷装置に備えられた磁気ディスク等の記憶手段は印刷資源を無制限に格納することができないため、適宜整理する必要がある。このとき、印刷資源格納手段から転送しておいた印刷資源を第三者が誤って削除することがあった。この場合には、新たにホストコンピュータ等からその印刷資源を転送する必要があり、印刷に要する時間を実質的に長引かせるといった問題があった。

【０００６】更に、印刷資源格納手段から転送しておいた印刷資源を第三者が勝手に内容を訂正し更新してしまう場合もあった。この場合には、文字パターンが異なってしまったり、所定の文字コードの組み合わせに対応する語句または文章の内容が異なってしまうというように印刷に使用する印刷資源自体が変質してしまう。したがって、更新されたことを知らずにその印刷資源を使用すると、印刷された内容自体が予期しないものになるといった問題を生じさせた。

【０００７】そこで本発明の目的は、ホストコンピュータ等の印刷資源格納手段から供給を受けた印刷資源について機密の必要なものを保護できるようにした印刷処理システムを提供することにある。

【０００８】本発明の他の目的は、ホストコンピュータ等の印刷資源格納手段から供給を受けた印刷資源のうち所定のものを第三者が勝手に削除できないようにした印刷処理システムを提供することにある。

【０００９】本発明の更に他の目的は、ホストコンピュータ等の印刷資源格納手段から供給を受けた印刷資源のうち所定のものを第三者が勝手に更新することのできないようにした印刷処理システムを提供することにある。

【００１０】

【課題を解決するための手段】請求項１記載の発明では、（イ）文字フォント等の印刷に必要とされる印刷資源を格納した送信側印刷資源格納手段と、印刷資源ごとにそれらを処理できる利用者を利用者アクセス権として規定した利用者管理テーブルと、印刷資源の処理要求者ごとに印刷資源のアクセス範囲を要求者アクセス権として規定した要求者管理テーブルと、送信側印刷資源格納手段に格納された印刷資源の送信が要求されたとき印刷資源ごとに対応する利用者アクセス権を付加して転送する印刷資源転送手段と、印刷資源の処理要求があったときその要求者アクセス権を組み込んだ印刷資源処理命令を発行する印刷資源使用命令発行手段とを備えたホストコンピュータと、（ロ）このホストコンピュータと回線によって接続され印刷資源が送られてきたときこれを格納する受信側印刷資源格納手段と、印刷資源処理命令を受信する印刷資源処理命令受信手段と、印刷資源処理命令が受信されたとき対象となる印刷資源の利用者アクセス権と受信した要求者アクセス権を比較する比較手段と、比較結果が一致しないときその印刷資源の処理を拒否する使用拒否手段とを備えた印刷装置とを印刷処理システムに具備させる。

【００１１】すなわち請求項１記載の発明では、ホストコンピュータから印刷資源を印刷装置に送信してこれに格納する際には、印刷資源と併せてそれを処理することのできる利用者を表わした利用者アクセス権を付加して転送しておく。また、その印刷資源の使用を要求する者に対しては、その者がアクセスすることのできる印刷資源を表わした要求者アクセス権を印刷資源処理命令に組み込んでホストコンピュータが印刷装置に送信することにする。印刷装置側ではこの印刷資源処理命令から抽出された要求者アクセス権と該当する印刷資源に付加された利用者アクセス権を照合し、要求者に印刷資源の処理の権限がない場合には、その処理を拒絶することにして、印刷資源の利用、削除、更新等の処理の安全を図るようになっている。

【００１２】請求項２記載の発明では、（イ）文字フォント等の印刷に必要とされる印刷資源を格納した送信側印刷資源格納手段と、印刷資源ごとにそれらを印刷時に使用できる利用者を利用者アクセス権として規定した利用者管理テーブルと、印刷資源の使用要求者ごとに印刷資源のアクセス範囲を要求者アクセス権として規定した要求者管理テーブルと、送信側印刷資源格納手段に格納された印刷資源の送信が要求されたとき印刷資源ごとに対応する利用者アクセス権を付加して転送する印刷資源転送手段と、印刷資源の使用要求があったときその要求者アクセス権を組み込んだ印刷資源使用命令を発行する印刷資源使用命令発行手段とを備えたホストコンピュータと、（ロ）このホストコンピュータと回線によって接

5

続され印刷資源が送られてきたときこれを格納する受信側印刷資源格納手段と、印刷資源使用命令を受信する印刷資源使用命令受信手段と、印刷資源使用命令が受信されたとき対象となる印刷資源の利用者アクセス権と受信した要求者アクセス権を比較する比較手段と、比較結果が一致しないときその印刷資源の使用を拒否する使用拒否手段とを備えた印刷装置とを印刷処理システムに具備させる。

【0013】すなわち請求項2記載の発明では、ホストコンピュータから印刷資源を印刷装置に送信してこれに格納する際には、印刷資源と併せてそれを利用することのできる利用者を表わした利用者アクセス権を付加して転送しておく。また、その印刷資源の使用を要求する者に対しては、その者がアクセスすることのできる印刷資源を表わした要求者アクセス権を印刷資源利用命令に組み込んでホストコンピュータが印刷装置に送信することにする。印刷装置側ではこの印刷資源利用命令から抽出された要求者アクセス権と該当する印刷資源に付加された利用者アクセス権を照合し、要求者に印刷資源の利用の権限がない場合には、その利用を拒絶することにして、印刷資源の利用の安全を図るようになっている。

【0014】請求項3記載の発明では、（イ）文字フォント等の印刷に必要とされる印刷資源を格納した送信側印刷資源格納手段と、印刷資源ごとにそれらを削除できる利用者を利用者アクセス権として規定した利用者管理テーブルと、印刷資源の削除要求者ごとに印刷資源のアクセス範囲を要求者アクセス権として規定した要求者管理テーブルと、送信側印刷資源格納手段に格納された印刷資源の送信が要求されたとき印刷資源ごとに対応する利用者アクセス権を付加して転送する印刷資源転送手段と、印刷資源の削除要求があったときその要求者アクセス権を組み込んだ印刷資源削除命令を発行する印刷資源使用命令発行手段とを備えたホストコンピュータと、（ロ）このホストコンピュータと回線によって接続され印刷資源が送られてきたときこれを格納する受信側印刷資源格納手段と、印刷資源削除命令を受信する印刷資源削除命令受信手段と、印刷資源削除命令が受信されたとき対象となる印刷資源の利用者アクセス権と受信した要求者アクセス権を比較する比較手段と、比較結果が一致しないときその印刷資源の削除を拒否する使用拒否手段とを備えた印刷装置とを印刷処理システムに具備させる。

【0015】すなわち請求項3記載の発明では、ホストコンピュータから印刷資源を印刷装置に送信してこれに格納する際には、印刷資源と併せてそれを削除することのできる削除者を表わした削除者アクセス権を付加して転送しておく。また、その印刷資源の使用を要求する者に対しては、その者がアクセスすることのできる印刷資源を表わした要求者アクセス権を印刷資源削除命令に組み込んでホストコンピュータが印刷装置に送信することにする。印刷装置側ではこの印刷資源削除命令から抽出された要求者アクセス権と該当する印刷資源に付加された削除者アクセス権を照合し、要求者に印刷資源の削除の権限がない場合には、その削除を拒絶することにして、印刷資源の削除の際の不都合の発生を防止している。

6

【0016】請求項4記載の発明では、（イ）文字フォント等の印刷に必要とされる印刷資源を格納した送信側印刷資源格納手段と、印刷資源ごとにそれらを更新できる利用者を利用者アクセス権として規定した利用者管理テーブルと、印刷資源の更新要求者ごとに印刷資源のアクセス範囲を要求者アクセス権として規定した要求者管理テーブルと、送信側印刷資源格納手段に格納された印刷資源の送信が要求されたとき印刷資源ごとに対応する利用者アクセス権を付加して転送する印刷資源転送手段と、印刷資源の更新要求があったときその要求者アクセス権を組み込んだ印刷資源更新命令を発行する印刷資源使用命令発行手段とを備えたホストコンピュータと、（ロ）このホストコンピュータと回線によって接続され印刷資源が送られてきたときこれを格納する受信側印刷資源格納手段と、印刷資源更新命令を受信する印刷資源更新命令受信手段と、印刷資源更新命令が受信されたとき対象となる印刷資源の利用者アクセス権と受信した要求者アクセス権を比較する比較手段と、比較結果が一致しないときその印刷資源の更新を拒否する使用拒否手段とを備えた印刷装置とを印刷処理システムに具備させる。

【0017】すなわち請求項4記載の発明では、ホストコンピュータから印刷資源を印刷装置に送信してこれに格納する際には、印刷資源と併せてそれを更新することのできる更新者を表わした更新者アクセス権を付加して転送しておく。また、その印刷資源の使用を要求する者に対しては、その者がアクセスすることのできる印刷資源を表わした要求者アクセス権を印刷資源更新命令に組み込んでホストコンピュータが印刷装置に送信することにする。印刷装置側ではこの印刷資源更新命令から抽出された要求者アクセス権と該当する印刷資源に付加された更新者アクセス権を照合し、要求者に印刷資源の更新の権限がない場合には、その更新を拒絶することにして、印刷資源の更新の際の不都合の発生を防止している。

【0018】

【実施例】以下実施例につき本発明を詳細に説明する。

【0019】第1の実施例

【0020】図1は本発明の第1の実施例における印刷処理システムの原理的な構成を表わしたものである。この印刷処理システムは、印刷資源を格納したホストコンピュータ11と、印刷装置12およびこれらを接続する回線13とによって構成されている。

【0021】ホストコンピュータ11側には、その制御

7

を行う制御部１５が配置されている。制御部１５は、ＣＰＵ（中央処理装置）１６を備えており、図示しないバスを介してＲＯＭ（リード・オンリ・メモリ）１７、ＲＡＭ（ランダム・アクセス・メモリ）１８と接続されている他、図示しない入出力回路を介して端末装置１９と接続されている。ここで端末装置１９は、ＣＲＴ等の出力装置やキーボード等の入力装置によって構成されている。

【００２２】制御部１５は、利用者アクセス権設定部２１、要求者アクセス権取得部２２およびアクセス権管理部２３の制御を行うようになっている。ここで、利用者アクセス権設定部２１とは、磁気ディスク装置２４内に格納されている図示しない利用者アクセス権のついていない印刷資源に対して利用者アクセス権を付け、アクセス権付き印刷資源２５とするための機能部品をいう。ここで利用者アクセス権とは、印刷資源を利用可能な者の範囲を定める情報である。印刷資源の中には、利用者を限定する必要がないもの、すなわち誰でも自由に利用することができる印刷資源も存在するが、本明細書ではこれに対しても“誰でもアクセスできるというアクセス権”を付けるものとして、その設定を利用者アクセス権設定部２１が行うようになっている。アクセス権付き印刷資源２５は、必要に応じて印刷資源送信部２６から回線１３を通じて印刷装置１２側に送出されるようになっている。

【００２３】利用者アクセス権および要求者アクセス権の設定に関する管理はアクセス権管理部２３が行うようになっている。この管理のために、アクセス権管理部２３は磁気ディスク装置２４内にアクセス権管理テーブル２８を備えている。なお、磁気ディスク装置２４には制御部１５が各種制御を行うために必要なプログラムも格納されている。

【００２４】要求者アクセス権取得部２２は、印刷資源の使用時に要求者アクセス権を取得する機能部品である。ここで要求者アクセス権とは、印刷資源の使用を要求するものが有しているアクセス権であり、文書等の印刷情報のアクセスを行うのに必要とされるアクセス権と等しい。印刷資源使用命令部２９は印刷資源の使用が要求されたとき、アクセス権管理部２３の管理の下で要求者アクセス権を取得し、これを付加した印刷資源使用命令を発行する。この印刷資源使用命令は回線１３を通じて印刷装置１２へ送出されるようになっている。

【００２５】次に印刷装置１２の説明を行う。印刷装置１２にも、その制御を行う制御部３１が配置されている。制御部３１は、ＣＰＵ３２を備えており、図示しないバスを介してＲＯＭ３３、ＲＡＭ３４と接続されている。ホストコンピュータ１１の印刷資源送信部２６から送られてきたアクセス権付き印刷資源は、印刷資源受信部３６で受信される。そして、制御部３１の制御を受ける印刷資源格納部３７によって磁気ディスク装置３８内

8

にアクセス権付き印刷資源３９として格納される。なお、磁気ディスク装置３８は、図示しないディスク制御装置を介して制御部３１と接続されており、アクセス権付き印刷資源３９の他に、この制御部３１を制御するための各種のプログラムも格納している。

【００２６】一方、ホストコンピュータ１１から回線１３を通じて送られてくる印刷資源使用命令は印刷資源使用命令応答部４１で受信され、印刷資源アクセス権確認部４２に送られるようになっている。印刷資源アクセス権確認部４２はこの要求者アクセス権をアクセス権付き印刷資源３９に付けられている利用者アクセス権と比較する。そして、要求者にその使用しようとする印刷資源の利用者アクセスが存在する場合にはその印刷資源の使用を許可する。これ以外の場合には、その印刷資源の使用が許可されない。その印刷資源の使用が許可された場合、印刷処理部４３はそのアクセス権付き印刷資源３９を使用して印刷情報の印刷を行う。

【００２７】その印刷資源の使用が許可されなかった場合には、要求者が使用できる範囲の印刷資源を用いて印刷情報の印刷が行われることになる。例えば特殊な文字フォントについて利用者が限定されていて印刷情報の印刷を行う者がその使用を要求した場合に、その使用が認められなかったとする。この際には、その要求者の使用できる他の文字フォントを用いれば印刷が可能である。

【００２８】図２は、この印刷処理システムで印刷資源が印刷装置側に転送されるときの処理の流れを表わしたものである。アクセス権なし印刷資源５１のうち利用者アクセスの付与が求められたものについては、利用者アクセス権設定部２１によって、アクセス権付き印刷資源２５に変更される。このとき、アクセス権管理部２３がその管理を行う。

【００２９】図３は、利用者アクセス権設定部の具体的な構成を表わしたものである。利用者アクセス権設定部２１は、利用者アクセス権の取得をアクセス権管理部２３に対して要求する利用者アクセス権取得要求部５３と、アクセス権なし印刷資源５１に対して利用者アクセス権を設定する印刷資源アクセス権設定部５４と、これらの制御を行う利用者アクセス権設定制御部５５から構成されている。利用者アクセス権設定制御部５５は、制御部１５の制御の下に利用者アクセス権設定部２１の制御を行うことになる。

【００３０】図４は、アクセス権管理部の具体的な構成を表わしたものである。アクセス権管理部２３は、アクセス権の取得要求を利用者アクセス権設定部２１から受信するアクセス権取得要求受信部５７と、アクセス権の取得要求があったときアクセス権管理テーブル２８を用いてアクセス権を取得するアクセス権取得部５８と、取得したアクセス権を図３の利用者アクセス権取得要求部５３に送出するアクセス権送信部５９を有している。アクセス権管理制御部６１は、制御部１５の制御の下で、

これら各部の制御を行うようになっている。

【0031】すなわち、アクセス権管理部23はアクセス権なし印刷資源51についてアクセス権の付加が要求されたときには、どのような印刷資源についてはどのようなアクセスを付与するかを示したアクセス権管理テーブル28を参照して該当するアクセス権を読み出し、これをその印刷資源に付加してアクセス権付き印刷資源25を作成することになる。このアクセス権付き印刷資源25は印刷資源送信部26から送出され、印刷装置12の印刷資源受信部36に受信される。そして、印刷資源格納部37によってアクセス権付き印刷資源39として磁気ディスク装置38（図1）内に格納されることになる。

【0032】図5は、印刷資源の使用時における印刷処理システムの処理の流れを表わしたものである。印刷装置12に印刷を行わせるために、ホストコンピュータ11で印刷資源使用要求が発生すると、要求者アクセス権取得部22はアクセス権管理部23の管理の下で要求者アクセス権を取得する。この要求者アクセス権は、印刷資源使用命令部29に送られ、その印刷資源使用命令発行部63で印刷資源使用命令が発行される。この印刷資源使用命令は、印刷資源使用命令送信部64から印刷装置12へ送信される。

【0033】図6は、要求者アクセス権取得部の具体的な構成を表わしたものである。要求者アクセス権取得部22は要求者アクセス権の取得をアクセス権管理部23に対して要求する要求者アクセス権取得要求部66と、取得された要求者アクセス権を印刷資源使用命令に付加して印刷資源使用命令部29に送出するための使用命令アクセス権設定部67を備えている。要求者アクセス権取得制御部68は、制御部15の制御の下でこれら要求者アクセス権取得要求部66および使用命令アクセス権設定部67の制御を行うようになっている。

【0034】図5に戻って説明を続ける。ホストコンピュータ11の印刷資源使用命令送信部64から送信された印刷資源使用命令は、印刷装置12の印刷資源使用命令応答部41内の印刷資源使用命令受信部71で受信され、印刷資源アクセス権確認部42に送られる。印刷資源アクセス権確認部42は、アクセス権付き印刷資源39を用いてアクセス権の確認を行い、アクセス権がないとされた場合には印刷資源使用命令応答部41内の印刷資源使用応答部72を介してホストコンピュータ11へそれを報告する。

【0035】ホストコンピュータ11はこれに基づいて代替の印刷資源を使用する等の措置を採るか、印刷を中止することになる。また、印刷資源アクセス権確認部42はその印刷資源のアクセス権があると判別した場合には、該当のアクセス権付き印刷資源39の使用を許可し、その印刷資源が印刷処理部43内の印刷資源使用部73に送られて印刷情報の印刷に使用されることになる。

【0036】図7は、印刷資源アクセス権確認部の具体的な構成を表わしたものである。印刷資源アクセス権確認部42は、印刷資源使用命令応答部41を介して印刷資源の使用要求者のアクセス権を取得する要求者アクセス権取得部71と、アクセス権付き印刷資源39を用いて利用者アクセス権を取得する利用者アクセス権取得部72の2つの取得部を備えている。アクセス権確認部73は、これら2種類のアクセス権を比較し、印刷資源についてのアクセス権が存在するかどうかの確認を行う。アクセス権確認結果送信部74はこの結果を印刷資源使用命令応答部41を介してホストコンピュータ11に送信することになる。印刷資源アクセス権確認部42内の印刷資源アクセス権確認制御部75は、印刷装置12側の制御部31によって制御され各部71～74の制御を行う他、印刷資源の使用が許可されたときには該当の印刷資源を印刷処理部43に送出する制御も行うようになっている。

【0037】図8は、このような印刷処理システムにおける印刷資源転送時の処理の流れを表わしたものである。印刷資源を図1に示したホストコンピュータ11から印刷装置12に転送する場合、図3に示した利用者アクセス権設定部21は利用者アクセス権取得要求部53にて利用者アクセス権の取得を要求する。アクセス権管理部23は、図4に示したアクセス権取得要求受信部57でこの要求を受信し、アクセス権取得部58でアクセス権管理テーブル28を参照して該当するアクセス権を取得する（図8ステップS101）。

【0038】アクセス権送信部59は、この利用者アクセス権を利用者アクセス権設定部21に返信した後、制御をこの利用者アクセス権設定部21に返す。利用者アクセス権設定部21では、アクセス権管理部23から受け取った利用者アクセス権を印刷資源アクセス権設定部54において転送対象の印刷資源に付加する。そして、印刷資源送信部26に印刷装置12への印刷資源の転送を依頼する。これにより、利用者アクセス権が付加された印刷資源が印刷装置12に転送される（ステップS102）。

【0039】印刷資源送信部26から転送されてきた利用者アクセス権付き印刷資源は、印刷資源受信部36で受信され（ステップS103）、磁気ディスク装置38に格納される（ステップS104）。

【0040】図9は、転送されてきた印刷資源を使用して印刷情報の印刷を行う場合の処理の流れを表わしたものである。印刷資源の使用要求が発生すると（ステップS201；Y）、要求者アクセス権取得部22はその要求者アクセス権取得要求部66にてアクセス権管理部23に対して要求者アクセス権の取得を要求する。アクセス権管理部23はそのアクセス権取得要求受信部57でこの要求を受信し、アクセス権取得部58でアクセス権

11

管理テーブル２８を用いて該当使用要求者に対する要求者アクセス権を取得する（ステップＳ２０２）。そしてアクセス権送信部５９から要求者アクセス権取得部２２にこれを返信して、制御を要求者アクセス権取得部２２に返す。

【００４１】要求者アクセス権取得部２２では、アクセス権管理部２３から受け取った要求者アクセス権を使用命令アクセス権設定部６７で印刷資源使用命令に付加する。そして、印刷資源使用命令部２９の印刷資源使用命令発行部６３に印刷装置１２への印刷資源使用命令の発行を依頼する。印刷資源使用命令発行部６３はこれを基に印刷資源使用命令を発行する（ステップＳ２０３）。

【００４２】この発行された要求者アクセス権付き印刷資源使用命令は、印刷資源使用命令応答部４１で受信され（ステップＳ２０４）、印刷資源アクセス権確認部４２に渡される。印刷資源アクセス権確認部４２では、要求者アクセス権取得部７１で印刷資源使用命令から要求者アクセス権を取得する（ステップＳ２０５）。また、利用者アクセス権取得部７２では印刷装置１２の磁気ディスク装置３８に格納されたアクセス権付き印刷資源３９から利用者アクセス権を取得する（ステップＳ２０６）。アクセス権確認部７３は以上により取得した要求者アクセス権と利用者アクセス権とを照合確認する（ステップＳ２０７）。そして、確認の結果、要求者が印刷資源を使用できると判別された場合には（ステップＳ２０８；Ｙ）、印刷処理部４３において該当の印刷資源を使用して印刷が行われる（ステップＳ２０９）。

【００４３】これに対して、その要求者が該当する印刷資源の使用を許されていない場合には（ステップＳ２０８；Ｎ）、その印刷資源の使用が拒否される（ステップＳ２１０）。いずれの場合にも、印刷資源使用命令応答部４１は要求に対する応答情報を作成し（ステップＳ２１１）、これを回線１３を通じてホストコンピュータ１１側に送信することになる（ステップＳ２１２）。印刷資源を使用することができないことで印刷を行うことができない場合には、応答情報としてその旨の情報が組み込まれることになる。

【００４４】第２の実施例

【００４５】図１０は本発明の第２の実施例における印刷処理システムの原理的な構成を表わしたものである。この印刷処理システムは、印刷資源を格納したホストコンピュータ１１Ａと、印刷装置１２Ａおよびこれらを接続する回線１３とによって構成されている。なお、この印刷処理システムはその構成が第１の実施例のそれと共通した箇所が多いので、回路装置の内容が同一の箇所には図１等と同一の符号を付し、これらの説明を適宜省略する。また、回路装置の内容自体は異なるが、第１の実施例と対応させて説明することが便利であるような回路装置についてはそれらの箇所に用いた符号の末尾に“Ａ”という符号を付加することにする。

12

【００４６】ホストコンピュータ１１Ａは、第１の実施例の印刷資源使用命令部２９の代わりに印刷資源削除命令部２９Ａを備えている。印刷装置１２Ａは第１の実施例の印刷資源使用命令応答部４１の代わりに印刷資源削除命令応答部４１Ａを、また印刷処理部４３の代わりに印刷資源削除部４３Ａを備えている。

【００４７】ここで、印刷資源削除命令部２９Ａは、印刷資源の削除が要求されたとき、アクセス権管理部２３の管理の下で要求者アクセス権を取得し、これを付加した印刷資源削除命令を発行するためのもである。この印刷資源削除命令は回線１３を通じて印刷装置１２Ａ送出されるようになっている。

【００４８】一方、ホストコンピュータ１１Ａから回線１３を通じて送られてくる印刷資源削除命令は印刷資源削除命令応答部４１Ａで受信され、印刷資源アクセス権確認部４２に送られるようになっている。印刷資源アクセス権確認部４２はこの要求者アクセス権をアクセス権付き印刷資源３９に付けられている利用者アクセス権と比較する。そして、要求者にその削除しようとする印刷資源の利用者アクセスが存在する場合にはその印刷資源の削除を許可する。これ以外の場合には、その印刷資源の削除が許可されない。その印刷資源の削除が許可された場合、印刷資源削除部４３Ａはアクセス権付き印刷資源３９内の該当する印刷資源を削除することになる。

【００４９】図１１は、この印刷処理システムで印刷資源が印刷装置側に転送されるときの処理の流れを表わしたものである。アクセス権なし印刷資源５１のうち利用者アクセスの付与が求められたものについては、利用者アクセス権設定部２１によって、アクセス権付き印刷資源２５に変更される。このとき、アクセス権管理部２３がその管理を行う。

【００５０】図１２は、利用者アクセス権設定部の具体的な構成を表わしたものである。利用者アクセス権設定部２１は、利用者アクセス権の取得をアクセス権管理部２３に対して要求する利用者アクセス権取得要求部５３と、アクセス権なし印刷資源５１に対して利用者アクセス権を設定する印刷資源アクセス権設定部５４と、これらの制御を行う利用者アクセス権設定制御部５５から構成されている。利用者アクセス権設定制御部５５は、制御部１５Ａの制御の下に利用者アクセス権設定部２１の制御を行うことになる。

【００５１】図１３は、アクセス権管理部の具体的な構成を表わしたものである。アクセス権管理部２３は、アクセス権の取得要求を利用者アクセス権設定部２１から受信するアクセス権取得要求受信部５７と、アクセス権の取得要求があったときアクセス権管理テーブル２８Ａを用いてアクセス権を取得するアクセス権取得部５８と、取得したアクセス権を図１２の利用者アクセス権取得要求部５３に送出するアクセス権送信部５９を有している。アクセス権管理制御部６１は、制御部１５Ａの制

御の下で、これら各部の制御を行うようになっている。

【0052】すなわち、アクセス権管理部23はアクセス権なし印刷資源51についてアクセス権の付加が要求されたときには、どのような印刷資源についてはどのようなアクセスを付与するかを示したアクセス権管理テーブル28Aを参照して該当するアクセス権を読み出し、これをその印刷資源に付加してアクセス権付き印刷資源25を作成することになる。このアクセス権付き印刷資源25は印刷資源送信部26から送出され、印刷装置12Aの印刷資源受信部36に受信される。そして、印刷資源格納部37によってアクセス権付き印刷資源39として磁気ディスク装置38（図10）内に格納されることになる。

【0053】図14は、印刷資源の削除時における印刷処理システムの処理の流れを表わしたものである。印刷資源の削除を行わせるために、ホストコンピュータ11Aで印刷資源削除要求が発生すると、要求者アクセス権取得部22Aはアクセス権管理部23の管理の下で要求者アクセス権を取得する。この要求者アクセス権は、印刷資源削除命令部29Aに送られ、その印刷資源削除命令発行部63Aで印刷資源削除命令が発行される。この印刷資源削除命令は、印刷資源削除命令送信部64から印刷装置12Aへ送信される。

【0054】図15は、要求者アクセス権取得部の具体的な構成を表わしたものである。要求者アクセス権取得部22Aは要求者アクセス権の取得をアクセス権管理部23に対して要求する要求者アクセス権取得要求部66と、取得された要求者アクセス権を印刷資源削除命令に付加して印刷資源削除命令部29Aに送出するための削除命令アクセス権設定部67を備えている。要求者アクセス権取得制御部68は、制御部15Aの制御の下でこれら要求者アクセス権取得要求部66および削除命令アクセス権設定部67の制御を行うようになっている。

【0055】図14に戻って説明を続ける。ホストコンピュータ11Aの印刷資源削除命令送信部64Aから送信された印刷資源削除命令は、印刷装置12Aの印刷資源削除命令応答部41A内の印刷資源削除命令受信部71Aで受信され、印刷資源アクセス権確認部42に送られる。印刷資源アクセス権確認部42は、アクセス権付き印刷資源39を用いてアクセス権の確認を行い、アクセス権がないとされた場合には印刷資源削除命令応答部41A内の印刷資源削除応答部72Aを介してホストコンピュータ11Aへそれを報告する。また、印刷資源アクセス権確認部42はその印刷資源のアクセス権があると判別した場合には、該当のアクセス権付き印刷資源39の削除を許可する。この場合にも、削除の実行結果がホストコンピュータ11Aに報告される。

【0056】図16は、印刷資源アクセス権確認部の具体的な構成を表わしたものである。印刷資源アクセス権確認部42は、印刷資源削除命令応答部41Aを介して

印刷資源の削除要求者のアクセス権を取得する要求者アクセス権取得部71Aと、アクセス権付き印刷資源39を用いて利用者アクセス権を取得する利用者アクセス権取得部72Aの2つの取得部を備えている。アクセス権確認部73は、これら2種類のアクセス権を比較し、印刷資源についてのアクセス権が存在するかどうかの確認を行う。

【0057】アクセス権確認結果送信部74はこの結果を印刷資源削除命令応答部41Aを介してホストコンピュータ11Aに送信することになる。印刷資源アクセス権確認部42内の印刷資源アクセス権確認制御部75は、印刷装置12A側の制御部31Aによって制御され各部71A～74の制御を行う他、印刷資源の削除が許可されたときには印刷資源削除部43Aが印刷資源を削除する際の制御も行うようになっている。

【0058】なお、先の第1の実施例ではその図8で印刷処理システムにおける印刷資源転送時の処理の流れを表わしたが第2の実施例ではこの点について実質的な変更がないので、これについての説明を省略する。

【0059】図17は、転送されてきた印刷資源を削除する場合の処理の流れを表わしたものである。印刷資源の削除要求が発生すると（ステップS301；Y）、要求者アクセス権取得部22Aはその要求者アクセス権取得要求部66にてアクセス権管理部23に対して要求者アクセス権の取得を要求する。アクセス権管理部23はそのアクセス権取得要求受信部57でこの要求を受信し、アクセス権取得部58でアクセス権管理テーブル28Aを用いて該当削除要求者に対する要求者アクセス権を取得する（ステップS302）。そしてアクセス権送信部59から要求者アクセス権取得部22Aにこれを返信して、制御を要求者アクセス権取得部22Aに返す。

【0060】要求者アクセス権取得部22Aでは、アクセス権管理部23から受け取った要求者アクセス権を削除命令アクセス権設定部67で印刷資源削除命令に付加する。そして、印刷資源削除命令部29Aの印刷資源削除命令発行部63Aに印刷装置12Aへの印刷資源削除命令の発行を依頼する。印刷資源削除命令発行部63Aはこれを基に印刷資源削除命令を発行する（ステップS303）。

【0061】この発行された要求者アクセス権付き印刷資源削除命令は、印刷資源削除命令応答部41Aで受信され（ステップS304）、印刷資源アクセス権確認部42に渡される。印刷資源アクセス権確認部42では、要求者アクセス権取得部71Aで印刷資源削除命令から要求者アクセス権を取得する（ステップS305）。また、利用者アクセス権取得部72Aでは印刷装置12Aの磁気ディスク装置38に格納されたアクセス権付き印刷資源39から利用者アクセス権を取得する（ステップS306）。アクセス権確認部73は以上により取得した要求者アクセス権と利用者アクセス権とを照合確認す

る（ステップS307）。そして、確認の結果、要求者が印刷資源を削除できると判別された場合には（ステップS308；Y）、印刷資源削除部43Aにおいて該当の印刷資源を削除する（ステップS309）。

【0062】これに対して、その要求者が該当する印刷資源の削除を許されていない場合には（ステップS308；N）、その印刷資源の削除が拒否される（ステップS310）。いずれの場合にも、印刷資源削除命令応答部41Aは要求に対する応答情報を作成し（ステップS311）、これを回線13を通じてホストコンピュータ11A側に送信することになる（ステップS312）。

【0063】第3の実施例

【0064】図18は本発明の第3の実施例における印刷処理システムの原理的な構成を表わしたものである。この印刷処理システムは、印刷資源を格納したホストコンピュータ11Bと、印刷装置12Bおよびこれらを接続する回線13とによって構成されている。なお、この印刷処理システムはその構成が第1の実施例のそれと共通した箇所が多いので、回路装置の内容が同一の箇所には図1等と同一の符号を付し、これらの説明を適宜省略する。また、回路装置の内容自体は異なるが、第1の実施例と対応させて説明することが便利であるような回路装置についてはそれらの箇所に用いた符号の末尾に“B”という符号を付加することにする。

【0065】ホストコンピュータ11Bは、第1の実施例の印刷資源使用命令部29の代わりに印刷資源確認命令部29Bを備えている。また、利用者アクセス権設定部21は印刷資源転送・更新部81を介して制御部15Bと接続されている。印刷装置12Bは第1の実施例の印刷資源使用命令応答部41の代わりに印刷資源確認命令応答部41Bを備えている。

【0066】ここで、印刷資源転送・更新部81は、印刷資源の転送指示と更新指示を行うようになっている。印刷資源確認命令部29Bは、印刷資源の更新が要求されたとき、アクセス権管理部23の管理の下で要求者アクセス権を取得し、これを付加した印刷資源確認命令を発行するためのものである。この印刷資源確認命令は回線13を通じて印刷装置12Bへ送出されるようになっている。

【0067】一方、ホストコンピュータ11Bから回線13を通じて送られてくる印刷資源確認命令は印刷資源確認命令応答部41Bで受信され、印刷資源アクセス権確認部42に送られるようになっている。印刷資源アクセス権確認部42はこの要求者アクセス権をアクセス権付き印刷資源39に付けられている利用者アクセス権と比較する。そして、要求者にその更新しようとする印刷資源の利用者アクセスが存在する場合にはその印刷資源の更新を許可する。これ以外の場合には、その印刷資源の更新が許可されない。

【0068】図19は、この印刷処理システムで印刷資源が印刷装置側に転送・更新されるときの処理の流れを表わしたものである。アクセス権なし印刷資源51のうち利用者アクセスの付与が求められたものについては、利用者アクセス権設定部21によって、アクセス権付き印刷資源25に変更される。このとき、アクセス権管理部23がその管理を行う。

【0069】図20は、利用者アクセス権設定部の具体的な構成を表わしたものである。利用者アクセス権設定部21は、利用者アクセス権の取得をアクセス権管理部23に対して要求する利用者アクセス権取得要求部53と、アクセス権なし印刷資源51に対して利用者アクセス権を設定する印刷資源アクセス権設定部54と、これらの制御を行う利用者アクセス権設定制御部55から構成されている。制御部15Bは、印刷資源転送・更新部81を介して利用者アクセス権設定制御部55を制御している。

【0070】図21は、アクセス権管理部の具体的な構成を表わしたものである。アクセス権管理部23は、アクセス権の取得要求を利用者アクセス権設定部21から受信するアクセス権取得要求受信部57と、アクセス権の取得要求があったときアクセス権管理テーブル28Bを用いてアクセス権を取得するアクセス権取得部58と、取得したアクセス権を図20の利用者アクセス権取得要求部53に送出するアクセス権送信部59を有している。アクセス権管理制御部61は、制御部15Bの制御の下で、これら各部の制御を行うようになっている。

【0071】すなわち、アクセス権管理部23はアクセス権なし印刷資源51についてアクセス権の付加が要求されたときには、どのような印刷資源についてはどのようなアクセスを付与するかを示したアクセス権管理テーブル28Bを参照して該当するアクセス権を読み出し、これをその印刷資源に付加してアクセス権付き印刷資源25を作成することになる。このアクセス権付き印刷資源25は印刷資源送信部26から送出され、印刷装置12Bの印刷資源受信部36に受信される。そして、印刷資源転送時には印刷資源格納部37によってアクセス権付き印刷資源39として磁気ディスク装置38（図18）内に格納される。これに対して印刷資源更新時には、印刷資源更新部83によって更新され、その内容がアクセス権付き印刷資源39として磁気ディスク装置38内に格納されることになる。

【0072】図22は、印刷資源の更新時における印刷処理システムの処理の流れを表わしたものである。印刷資源の更新を行わせるために、ホストコンピュータ11Bで印刷資源更新要求が発生すると、要求者アクセス権取得部22Bはアクセス権管理部23の管理の下で要求者アクセス権を取得する。この要求者アクセス権は、印刷資源確認命令部29Bに送られ、その印刷資源確認命令発行部63Bで印刷資源確認命令が発行される。この印刷資源確認命令は、印刷資源確認命令送信部64から

17

印刷装置１２Ｂへ送信される。

【００７３】図２３は、要求者アクセス権取得部の具体的な構成を表わしたものである。要求者アクセス権取得部２２Ｂは要求者アクセス権の取得をアクセス権管理部２３に対して要求する要求者アクセス権取得要求部６６と、取得された要求者アクセス権を印刷資源確認命令に付加して印刷資源確認命令部２９Ｂに送出するための確認命令アクセス権設定部６７を備えている。要求者アクセス権取得制御部６８は、制御部１５Ｂの制御の下でこれら要求者アクセス権取得要求部６６および確認命令アクセス権設定部６７Ｂの制御を行うようになっている。

【００７４】図２２に戻って説明を続ける。ホストコンピュータ１１Ｂの印刷資源確認命令送信部６４Ｂから送信された印刷資源確認命令は、印刷装置１２Ｂの印刷資源確認命令応答部４１Ｂ内の印刷資源確認命令受信部７１Ｂで受信され、印刷資源アクセス権確認部４２に送られる。印刷資源アクセス権確認部４２は、アクセス権付き印刷資源３９を用いてアクセス権の確認を行い、アクセス権がないとされた場合には印刷資源確認命令応答部４１Ｂ内の印刷資源更新応答部７２Ｂを介してホストコンピュータ１１Ｂへそれを報告する。また、印刷資源アクセス権確認部４２はその印刷資源のアクセス権があると判別した場合には、印刷資源更新応答部７２Ｂを介してホストコンピュータ１１Ｂへ更新する印刷資源の転送を要求する。

【００７５】図２４は、印刷資源アクセス権確認部の具体的な構成を表わしたものである。印刷資源アクセス権確認部４２は、印刷資源確認命令応答部４１Ｂを介して印刷資源の更新要求者のアクセス権を取得する要求者アクセス権取得部７１Ｂと、アクセス権付き印刷資源３９を用いて利用者アクセス権を取得する利用者アクセス権取得部７２Ｂの２つの取得部を備えている。アクセス権確認部７３は、これら２種類のアクセス権を比較し、印刷資源についてのアクセス権が存在するかどうかの確認を行う。アクセス権確認結果送信部７４はこの結果を印刷資源確認命令応答部４１Ｂを介してホストコンピュータ１１Ｂに送信することになる。印刷資源アクセス権確認部４２内の印刷資源アクセス権確認制御部７５は、印刷装置１２Ｂ側の制御部３１Ｂによって制御され各部７１Ｂ～７４の制御を行う。

【００７６】なお、先の第１の実施例ではその図８で印刷処理システムにおける印刷資源転送時の処理の流れを表わしたが第３の実施例でもこの点について実質的な変更がないので、これについての説明を省略する。

【００７７】図２５は、転送されてきた印刷資源を更新する場合の処理の流れを表わしたものである。印刷資源の更新要求が発生すると（ステップＳ４０１；Ｙ）、要求者アクセス権取得部２２Ｂはその要求者アクセス権取得要求部６６にてアクセス権管理部２３に対して要求者アクセス権の取得を要求する。アクセス権管理部２３は

18

そのアクセス権取得要求受信部５７でこの要求を受信し、アクセス権取得部５８でアクセス権管理テーブル２８Ｂを用いて該当更新要求者に対する要求者アクセス権を取得する（ステップＳ４０２）。そしてアクセス権送信部５９から要求者アクセス権取得部２２Ｂにこれを返信して、制御を要求者アクセス権取得部２２Ｂに返す。

【００７８】要求者アクセス権取得部２２Ｂでは、アクセス権管理部２３から受け取った要求者アクセス権を確認命令アクセス権設定部６７Ｂで印刷資源確認命令に付加する。そして、印刷資源確認命令部２９Ｂの印刷資源確認命令発行部６３Ｂに印刷装置１２Ｂへの印刷資源確認命令の発行を依頼する。印刷資源確認命令発行部６３Ｂはこれを基に印刷資源確認命令を発行する（ステップＳ４０３）。

【００７９】この発行された要求者アクセス権付き印刷資源確認命令は、印刷資源確認命令応答部４１Ｂで受信され（ステップＳ４０４）、印刷資源アクセス権確認部４２に渡される。印刷資源アクセス権確認部４２では、要求者アクセス権取得部７１Ｂで印刷資源確認命令から要求者アクセス権を取得する（ステップＳ４０５）。また、利用者アクセス権取得部７２Ｂでは印刷装置１２Ｂの磁気ディスク装置３８に格納されたアクセス権付き印刷資源３９から利用者アクセス権を取得する（ステップＳ４０６）。アクセス権確認部７３は以上により取得した要求者アクセス権と利用者アクセス権とを照合確認する（ステップＳ４０７）。そして、確認の結果、要求者が印刷資源を更新できると判別された場合には（ステップＳ４０８；Ｙ）、印刷資源の更新を許可する（ステップＳ４０９）。

【００８０】これに対して、その要求者が該当する印刷資源の更新を許されていない場合には（ステップＳ４０８；Ｎ）、その印刷資源の更新が拒否される（ステップＳ４１０）。印刷資源の更新が拒否された場合には、要求に対する応答情報を作成してホストコンピュータ１１Ｂへ報告し（ステップＳ４１１）、処理を終了させる（エンド）。

【００８１】印刷資源の更新が許可された場合には（ステップＳ４０９）、この要求に対する応答情報が作成されてアクセス権確認結果送信部７４からホストコンピュータ１１Ｂへの報告が行われ（ステップＳ４１２）、印刷資源転送・更新部８１にて該当する印刷資源の更新が依頼される。これにより、印刷資源の更新が行われることになる（ステップＳ４１３）。

【００８２】なお、以上説明した第１～第３の実施例ではアクセス権管理テーブルから要求者アクセス権と利用者アクセス権双方を取得するようにしたが、２つのテーブルを用意し、取得先を異ならせるようにしてもよい。

【００８３】また、実施例では印刷資源の印刷時の使用と、印刷資源自体の削除あるいは更新を例に挙げて説明したが、これ以外の印刷資源の処理についても本発明が

適用されることはもちろんである。

【0084】

【発明の効果】以上説明したように請求項1記載の発明によれば、ホストコンピュータが必要な印刷資源を印刷装置に転送してその受信側印刷資源格納手段に格納された印刷資源を処理させるようにした印刷処理システムにおいて、印刷資源には利用者アクセス権を付けて印刷装置に転送するようにし、印刷資源の処理の要求があったときにはその要求者のアクセス権を印刷装置に送出するようにした。したがって、印刷装置側ではこれらのアクセス権を照合するだけで該当する印刷資源の処理が妥当であるかどうかを判別することができ、アクセス権を持たない第三者の不適当な要求を拒絶することができる。これにより、第三者の印刷資源処理の乱用を防止し、印刷資源適切な保護が可能になる。

【0085】また請求項2記載の発明によれば、ホストコンピュータが必要な印刷資源を印刷装置に転送してその受信側印刷資源格納手段に格納された印刷資源を印刷のために使用させるようにした印刷処理システムにおいて、印刷資源には利用者アクセス権を付けて印刷装置に転送するようにし、印刷資源の使用の要求があったときにはその要求者のアクセス権を印刷装置に送出するようにした。したがって、印刷装置側ではこれらのアクセス権を照合するだけで該当する印刷資源の使用が妥当であるかどうかを判別することができ、アクセス権を持たない第三者の不適当な要求を拒絶することができる。これにより、第三者の印刷資源利用の乱用を防止し、印刷資源適切な保護が可能になる。

【0086】更に請求項3記載の発明によれば、ホストコンピュータが必要な印刷資源を印刷装置に転送してその受信側印刷資源格納手段に格納された印刷資源を必要に応じて削除させるようにした印刷処理システムにおいて、印刷資源には利用者アクセス権を付けて印刷装置に転送するようにし、印刷資源の削除の要求があったときにはその要求者のアクセス権を印刷装置に送出するようにした。したがって、印刷装置側ではこれらのアクセス権を照合するだけで該当する印刷資源の削除が妥当であるかどうかを判別することができ、アクセス権を持たない第三者の不適当な要求を拒絶することができる。これにより、第三者の印刷資源利用の乱用を防止し、印刷資源適切な保護が可能になる。

【0087】また請求項4記載の発明によれば、ホストコンピュータが必要な印刷資源を印刷装置に転送してその受信側印刷資源格納手段に格納された印刷資源を必要に応じて更新させるようにした印刷処理システムにおいて、印刷資源には利用者アクセス権を付けて印刷装置に転送するようにし、印刷資源の更新の要求があったときにはその要求者のアクセス権を印刷装置に送出するようにした。したがって、印刷装置側ではこれらのアクセス権を照合するだけで該当する印刷資源の更新が妥当であるかどうかを判別することができ、アクセス権を持たない第三者の不適当な要求を拒絶することができる。これにより、第三者の印刷資源利用の乱用を防止し、印刷資源適切な保護が可能になる。

【図面の簡単な説明】

【図1】 本発明の第1の実施例における印刷処理システムの原理的な構成を表わしたブロック図である。

【図2】 この印刷処理システムで印刷資源が印刷装置側に転送されるときの処理の流れを表わした説明図である。

【図3】 第1の実施例の利用者アクセス権設定部の具体的な構成を表わしたブロック図である。

【図4】 第1の実施例のアクセス権管理部の具体的な構成を表わしたブロック図である。

【図5】 印刷資源の使用時における印刷処理システムの処理の流れを表わした説明図である。

【図6】 第1の実施例の要求者アクセス権取得部の具体的な構成を表わしたブロック図である。

【図7】 第1の実施例の印刷資源アクセス権確認部の具体的な構成を表わしたブロック図である。

【図8】 第1の実施例の印刷処理システムにおける印刷資源転送時の処理の流れを表わした流れ図である。

【図9】 第1の実施例で転送されてきた印刷資源を使用して印刷情報の印刷を行う場合の処理の流れを表わした流れ図である。

【図10】 本発明の第2の実施例における印刷処理システムの原理的な構成を表わしたブロック図である。

【図11】 第2の実施例の印刷処理システムで印刷資源が印刷装置側に転送されるときの処理の流れを表わした説明図である。

【図12】 第2の実施例の利用者アクセス権設定部の具体的な構成を表わしたブロック図である。

【図13】 第2の実施例のアクセス権管理部の具体的な構成を表わしたブロック図である。

【図14】 印刷資源の削除時における第2の実施例の印刷処理システムの処理の流れを表わした説明図である。

【図15】 第2の実施例の要求者アクセス権取得部の具体的な構成を表わしたブロック図である。

【図16】 第2の実施例の印刷資源アクセス権確認部の具体的な構成を表わしたブロック図である。

【図17】 第2の実施例で転送されてきた印刷資源を削除する場合の処理の流れを表わした流れ図である。

【図18】 本発明の第3の実施例における印刷処理システムの原理的な構成を表わしたブロック図である。

【図19】 この印刷処理システムで印刷資源が印刷装置側に転送・更新されるときの処理の流れを表わした説明図である。

【図20】 第3の実施例の利用者アクセス権設定部の具体的な構成を表わしたブロック図である。

21　　22

【図２１】　第３の実施例のアクセス権管理部の具体的な構成を表わしたブロック図である。

【図２２】　印刷資源の更新時における第３の実施例の印刷処理システムの処理の流れを表わした説明図である。

【図２３】　第３の実施例の要求者アクセス権取得部の具体的な構成を表わしたブロック図である。

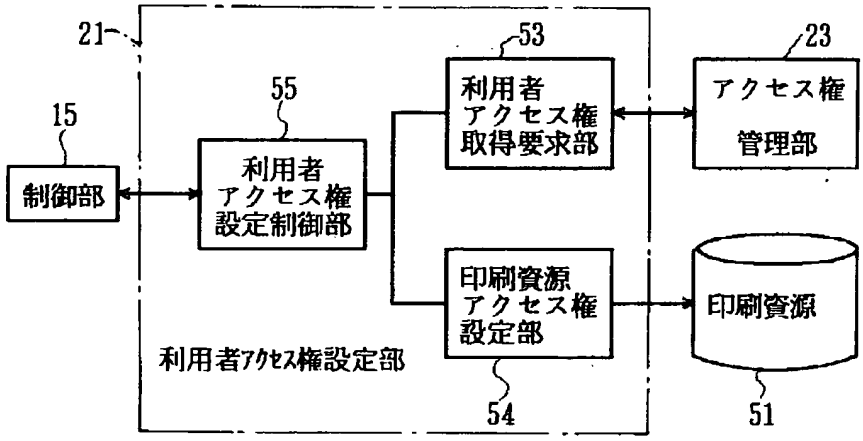
【図２４】　第３の実施例の印刷資源アクセス権確認部の具体的な構成を表わしたブロック図である。

【図２５】　第３の実施例で転送されてきた印刷資源を更新する場合の処理の流れを表わした流れ図である。

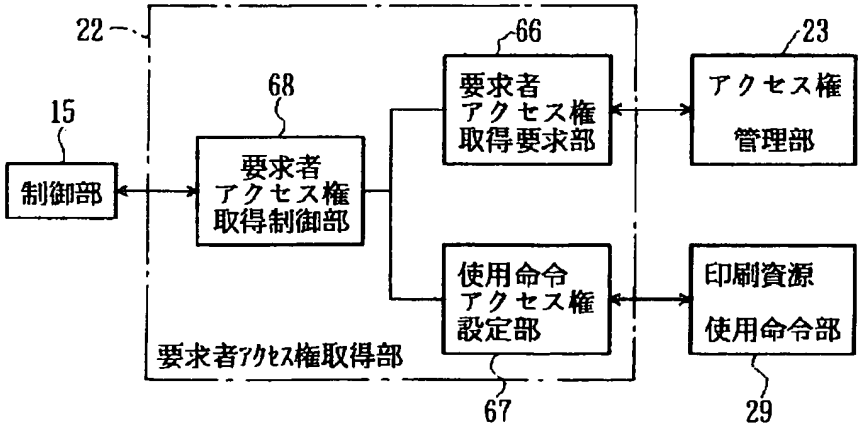
【符号の説明】

１１、１１Ａ、１１Ｂ…ホストコンピュータ、１２、１２Ａ、１２Ｂ…印刷装置、１３…回線、１５、１５Ａ、１５Ｂ…（ホストコンピュータの）制御部、１６…ＣＰＵ、１７…ＲＯＭ、１８…ＲＡＭ、１９…端末装置、２１…利用者アクセス権設定部、２２…要求者アクセス権設定部、２３…アクセス権管理部、２４、３８…磁気ディスク装置、２５…アクセス権付き印刷資源、２６…印刷資源送信部、２８、２８Ａ、２８Ｂ…アクセス権管理テーブル、２９…印刷資源使用命令部、２９Ａ…印刷資源削除命令部、２９Ｂ…印刷資源確認命令部、３１、３１Ａ、３１Ｂ…（印刷装置の）制御部、３６…印刷資源受信部、３７…印刷資源格納部、３９…アクセス権付き印刷資源、４１…印刷資源使用命令応答部、４１Ａ…印刷資源削除命令応答部、４１Ｂ…印刷資源確認命令応答部、４２…印刷資源アクセス権確認部、４３…印刷処理部、４３Ａ…印刷資源削除部、５１…アクセス権なし印刷資源、７１…要求者アクセス権取得部、７２…利用者アクセス権取得部、８１…印刷資源転送・更新部、８３…印刷資源更新部

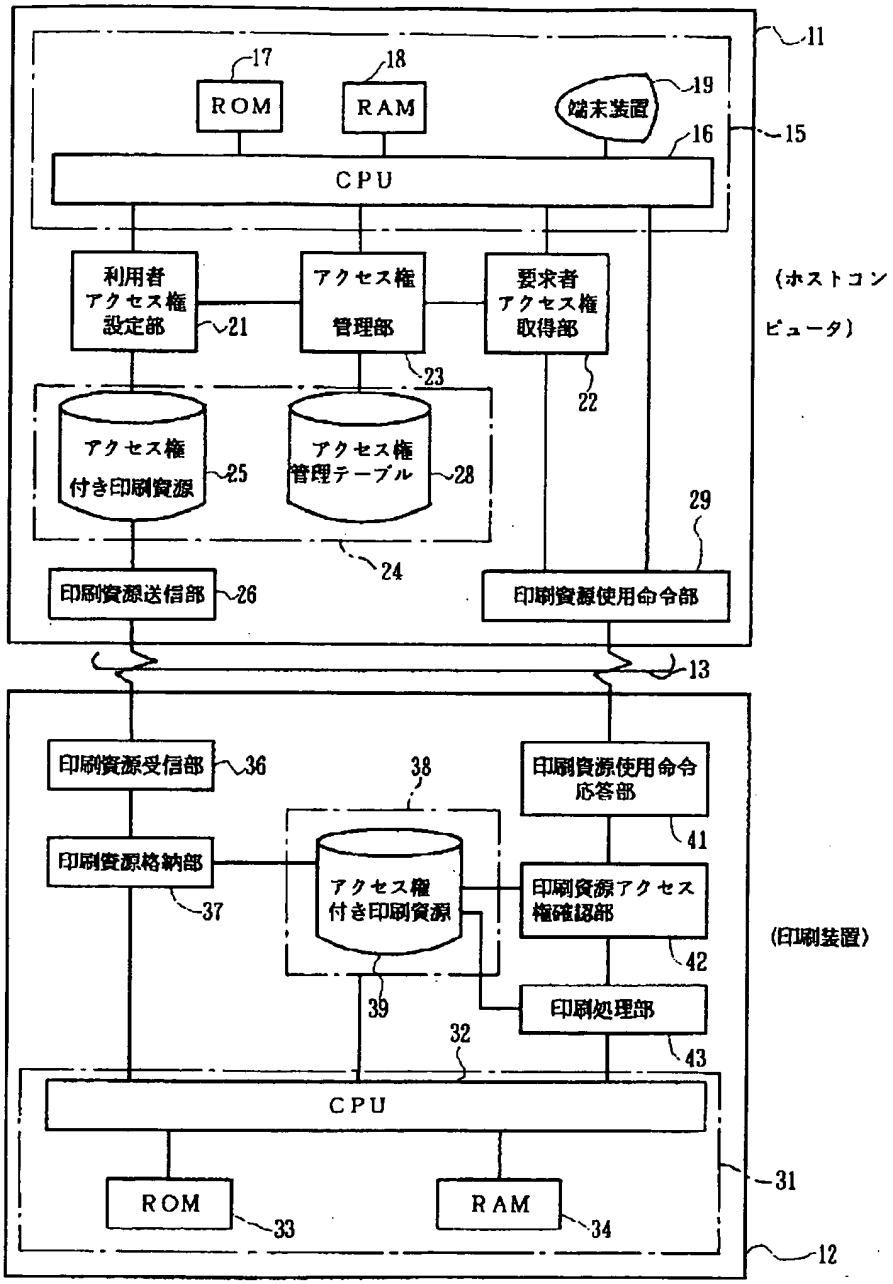
【図３】



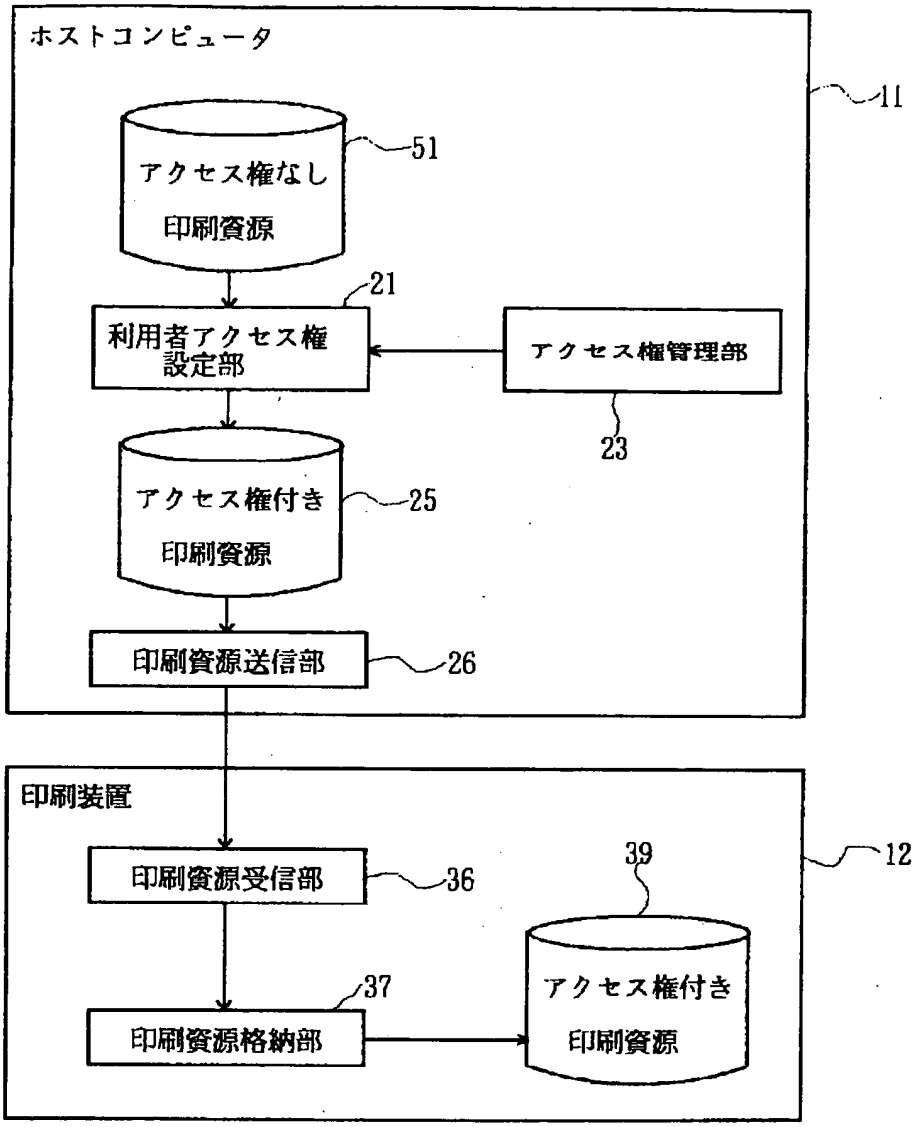
【図６】



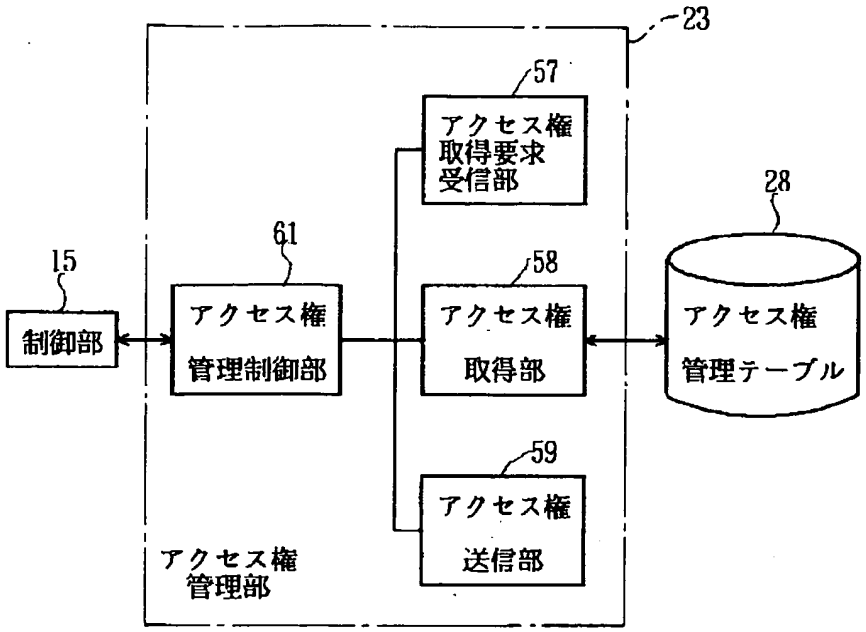
【図1】



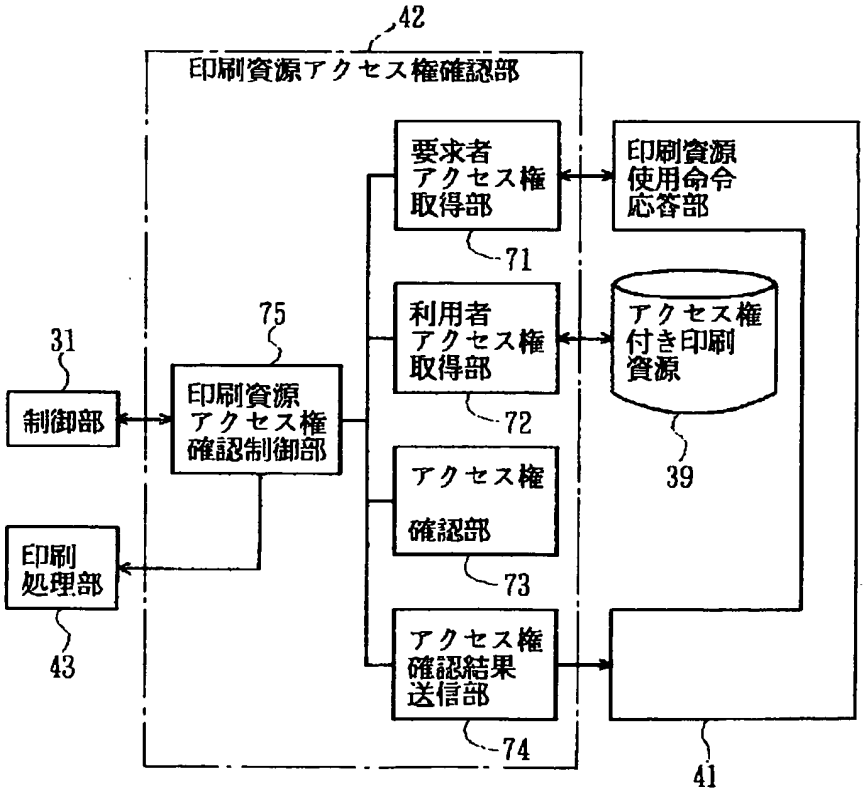
【図2】



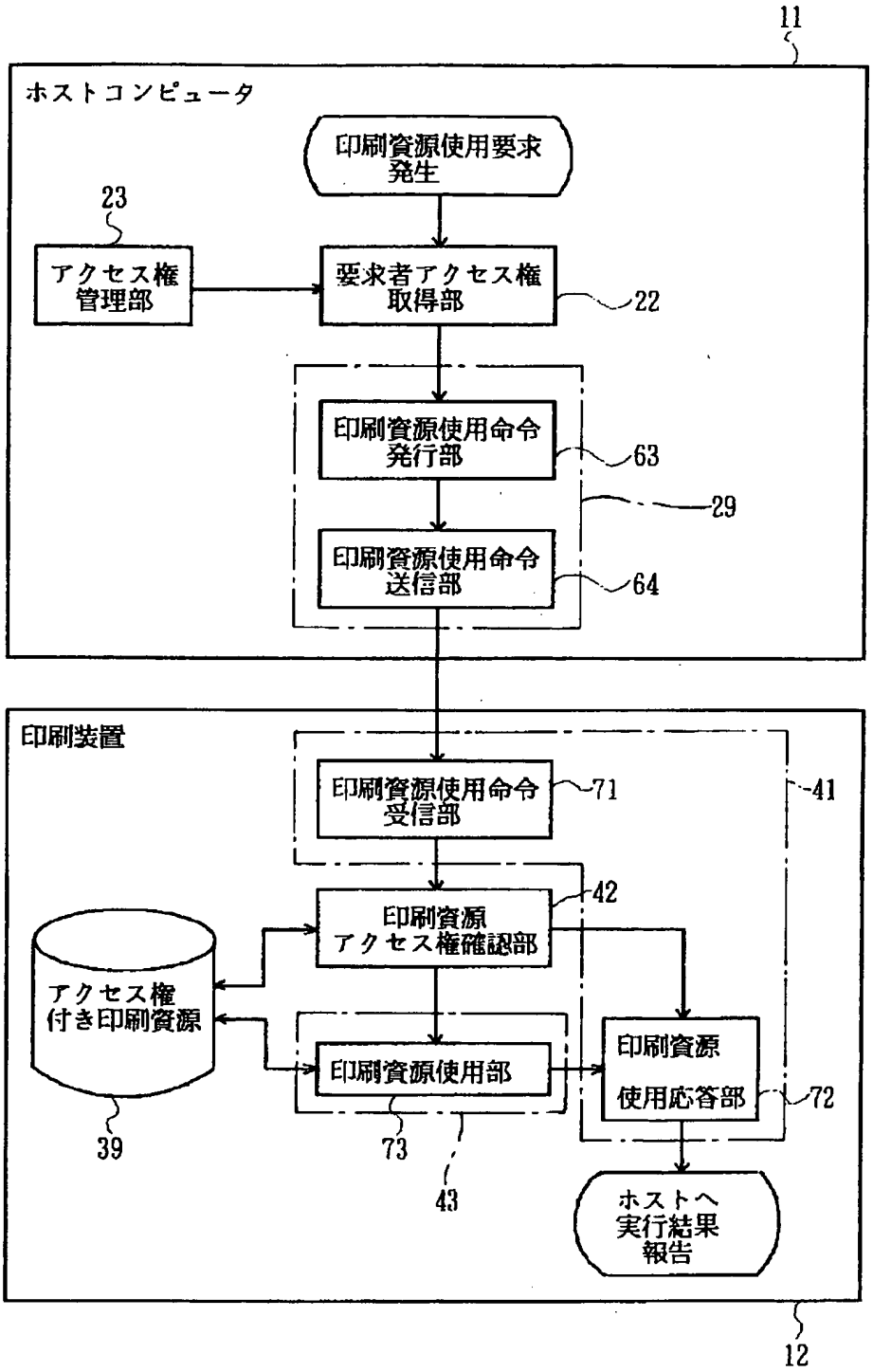
【図４】



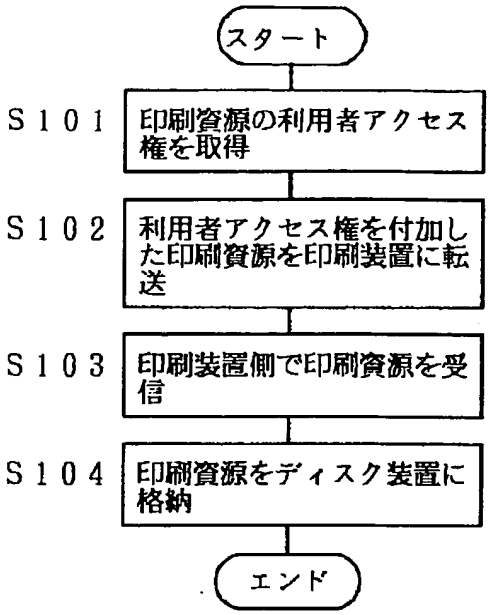
【図７】



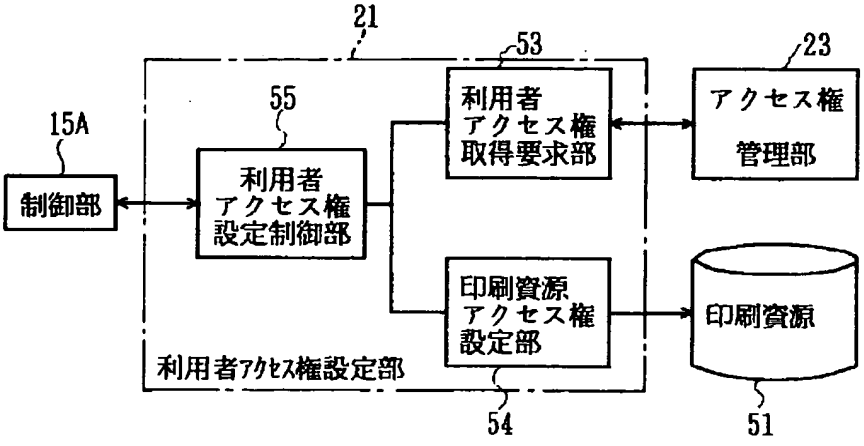
【図5】



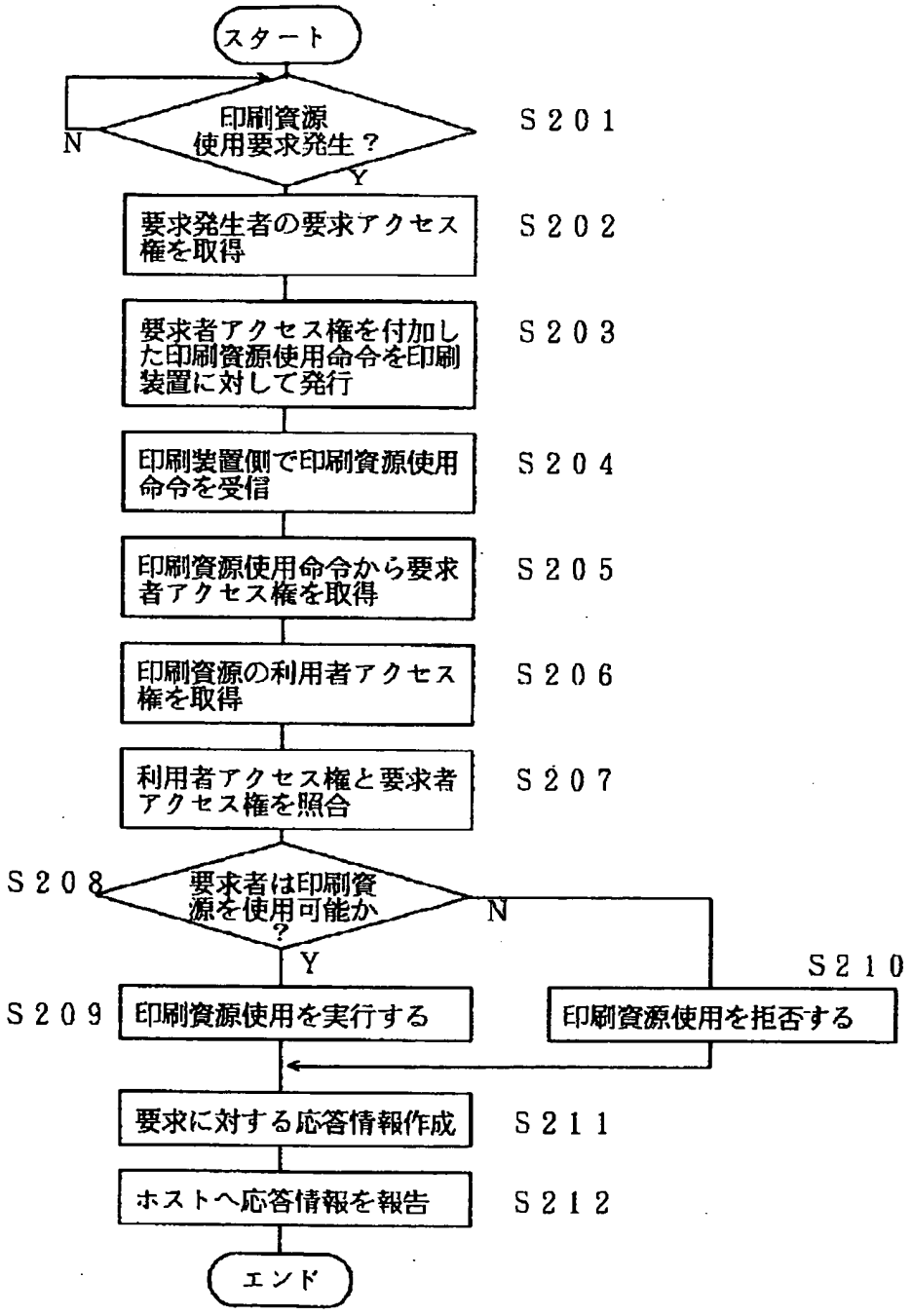
【図8】



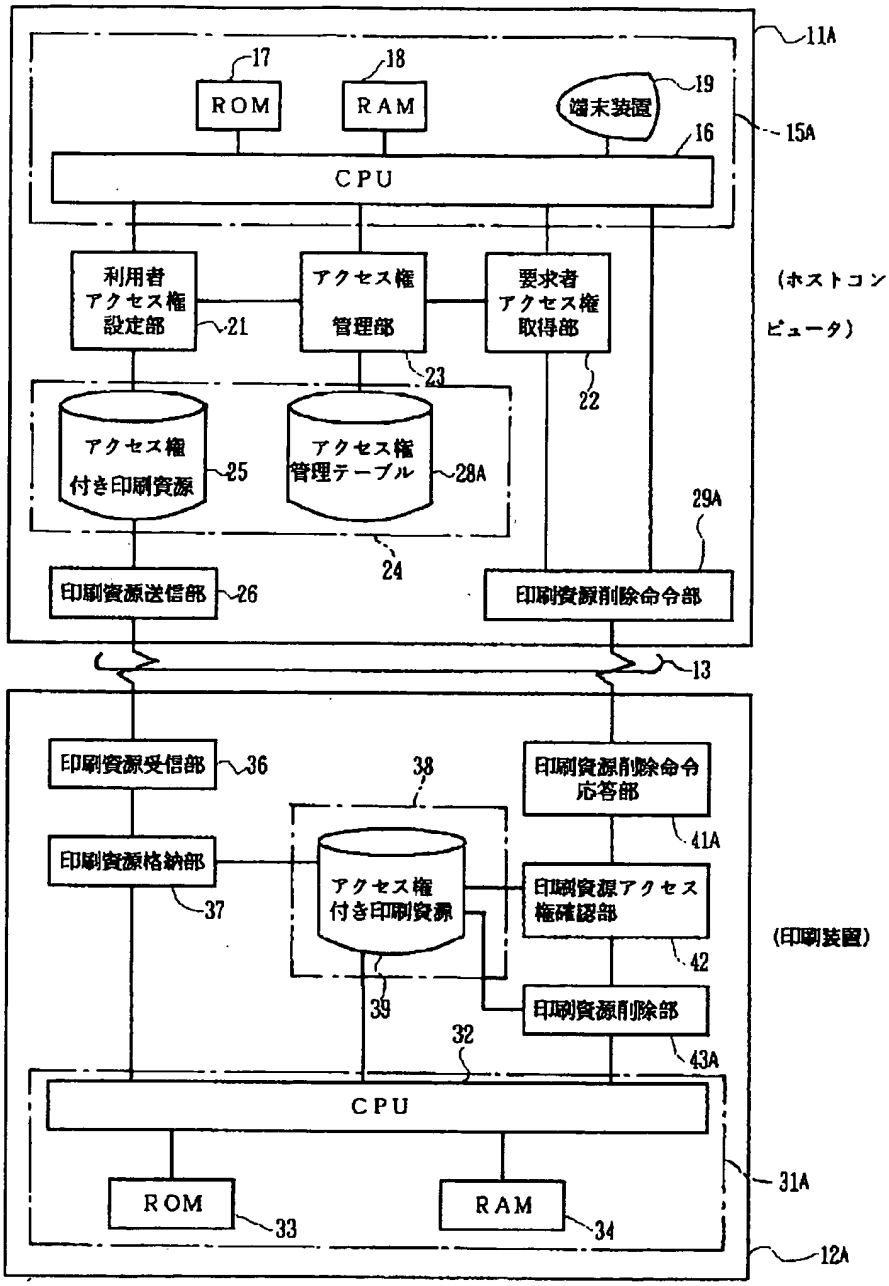
【図12】



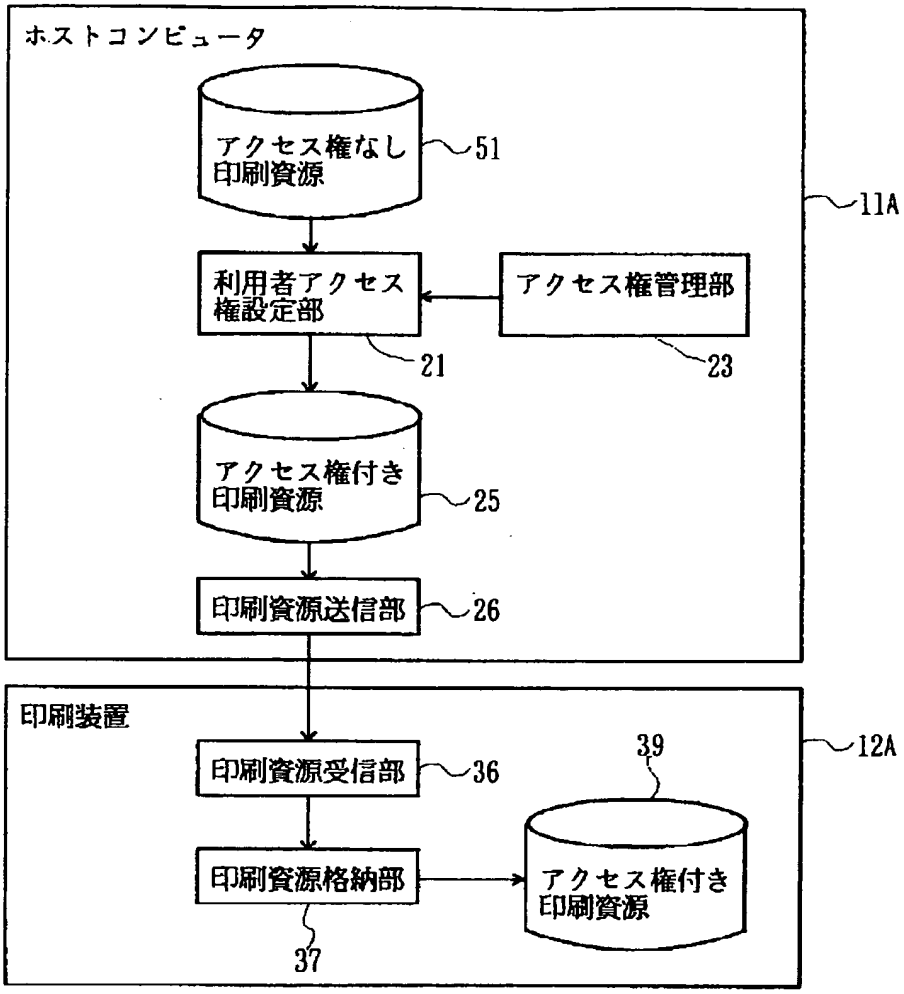
【図9】



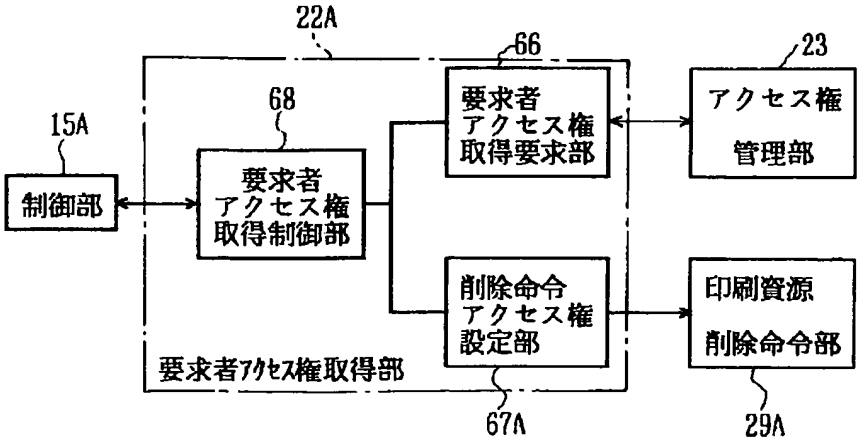
【図10】



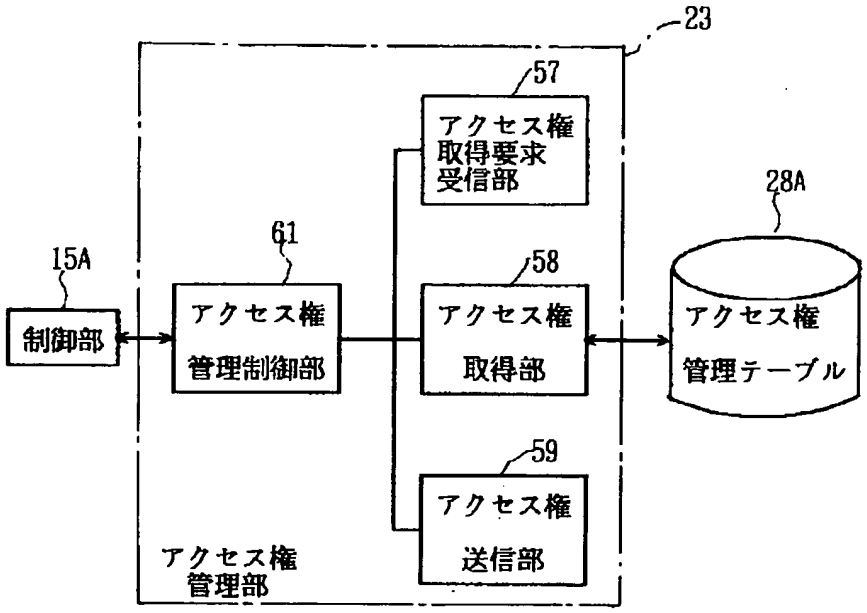
【図１１】



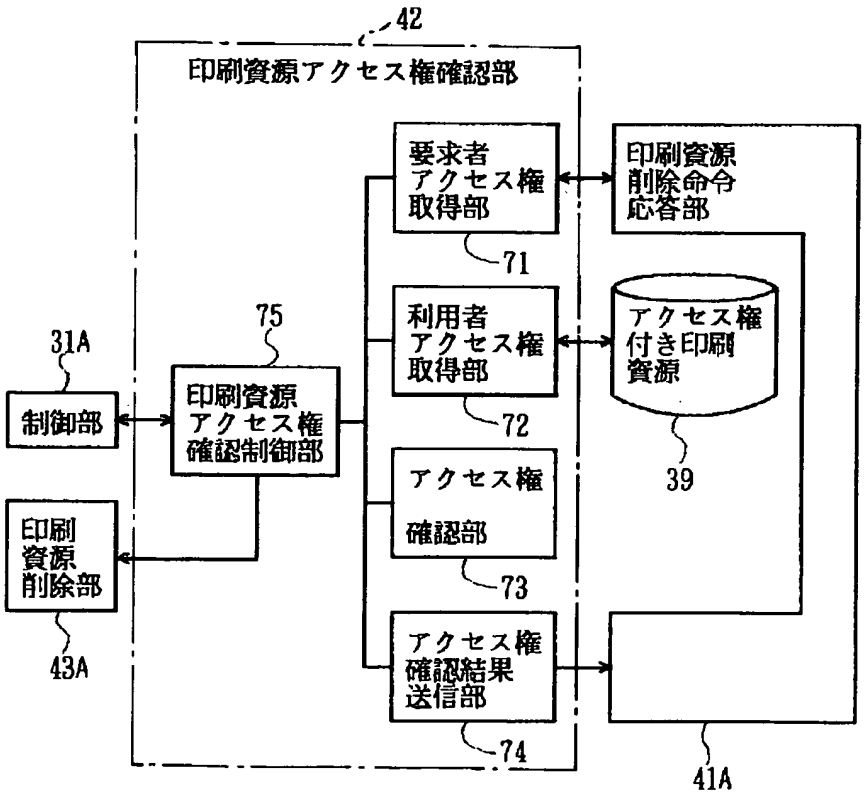
【図１５】



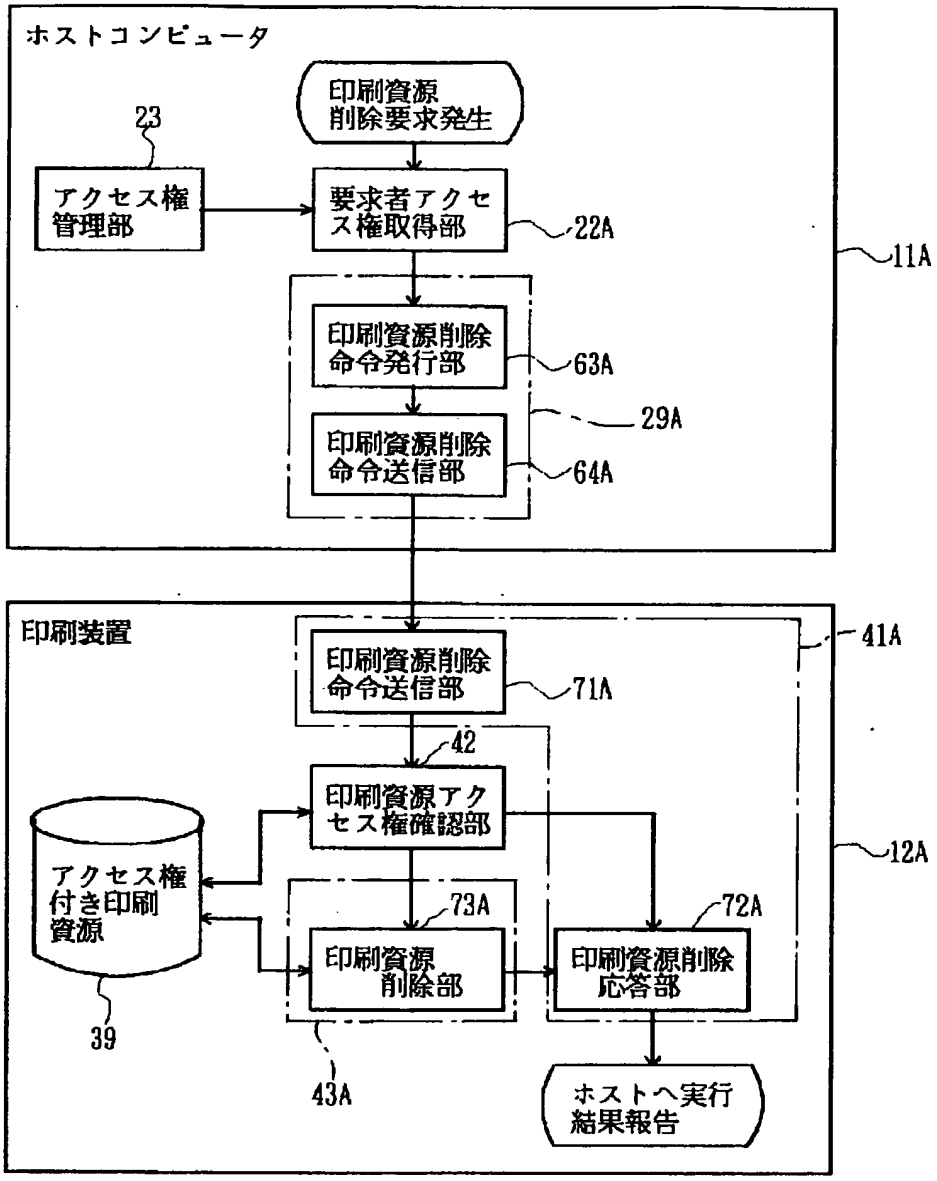
【図13】



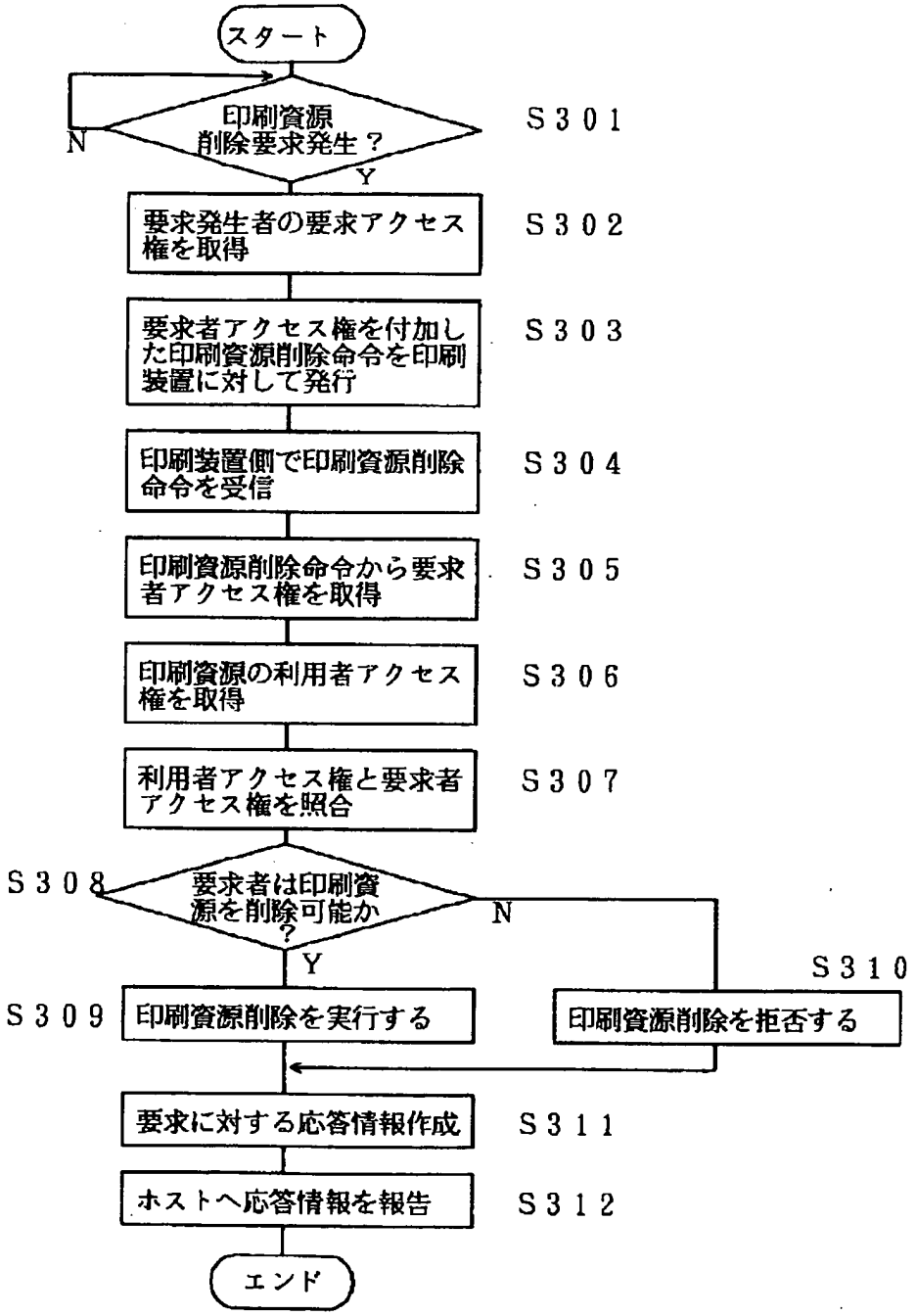
【図16】



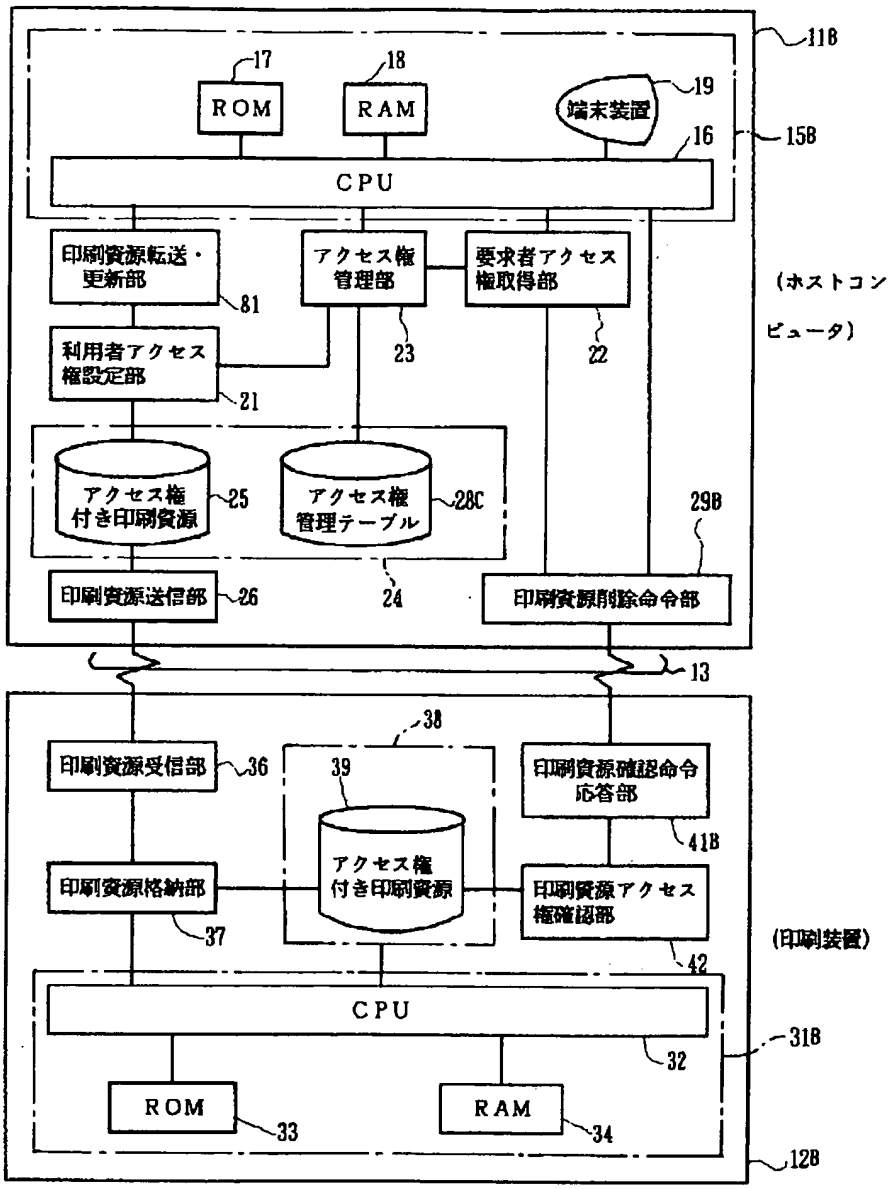
【図14】



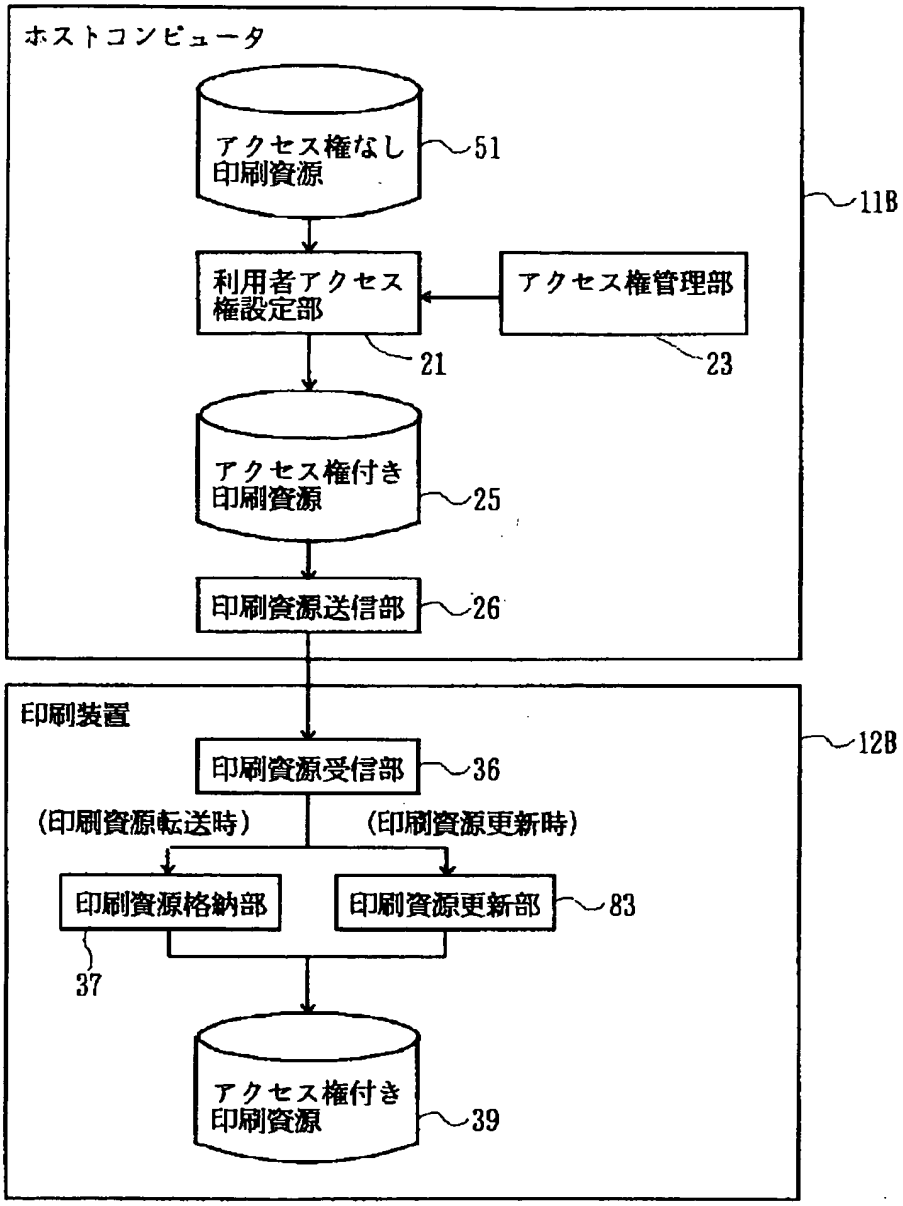
【図17】



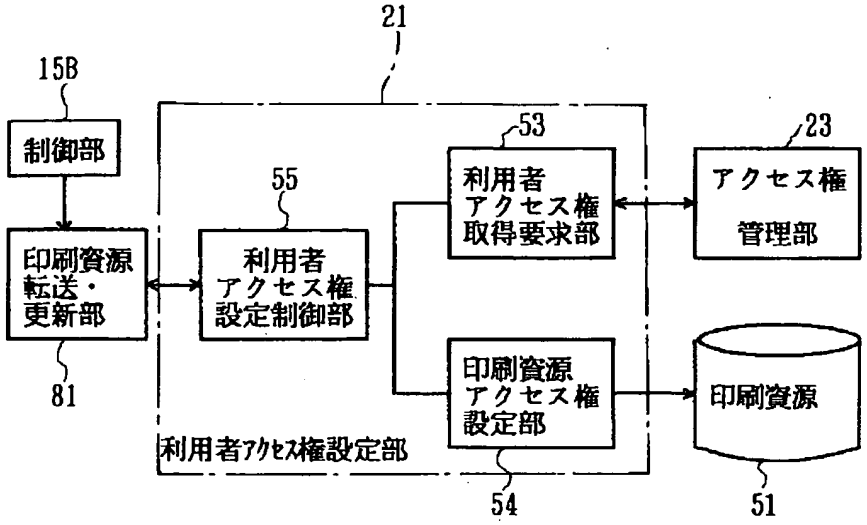
【図１８】



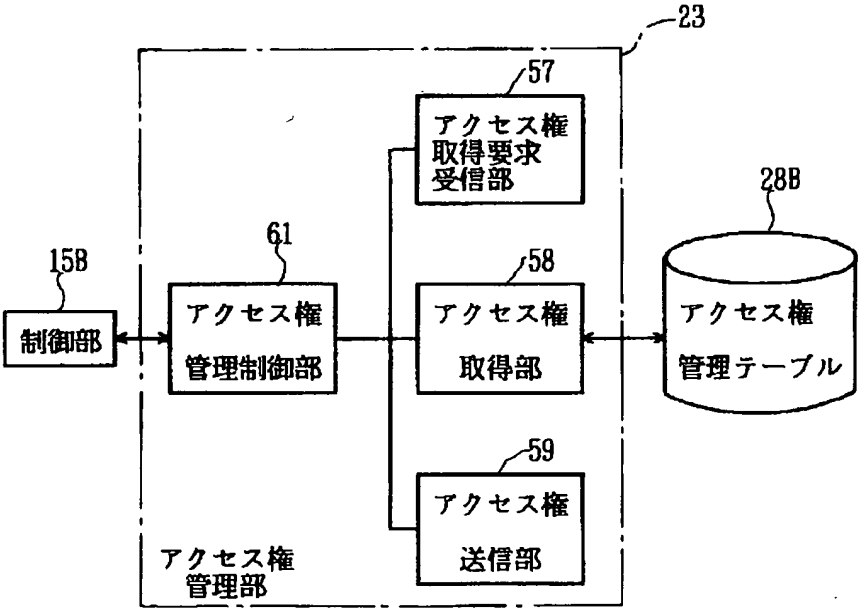
【図19】



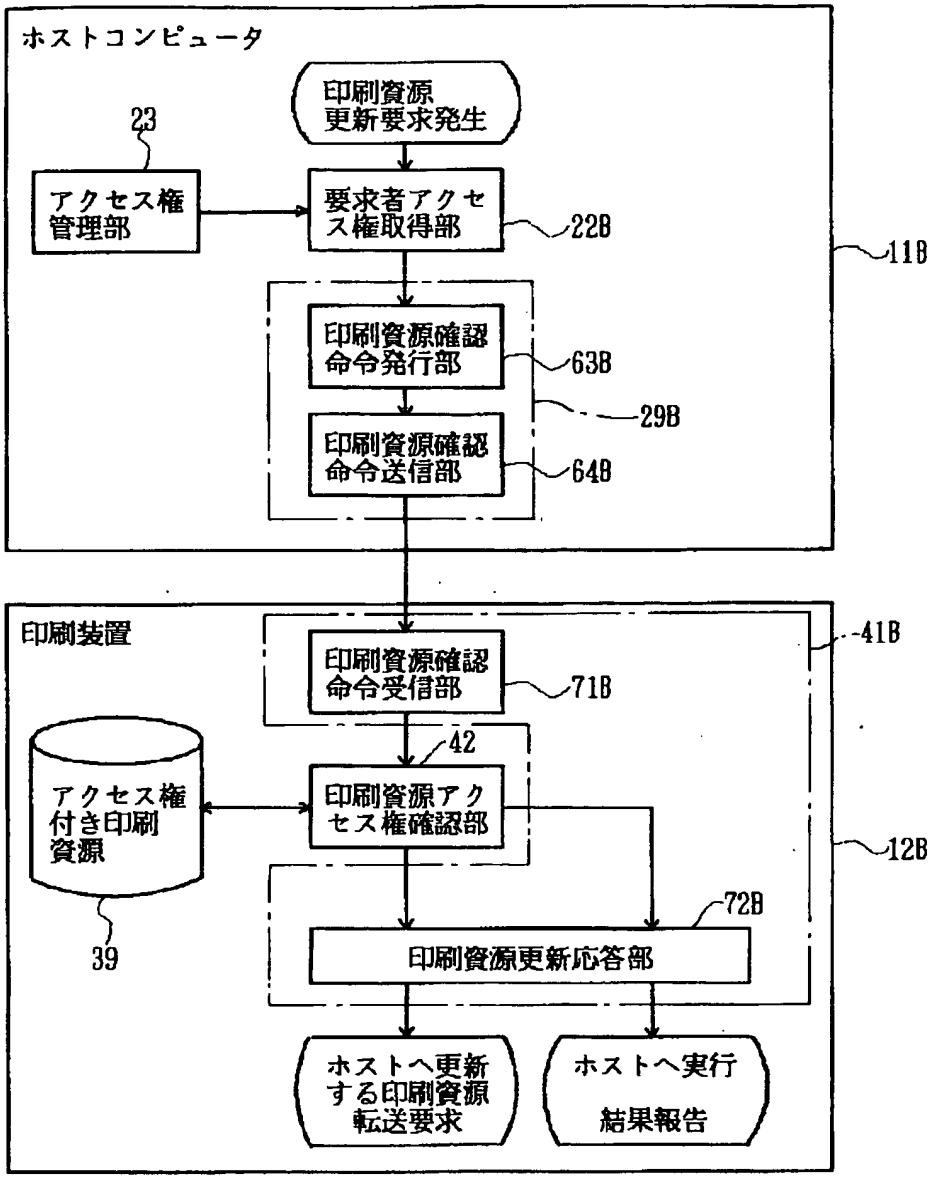
【図２０】



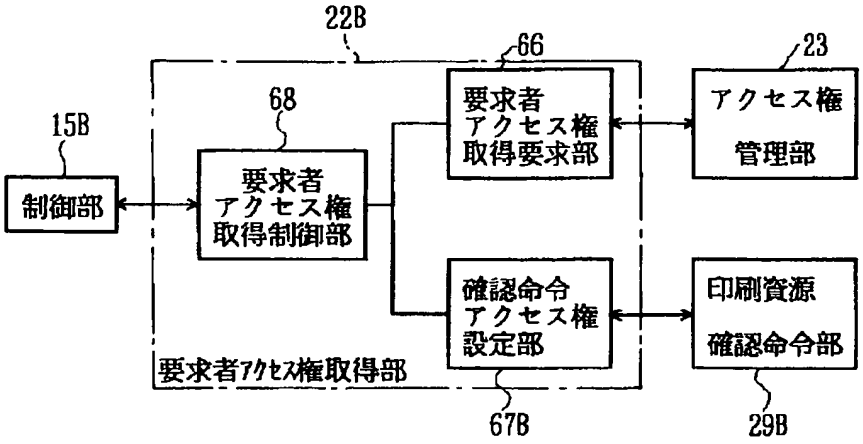
【図２１】



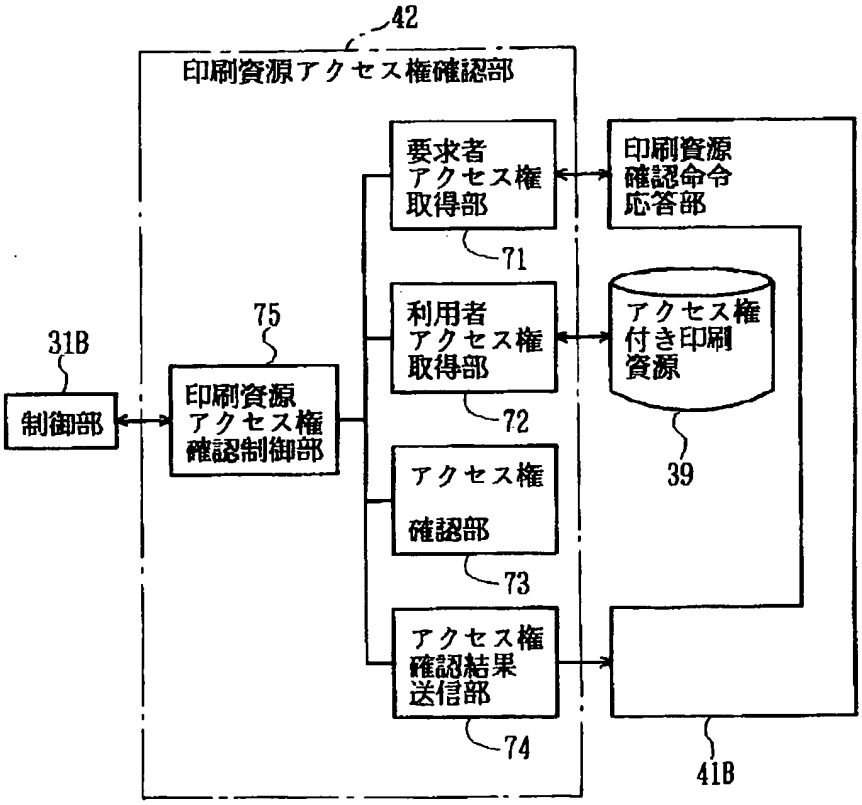
【図22】



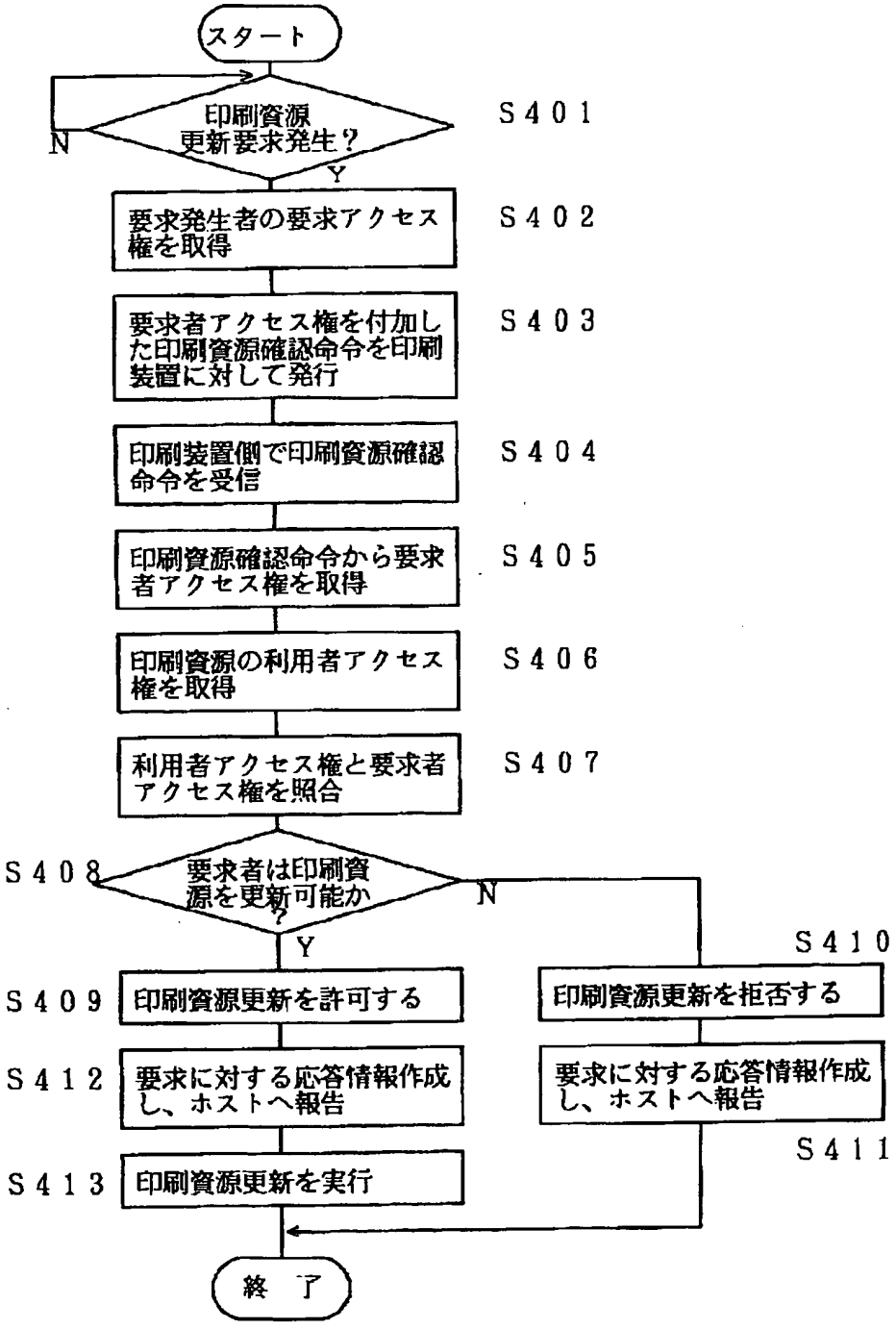
【図23】



【図24】



【図25】



Requested Patent: JP6215010A

Title: INFORMATION ACQUIRING DEVICE ;

Abstracted Patent: JP6215010 ;

Publication Date: 1994-08-05 ;

Inventor(s): TSUTSUI KIYOUYA ;

Applicant(s): SONY CORP ;

Application Number: JP19930021729 19930114 ;

Priority Number(s): ;

IPC Classification: G06F15/21; G06F7/58; G06K17/00; G06K19/00; G07F7/08 ;

Equivalents: ;

ABSTRACT:

PURPOSE:To copy the information of news or music by inserting an information recording and reproducing device such as an IC card to an information acquiring device.

CONSTITUTION:The information acquiring device copies the information of news or music to an information recorder A by inserting the information recorder A such as the IC card to the information acquiring device installed at a station or on the street of a shopping town. The information recorder A copying these kinds of information is mounted at an information reproducing device B and viewed by a display B2 and an earphone B4, etc. Prepaid information is previously written in the information recorder A by a right management information updating device and therefore, it is not necessary to adjust compensation in the case of copying information from the information acquiring device to the information recorder A.

(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（A）　(11)特許出願公開番号

特開平6－215010

(43)公開日　平成６年(1994)８月５日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　15/21	３５０	8724－5L		
7/58	Ｚ	9188－5B		
Ｇ０６Ｋ　17/00	Ｌ	7459－5L		
		8623－5L	Ｇ０６Ｋ　19/ 00　Ｑ	
		9256－3E	Ｇ０７Ｆ　7/ 08　Ｓ	

審査請求　未請求　請求項の数10　ＦＤ　（全 14 頁）　最終頁に続く

(21)出願番号　特願平5－21729

(22)出願日　平成５年(1993)１月14日

(71)出願人　000002185
ソニー株式会社
東京都品川区北品川６丁目７番35号

(72)発明者　筒井　京弥
東京都品川区北品川６丁目７番35号　ソニー株式会社内

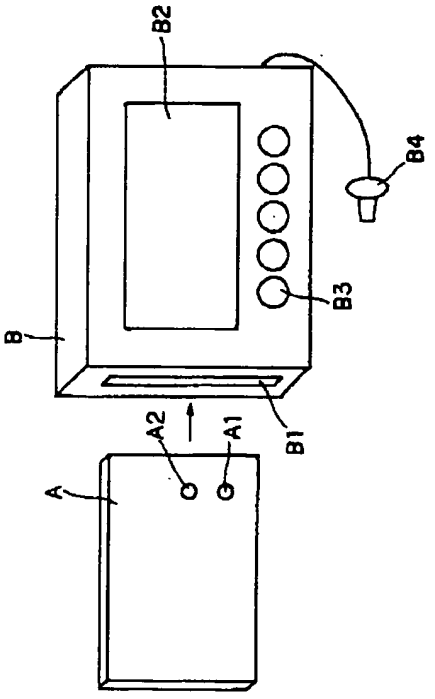
(74)代理人　弁理士　稲本　義雄

(54)【発明の名称】　情報取得装置

(57)【要約】

【目的】　情報提供装置に対してＩＣカード等の情報記録再生装置を挿入することで、ニュース或いは音楽等の情報をコピー出来るようにする。

【構成】　駅或いはショッピング街等の街頭に設置された情報提供装置に対して、ＩＣカード等の情報記録装置Ａを挿入することで、情報提供装置より情報記録装置Ａに対してニュース或いは音楽等の情報をコピーさせる。これらの情報をコピーした情報記録装置Ａは、情報再生装置Ｂに装着され、表示体Ｂ２およびイアホンＢ４等で視聴される。情報記録装置Ａには、予め権利管理情報更新装置によってプリペイド情報が書き込まれており、従って情報提供装置より情報記録装置Ａに対して情報をコピーする際には対価の精算の必要はない。



1

【特許請求の範囲】

【請求項1】　識別情報に基づいて権利管理情報を送出する権利管理情報更新手段と、識別情報に基づいて提供すべき情報を送出する情報提供手段との間でそれぞれ情報交換が成される情報取得装置であって、

前記権利管理情報更新手段と情報交換を行う際に、識別情報に基づいて権利管理情報更新手段からの権利管理情報を管理情報記録媒体に格納する管理情報格納制御手段と、

前記管理情報記録媒体に権利管理情報が格納された状態において、前記情報提供手段と情報交換を行う際に、識別情報に基づいて情報提供手段から提供される情報を情報記録媒体に格納する情報格納制御手段とを具備したことを特徴とする情報取得装置。

【請求項2】　前記情報記録媒体に蓄積された情報を読み出し再生する情報再生手段をさらに具備したことを特徴とする請求項1に記載の情報取得装置。

【請求項3】　前記管理情報記録媒体には、前記権利管理情報に加え、さらに識別情報と、第1の鍵情報と、第2の鍵情報とが格納されるように成されたことを特徴とする請求項1または請求項2に記載の情報取得装置。

【請求項4】　前記権利管理情報更新手段には、前記管理情報記録媒体に格納された第1の鍵情報と同一の鍵情報が格納された鍵情報テーブルが具備されて成ることを特徴とする請求項3に記載の情報取得装置。

【請求項5】　前記権利管理情報更新手段は、前記管理情報記録媒体に格納された第1の鍵情報と同一の鍵情報を通信手段によって管理センタより取得するようにして成る請求項3に記載の情報取得装置。

【請求項6】　前記管理情報記録媒体に格納される権利管理情報は前記権利管理情報更新手段との情報交換によって書き替え可能に成されていることを特徴とする請求項1乃至請求項5に記載の情報取得装置。

【請求項7】　前記権利管理情報の書き替えは前記第1の鍵情報に基づいて正当であると認証された権利管理情報更新手段からの情報に基づいてのみ可能であることを特徴とする請求項6に記載の情報取得装置。

【請求項8】　前記正当であると認証する手段は、情報取得装置に内蔵された乱数発生手段によって発生された乱数に基づいて行われることを特徴とする請求項7に記載の情報取得装置。

【請求項9】　前記情報提供手段には、前記管理情報記録媒体に格納された第2の鍵情報と同一の鍵情報が格納された鍵情報テーブルが具備されて成ることを特徴とする請求項3に記載の情報取得装置。

【請求項10】　前記情報提供手段は、前記管理情報記録媒体に格納された第2の鍵情報と同一の鍵情報を通信手段によって管理センタより取得するようにして成る請求項3に記載の情報取得装置。

【発明の詳細な説明】

2

【0001】

【産業上の利用分野】本発明はニュース、或いは音楽等の情報を迅速に取得することができる情報取得装置に関する。

【0002】

【従来の技術】例えば、特開平3－118690号には、ニュース或いは音楽等の各種の情報が格納された装置に対して、手持ちのICカード或いはカセットテープ等の情報記録媒体を装着し、対価として必要な金額に相当するコインを装置のコインボックスに投入することで、前記情報記録媒体に対して装置よりニュース或いは音楽等の各種の情報を選択的にダビングできるように成された情報提供装置が開示されている。

【0003】従って、これらの情報提供装置を駅前或いはショッピング街等の人通りの多い各所に設置しておけば、利用者が必要な時に前記情報提供装置に向かって手持ちのICカード或いはカセットテープ等の情報記録媒体を装着し、情報の対価としてのコインをコインボックスに投入することで、即座に必要な情報を得ることができるようになる。

【0004】

【発明が解決しようとする課題】ところで、前記した従来の装置においては、手持ちのICカード或いはカセットテープ等の情報記録媒体に対して必要な情報をダビングさせる際に、所定の金額をコインボックスに対してその都度投入する等の操作が必要である。情報記録媒体としては、将来ICカードの普及が見込まれており、情報記録媒体としてICカードを利用するようにした場合には、ICカードに対しての情報のダビングは略瞬時に行うことが可能であるにもかかわらず、対価として必要なコインを投入するなどの操作を余儀無くされることとなる。従って、例えば駅構内において、電車を待つ間の限られた時間内にダビング操作を行おうとしても、コインボックス等にコインを投入する等の金額的な精算が必要となり、この精算による時間が必要となるため、情報を取得できる人数は限られたものになる。

【0005】本発明は、この様な点に着目して成されたものであり、情報提供手段としての情報提供装置に対して、情報記録（再生）手段を装着させることで、必要な情報を即座にダビング出来るようにし、前記した従来のものの不都合を解消した情報取得装置を提供することを課題とするものである。

【0006】

【課題を解決するための手段】前記課題を達成するために成された請求項1に記載の情報取得装置は、識別情報に基づいて権利管理情報を送出する権利管理情報更新手段としての権利管理情報更新装置と、識別情報に基づいて提供すべき情報を送出する情報提供手段としての情報提供との間でそれぞれ情報交換が成される情報取得装置であって、権利管理情報更新装置と情報交換を行う際

に、識別情報に基づいて権利管理情報更新装置からの権利管理情報を管理情報記録媒体に格納する管理情報格納制御手段と、管理情報記録媒体に権利管理情報が格納された状態において、情報提供手段としての情報提供装置と情報交換を行う際に、識別情報に基づいて情報提供装置から提供される情報を情報記録媒体に格納する情報格納制御手段とを備えることを特徴とする。

【０００７】また、請求項２に記載の情報取得装置は、情報記録媒体に蓄積された情報を読み出し再生する情報再生手段をさらに具備することを特徴とする。

【０００８】また、請求項３に記載の情報取得装置は、管理情報記録媒体に対して権利管理情報に加え、識別情報と、第１の鍵情報と、第２の鍵情報とをさらに格納するようにした点を特徴とする。

【０００９】また、請求項４に記載の情報取得装置は、権利管理情報更新手段としての権利管理情報更新装置に、管理情報記録媒体に格納された第１の鍵情報と同一の鍵情報が格納された鍵情報テーブルを備えることを特徴とする。

【００１０】また、請求項５に記載の情報取得装置は、権利管理情報更新手段としての権利管理情報更新装置が、管理情報記録媒体に格納された第１の鍵情報と同一の鍵情報を通信手段によって管理センタより取得するよう構成した点を特徴とする。

【００１１】また、請求項６に記載の情報取得装置は、管理情報記録媒体に格納される権利管理情報が権利管理情報更新手段としての権利管理情報更新装置との情報交換によって書き替え可能に成されるよう構成した点を特徴とする。

【００１２】また、請求項７に記載の情報取得装置は、権利管理情報の書き替えが第１の鍵情報に基づいて正当であると認証された権利管理情報更新手段としての権利管理情報更新装置からの情報に基づいてのみ可能であるよう構成した点を特徴とする。

【００１３】また、請求項８に記載の情報取得装置は、前記正当であると認証する手段が、情報取得装置に内蔵された乱数発生手段としての乱数発生器によって発生される乱数に基づいて行われるよう構成した点を特徴とする。

【００１４】また、請求項９に記載の情報取得装置は、管理情報記録媒体に格納された第２の鍵情報と同一の鍵情報が格納された鍵情報テーブルが情報提供手段としての情報提供装置に具備されたことを特徴とする。

【００１５】また、請求項１０に記載の情報取得装置は、情報提供手段としての情報提供装置に、管理情報記録媒体に格納された第２の鍵情報と同一の鍵情報を通信手段によって管理センタより取得できるように構成した点を特徴とする。

【００１６】

【作用】請求項１に記載の情報取得装置においては、は

じめに権利管理情報更新装置と情報交換を行う際に、識別情報に基づいて権利管理情報更新装置からの権利管理情報を管理情報記録媒体に予め格納する。この時、情報に対する対価をプリペイドする。次いで管理情報記録媒体に権利管理情報が格納された状態において、情報提供装置と情報交換を行うように成され、情報提供装置からの情報を取得する毎に例えば権利管理情報が残度数情報として書き替えられる。

【００１７】また請求項２に記載の情報取得装置においては、情報記録媒体に蓄積された情報を読み出し再生する情報再生手段としての再生装置がさらに具備されており、この再生装置によって情報を視聴することが可能となる。

【００１８】また請求項３に記載の情報取得装置においては、管理情報記録媒体に対して権利管理情報に加え、識別情報と、第１の鍵情報と、第２の鍵情報とがさらに格納され、これらの識別情報および鍵情報によって情報の授受の安全性が確保される。

【００１９】また請求項４に記載の情報取得装置においては、権利管理情報更新手段としての権利管理情報更新装置に、管理情報記録媒体に格納された第１の鍵情報と同一の鍵情報が格納された鍵情報テーブルが備えられる。

【００２０】また請求項５に記載の情報取得装置においては、権利管理情報更新手段としての権利管理情報更新装置が、管理情報記録媒体に格納された第１の鍵情報と同一の鍵情報を通信手段によって管理センタより取得するよう構成される。

【００２１】また請求項６に記載の情報取得装置においては、管理情報記録媒体に格納される権利管理情報が権利管理情報更新手段としての権利管理情報更新装置との情報交換によって書き替え可能に成される。

【００２２】また請求項７に記載の情報取得装置においては、権利管理情報の書き替えが第１の鍵情報に基づいて正当であると認証された権利管理情報更新手段としての権利管理情報更新装置からの情報に基づいてのみ可能であるよう構成される。

【００２３】また請求項８に記載の情報取得装置においては、前記正当であると認証する手段が、情報取得装置に内蔵された乱数発生手段としての乱数発生器によって発生される乱数に基づいて行われるよう構成される。

【００２４】また請求項９に記載の情報取得装置においては、管理情報記録媒体に格納された第２の鍵情報と同一の鍵情報が格納された鍵情報テーブルが情報提供手段としての情報提供装置に具備される。

【００２５】また請求項１０に記載の情報取得装置においては、情報提供手段としての情報提供装置に、管理情報記録媒体に格納された第２の鍵情報と同一の鍵情報を通信手段によって管理センタより取得できるように構成される。

5

【0026】

【実施例】本発明の情報取得装置は、ニュース、或いは音楽等の情報を視聴しようとする利用者が、自分が所持している情報取得装置、すなわち情報記録（再生）装置に対して情報提供手段としての情報提供装置より必要な情報をコピー（ダビング）し、即座にその情報を視聴できるようにするものである。この場合、情報提供装置より情報を受ける権利、すなわち権利管理情報を権利管理情報更新手段としての権利管理情報更新装置により予め更新しておくことで、情報記録（再生）装置は情報提供装置より必要な情報を入手するに際し、その都度現金等の情報料の精算を行う必要をなくすように成される。

【0027】ここで言う権利管理情報とは、その情報使用者が情報提供装置から情報を入手する権利を表す情報のことを言い、例えばその情報使用者の会員情報や、情報入手時毎に減じられる残度数情報等をその例として挙げることができる。

【0028】以下、本発明の実施例について図面を参照して説明する。図1は、本発明の情報取得装置を構成する情報記録装置Aと、この情報記録装置が装着され、情報記録装置Aに記録された情報を再生することができる情報再生装置Bの外観図を示したものである。

【0029】前記情報記録装置Aは、例えばICカードにより構成されており、この情報記録装置Aには、情報の再生時に、情報再生装置Bとの間でデータおよび制御信号の交換をするための端子として情報再生装置結合端子A1が設けられている。また、情報記録装置AとしてのICカードの一部には、後述する情報提供装置との間でデータおよび制御信号の交換をするための端子として情報提供装置結合端子A2も設けられている。前記情報再生装置結合端子A1および情報提供装置結合端子A2は、実際には一つの端子を切り替えて使用するように構成することもできる。

【0030】前記情報再生装置Bは、携帯に便利なように偏平箱型形状に成され、その左側壁部には、ICカードより成る前記情報記録装置Aが挿入されるスリット状の挿入口B1が形成されており、また正面中央部には、表示手段としての例えば液晶表示体B2が配置され、さらに正面下側部には、再生選択手段としての複数個の押釦スイッチB3が横一線に配置されている。そして、その右側壁部からはイアホンB4が導出されている。

【0031】前記液晶表示体B2は、前記情報記録装置A内に記録された情報の内容を表示することができ、この液晶表示体B2に表示される、例えば情報リストに基づいて押釦スイッチB3のいずれかを押圧操作することで、選択された情報が液晶表示体B2に視覚情報として再生され、また前記イアホンB4によって音声情報として再生される。情報の内容は、テキスト情報、映像情報、コンピュータプログラムおよび音声情報等を含み、特に限定されない。

6

【0032】なお、図1の実施例には図示していないが、イアホンの代わりに、或いはイアホンに加えて情報再生装置Bにスピーカを装備していてもよく、その場合には、スピーカに音声情報の再生結果を出力してもよい。さらに再生信号は、やはり図1には描かれていないが、外部出力端子を経由させて、外部のCRTやスピーカ等に接続してもよい。

【0033】図2は、本発明の情報取得装置の他の実施例である情報記録再生装置の外観図を示したものである。すなわち、図1に示した情報記録装置Aに相当するものが、情報再生装置Bに内臓された構成となっており、外観上は前記図1における情報再生装置Bと略同一形状に成されている。従って、図2において、図1に相当する部分は同一符号を付し、その詳細な説明は省略する。

【0034】ただし、図2に示す情報記録再生装置Cにおいては、その左側壁部に後述する情報提供装置への結合端子C1が設けられており、この端子C1を経由して情報が取得される。なお、図2に示した情報記録再生装置Cに内蔵される情報記録媒体の種類は特に限定されることはないが、高速にコピー（ダビング）が可能であり、かつ、記録媒体へのランダムアクセスが容易で、携帯性にも優れたICメモリを使用すると便利である。

【0035】図3は、本発明の情報取得装置に対して情報を提供する情報提供手段としての情報提供装置の外観図を示している。情報提供装置D内には、後述する情報記録媒体が内蔵され、情報取得装置に対して提供するための各種の情報が蓄積されている。そして、箱型の情報提供装置Dの正面パネルには、蓄積されている情報の内容や提供価格等を表示する表示手段としての液晶表示体D11乃至D16が配置され、どの情報を情報提供装置Dから供給を受けるかを選択する出力選択手段としての複数個の押釦スイッチD11乃至D16がそれぞれ前記各液晶表示体D11乃至D16の近傍に配置されている。そして、その正面右下側部には、ICカードより成る前記情報記録装置Aの挿入排出口D3が配置されている。

【0036】従って、情報を入手しようとする場合には、ICカードより成る情報記録装置Aを前記挿入排出口D3に挿入し、液晶表示体D11乃至D16に表示された入手しようとする情報に対応する前記押釦スイッチD11乃至D16を択一的に押圧操作することで、情報記録装置Aに対して情報がコピーされる。そしてコピー済みの情報記録装置Aは、挿入排出口D3より排出される。

【0037】なお、図3に示す例は、ICカードより成る情報記録装置Aを直接挿入し、コピーを行うようにする場合を示しているが、例えば前記図2に示したような情報記録再生装置Cに対しては、ワイヤ（図示せず）を情報提供装置結合端子C1に接続し、リクエストされた

情報を電気的に供給するように成される。

【００３８】図４は、本発明の情報取得装置に対して情報を提供する情報提供装置の他の例を示した外観図である。この図４に示した情報提供装置Ｅにおいては、ＩＣカードより成る情報記録装置Ａに対して情報を供給するものであり、情報記録装置Ａの挿入口Ｅ１と排出口Ｅ２とが距離をおいて分離されており、情報入手希望者Ｆは矢印方向に歩きながら、ＩＣカードより成る情報記録装置Ａに対して情報をコピーさせることができる。この例は、多くの人に対して迅速に情報を提供する場合に便利である。

【００３９】図５は、権利管理情報更新手段としての権利管理情報更新装置の外観図を示している。箱型の権利管理情報更新装置Ｇの正面パネルには、ＩＣカードより成る情報記録装置Ａの挿入排出口Ｇ１およびコイン投入口Ｇ２が設けられている。権利管理情報の更新を希望する場合には、情報記録装置Ａを挿入排出口Ｇ１に挿入すると共に、コイン投入口Ｇ２に必要な対価を投入することで情報記録装置Ａの権利管理情報の更新が成される。

【００４０】図６は、権利管理情報更新装置の他の例を示した外観図である。この例に示す権利管理情報更新装置Ｈは人手によって管理されるものである。すなわち、管理者が、更新希望者より対価ならびに情報記録装置Ａを受けとり、その対価に応じた更新情報を入力手段としてのテンキーＨ１を操作する。テンキーＨ１の操作により、更新情報が表示装置Ｈ２に表示され、情報記録装置Ａを挿入排出口Ｈ２に挿入することで、情報記録装置Ａは対価に応じた権利管理情報の更新が成される。

【００４１】なお、図５および図６に示した権利管理情報更新装置は、いずれも図１に示したようなＩＣカードより成る情報記録装置Ａに対して権利管理情報を更新させるタイプであるが、例えば前記図２に示したような情報記録再生装置Ｃに対しては、ワイヤ（図示せず）によって電気的に接続し、権利管理情報を更新させるように成される。

【００４２】次に、図７は、情報記録再生装置の電気的な内部構成をブロック図で示したものである。この例は、例えば前記図２に示すような情報記録装置と情報再生装置が一体となった状態を示しており、図１においては、情報再生装置Ｂに対して情報記録装置Ａが装着された状態で示している。

【００４３】図７において、情報記録再生装置１００１は、情報提供装置からライン１０３を介して転送された情報を情報記録媒体１００３に格納し、格納された情報は伝送ライン１０２を介して情報再生手段１００２によって読み出され、再生される。そして、ライン１０１を介して、図１または図２に示す表示手段Ｂ２およびイアホンＢ４等に出力される。

【００４４】情報提供装置からの情報を受けるに先立って、制御手段としての制御装置１００４は、情報格納制御手段として作用し、後に詳述するが、情報提供装置との間でライン１０４を介して通信（情報交換）を行い、管理情報記録媒体１００６に記録された情報に基づいて、その情報記録再生装置が情報の伝送を受けることができることを証明する。ここで、管理情報記録媒体１００６に格納された権利管理情報は、ライン１０７を介して制御装置１００４に接続されており、さらに権利管理情報は、後に詳述するが、ライン１０５を介して権利管理情報更新装置との間で記録／更新される。この場合、前記制御装置１００４は、管理情報格納制御手段として作用する。

【００４５】前記ライン１０４および１０５を介した情報提供装置および権利管理情報更新装置との通信には、管理情報記録媒体１００６に格納された識別情報Ｉ、第１の鍵情報１、第２の鍵情報２の情報が使用される。これらの値は、全ての情報記録再生装置について同一であってもよいが、後で述べるように、情報記録再生装置によって異なっていると都合が良い。また前記制御装置１００４には、ライン１０６を介して接続された乱数発生手段としての乱数発生回路１００５より乱数が供給されている。

【００４６】次に、図８は、権利管理情報更新装置の電気的な内部構成をブロック図で示したものである。権利管理情報更新装置１０１１には、制御手段としての制御装置１０１２が内蔵されており、この制御装置１０１２には、第１の鍵情報が格納された鍵情報１テーブル１０１４がライン１１２を介して接続され、また更新情報記録媒体１０１３がライン１１１を介してそれぞれ接続されている。そして権利管理情報更新装置１０１１は、後に詳述するが、情報記録再生装置との間でライン１０５を介して通信（情報交換）を行うことにより、情報記録再生装置に記録されている権利管理情報を更新することができる。また、この更新手続の記録は更新情報記録媒体１０１３に記録され、これにより権利管理情報の更新に伴う金銭の授受が人手によって行われた場合にも、正確な決済処理を行うことができる。また、この更新情報をライン１１３を介して管理センタＪに送信することにすれば、権利管理更新装置が設置されている場所まで行って決済のための手続きを行わなくても、金銭授受者との決済を行うことが可能である。

【００４７】図９は、情報提供装置の電気的な内部構成をブロック図で示したものである。情報提供装置１０２１には、制御手段としての制御装置１０２３が内蔵されており、この制御装置１０２３には、まず第２の鍵情報が格納された鍵情報２テーブル１０２５がライン１２４を介して接続され、また乱数発生手段としての乱数発生回路１０２５がライン１２３を介して接続されている。さらに制御装置１０２３には、ライン１２５を介して情報伝送記録媒体１０２６が接続されている。そして前記制御装置１０２３からは、ライン１２２によって情報記

録媒体１０２２に対して制御信号が出力される構成となっている。

【００４８】情報提供装置の情報記録媒体１０２２には、ライン１２１によって管理センタＪから伝送された情報が記録され、情報記録装置に対してライン１０３を介して情報の転送がなされる。その転送に先立ち、制御手段１０２３は、情報記録再生装置との間でライン１０４を介して通信（情報交換）を行ない、その情報転送要求が正当なものであるかどうかの確認を行なう。また、この情報転送に関する記録を情報転送記録媒体１０２６に記録することができ、この情報転送に関するデータをライン１２６を介して管理センタＪに送信することも可能である。

【００４９】ここで、権利管理情報の記録／更新に関する処理フローについて、図１０に示すフローチャートに基づいて説明する。権利管理情報の記録／更新は、正当に管理されている権利管理情報更新装置、すなわち各情報記録装置に記録されている秘密の鍵情報１の内容を知っている権利管理情報更新装置のみによって行なうことができるものでなければならない。

【００５０】図１０において、権利管理情報更新装置１０１１に挿入された情報記録再生装置１００１は、まず識別情報Ｉを権利管理情報更新装置に送信（ステップＳ１）すると共に、乱数Ｒを発生（ステップＳ２）して、権利管理情報更新装置１０１１に送信（ステップＳ３）する。前記識別情報Ｉを受信（ステップＳ４）し、また、乱数Ｒを受信（ステップＳ５）した権利管理情報更新装置１０１１は、まず識別情報Ｉの値から鍵情報１のテーブル１０１４を検索して、その情報記録再生装置１００１に対応する鍵情報１、Ｋ１［Ｉ］を得、Ｋ１［Ｉ］と乱数ＲからＥ２を算出（ステップＳ６）して、情報記録再生装置１００１に送信（ステップＳ７）する。

【００５１】一方、情報記録再生装置１００１においても、Ｋ１［Ｉ］と乱数Ｒから権利管理情報更新装置が成した方法と同一の方法でＥ１を算出（ステップＳ８）する。そして、権利管理情報更新装置１０１１よりＥ２を受信（ステップＳ９）し、ステップＳ８で算出したＥ１と、ステップＳ９で受信したＥ２の比較（ステップＳ１０）を行なう。この比較結果が一致しない（Ｎｏ）場合には、不正な権利管理情報更新装置１０１１からの記録／更新要求があったものと判断して、権利管理情報の記録／更新を停止（ステップＳ１１）させる。また、この比較結果が一致した（Ｙｅｓ）ならば、正当な権利管理情報更新装置１０１１からの記録／更新要求があったものと判断する。この場合には、権利管理情報更新装置１０１１より権利管理情報が送信（ステップＳ１２）され、これを受けて、情報記録再生装置１００１は権利管理情報の記録／更新を行なう（ステップＳ１３）。

【００５２】なお、この時、図８において、権利管理情報更新の記録を更新情報記録媒体１０１３に記録したり、ライン１１３を介して管理センタＪに送信することで、決済処理をセンタで行なわせ、また統計処理の集計も行なわせることが可能となる。

【００５３】次に、情報提供装置より情報記録再生装置に対して情報を転送する処理フローについて、図１１に示すフローチャートに基づいて説明する。情報の提供は、情報提供を受ける権利を証明することのできる情報記録再生装置、すなわち識別情報Ｉに対応した鍵情報２の値を知っている事実と、情報提供を受けることを表わす権利管理情報の両者を提示できる情報記録再生装置のみに対して行なわれるものでなければならない。

【００５４】情報提供装置１０２１に挿入された情報記録再生装置１００１は、まず識別情報Ｉを情報提供装置１０２１に送信（ステップＳ２１）し、情報提供装置１０２１は、これを受信（ステップＳ２２）する。次いで情報提供装置１０２１は、乱数Ｐを発生（ステップＳ２３）し、これを情報記録再生装置１００１に送信（ステップＳ２４）する。情報記録再生装置１００１では、乱数Ｐを受信（ステップＳ２５）し、受信した乱数Ｐと鍵情報２の値、Ｋ２［Ｉ］からＦ２を算出（ステップＳ２６）して、Ｆ２を情報提供装置１０２１に送信（ステップＳ２７）する。

【００５５】一方、情報提供装置１０２１においても、受信した識別情報Ｉの値から鍵情報２のテーブル１０２５を検索して、その情報記録再生装置１００１に対応する鍵情報２、Ｋ２［Ｉ］を得、Ｋ２［Ｉ］と乱数Ｐから情報記録再生装置がＦ２を算出したのと同じ方法でＦ１を算出（ステップＳ２８）する。そして、情報記録再生装置１００１よりＦ２を受信（ステップＳ２９）し、ステップＳ２８で算出したＦ１と、ステップＳ２９で受信したＦ２の比較（ステップＳ３０）を行なう。

【００５６】この比較結果が一致しない（Ｎｏ）場合には、不正な情報記録再生装置１００１からの情報提供要求であったと判断して、情報提供装置１０２１から情報記録再生装置１００１への情報の送信を停止（ステップＳ３１）させる。また、この比較結果が一致した（Ｙｅｓ）ならば、正当な情報記録再生装置１００１からの情報提供要求であったと判断する。この場合には、情報記録再生装置１００１より送信（ステップＳ３２）される権利管理情報を受信（ステップＳ３３）して、その情報記録再生装置１００１が実際にその情報提供を受ける権利があるのかどうか、会員情報等の権利管理情報を調べる（ステップＳ３４）。

【００５７】ステップＳ３４において、権利が有効でない（Ｎｏ）と判断された場合には、情報の送信を停止（ステップＳ３１）させる。権利が有効である（Ｙｅｓ）と判断されれば、情報の送信（ステップＳ３５）を行ない、情報記録再生装置１００１は、情報の受信／記録（ステップＳ３６）がなされる。なお、この情報転送

の記録を情報転送記録媒体１０２６に記録したり、ライン１２５を介して管理センタＪに送信することで、各種の統計処理に役立たせることができる。

【００５８】以上の構成および作用において、ライン１０４および１０５の通信処理で、鍵情報１や鍵情報２を直接送信せず、鍵情報で乱数を符号化したものを送信するのは、送信時にこれらの秘密情報が盗まれることを防ぐためのものであり、もしそうした危険性を考慮しなくてよい場合には、これらの情報を直接送信して正当性を証明するようにしてもよい。また逆に、さらに安全性を高めるために、識別情報等の情報に関しても別の鍵情報を使って送信することにしてもよい。また、鍵情報１と２は別々なものとして説明したが、これらは勿論同一のものであってもよい。

【００５９】ここで、前記した実施例における情報の授受の安全性について述べるが、情報提供装置から不正に情報が転送されるのは次の二つの場合である。

（１）不正な情報記録再生装置に情報の転送が行なわれる。

（２）正当な情報記録再生装置に不正な権利管理情報が記録され、情報提供装置から情報の転送が行なわれる。

【００６０】前記（１）に関して言えば、情報記録再生装置は大量に生産され、不特定多数の使用者に比較的安価に供給されるもので、ハードウエア的にその偽造を行なうことは困難である。しかし万一情報記録再生装置が偽造された場合に、その情報記録再生装置には少なくともある識別情報に対応した鍵情報２が記録されていなければ、情報提供装置からの情報の転送を受けることはできない。

【００６１】そこで偽造者は鍵情報２を盗み出す必要があるが、情報提供装置は厳重に管理することが比較的容易であるので、鍵情報２は情報記録再生装置から盗み出される危険性の方が高い。しかし鍵情報２は、信号としてそのまま情報記録再生装置から外部に出ることはないので、これを盗み出すのは非常に困難である。また万一これを盗み出し、その鍵情報２を持った情報記録再生装置が大量に出回った場合には、情報提供装置および管理センタでの統計的な処理により、そのことを認知し、その情報記録再生装置への情報転送を一時的に中断する等の処理を施すことができる。

【００６２】また、前記（２）に関して言えば、不正な権利管理情報更新装置を作るためには、その装置によって権利管理情報の更新が行なわれるすべての情報記録再生装置の鍵情報１についての情報が必要になる。しかし権利管理情報更新装置は厳重に管理することが比較的容易であり、ここから鍵情報１を盗み出すことは難しい。また鍵情報１の内容は直接外部に出力されない上、各情報記録再生装置によって異なっているので、結局不特定多数の情報記録再生装置に対して不正に権利管理情報の記録／更新を行なう装置を構成することは極めて困難なものとなる。

【００６３】

【発明の効果】以上の説明で明らかなように、請求項１に記載の情報取得装置によれば、はじめに権利管理情報更新装置と情報交換を行う際に、識別情報に基づいて権利管理情報更新装置からの権利管理情報を管理情報記録媒体に予め格納する。この時、情報に対する対価の精算が成される。次いで管理情報記録媒体に権利管理情報が格納された状態において、情報提供装置と情報交換を行うように成され、情報提供装置からの情報を取得（コピー）する毎に例えば権利管理情報が残度数情報として書き替えられる。従って情報を取得する毎に、その場で情報の対価として金銭を授受する必要がなく情報提供装置より情報取得装置に対して能率良く情報を提供することが可能となる。

【００６４】また請求項２に記載の情報取得装置によれば、情報記録媒体に蓄積された情報を読み出し再生する情報再生手段としての再生装置がさらに具備されており、この再生装置によって情報を即座に視聴することが可能となる。

【００６５】また請求項３に記載の情報取得装置によれば、管理情報記録媒体に対して権利管理情報に加え、識別情報と、第１の鍵情報と、第２の鍵情報とがさらに格納される。従ってこれらの識別情報および鍵情報によって情報の授受の安全性が確保される。

【００６６】また請求項４に記載の情報取得装置によれば、権利管理情報更新手段としての権利管理情報更新装置に、管理情報記録媒体に格納された第１の鍵情報と同一の鍵情報が格納された鍵情報テーブルが備えられており、従って正当な情報取得装置に対してのみ権利管理情報の更新が約束される。

【００６７】また請求項５に記載の情報取得装置においては、権利管理情報更新手段としての権利管理情報更新装置が、管理情報記録媒体に格納された第１の鍵情報と同一の鍵情報を通信手段によって管理センタより取得するよう構成される。この構成によると、権利管理情報更新装置には相当な数の複数の第１の鍵情報を格納するための鍵情報記録媒体を具備する必要がなくなる。

【００６８】また請求項６に記載の情報取得装置においては、管理情報記録媒体に格納される権利管理情報が権利管理情報更新手段としての権利管理情報更新装置との情報交換によって書き替え可能に成される。従って情報取得装置におけるＩＣカード等の記録媒体は、反復して使用可能となる。

【００６９】また請求項７に記載の情報取得装置においては、権利管理情報の書き替えが第１の鍵情報に基づいて正当であると認証された権利管理情報更新手段としての権利管理情報更新装置からの情報に基づいてのみ可能であるよう構成される。

【００７０】また請求項８に記載の情報取得装置におい

13

ては、前記正当であると認証する手段が、情報取得装置に内蔵された乱数発生手段としての乱数発生器によって発生される乱数に基づいて行われるよう構成される。従って前記鍵情報が表面上現れることがないため、鍵情報の秘密性が向上する。

【００７１】また請求項９に記載の情報取得装置においては、管理情報記録媒体に格納された第２の鍵情報と同一の鍵情報が格納された鍵情報テーブルが情報提供手段としての情報提供装置に具備される。従って正当な情報取得装置に対してのみ情報の提供が約束される。

【００７２】また請求項１０に記載の情報取得装置においては、情報提供手段としての情報提供装置に、管理情報記録媒体に格納された第２の鍵情報と同一の鍵情報を通信手段によって管理センタより取得できるように構成される。この構成によると、情報提供手段には相当な数の複数の第２の鍵情報を格納するための鍵情報記録媒体を具備する必要がなくなる。

【図面の簡単な説明】

【図１】本発明における情報取得装置の一実施例の示す外観図である。

【図２】本発明における情報取得装置の他の実施例を示す外観図である。

【図３】本発明における情報取得装置と情報交換される情報提供装置を示した外観図である。

【図４】本発明における情報取得装置と情報交換される情報提供装置の他の例を示した外観図である。

14

【図５】本発明における情報取得装置と情報交換される権利管理情報更新装置を示した外観図である。

【図６】本発明における情報取得装置と情報交換される権利管理情報更新装置の他の例を示した外観図である。

【図７】図１または図２に示した情報取得装置としての情報記録再生装置の電気的な構成を示したブロック図である。

【図８】図５または図６に示した権利管理情報更新装置の電気的な構成を示したブロック図である。

【図９】図５または図６に示した情報提供装置の電気的な構成を示したブロック図である。

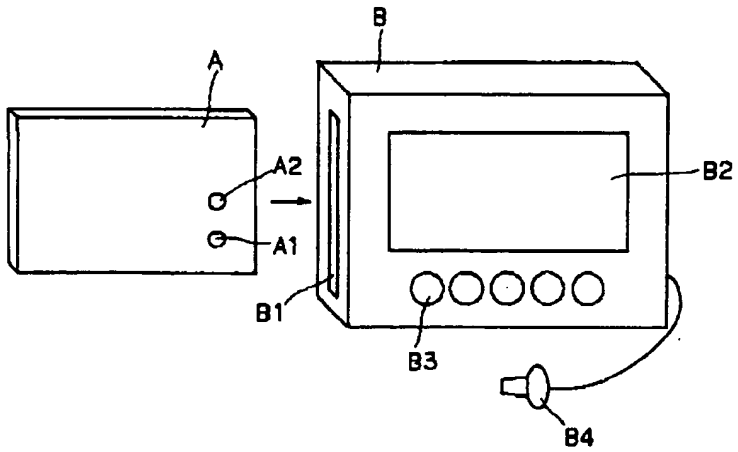
【図１０】本発明における情報取得装置と権利管理情報更新装置との交信作用を説明するフローチャートである。

【図１１】本発明における情報取得装置と情報提供装置との交信作用を説明するフローチャートである。

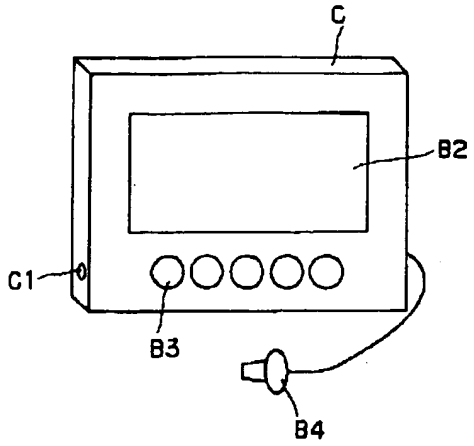
【符号の説明】

A　情報取得装置（情報記録装置）
B　情報再生装置
C　情報取得装置（情報記録再生装置）
D　情報提供装置（情報提供手段）
E　情報提供装置（情報提供手段）
F　情報入手希望者
G　権利管理情報更新装置（権利管理情報更新手段）
H　権利管理情報更新装置（権利管理情報更新手段）
J　管理センタ

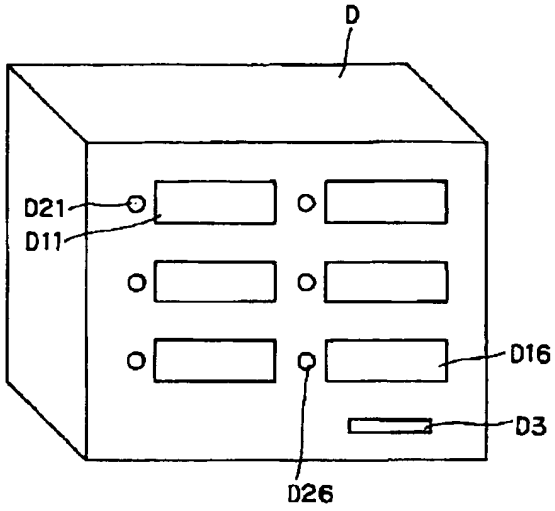
【図１】



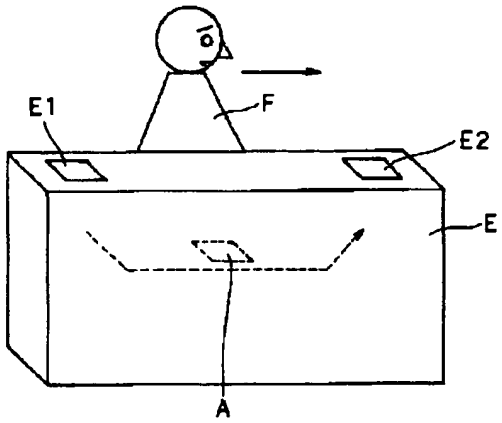
【図2】



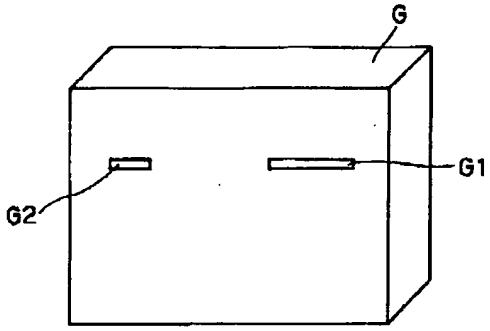
【図3】



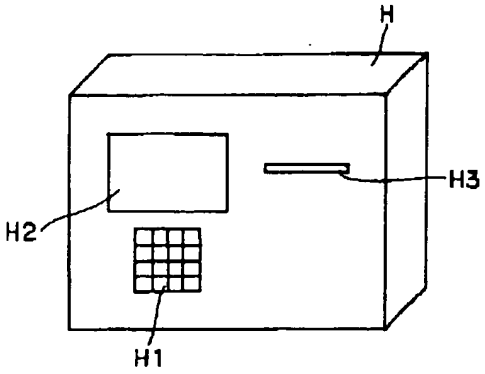
【図4】



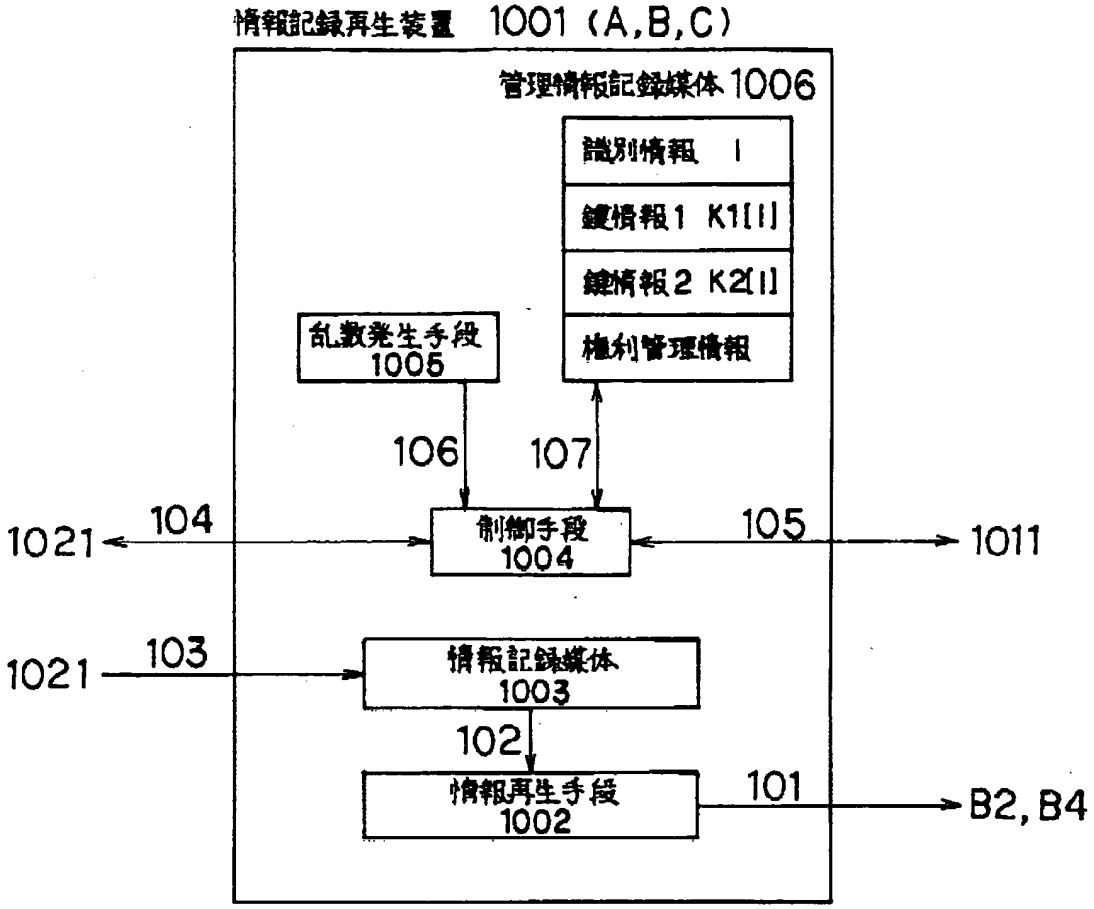
【図5】



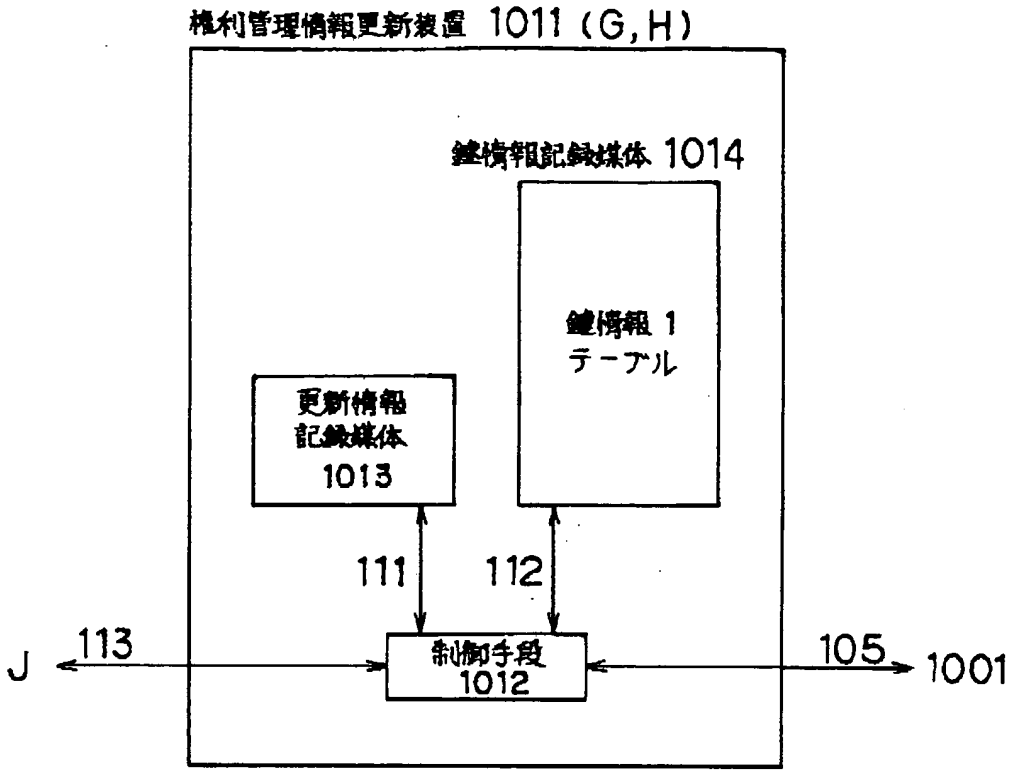
【図6】



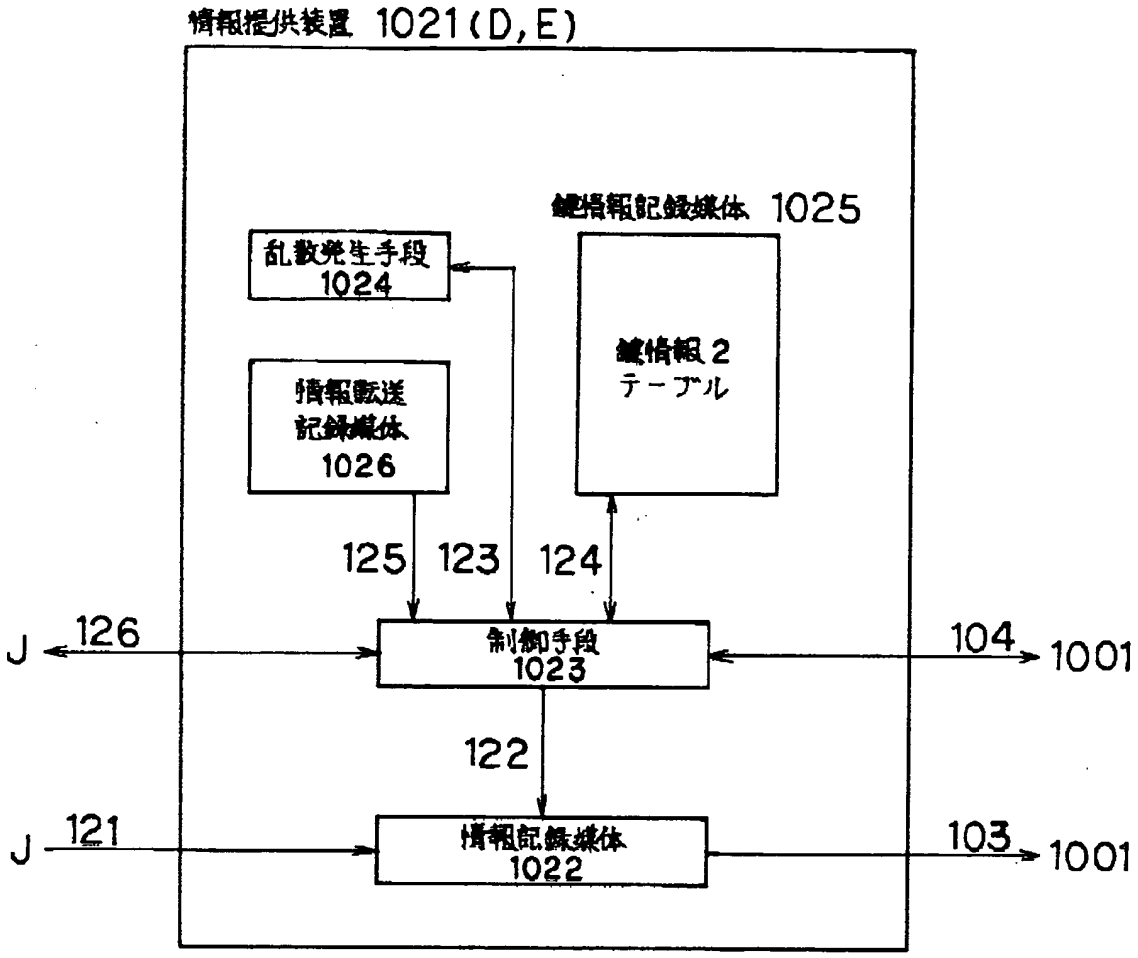
【図7】



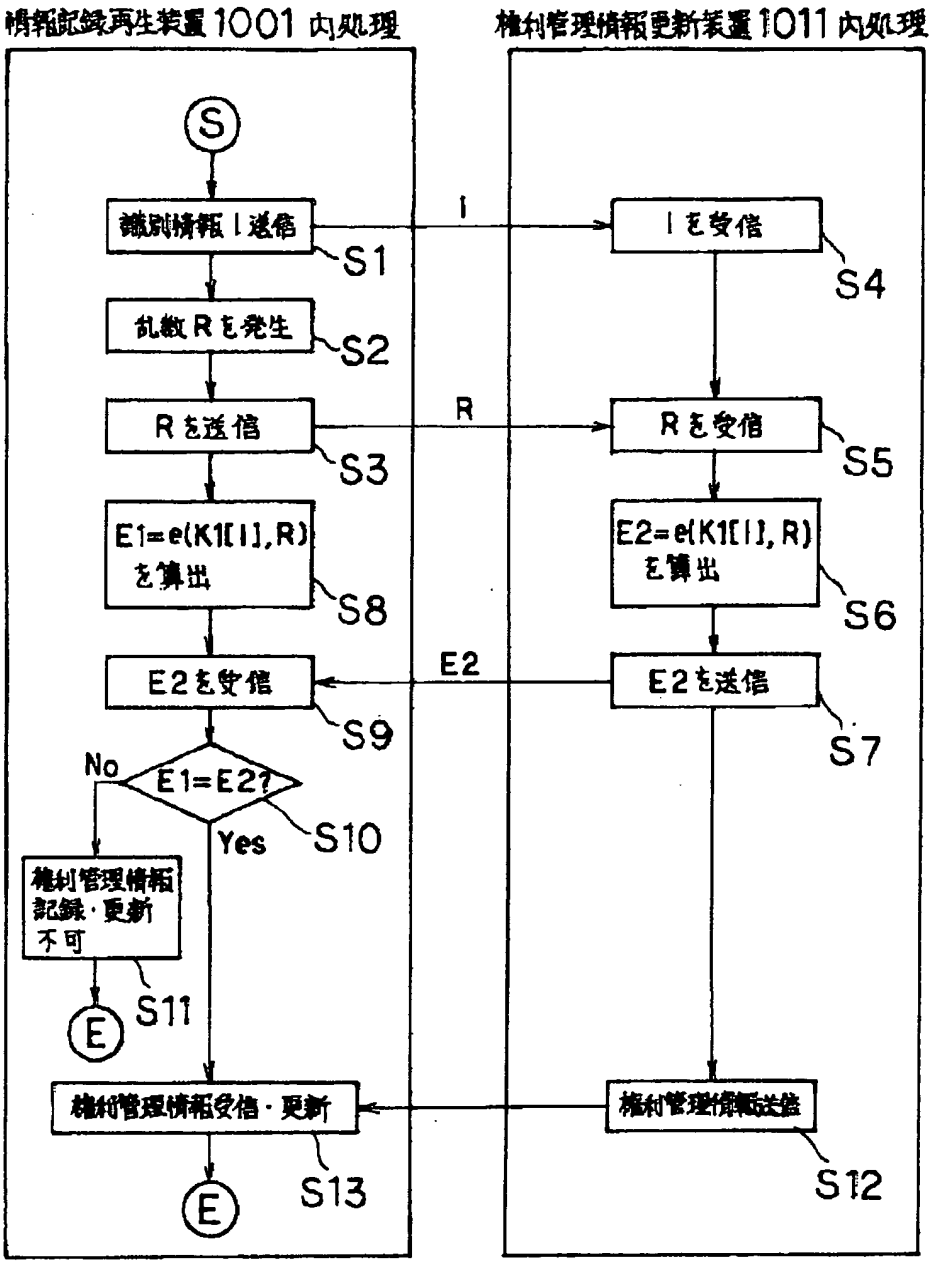
【図8】



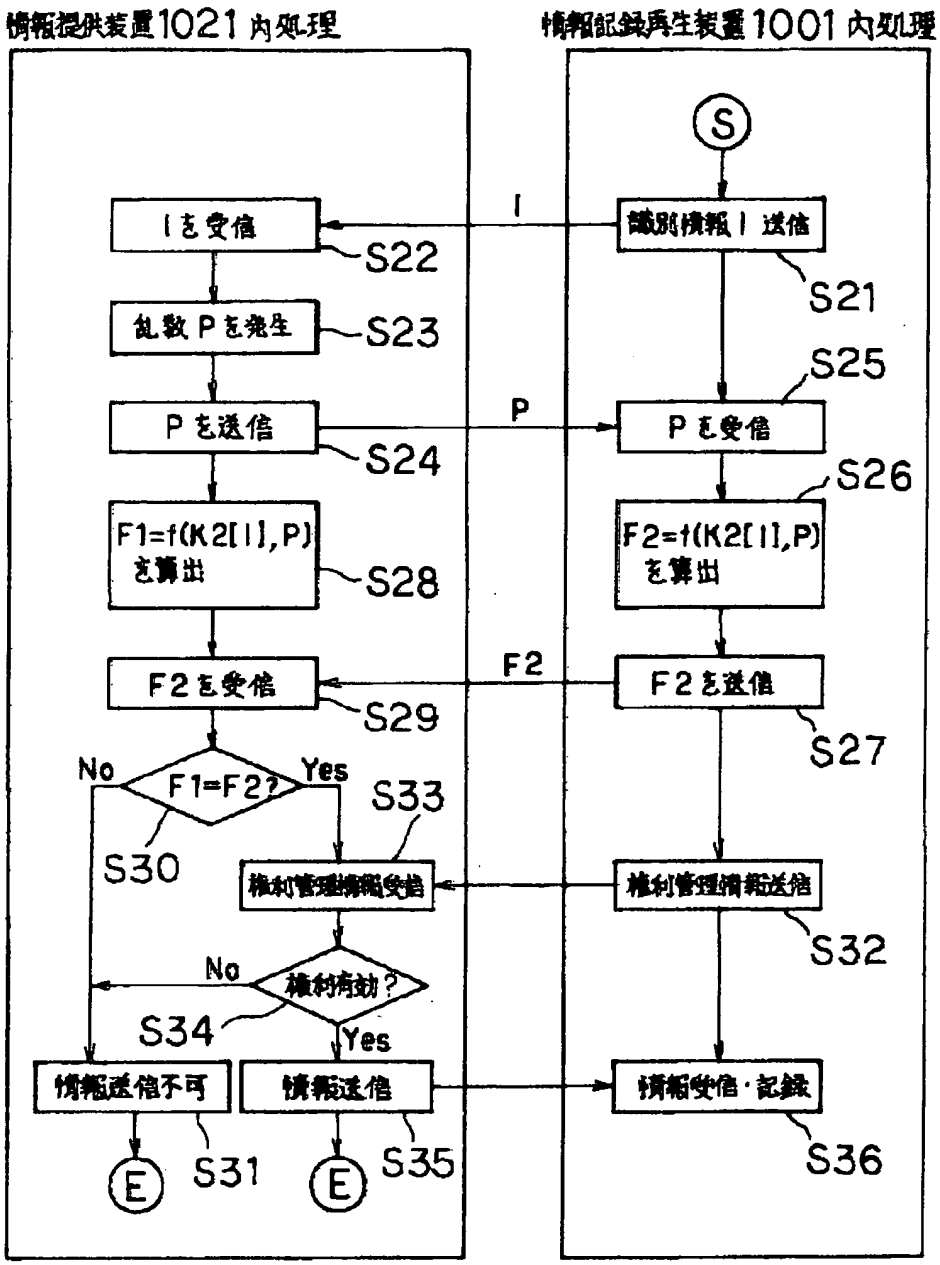
【図9】



【図10】



【図11】



フロントページの続き

(51)Int.Cl.⁵	識別記号	庁内整理番号	F I	技術表示箇所
G06K 19/00				
G07F 7/08				

Requested Patent: JP7084852A

Title: SECURITY SYSTEM FOR INFORMATION ;

Abstracted Patent: JP7084852 ;

Publication Date: 1995-03-31 ;

Inventor(s): TANAKA KAZUAKI; others: 02 ;

Applicant(s): HITACHI LTD ;

Application Number: JP19930225440 19930910 ;

Priority Number(s): ;

IPC Classification: G06F12/00; G06F3/14; G06F12/14 ;

Equivalents: ;

ABSTRACT:

PURPOSE:To offer information to limited users and to restrict the takeout of a copy of the information and also prevent it by providing a program, which controls a read of displayed screen data, with a function which limits the read.

CONSTITUTION:This system is provided with a database server 1, a client work station 2, and a network 3, and the window server 22 of the client work station 2 performs an input/output process on a display screen at a request made by a user interface program 13 in the server 1. Then a document management file while managing a document file itself wherein substance data on a document are stored manages whether the document can be referred to or copied. At this time, a reference management program checks a reference right management file for a document file whose reference management attribute indicates 'limited' to check whether or not a referring person has the right to refer or copy, thereby deciding whether the the person is allowed to refer to or copy the document file.

(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（A）　(11)特許出願公開番号

特開平7－84852

(43)公開日　平成７年(1995)３月31日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ	12/00	５３７　Ｍ	8944－5B		
	3/14	３４０　Ａ			
	12/14	３２０　Ａ			

審査請求　未請求　請求項の数９　ＯＬ　（全７頁）

(21)出願番号　特願平5－225440

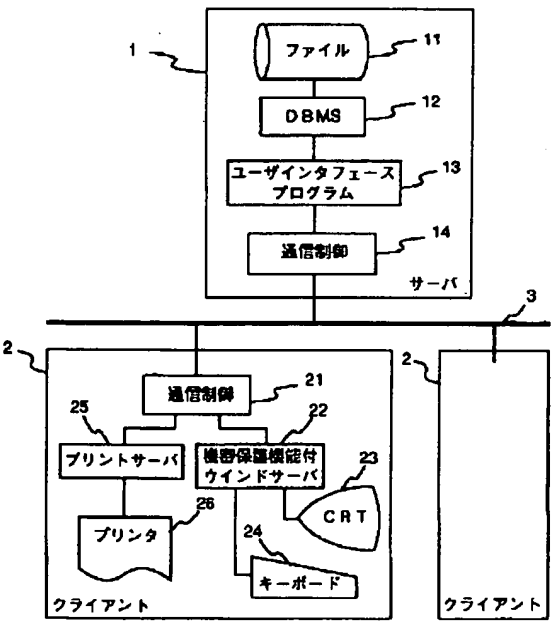
(22)出願日　平成５年(1993)９月10日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　田中　和明
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内
(72)発明者　能見　誠
茨城県勝田市市毛1070番地　株式会社日立製作所水戸工場内
(72)発明者　岩崎　一正
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内
(74)代理人　弁理士　小川　勝男

(54)【発明の名称】　情報の機密保護方式

(57)【要約】
【構成】データベースサーバ１，クライアントワークステーション等により構成し、ファイル参照管理で参照の可否，複写の可否を管理し、特に複写の可否によって他の媒体への複写を禁止する機能を備え、更に、画面表示を司るウインドサーバ２２に画面のハードコピーを制限する機能を設けた。
【効果】限られた範囲のユーザにデータベースの情報を提供し、かつ、その情報の複写持出しを制限，防止する上に大きな効果がある。

図１



1

【特許請求の範囲】

【請求項1】文書，データのファイルを蓄積し、それを利用する計算機システムにおいて、ファイルの参照と共に複写の可否を登録し、その可否によって参照，複写を制限することを特徴とする情報の機密保護方式。

【請求項2】請求項1において、情報を表示する端末の表示制御機能に表示画面の読みだしを制限する機能を付加することによって表示画面のハードコピーを制限する情報の機密保護方式。

【請求項3】請求項1において、情報の複写媒体ごとに複写の可否を制限する機能を備えた情報の機密保護方式。

【請求項4】請求項1において、複写の可否を参照者もしくは参照者グループ対応に登録し、その登録情報に基づいて複写を制限する情報の機密保護方式。

【請求項5】請求項1において、複写の可否を情報種別対応に登録し、その登録情報に基づいて複写を制限する情報の機密保護方式。

【請求項6】請求項2において、マルチウインドを表示する機能を備えた端末を具備し、前記端末においてウインド毎に読みだし可否の属性を管理する機能を備え、少なくも一つの読みだし禁止ウインドを表示している間は画面の表示データを読みだすことを禁止する機能を備えた情報の機密保護方式。

【請求項7】請求項6において、情報を送り出すプログラムより端末の表示ウインドに読みだし禁止属性を設定した後に、その読みだしを指示し、読みだしが不可能であれば禁止機能がはたらいていると判断して、複写禁止の対象となっている情報を表示する情報の機密保護方式。

【請求項8】請求項6において、情報を送り出す機能と表示する端末を一つの計算機で実現した情報の機密保護方式。

【請求項9】計算機のマルチウインドシステムのウインドサーバにおいて、各ウインド単位のウインド管理情報の中に複写可否を指定するための複写可否属性を付与し、ウインドサーバへの表示画面データ読みだし要求があったとき、表示中の全てのウインドの複写可否属性をチェックし、少なくも一つのウインドが複写可否属性が否を示す場合、画面表示データの読みだしを禁止する機能をこと備えたことをを特徴とするウインドサーバシステム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明はコンピュータ上に蓄積され、端末等によってそのデータを参照する情報システムに係り、その参照，複写等の制限を設けて情報の保安を管理する方法に関する。

【0002】

【従来の技術】従来、計算機上の情報の参照を制限する

2

最も単純な方法は、例えばワークステーションのオペレーティングシステム（以下ＯＳと略す）であるユニックス（Unix）の場合、利用者に識別コード、いわゆる、ＩＤコードを与え、かつ、そのＩＤコードに対応する個人のパスワードを与え、このパスワードを当人、あるいは特定のグループ以外に秘密とすることで、他人が管理する情報を許可無く参照することを禁止する方法がある。この方法では、このＩＤコード，パスワードの他に、プログラム，情報ファイルに参照権利属性を付与して書き込み，読みだしの権利を個人，グループ，その他の複数のレベルで管理できるようにしているのが一般的である。

【0003】この方法の拡張として、更に、特定の情報、例えば、データベース等を限定的に参照許可を与える方法として、ファイルのアクセス権ではなく、データベースへの接続権を限定し、更に特定のＩＤとパスワードを与え、階層的な保護を図る方法が利用されている。

【0004】一方、重要な情報の機密保護等を必要とする場合は、上記一般的な方法に加えて、参照の手続き、及び情報そのものを暗号化して特定の暗号化、復号機構を備えた端末等によりアクセスし、かつ暗号の鍵を特定者に与えることによってそれ以外の者が参照，解読，盗聴不可能なシステムが提供されている。

【0005】一方、著作権に基づく不法なプログラム，データの複写を防止するためには、コピープロテクトのための情報を付加し、その格納された媒体上に参照プログラムを同居させ、その参照プログラムによって目的とするプログラム，データを参照するもので、その参照プログラム以外のＯＳ等では直接参照出来ないようにすることによってプロテクトを実現する方法が一般的である。

【0006】

【発明が解決しようとする課題】以上のような保護方法を用い、汎用のUnixワークステーションあるいは汎用パーソナルコンピュータを用いて、例えば、企業内の情報システムを構築した場合、企業内の社員は自由、あるいは緩やかな制限で参照できる必要のある情報を扱った場合、参照そのものは問題無いが、それを物理的な媒体、例えばフロッピーディスクにコピーしたもの、あるいはプリントアウトした書類は外部には持ち出してはならない情報が存在する。製品の開発，設計，製造に関わる技術情報はその種の情報の一例である。

【0007】このような情報は、他の媒体に複写，印刷するのを防止することは、それを参照するプログラムを専用のものにすることにより比較的容易に実現することが可能である。

【0008】しかし、一般のワークステーション，パーソナルコンピュータは表示された画面をそのまま印刷する機能を持っているか、持っていなくても他のプログラムを動作させて表示されている画面を容易に印刷するこ

3

とが出来るようになっているため、自由に表示画面を印刷可能となり、一方、その画面印刷機能が無ければ通常の利用で不都合が生じることになり、その機能を無くすことは出来ないという問題がある。

【0009】

【課題を解決するための手段】以上の問題を解決するために、表示された画面データの読みだしを司るプログラムに読みだしを制限する機能を設け、かつ、読みだしを制限すべき情報の表示領域（ウインドー）に読みだし制限の属性を持たせることによって解決する。

【0010】

【作用】以上のような手段を設け、特定情報を表示するプログラムは上記読みだし制限の属性を表示ウインドーに与えて、表示し、一方、画面の読みだしを行うプログラムは読みだしを制限されたウインドーが少なくも一つ表示画面上に存在する場合は、その全面を読みだし禁止とするか、あるいは、読みだしを制限されたウインドーの領域のみ読みだし不可能として白地に抜いたデータを読みだし要求のあったプログラムに渡す等の手続きを実行することによって機密保護情報の印刷を防ぐことが出来る。

【0011】

【実施例】以下、本発明の実施例を図に従って説明する。

【0012】図1は本発明を実施する情報システム全体の構成例を示したもので、1は情報を蓄積し、要求に応じて情報を提供するデータベースサーバ、2はその情報を参照するためのクライアントワークステーション、3はネットワークである。

【0013】更に、データベースサーバ1の内部を機能的に表現したものとして、11はデータベースファイルであり文書等のデータ及びその管理情報が格納されている。12はデータベース管理システム（以下ＤＢＭＳ）、13はユーザインタフェースプログラム、14は通信制御機能である。

【0014】一方、クライアントワークステーション2の内部を機能的に表現したものとして、21はサーバ内の通信制御と対になってコマンド，データの授受を行うための通信制御機能である。22はウインドーサーバでUnixシステムのＸ－windowサーバ等がこれに相当し、サーバ1の中のユーザインタフェースプログラム13の要求に応じて表示画面上での入出力処理を行う。又、23はプリントサーバで13の要求に応じて文書や表示画面の印刷処理を行う。

【0015】このようなデータベースシステムにおいて、情報の機密管理は、通常計算機システムを利用する上でのＩＤ，パスワード管理と、データベースを利用する上でのＩＤ，パスワード管理が多重に行われ、情報の機密管理はＤＢＭＳ12又はユーザインタフェースプログラム13、あるいはその双方において後者を主体に行

4

われているが、参照の可否を管理しているに過ぎず、本発明では参照可否の他に参照結果の複写をも管理する。

【0016】図2はその複写管理方法の例を説明したもので、文書管理ファイルは文書の実体データの格納されている文書ファイル自身を管理すると同時に、その参照及び複写可否の管理を行う。

【0017】この時、参照管理プログラムはファイルの参照管理属性が「制限」を示している文書ファイルＷａの場合は、参照権管理ファイルをチェックして参照者が参照権，複写権を得ているかどうかをチェックしてその参照，複写を許可するかどうかを判定する。

【0018】図2の例の場合はＩＤが「ＮＮＮＮＮ」の参照者は参照権が「許可」であるが、複写権が「不許可」となっているため複写要求を出してもそれは受け付けられないのに対して、ＩＤが「ＭＭＭＭ」の参照者は複写権が「許可」と登録されているため、複写要求は受け付けられ、複写サービスを受けることが出来る。

【0019】一方、文書ファイルＦｂは参照管理属性がファイルそのものを複写不可とする「複写禁止」となっているため、参照者の権利のいかんに関わらず複写は許可されない。

【0020】他方、参照管理属性が「無制限」となっている文書ファイルＦｃとＦｄは参照のみならず複写の制限もなく、無条件に複写サービスを受けることが可能であることを示している。

【0021】このように、ファイルの参照，複写サービスを提供するか否かの管理を行うことが可能となる。

【0022】この場合、「複写」の持つ意味は、データとしてファイルをフロッピーディスク等の他のメディアや他のワークステーションの磁気ディスク等の外部記憶装置への複写のみならず、プリンタへの出力も含まれるが、何に複写するかまでも細かく管理したい場合は、複写権管理ファイルの管理項目をきめ細かくすることによって可能である。

【0023】しかし、以上のような管理を行っても、Unix等のＯＳ，Ｘ－windowのウインドサーバを搭載したワークステーションシステムの場合は、全く異なるプログラムを並行して動作させ、かつ、同一のディスプレイにマルチウインドで複数の画面を同時に表示することが出来、複写禁止の文書を一つのウインドで表示し、それとは関係のないプログラムを動かし今一つのウインドを表示し、そこから表示画面のハードコピーをウインドサーバ，プリントサーバを経由して複写禁止の画面をコピーすることは可能で、従来のウインドサーバ，プリントサーバをそのまま利用する限りこれを防止することは出来ない。

【0024】それを示したのが図3で、ウインドＷａは複写禁止の文書を表示しており、他方、ウインドＷｂ，Ｗｃ，Ｗｄは複写を禁止されていない情報を表示している場合を示したもので、例えば、ウインドＷｂより画面

5

のハードコピーを要求するプログラムを動かせば、複写禁止ウインドを含めて画面全体をプリントアウトすることが出来る。

【0025】もちろん、ウインドWaを全面に表示し、ウインドWbを一時消去して、複写禁止文書の画面のみをプリントアウトすることも可能である。

【0026】そこで本発明は更に、画面（ウインド）を表示する時に、ウインドの属性として複写管理属性を持たせ、それが「複写禁止」となっているウインドが表示されている場合は、表示画面の読み取り，複写を禁止する機構を設けてそれを防止するものである。

【0027】図4はその属性を付加したウインド管理テーブルの一例を示したもので、一つのウインド管理テーブルには複写権の有無を設定する項目を追加し、既にある表示状態管理情報とを用いてその管理制御を行う。

【0028】図4で示す例は、ウインドWaが複写禁止属性を持ち、他のウインドWb，Wc，Wdは複写禁止となっていない。

【0029】従って、この場合は、ウインドWaが画面上に表示されている限りは他のウインドから画面コピーを要求しても、ウインドを管理しているウインドサーバが画面データを渡さないようにしておけば以上の問題を解決することが出来る。

【0030】図5は、参照者の要求に基づいて例えば文書ファイルを検索し、それを表示するユーザインタフェースプログラムのフローチャートの一例を示したもので、ステップ301，302は参照権チェックのためのユーザID，パスワードの入力である。この場合、ユーザID、及びパスワードの入力はワークステーションを利用開始時に入力するものをそのまま使用しても良いし、管理を厳しくするために別に設けても良い。又、ステップ301，302でユーザID、及びパスワードの入力だけではなく、このデータベースシステムのサービスを提供すべきユーザであるかどうかをチェックしてもよい。

【0031】次に303で検索項目を入力し、それに基づいてステップ304でデータベースに登録されたファイルを検索し、該当なファイルが存在しない場合は終了とし、ファイルがあった場合は次のステップ306，307で参照権，複写権のチェックを行う。

【0032】この時のチェックは図2に示す文書管理ファイル，参照管理ファイルのデータに基づいて行う。

【0033】ここで一般のデータベースと同様に、ユーザに参照権が与えられていない場合は、ここでは省略しているが、許可されていないことをユーザにメッセージを出して終了する。

【0034】一方、参照権が与えられている場合は、更にステップ307で複写権が与えられているかチェックし、複写権が与えられている場合はステップ310で複写許可の属性を持つウインドを生成し、他方、複写権の

6

無い場合はステップ309でウインドサーバが画面読み出しを禁止出来ているかどうかを確認し、読み出し可能であれば禁止機能がないとして表示を打切り、禁止出来ていれば読みだし保護が可能であると判断してステップ311以降の処理に入る。

【0035】そのあとでステップ311でファイルを読みだし、ステップ312でその表示データをクライアントワークステーション2のウインドサーバに送信することでユーザが利用しているワークステーションの表示画面上に文書ファイルが図3の例のように表示されることになる。

【0036】次に、図6はこの画面のハードコピーを他のプログラム等からプリントサーバに要求したときのプリントサーバのハードコピープログラムの一例を示したもので、印刷要求によって処理を開始し、ステップ401で表示画面の読みだし受渡しをウインドサーバに要求し、ステップ402でリターン情報でそれが正常に受け渡されたか、あるいは、複写禁止ウインドが含まれているかをチェックし、禁止されていなければ、ステップ403でその画面データをプリンタへ送り、ステップ404で印刷完了を待って終了する。

【0037】一方、画面の表示データを要求されたウインドサーバは、図7の例に示すように、ステップ501でウインド管理テーブルをチェックし、複写禁止属性を持つウインドが表示されていないときのみステップ502で画面データを読みだし、ステップ503で要求元にそのでデータを受け渡す。

【0038】以上のように、ウインドサーバが表示画面データの読みだしをウインド管理情報の複写権属性と表示状態を監視し、その読みだし可否によって受渡しを管理制御出来るようにすることによって表示画面の複写を制限する機能を実現することが可能となる。

【0039】次に、図8は文書ファイルをデータベースに登録する際の参照管理を行うときの処理を示したもので、ステップ601で参照属性を入力し、ステップ602でその属性を判断し、参照が禁止の場合はステップ603で参照禁止属性を付与し、参照制限であればステップ604で参照制限属性を付与し、ステップ605でその参照を管理するための参照管理ファイルを生成する。無制限であればステップ606で参照無制限属性を付与して、ステップ607でファイルを格納する。

【0040】この場合、基本の管理は参照のみとしそのもとで複写の可否を制限したものを示したが、情報の性格から無条件に複写が禁止されるべきものもあり、この場合は属性情報に複写禁止属性を追加すればその管理を行うことが出来る。

【0041】更にユーザごとの参照権登録処理のフローチャートとを示したものが図9である。図9において、データベース管理者は登録対象者と参照対象文書ファイルを指定し、システムはステップ701，702でそれ

7

を入力する。次にそれに対応した参照権をステップ７０３で入力し、ステップ７０４でそれをチェックし、参照権が与えられている場合はステップ７０５で参照許可属性を設定し、不許可であればステップ７０９で不許可属性を設定する。

【００４２】更に、参照権が与えられている場合は複写権をチェックし、複写権を与えている場合はステップ７０７で複写許可を設定し、与えない場合は不許可を設定する。その結果をステップ７１０で参照管理ファイルに登録して終了する。

【００４３】尚、図９では文書，参照者毎に参照権の設定を行っているが、文書の種類，参照者グループ毎の単位で設定，管理を行っても良い。

【００４４】

【発明の効果】本発明によれば、限られた範囲のユーザにデータベースの情報を提供し、かつ、その情報の複写持出しを制限，防止する上で大きな効果がある。

【図面の簡単な説明】

【図１】本発明の一実施例のブロック図。

【図２】本発明の文書管理ファイルの実施例の説明図。

【図３】本発明の画面表示例の説明図。

8

【図４】本発明のウインド管理テーブルの実施例の説明図。

【図５】本発明のデータベース検索処理実施例のフローチャート。

【図６】本発明の一実施例の画面印刷プログラムのフローチャート。

【図７】本発明の一実施例のウインドサーバの画面読み出し機能のフローチャート。

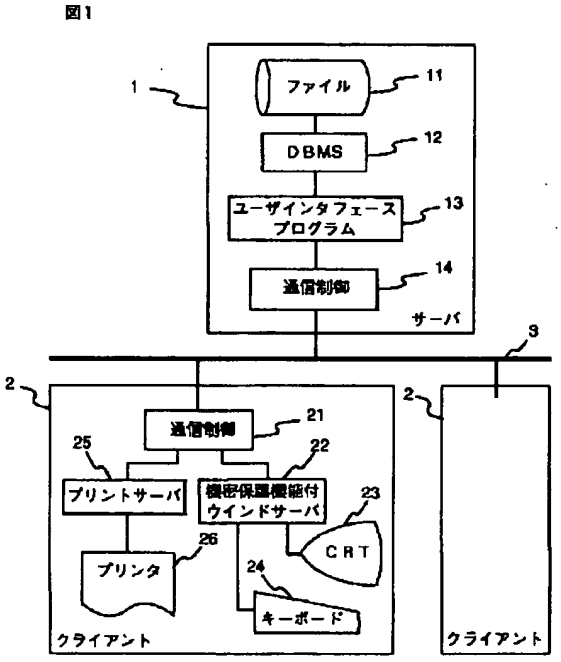
【図８】本発明の一実施例のデータベース参照，複写管理属性設定処理のフローチャート。

【図９】本発明の一実施例の参照者単位の参照権，複写権登録プログラムのフローチャート。

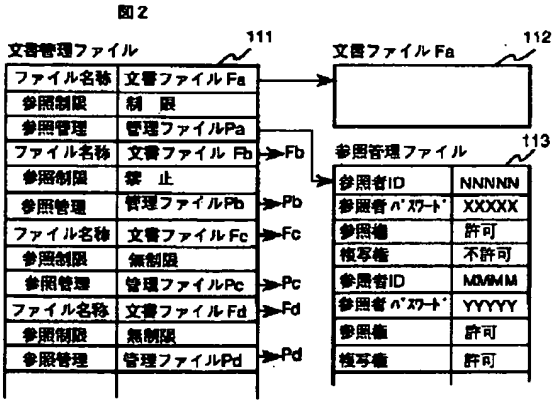
【符号の説明】

１…データベースサーバ、１１…その中のファイルシステム、１２…データベース管理システム、１３…ユーザインタフェースプログラム、１４…通信制御機能、２…クライアントシステム、２１…通信制御機能、２２…機密保護付きウインドサーバ、２３…ディスプレイ装置、２４…キーボード、２５…プリントサーバ、２６…プリンタ、３…サーバとクライアントを結ぶネットワーク。

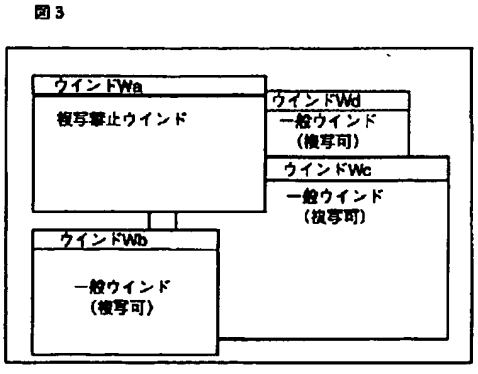
【図１】



【図２】

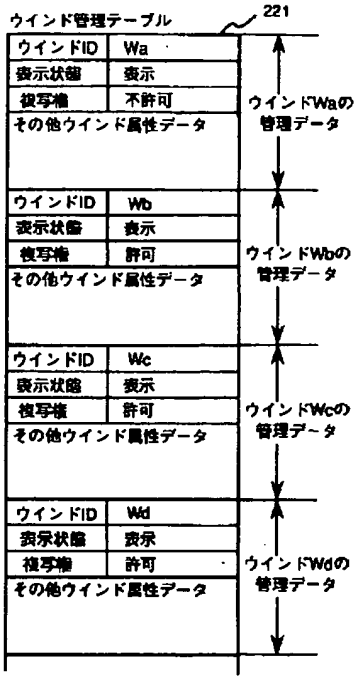


【図３】



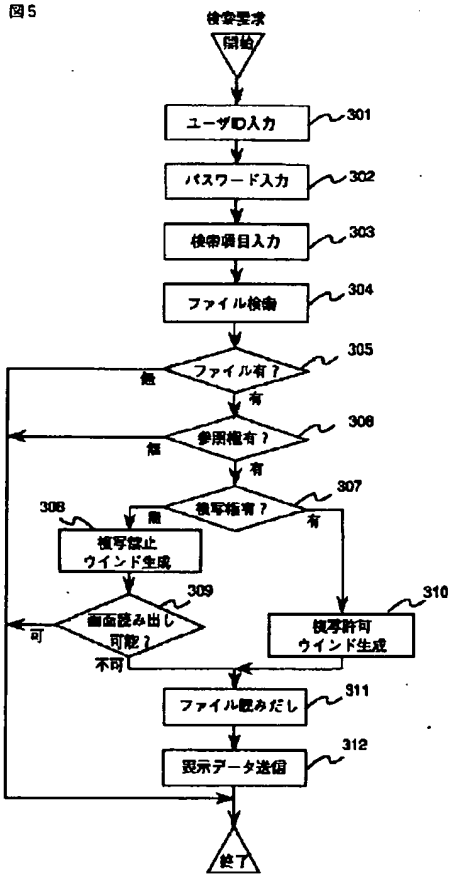
【図４】

図4



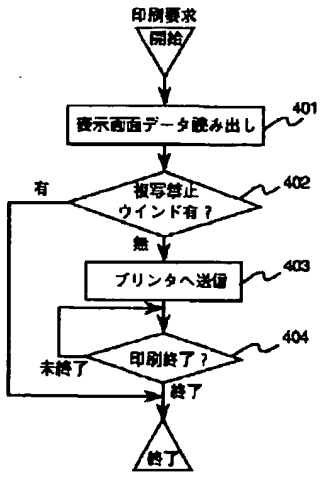
【図５】

図5



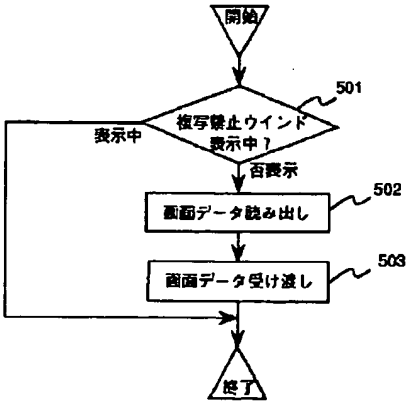
【図６】

図6

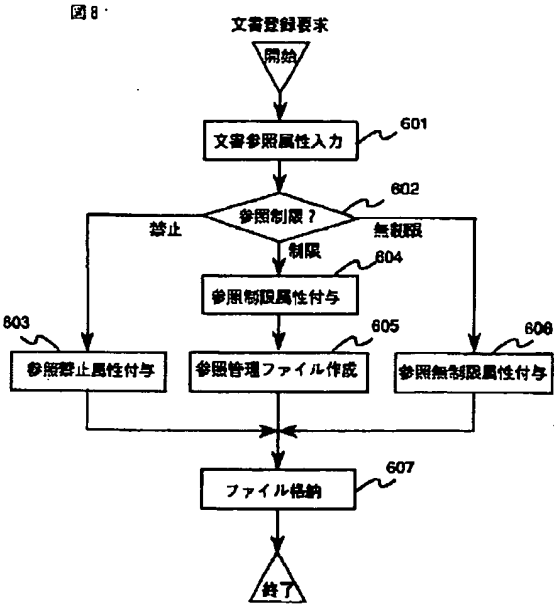


【図７】

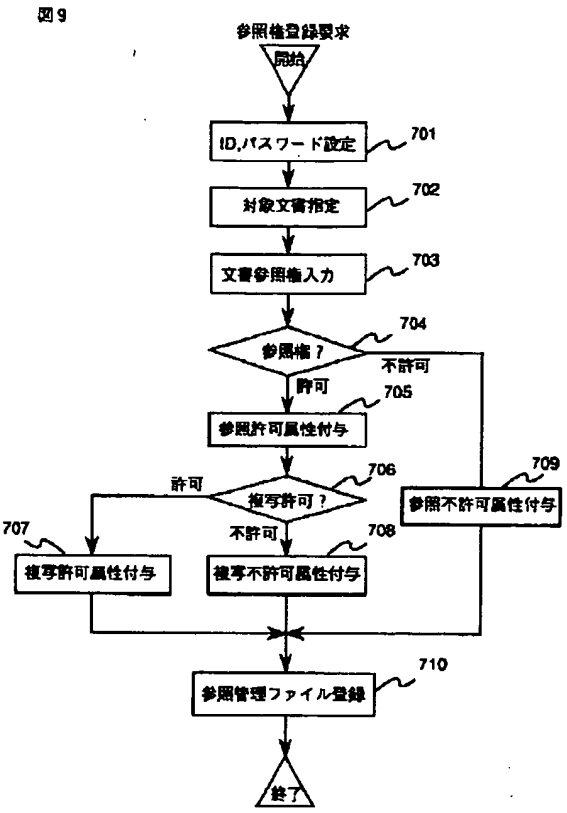
図7



【図８】



【図９】



Requested Patent: JP7200317A

Title: OPERATION RIGHT MANAGEMENT EQUIPMENT ;

Abstracted Patent: JP7200317 ;

Publication Date: 1995-08-04 ;

Inventor(s): TAKAHASHI TOSHINARI; others: 04 ;

Applicant(s): TOSHIBA CORP ;

Application Number: JP19930349335 19931228 ;

Priority Number(s): ;

IPC Classification: G06F9/46; G06F1/00; G06F12/14 ;

Equivalents: ;

ABSTRACT:

PURPOSE:To attain flexible operation right management for threads by providing a means storing revision right information and a means referencing the stored revision right information before execution of revision of operation right information and verifying whether or not the revision is permitted to the equipment.

CONSTITUTION:The equipment is provided with a means storing revision right information representing whether or not revision of operation right information is permitted based on a thread or the user being a subject of the revision or a memory area or a program in which the subject of revision is in existence and a means referencing the stored revision right information before execution of the revision of the operation right information and verifying whether or not the revision is permitted. That is, an operation right list storage section 7 stores a table relating to thread protection and storing a fact of the right executing the operation to each thread and used for an operation right discrimination section 12. Thus, not only the management of the operation right to the thread but also the revision right is flexibly expressed and managed by the combination of various conditions.

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（A）**　(11)特許出願公開番号

特開平7－200317

(43)公開日　平成７年(1995)８月４日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ	9/46	３４０　Ｂ	7629－5Ｂ		
	1/00	３７０　Ｅ			
	12/14	３１０　Ｋ			

審査請求　未請求　請求項の数３　ＦＤ　（全 14 頁）

(21)出願番号　特願平5－349335

(22)出願日　平成５年(1993)12月28日

(71)出願人　000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地

(72)発明者　高橋　俊成
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内

(72)発明者　岡本　利夫
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内

(72)発明者　福本　淳
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内

(74)代理人　弁理士　鈴江　武彦

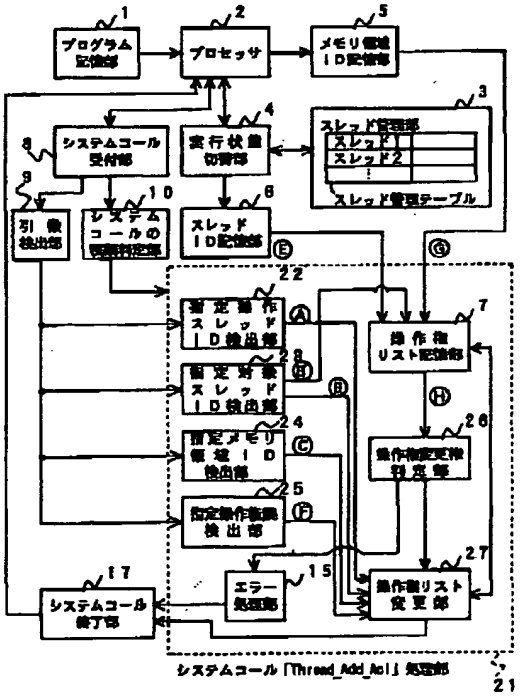
最終頁に続く

(54)【発明の名称】　操作権管理装置

(57)【要約】

【目的】　スレッドの操作権管理の柔軟化を目的とする。

【構成】　本発明は、複数の並行して実行されるスレッドによりプログラムを実行する手段と、各スレッドに対する操作を、この操作を行う主体となるスレッドまたはユーザ、もしくはこの操作を行う主体が存在するメモリ領域またはプログラムの少なくともいずれかに基づいて、許可するか否かを指示する操作権情報を記憶する手段とを備える操作権管理装置において、前記操作権情報の変更を、この変更を行う主体となるスレッドまたは、ユーザもしくはこの変更を行う主体が存在するメモリ領域またはプログラムの少なくともいずれかに基づいて、許可するか否かを示す変更権情報を記憶する手段と、前記操作権情報の変更を実行する前に、記憶された前記変更権情報を参照して、この変更が許可されるか否かを検証する手段とを具備したことを特徴とする。



1

【特許請求の範囲】

【請求項1】複数の並行して実行されるスレッドによりプログラムを実行する手段と、

各スレッドに対する操作を、この操作を行う主体となるスレッドまたはユーザ、もしくはこの操作を行う主体が存在するメモリ領域またはプログラムの少なくともいずれかに基づいて、許可するか否かを指示する操作権情報を記憶する手段とを備える操作権管理装置において、

前記操作権情報の変更を、この変更を行う主体となるスレッドまたは、ユーザもしくはこの変更を行う主体が存在するメモリ領域またはプログラムの少なくともいずれかに基づいて、許可するか否かを示す変更権情報を記憶する手段と、

前記操作権情報の変更を実行する前に、記憶された前記変更権情報を参照して、この変更が許可されるか否かを検証する手段とを具備したことを特徴とする操作権管理装置。

【請求項2】実メモリまたは仮想メモリを複数のメモリ領域の集合として管理する手段と、

各メモリ領域に対する操作を、この操作を行う主体となるスレッドまたはユーザ、もしくはこの操作を行う主体が存在するメモリ領域またはプログラムの少なくともいずれかに基づいて、許可するか否かを指示する操作権情報を記憶する手段とを備える操作権管理装置において、

前記操作権情報の変更を、この変更を行う主体となるスレッドまたは、ユーザもしくはこの変更を行う主体が存在するメモリ領域またはプログラムの少なくともいずれかに基づいて、許可するか否かを示す変更権情報を記憶する手段と、

前記操作権情報の変更を実行する前に、記憶された前記変更権情報を参照して、この変更が許可されるか否かを検証する手段とを具備したことを特徴とする操作権管理装置。

【請求項3】前記変更権情報を前記操作権情報に付随されて記憶することを特徴とする請求項1または2に記載の操作権管理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、計算機システム中でのメモリ領域またはスレッドへの操作権を管理する操作権管理装置に関するものである。

【0002】

【従来の技術】近年マイクロプロセッサの性能向上により複数の小型のコンピュータをネットワークでつないで使用することが通常のこととなり、ネットワーク結合された計算機やユーザー同士での協調作業が可能となり、操作権の管理を厳密に行う必要が出てきた。

【0003】またオブジェクト指向技術の進歩やサーバ・クライアント・モデルのような新しいパラダイムの出現により、従来の記憶装置やファイルを単位とした操作

2

権管理では十分な性能が得られなくなってきた。

【0004】こういった問題を解決するために、単一仮想記憶などの技術により、メモリやCD－ROMやハードディスク・ドライブのような記憶装置、さらにはディスプレイやキーボードといったあらゆる計算機資源を同一の概念で管理するオペレーティング・システムの設計が試みられている。中でも、各種の資源を同一の仮想空間上に配置する単一仮想記憶の技術は特徴的なものである。さらに、この単一仮想記憶をいくつかのブロックに分割し、ブロック単位で操作権を管理する方法が提案されている。この管理単位を「メモリ領域」と呼び、あらゆる記憶装置の操作権管理の単位として用いられる。

【0005】一方、オペレーティング・システムはマイクロ・カーネル技術の定着により、プログラムの実行を管理するスレッドという単位を使って管理するのが一般的になりつつある。すなわち、プログラム実行の制御対象としてプロセスまたはスレッドと呼ばれる制御単位が用いられる。個々のスレッドは独立にプロセッサのレジスタ値やスタックなどを持ち、オペーレーティング・システムによりプロセッサを割り当てられた間だけプログラムを実行し、一定時間後またはハードウェア割り込みなどが発生したときにオペレーティング・システムはスレッドに対するプロセッサの割り当てを解除して他のスレッドへプロセッサを割り当てる。個々のスレッドは独立したレジスタ値を持っているため、仮想的に並列に動いているプロセッサであるとみなすことができる。

【0006】このようなオペレーティングシステムでは、メモリ領域とスレッドという二つの概念を単位として計算機資源を構成し、そこへの操作権を管理する情報もACL (Access Control List)という形で各メモリ領域および各スレッドに付随させて管理することが行われる。

【0007】例えば、スレッドの場合を例に説明すると、スレッド自体を操作したいとき、すなわちスレッドの実行を一時停止したり再開したりしたいときやスレッドを強制的に消滅させたいとき、またはスレッドの情報（レジスタ値やスタック等の内部状態）を調べたいときのため、オペレーティングシステムはスレッド操作命令群を備えており、あるスレッドから別のスレッドを操作することが可能である。このようなスレッド操作命令群は、ユーザプログラムからシステムコールとしてオペレーティング・システムに指示し、利用できるようになっている。

【0008】このようなスレッド操作命令があるときに、どのスレッドからでもスレッド操作命令を発行できるようにすると、不当な操作もできることになってしまうので保護の機能が必要になる。たとえばスレッドを強制的に消滅させるようなことは特定の権限を持つスレッドだけから行えるようにすることが必要になる。

【0009】さらに、仮想空間が複数あり、仮想空間ご

3

とに1つのプログラムやデータが配置されており、スレッドもその空間内しか動きまわれないようなＭａｃｈ（参考文献：「分散オペレーティング・システム」前川守他編、共立出版）をはじめとするオペレーティング・システムの場合、同じ空間内に存在するスレッドの保護を同一とし、スレッドの保護をメモリ空間の保護に依存する方法をとっている。

【００１０】しかし、それらのスレッドを区別して保護することはできない。また、単一仮想空間のシステムのように、１つのアドレス空間内にすべてのプログラムが配置され、複数のスレッドが存在する場合、スレッドの保護はメモリ保護だけではなくオペレーティングシステムによる権限の設定が必要になる。

【００１１】この問題のひとつの解決法は、スレッドの所有者を決めて、同一の所有者のスレッド間ではスレッドの操作を許す方式であり、Unixオペレーティングシステムのsignalはこれに相当する。しかしこれでは同一の所有者のスレッド間の保護の柔軟性に欠け、さらに、操作権限を他の所有者のスレッドに渡すことができない。また、複数の所有者のスレッドからのスレッド操作を可能にしたりできないなど、柔軟にスレッドの操作ができるようなスレッド保護機能が不足していた。

【００１２】また、一方ではデータ・ファイルへの操作権の多彩な設定ができるように、ファイルやディレクトリに対するＡＣＬに、情報の追加権、削除権、ＡＣＬの変更権などのさまざまな情報を付加して柔軟性を高めようとするＤＣＥ（参考文献：ＯＳＦ　ＤＣＥ技術解説、ソフトリサーチセンター）のような試みもあるが、やはり複数の所有者にメモリ領域のＡＣＬの管理を行わせるような柔軟な保護機能は不足している。

【００１３】

【発明が解決しようとする課題】以上述べてきたことに基づいて従来の操作権管理装置の主要な課題をまとめると次のようになる。

【００１４】（1）従来、スレッドに対する操作を、この操作を行う主体や、この操作を行う主体が存在するメモリ領域などに基づいて、許可するか否かを指示する方法は試みられている。しかし、スレッドに対するこの操作権情報を変更する権利については、例えば、そのスレッドの所有者には認める、あるいはルートと呼ばれる特別のユーザにのみ認める、といった単純な管理しか行うことができなかった。

【００１５】（2）また、従来、実メモリ、仮想メモリ、データ・ファイルなどのメモリ領域に対する操作を行う主体や、この操作を行う主体が存在するメモリ領域などに基づいて、許可するか否かを指示する方法は試みられている。しかし、メモリ領域に対するこの操作権情報を変更する権利については、例えば、そのメモリ領域の所有者には認める、あるいはルートと呼ばれる特別のユーザにのみ認める、といった単純な管理しか行うこと

4

ができなかった。

【００１６】（3）さらに、スレッドやメモリに対する操作権を柔軟に記述し、かつ、同様に変更権をも柔軟に記述すると、柔軟な記述ができるようになる反面、スレッドやメモリに対する保護のためのデータ量が多くなり、その管理が複雑になるという欠点を持つという問題がある。

【００１７】このように、従来の操作権管理装置では、メモリ領域単位での操作権限の設定ができなかったり、操作権限の設定の面で柔軟性に欠けたり、または、同一の仮想アドレス空間を共有しているスレッド間では保護が無かったか、または制約が強かった。

【００１８】本発明は、上記課題を考慮してなされたものであり、別々の仮想空間にいるスレッド間はもとより、同一仮想アドレス空間を共有しているスレッド間でのスレッド操作命令あるいは個々のメモリ領域に対して、同一の方法で柔軟性があり、かつ十分な保護を行う操作権管理装置を提供することを目的とする。

【００１９】

【課題を解決するための手段】上記目的を達成するために、本発明（請求項1）では、複数の並行して実行されるスレッドによりプログラムを実行する手段と、各スレッドに対する操作を、この操作を行う主体となるスレッドまたはユーザ、もしくはこの操作を行う主体が存在するメモリ領域またはプログラムの少なくともいずれかに基づいて、許可するか否かを指示する操作権情報を記憶する手段とを備える操作権管理装置において、前記操作権情報の変更を、この変更を行う主体となるスレッドまたは、ユーザもしくはこの変更を行う主体が存在するメモリ領域またはプログラムの少なくともいずれかに基づいて、許可するか否かを示す変更権情報を記憶する手段と、前記操作権情報の変更を実行する前に、記憶された前記変更権情報を参照して、この変更が許可されるか否かを検証する手段とを具備したことを特徴とする。

【００２０】また、本発明（請求項2）では、実メモリまたは仮想メモリを複数のメモリ領域の集合として管理する手段と、各メモリ領域に対する操作を、この操作を行う主体となるスレッドまたはユーザ、もしくはこの操作を行う主体が存在するメモリ領域またはプログラムの少なくともいずれかに基づいて、許可するか否かを指示する操作権情報を記憶する手段とを備える操作権管理装置において、前記操作権情報の変更を、この変更を行う主体となるスレッドまたは、ユーザもしくはこの変更を行う主体が存在するメモリ領域またはプログラムの少なくともいずれかに基づいて、許可するか否かを示す変更権情報を記憶する手段と、前記操作権情報の変更を実行する前に、記憶された前記変更権情報を参照して、この変更が許可されるか否かを検証する手段とを具備したことを特徴とする。

【００２１】また、望ましくは、上記発明（請求項1）

において、スレッド作成時には、あらかじめ定められたデフォルトの変更権情報が設定されることを特徴とする。

【0022】また、望ましくは、上記発明（請求項2）において、メモリ領域作成時には、あらかじめ定められたデフォルトの変更権情報が設定されることを特徴とする。

【0023】さらに、上記各操作権管理装置において、前記操作権情報を記憶する手段の情報を変更することが不可能になるような、前記変更権情報を記憶する手段の情報の変更を認めないようにしても良い。

【0024】また、本発明（請求項3）では、上記発明（請求項1または2）において、前記変更権情報を前記操作権情報に付随されて記憶することを特徴とする。

【0025】さらに、上記各操作権管理装置において、前記操作権情報を記憶する手段の情報を変更することのできる複数の条件を記憶する操作権管理権限記憶手段と、前記変更権情報を記憶する手段の情報に変更を加えようとする際に該操作権管理権限記憶手段の情報によりこの変更を行う権限があるか否かを検証する手段と、この権限があると検証された場合に前記変更を許可する手段とをさらに設けても良い。また、この場合、前記変更権情報を記憶する手段の情報を変更することが不可能になるような、前記操作権変更権管理権限記憶手段の情報の変更を認めないようにしても良い。

【0026】

【作用】本発明（請求項1）によれば、スレッドへの操作権の管理のみならず、その操作権の管理の権限である変更権についても、さまざまな条件の組み合わせによって柔軟に表現、管理することができるようになる。

【0027】この機構を用いれば、スレッドの操作権に関して、特定のスレッドにのみその変更を許可したり、またそれらの複数の条件を組み合わせて表現したりすることができる。

【0028】特に、近年多く利用されるようになったグループウェアと呼ばれる計算機の協調作業アプリケーションを作成する際には、スレッドの共有や、排他制御、他のプログラムからの保護、などのきめ細かな制御を容易に記述することが必要になるが、本発明を用いればこの記述を柔軟に行うことができる。

【0029】また、プログラムの誤りや、システム・コールの不正な呼び出しなどによって、スレッドへの操作権が異常に書き換えられてしまうことを未然に防ぐことができるので、スレッドの共有を行いつつもなおスレッドの動作に関する信頼性が飛躍的に向上する。

【0030】また、本発明（請求項2）によれば、メモリ領域への操作権の管理のみならず、その操作権の管理の権限である変更権についても、さまざまな条件の組み合わせによって柔軟に表現、管理することができるようになる。

【0031】この機構を用いれば、メモリ領域への操作権に関して、特定のスレッドにのみその変更を強化したり、特定のメモリ領域に存在するスレッドにのみその変更を許可したり、またそれら複数の条件を組み合わせて表現したりすることができる。

【0032】特に、近年利用が試みられている単一仮想記憶と呼ばれる、複数のアプリケーションが同一の仮想空間上で動作するオペレーティング・システムにおいては、メモリ領域の共有や、排他制御、他のプログラムからの保護、などのきめ細かな制御を記述することが必要となるが、本発明を用いればこの記述を柔軟に行うことができる。

【0033】また、プログラムの誤りや、システム・コールの不正な呼び出しなどによって、メモリ領域への操作権が異常に書き換えられてしまうことを未然に防ぐことができるので、メモリ領域に書かれたデータが不用意に書き換えられてしまうことがおこらず、メモリ領域の共有を行いつつもなおメモリ領域に書かれたデータに対する信頼性が飛躍的に向上する。

【0034】さらに、本発明（請求項3）によれば、スレッドやメモリに対する操作権情報と、その操作権情報の変更を行う変更権情報とは、一般に共通点が多く、データ構造も似通ったものになることが多い。本発明においては、変更権情報を操作権情報に付随させて記憶することにより、共通のデータをなるべく一つのものとして管理するため、データ量を少なくすることができ、変更権および操作権を検証するための手続きも共通化する。

【0035】また、スレッドやメモリに対する操作権や変更権などの管理は、一般にメモリ管理ユニットと呼ばれるハードウェアを用いることによって、実行性能を向上させることができる。そのような場合に、操作権および変更権それぞれに別のハードウェアを用意することは、計算機の構成を複雑にする可能性がある。しかし、本発明によって、操作権と変更権を同一に管理することにより、必要なハードウェアを単純化することができる。

【0036】

【実施例】以下、図面を参照して本発明の実施例について説明する。

【0037】本発明に係る操作権管理装置は、オペレーティング・システム（OS）と呼ばれる計算機システム全体の実行制御を行うものの構成要素の一部である。OSは計算機資源であるメモリ、プロセッサ、周辺機器を管理し、また、ユーザプログラムとよばれるユーザが指示した実務上の処理を行うプログラムの実行を制御する。一つの装置上には複数のユーザプログラムがのっており、OSがそれらのプログラム間の制御も行う。

【0038】ユーザプログラムは、実行上に必要となる物理メモリなどの資源を確保したり、周辺機器にデータを入出力したり、他のユーザプログラムにデータを転送

7

したり、処理を依頼したり、実行を制御するためにＯＳに対して指示を行うことが必要となる。このＯＳに対する指示をシステムコールを呼ばれる特別な手段で行う。本実施例では、システムコールのうち、操作権を考慮するユーザプログラムの制御に関して説明する。

【００３９】操作権については、スレッドの管理とメモリ領域の管理がある。これらメモリ領域の管理とスレッドの管理については同一の部分が多いので、同様の機構を重複して説明することは避け、主にスレッドに対する操作権管理を例として、同一の部分には同一の用語を用いて説明する。

【００４０】なお、本実施例では、メモリの「操作」とは、アクセス（読み書き実行）のことを表すものとする。

【００４１】最初に、本実施例の特徴を概略的に述べる。本実施例では、個々のメモリ領域またはスレッドに対して、どのような条件で操作を許可するかという情報を登録したリストを設け、このリストにより許可されている場合に限り、操作を許可するとともに、この情報自体を変更することのできる権限を、前記リストと同一の形式、類似の形式あるいは共通の形式で管理する。また、特定の条件の時にのみこの変更を許す機構を付加する。また、この変更をする権限の変更についても同一の形式、類似の形式あるいは共通の形式で管理する。

【００４２】この結果、本実施例は、不当な操作命令やプログラムの誤りからメモリ領域やスレッドを保護し、またこのような誤った操作命令を検出することによってプログラムの誤りの検出を容易とするために、メモリ領域またはスレッドごとに設けた操作の可否を示す情報を変更する権利自体もそのメモリ領域またはスレッドが持つので、操作権の管理のみならず、操作権の管理の権限すらも同一の形式、類似の形式あるいは共通の形式で管理し、削除し、追加し、または変更することを柔軟に行うことができるようになる、という作用効果を奏するものである。

【００４３】また、この操作権の管理の権限は、メモリ領域やスレッドの所有者とは全く別の機構より、通常のアクセス制御リストと同様の形式で複数持つので、操作権管理の権限を複数のスレッドで共有したり、自分の作ったメモリ領域やスレッドの管理権を他のスレッドに譲渡したりするなどの柔軟な操作権管理が可能となる。

【００４４】（第１の実施例）以下、本発明の第１の実施例について説明する。

【００４５】図１は、本実施例の全体構成図である。プログラム記憶部１には、計算機で処理する命令が記述されたユーザプログラムや処理する対象のデータ、計算機で処理途中の結果を一時的に記憶するスタックなどのデータがメモリ空間上に蓄えられている。メモリ空間は、複数の領域（メモリ領域）に区切られて管理され、それぞれの領域には、お互いを区別するための識別子がついており、メモリ領域ごとにプログラムやデータやスタックを配置している。

8

【００４６】さらに、プログラム記憶部１に蓄えられているプログラムを解釈実行し、データを処理する実行主体であるプロセッサ２と、プロセッサ内の状態を管理するスレッド管理部３、プロセッサの実行状態の切り替え処理を行う実行状態切り替え部４がある。

【００４７】プロセッサ２内には、命令の実行途中に一時的に利用するレジスタ類、現在実行している命令のプログラム記憶部での位置を示すためのプログラムカウンタ（ＰＣ）、スタックデータの先頭位置を示すスタックポインタ（ＳＰ）などが存在し、これら一連のデータを入れ替えることで、複数の別々のユーザプログラムの処理を時分割で並行して進めることが可能となる。

【００４８】スレッドとは、上記プロセッサ内で使用する一連のデータのことを呼び、これを複数組用意し、このスレッドを切り替えてプロセッサにセットし実際に実行させることにより、時分割処理が可能となる。

【００４９】メモリ領域ＩＤ記憶部５は、現在実行中のプログラムがプログラム記憶部１のどのメモリ領域に存在しているかを、そのメモリ領域のＩＤとして記憶しているものである。これは、プロセッサ２内のＰＣの値を利用し、現在実行中のメモリ領域ＩＤを求めて、記憶している。このメモリ領域ＩＤは、現在実行中のプログラムＩＤとみなすこともできる。

【００５０】スレッド管理部３の中にはスレッド管理テーブルと呼ぶ表があり、プロセッサ２にセットするレジスタの情報やそのほかの情報がスレッドごとに管理されている。スレッドには、お互いを区別するために識別子（スレッドＩＤ）がついている。

【００５１】実行状態切替部４は、現在プロセッサ２上で実行しているスレッド（例えばスレッドＩＤ＝１）を中断し、スレッド（スレッドＩＤ＝１）の状態をスレッド管理テーブルの所定の位置（スレッドＩＤ＝１の格納領域）に退避し、別のスレッド（例えばスレッドＩＤ＝２）を起動すべく、スレッド管理テーブルのスレッド（スレッドＩＤ＝２）の情報をプロセッサ２に再セットし、スレッド（スレッドＩＤ＝２）の実行を再開させる。また、実行状態切替部４には、さらに、どのスレッドをセットし退避するかを決定するスレッドスケジューリング機構（図示せず）と、いつスケジューリングするか指示するタイマ部（図示せず）が存在する。

【００５２】スレッドＩＤ記憶部６は、スレッドＩＤを記憶するためのものである。

【００５３】操作権リスト記憶部７は、スレッド保護に関するテーブルを保持し、スレッドごとにスレッドに対する操作を実行する権限がどうなっているかを記憶するもので、操作権限判定部１２で使用される。本実施例はスレッドの操作権管理を例として説明しているが、メモリ領域の操作権管理の場合には、この操作権リスト記憶

9

部7はメモリ領域の保護に関するテーブルを保持し、スレッドごとにメモリ領域に対する操作を実行する権限がどうなっているかを記憶するものである。以下同様にスレッドに関する機構とメモリ領域に関する機構の同一の部分については説明を省略し、スレッドに関するものを代表して記述する。

【００５４】さて、システムコールとしてユーザプログラムを実行しているあるスレッドが他のスレッドに対して制御しようとする例として、Thread_Get _Statusシステムコールを発行する場合について説明する。

【００５５】このシステムコールは、スレッドＩＤで指示されたスレッドの状態、つまりそのスレッドのレジスタ類の情報を、このシステムコールの呼び出し元のユーザ・プログラムに通知するものである。いま、スレッド１（スレッドＩＤ＝１）がプログラムＡ（メモリ領域ＩＤ＝Ａ）を実行中、そこでスレッド２（スレッドＩＤ＝２）の情報をこのシステムコールで得ようとしたとする。Ｃ言語で記載されているプログラムＡでは、次のように記されている。

err = Thread_Get _Status(2, &Thread _Status);ここで、関数Thread_Get _Statusがスレッドの情報を得るシステムコールであって、第１引数の２は、スレッド２（スレッドＩＤ＝２）の情報を得ることを指示しており、第２引数の&Thread _Statusはシステムコールの結果得られたスレッド２の情報を格納するメモリ領域の先頭位置を指示している。また、このシステムコールの実行が成功したかどうかは、この関数の返り値として変数 err に入る。

【００５６】この関数をプログラムＡのスレッド１が実行すると、スレッド１はシステムコールにより実行モードがユーザレベルから特権レベルに遷移し、ＯＳを呼び出す。特権レベルとは、ＯＳが処理を行う専用レベルのことである。

【００５７】呼び出されたＯＳでは、ＯＳ内のシステムコール受付部８に処理が移る。ここでは、このシステムコールを呼びだした元のユーザプログラムの中断処理を行い、要求されたシステムコールの種類と指示された引き数を受取り、それぞれその情報を引数検出部９とシステムコールの種類判別部１０に送る。

【００５８】本例では、指示されたシステムコールは、Thread_Get _Statusなので、システムコールの種類判定部１０で、次には、Thread_Get _Status処理部１１を呼び出す。

【００５９】そこでの処理は、図２に示した操作権リスト記憶部７を使って行われる。この例では操作権リスト記憶部７はスレッドに対する操作権リストを記憶するものである。この記憶部には、図２に示したようなテーブルを保持している。テーブルには、操作スレッドＩＤ、対象スレッドＩＤ，操作スレッドが存在するメモリ領域ＩＤ、スレッド操作権限、スレッド操作権変更権限の各

10

エリアをもっている。

【００６０】同様のエリアとしては、対象スレッドが存在するメモリ領域ＩＤを指定することも考えられるが、本例ではかかるエリアを持たないものを例として説明する。

【００６１】メモリ領域に対する操作権リスト記憶部も同様であるが、テーブルの内容は若干異なる。これについての詳細は第２の実施例で説明する。

【００６２】操作スレッドＩＤとはスレッド操作のシステムコールを実行しようとしているスレッドのスレッドＩＤのことで、対象スレッドＩＤとはこのシステムコールの引数で指定された操作対象のスレッドＩＤを示す。また、操作スレッドが存在するメモリ領域ＩＤとは、この実行されようとしているシステムコールを含むプログラムの存在するメモリ領域ＩＤのことであり、スレッド操作権限とは、どういうスレッド操作が可能かを示すもので、スレッドの内部状態を読み出す権限、スレッドの内部状態を変更する権限、スレッドの実行を停止／再開する権限の３つの権限を独立に指定することができる。スレッド操作権変更権限は、スレッド操作権限の記入されたフィールドの情報を変更する権限のことであり、この権限が指定されている場合、その列に書かれているスレッド操作権限のフィールドの情報が変更できるだけでなく、その列に書かれている対象スレッドＩＤに関する全てのスレッド操作権限フィールドの情報を変更することができる。

【００６３】たとえば図の例では、スレッド１（スレッドＩＤ＝１）が発したシステムコールでは、スレッド２（スレッドＩＤ＝２）の内部状態の参照と変更は可能だが、スレッド２の実行の停止／再開およびスレッド２の操作権限の変更はできないことを示している。また、スレッド１は、スレッド３（スレッドＩＤ＝３）に対しては実行の停止／再開はできるがそれ以外の操作はできない。さらに、スレッド１は、スレッド２とスレッド３以外のスレッドに対しては、何の操作もできないことを示している。また、スレッド１００（スレッドＩＤ＝１００）はスレッド２の内部状態の読み出しおよび操作権限の変更ができることを示している。

【００６４】スレッド１が対象スレッドの場合の例で、テーブル中の操作スレッドが存在するメモリ領域ＩＤの部分にａ１１という情報が保持されているが、これは、このシステムコールがどのユーザプログラムから行われても同等で、システムコールを実行するスレッドＩＤのみに実行権限が依存することを示している。

【００６５】メモリ領域ＩＤの欄がａ１１でないときは、そのＩＤのメモリ領域に存在するプログラムからシステムコールが実行された時の操作権限を規定する。たとえば、スレッド３が、プログラムＡ（メモリ領域ＩＤ＝Ａ）の実行中、スレッド１に対しては、すべての操作が可能であるが、それ以外のときは、操作不可である。

つまり、スレッド3がプログラムB（メモリ領域ＩＤ＝B）の実行中、スレッド1に対して同様なシステムコールを実行しても操作できないし、スレッド5（スレッドＩＤ＝5）がプログラムAにおいてスレッド1に対して同じシステムコールを実行しても操作できない。すなわち、このテーブルに記載のないスレッド操作は一切できない。

【００６６】図2の例では、スレッド操作権限変更権限のフィールドは1つだけであるが、前記3つの権限ごとに操作権の変更権限を別々に指定できるようにしても良い。

【００６７】また、スレッド操作権限には、本前の例で挙げた3つの権限の他にもあってよい。例えば、操作をするユーザのユーザＩＤ、該当フィールドを時間制限するための有効期間開始時刻、有効時間終了時刻などが考えられる。その場合にも、それら権限の変更権を設定することができる。

【００６８】また、本実施例では、スレッド操作権限とスレッド操作権の変更権限とを、同一のテーブルの中で操作スレッドＩＤ、対象スレッドＩＤ、操作スレッドが存在するメモリ領域ＩＤを共有して表現しているが、スレッド操作権限と、スレッド操作権の変更権限とを別々のテーブルに記述し、それぞれについて操作スレッドＩＤ、対象スレッドＩＤ、操作スレッドが存在するメモリ領域ＩＤを指定しても良い。

【００６９】次に、Thread_Get _Status処理部１１の処理の流れを図3に示す。

【００７０】まず、ステップＳ１にて、このシステムコールを実行した操作スレッドＩＤ、メモリ領域ＩＤをそれぞれスレッドＩＤ記憶部6とメモリ領域ＩＤ記憶部5から情報を得る。上述したように、現在実行中のスレッドＩＤは、実行状態切替部4でセットされ、現在実行中のメモリ領域ＩＤは、プロセッサの内部にあるＰＣの情報からセットされている。さらに、システムコールの第一引数から対象スレッドＩＤ検出部１３が対象スレッドＩＤを得ている。第二引数からは、スレッド情報格納位置検出部１４が、このシステムコールの実行結果を格納するユーザ・プログラムのメモリ位置を検出する。

【００７１】次にステップＳ２にて、以上の3つの情報をもとに操作権記憶部７１の表を引き、該当エントリをサーチし、スレッド操作権限を調べる。もし、見つかれば、ステップＳ３に進み、なければステップＳ８を行うエラー処理部１５へ進む。

【００７２】サーチは、テーブルの上から順に比較を行い、最初にマッチしたエントリの情報を次のステップ以降で利用する。テーブルのフィールドのａｌｌ部分は、すべてにマッチすることを意味する。もし、テーブルの最後のエントリまで比較し、マッチしなかったらサーチの失敗となる。

【００７３】ステップＳ３では、見つかったエントリのスレッド操作権限を表す部分のうち、スレッド内部状態読出権限に該当する部分の情報を読み出し、許可か否かを判断する。この判定は操作権限判定部１２で行う。もし判定で許可されればステップＳ４へ進み、不許可になればステップＳ８のエラー処理へ移る。

【００７４】次のステップＳ４では、対象スレッドに対して、許可されたスレッドの内部状態を読み出すためのスレッド管理テーブルの参照操作を実行する。この処理はスレッド状態参照実行部１６にて行う。ここでは、該当スレッドの内部状態であるスレッド情報を、スレッド情報格納位置検出部１４で指示されたメモリ中に格納する。

【００７５】ステップＳ５にて、ステップＳ４での処理結果を判定する。処理が成功するとステップＳ６にて、成功を示すコードをシステムコール終了部１７へ伝える。失敗するとステップＳ７にて、失敗を示すコードをシステムコール終了部１７へ伝える。

【００７６】ステップＳ８では、エラーコードをシステムコール終了部１７へ伝える。

【００７７】Thread_Get _Status処理部１１での処理を終えると、最後に、システムコール終了部１７に処理が進む。このシステムコールを実行したユーザプログラムに返す返値をシステムコールの返値として戻るように処理を行い、中断していた呼出元のユーザプログラムの実行ができるように復帰処理を行い、実行レベルを特権レベルからユーザレベルへ戻す処理を行う。

【００７８】次に、操作権記憶部７１に記入されたスレッド操作権限を変更するシステムシステムコールについて説明する。

【００７９】Thread_Add _Acl(指定操作スレッドＩＤ、指定対象スレッドＩＤ、指定メモリ領域ＩＤ、追加する指定操作権限）なるシステムコールは、上記で説明したスレッド操作権限を、新たに追加するものである。

【００８０】いま、スレッド１００（スレッドＩＤ＝１００）がスレッド1（スレッドＩＤ＝1）に対し、スレッド2（スレッドＩＤ＝2）の実行の停止／再開権限を付与しようとしたとする。Ｃ言語で記載されているプログラムＣでは、次のように記されている。

【００８１】err = Thread_Add _Acl(1, 2, all, THREAD _SUSPEND);ここで、関数Thread_Add _Acl は、スレッドに対する操作権を追加するためのシステムコールであって、第1引数の1は、指定した操作スレッド番号（対象スレッドＩＤ）が1であること、すなわちスレッドＩＤ＝1に対して権限を与えることを指示している。第2引数の2は、指定した対象スレッド番号（指定対象スレッドＩＤ）が2であること、すなわちスレッドＩＤ＝2を操作する権限であることを指示している。第3引数のａｌｌは、指定した操作スレッドが存在するメモリ領域ＩＤ（指定メモリ領域ＩＤ）がａｌｌであること、すなわち操作スレッドが存在するメモリ領域ＩＤに

よって与える権限を制限することはしないということを指示している。第４引数のTHREAD_SUSPEND は、これら３つの指定した組み合わせに対して、付加する操作権限（指定操作権限）を指示している。この例では、スレッドの実行の停止／再開権限を付加することを意味している。第４引数には複数種類の権限を同時に指定することもできる。

【００８２】この関数をプログラムＣのスレッド１００が実行すると、スレッド１００はシステムコールによりプロセッサの実行モードをユーザレベルから特権レベルに遷移させ、ＯＳを呼び出す。特権レベルとは、ＯＳが処理を行う専用レベルのことである。

【００８３】呼び出されたＯＳでは、ＯＳ内のシステムコール受付部８に処理が移る。ここでは、このシステムコールを呼び出した元のユーザプログラムの中断処理を行い、要求されたシステムコールの種類と指示された引数を受け取り、それぞれその情報を引数検出部９とシステムコールの種類判別部１０に送る。

【００８４】本例では、指示されたシステムコールは、Thread_Add _Acl なので、システムコールの種類判別部１０で、Thread_Add _Acl 処理部２１を呼び出す。Thread_Add _Acl 処理部２１は図１におけるThread_Get _Status処理部１１を図４のように置き換えたものである。なお、Thread_Add _Acl 処理部２１とThread_Get _Status処理部１１は本来一体のブロック図として示すべきものであるが、図が煩雑になるため説明の便宜をはかり図１および図４の別々の図としてある。

【００８５】そこでの処理は、図２に示したスレッド操作権リスト記憶部７を使って行われる。その内容および意味は前に説明した通りである。

【００８６】次に、図５にThread_Add _Acl 処理部２１の処理の流れを示す。

【００８７】まず、ステップＳ１１にて、このシステムコールを実行した実行スレッドＩＤ、メモリ領域ＩＤをそれぞれスレッドＩＤ記憶部６とメモリ領域ＩＤ記憶部５から情報を得る。さらに、システムコールの第２引数から指定対象スレッドＩＤ検出部２３が、このシステムコールによってスレッド操作権限の変更を行う対象としたスレッドのＩＤを得る。

【００８８】次にステップＳ１２にて、以上の３つの情報をもとにスレッド操作権リスト記憶部７のテーブルを引き、該当エントリをサーチする。もし、見つかれば、ステップＳ１３に進み、なければステップＳ１９を行うエラー処理部１５へ進む。サーチは、テーブルの上から順に比較を行い、最初にマッチしたエントリの情報を次のステップ以降で利用する。テーブルのフィールドのａｌｌ部分は、全てにマッチすることを意味する。もし、テーブルの最後のエントリまで比較し、マッチしなかったらサーチの失敗となる。

【００８９】ステップＳ１３では、見つかったエントリのスレッド操作権変更権限に該当する部分の情報を読み出し、許可か否かを判断する。この判定は操作権変更権判定部２６で行う。もし判定で許可されればステップＳ１４へ進み、不許可になればステップＳ１９のエラー処理へ移る。

【００９０】ステップＳ１４では、システムコールの第１引数から、指定対象スレッドＩＤ検出部２３が指定した対象スレッドＩＤを得る。また、システムコールの第３引数から、指定メモリ領域ＩＤ検出部２４が指定した操作スレッドが存在するメモリ領域ＩＤを得る。さらに、システム・コールの第４引数から、指定操作権限検出部２５が指定した追加すべき操作権限を得る。

【００９１】ステップＳ１５では、ステップＳ１１およびステップＳ１４で得られた対象スレッドＩＤ、指定対象スレッドＩＤ、指定メモリ領域ＩＤ、指定操作権限をもとに操作権リスト変更部２７が操作権リスト記憶部７を書き換える。つまり、まず操作権リスト記憶部７のテーブルを順にサーチし、対象スレッドＩＤと指定対象スレッドＩＤと指定メモリ領域ＩＤの３つが全て一致する列があれば、その列のスレッド操作権限のフィールドを書き換え、もしなければ、指定した３つのＩＤを示す列を新たに操作権リスト記憶部７のテーブルに追加する。この例では、指定した３つのＩＤと一致する列が最初に見付かるので、その列のスレッド操作権限のフィールドにあるスレッドの実行の停止／再開権限を×から○に変更する。

【００９２】ステップＳ１６にて、ステップＳ１５での処理結果を判定する。処理が成功するとステップＳ１７にて、成功を示すコードをシステムコール終了部１７へ伝える。失敗するとステップＳ１８にて、失敗を示すコードをシステムコール終了部１７へ伝える。ステップＳ１９では、エラーコードをシステムコール終了部１７へ伝える。

【００９３】Thread_Add _Acl 実行部２１での処理を終えると、最後にシステムコール終了部１７に処理が進む。このシステムコールを実行したユーザプログラムに返す返値をシステムコールの返値として戻るように処理を行い、中断していた呼出元のユーザプログラムの実行ができるように復帰処理を行い、実行レベルを特権レベルからユーザレベルへ戻す処理を行う。

【００９４】本例のようなThread_Add _Acl システム・コールの実行が成功した結果、図２で示された操作権リスト記憶部７のデータは図６のように変更される。

【００９５】本例では操作権リスト変更部２７において、書換を必要とする該当フィールドを書き換えるだけで、Thread_Add _Acl システム・コールからの要求を実行することができたが、Thread_Add _Acl システム・コールに与えられた引数の内容によっては、該当フィールドを書き換えるだけでは表現できず、操作権リスト記憶部７のデータを増やさなくてはならない。

【００９６】例えば、操作権リスト記憶部７のデータが先の図２に示したとおりであるとき、 Thread_Add _Acl (3, 2, A, THREAD_SUSPEND*THREAD_GET _STATUS) を実行した場合を考える。

【００９７】ここで、第１引数の３は、指定した操作スレッド番号（対象スレッドＩＤ）が３であること、すなわちスレッドＩＤ＝３に対して権限を与えることを指示しており、第２引数の２は、指定した対象スレッド番号（指定対象スレッドＩＤ）が２であること、すなわちスレッドＩＤ＝２を操作する権限であることを指示しており、第３引数のＡは、指定メモリ領域ＩＤがＡであること、すなわち、このシステムコールの実行により変更される権限は、プログラムＡ（メモリ領域ＩＤ＝Ａ）を実行中にのみ有効であることを指示しており、第４引数のTHREAD_SUSPEND は、これら３つの指定した組み合わせに対して、スレッドの実行の停止／再開権限を付加すること、そしてTHREAD_GET _STATUSはスレッドの内部状態読出権限を付加することを指示している。

【００９８】このシステム・コールが発せられると、操作権リスト変更部２７が、操作権リスト記憶部７のテーブルを順にサーチし、対象スレッドＩＤと指定対象スレッドＩＤと指定メモリ領域ＩＤの３つが全て一致する列をさがすが、この場合そのような列は存在しないので、このシステム・コールは、操作権リスト記憶部７の持つテーブルの一部のフィールドを書き換えるだけでは実行できない。指定した３つのＩＤを示す列を新たに操作権リスト記憶部７のテーブルに追加する。このような手順を踏むことにより、このシステム・コールの実行が終了すると、図２で示された操作権リスト記憶部７のデータは図７のように変更され、操作権リスト記憶部７の列の長さが１つ長くなっている。

【００９９】以上の説明においては、最初に操作権変更権限をどこに指定するかということについては記述されていない。一般にはＯＳの起動時に、一部の特権的なスレッドに対してこの権限を与え、以下階層的に権限を増やしてゆくことにより実現することが可能である。

【０１００】しかし、スレッドの作成時に必ず操作権変更権を指定しなければならないのは、プログラミングの手間を増やすおそれがあるので、メモリ領域またはスレッドの作成時には、あらかじめ定められたデフォルトの操作権変更権が設定されるように構成しても良い。デフォルトを何にするかについては各種考えられるが、最も簡単には、そのメモリ領域またはスレッドを起動したスレッドに対して操作権変更権を与えることが考えられる。この方法は、デフォルトとして従来ＯＳにおけるオーナーの概念を用いたことになり、さらに本実施例で説明した方法により、本発明の操作権管理装置はより柔軟な操作権管理を実現するものである。

【０１０１】また、先の例ではThread_Add _Acl の第４引数に指定することのできる指定操作権限に制約はないものとして記述したので、第４引数にTHREAD_CHANGE _ACL を指定することができる。すなわち、スレッド操作権変更権限を変更させることもできる。また、以上説明したThread_Add _Acl と同様に、Thread_DelAcl なるシステム・コールを用いて、例えば

Thread_Del _Acl(3,1,A,THREAD_CHANGE _ACL)

は、スレッドが、メモリ領域ＩＤ＝Ａのメモリ領域から、スレッドＩＤ＝１のスレッドのスレッド操作権限を取り除くことを意味している。この場合、永遠に誰にも操作権変更権限を変更することのできないようなスレッドを作ってしまうおそれがあるが、そういった変更が行われることが問題となるオペレーティングシステムにおいては、操作権変更権記憶部の情報を変更する際に、操作権記憶部の情報を変更することが不可能になるような変更を禁止するようにしても良い。例えば、先にあげた

Thread_Del _Acl(3,1,A,THREAD_CHANGE _ACL)

を実行すると、もはやスレッドＩＤ＝１のスレッドに対し、スレッド操作権限を変更することのできるスレッドが１つもなくなる。このようなシステム・コールが実行されると、操作権リスト変更部２７はエラーを検出したことになり、失敗を示すコードをシステムコール終了部１７へ伝え、操作権リスト記憶部のデータは変更されない。

【０１０２】また、操作権変更権を変更する権利を別個のものとして扱い、操作権リスト記憶部７に操作権変更権管理権限のフィールド（操作権変更権管理権限記憶部）を用意し、より確実な操作権管理を行っても良い。

【０１０３】さらには、操作権変更権記憶部の情報を変更することが不可能になるような、操作権変更権管理権限記憶部の情報の変更を禁止するようにしても良い。

【０１０４】なお、上記「スレッドＩＤ」を「ユーザＩＤ」に置き換えて構成することも可能である。この場合、図１におけるスレッドＩＤ記憶部６を、ユーザＩＤ記憶部と置き換える修正を施せば良い。

【０１０５】また、前述した「操作スレッドが存在するメモリ領域ＩＤ」を「操作スレッドが実行しているプログラムＩＤ」に置き換えて構成することも可能である。この場合、図１におけるメモリ領域ＩＤ記憶部５は、プログラムＩＤ記憶部と置き換える修正を施せば良い。

【０１０６】（第２の実施例）次に、本発明の第２の実施例について説明する。

【０１０７】本発明の操作権管理装置は、メモリ領域やスレッドに対する操作権管理を、例えばオーナーといった固定的な概念により行うのではなく、操作権変更権を操作権リスト自体に持たせ、自由に削除、追加させることによって柔軟に行うことを目的としている。ここでは、この機構を生かした例として、複数人による共同作業のアプリケーションが動作する場合に、操作権管理が柔軟に行われることを示す実施例について述べる。

【０１０８】ユーザーＡは、計算機を使った電子会議を

開催すべく、新たにスレッド１０（スレッドＩＤ＝１０）を作成した。より具体的には、ユーザーＡの動かしていたログイン・シェルであるスレッド１１がスレッド１０を作成した。このとき、操作権リスト記憶部７の該当部分のデータは図８のようになっている。以下図には操作権リスト記憶部７のテーブルのうち、本実施例を説明するのに必要な部分のみを記述し、他の部分については記述を省略する。

【０１０９】スレッド１１（スレッドＩＤ＝１１）はスレッド１０（スレッドＩＤ＝１０）の内部状態読み出しおよびスレッド操作権の変更ができる。しかし、内部状態を変更したり、実行の停止／再開を行うことはできない。つまり、既に動き始めたスレッド１０（スレッドＩＤ＝１０）は、それを作成したスレッド１１（スレッドＩＤ＝１１）によって誤って実行を妨害されることはない。これらの初期状態はスレッドを作成するシステムコールの引数として指定するものとする。

【０１１０】作成されたスレッド１０は電子会議のプログラムを実行し、ユーザーＡとのインターフェースを司る。また、スレッド１０は電子会議をサポートするためのメモリ領域として、メモリ領域３０（メモリ領域ＩＤ＝３０）を作成する。

【０１１１】メモリ領域３０に関する操作権の管理には、スレッドの操作権の管理と同様に操作権リスト記憶部７を用いる。ただし、スレッドの操作権管理の説明で述べた対象スレッドＩＤ、スレッド操作権限、スレッドの内部状態読出権限、スレッドの内部状態変更権限、スレッドの実行の停止／再開権限、スレッド操作権変更権限は、メモリ領域の操作権管理においてはそれぞれ対象メモリ領域ＩＤ、メモリ領域操作権限、メモリ領域の内部データ読出権限、メモリ領域の内部データ変更権限、メモリ領域にあるプログラムの実行権限、メモリ領域操作権変更権限、となる。

【０１１２】図９に示した操作権リスト記憶部は、スレッド１０がメモリ領域３０の内部データの読み出しおよび変更、操作権変更ができることを意味している。

【０１１３】次にユーザーＢおよびユーザーＣが電子会議への参加を表明し、スレッド１０は新たにユーザーＢおよびユーザーＣのためのスレッド１２（スレッドＩＤ＝１２）およびスレッド１３（スレッドＩＤ＝１３）を作成した。その結果、図８の操作権リスト記憶部７のデータは図１０のように追加される。また電子会議をサポートするためのメモリ領域であるメモリ領域３０は、スレッド１２およびスレッド１３からも使われるので、図９の操作権リスト記憶部７のデータは図１１のように追加される。

【０１１４】つまり、スレッド１２およびスレッド１３はスレッド１０によりコントロールされており、スレッドに関するあらゆる操作ができるが、スレッド１２およびスレッド１３は他のスレッドのコントロールをすることはできない。これにより、スレッド１０が先導して電子会議の進行をプログラムされた通りに実行することができる。

【０１１５】共通のメモリ領域として使われるメモリ領域３０は、スレッド１０、スレッド１２、スレッド１３のいずれからもデータの読み出し、変更ができるが、その領域が誰から（どのスレッドから）どのような操作ができるかということを変更することができるのはスレッド１０だけである。

【０１１６】このような仕組みで、複雑な依存関係のある電子会議のようなコミュニケーション・プログラムもＯＳの操作権管理機構を用いて確実に記述することができる。

【０１１７】次に、会議を先導していたユーザーＡのスレッド１０が、途中で会議を抜け、会議はそのまま進行している場合を考える。

【０１１８】単純にスレッド１０が自分自身の実行を終了させたのでは、メモリ領域３０の操作権を変更することのできるスレッドがいなくなり、会議を正しく進行させることができなくなる。このような場合には、ユーザーＡは一旦会議の進行権を他の人（例えばユーザーＢ）に譲らなければならない。

【０１１９】まず、メモリ領域３０の操作権をスレッド１２（ユーザーＢのスレッド、スレッドＩＤ＝１２）に設定する。そして、スレッド１３の管理権をスレッド１２に譲り、そのあとで実行を終了する。それ以外のもので、操作スレッドＩＤが１０であるものは、スレッド１０の終了と同時に自動的に消滅させる。

【０１２０】これによって、図１０および図１１の操作権リスト記憶部はそれぞれ図１２および図１３のようになる。すなわち、メモリ領域３０へのアクセスはスレッド１２とスレッド１３の両方からできるが、スレッドの管理自体の権利はスレッド１２が継承する。注意すべき点は、スレッド１０の実行が終了したために、ユーザーＡのlogin shell であるスレッド１１からは、電子会議を実行するいずれのプログラムに対しても何の権利もなくなっていることである。

【０１２１】ここに示した例は電子会議プログラムを実現する一方法を示したものであり、必ずしもこのような手順で操作権を譲渡してゆかなければプログラムが記述できないということは意味していない。

【０１２２】このようにして、スレッドおよびメモリ領域への操作権を管理しつつ、管理者の変更を実現することができる。

【０１２３】本発明の操作権管理装置を使えば、「会議に参加する」あるいは「会議から退室する」という、本実施例で具体的に例をあげて説明したような場合における、曖昧な操作権管理も、アプリケーションで逐一記述することなく、ＯＳのサポートにより柔軟かつ確実に実現することができる。

19

【０１２４】また、本発明は上述した実施例に限定されるものではなく、その要旨を逸脱しない範囲で、種々変形して実施することができる。

【０１２５】

【発明の効果】本発明の操作権管理装置によれば、スレッドまたはメモリ領域に対する操作を保護するための操作権管理を柔軟かつ確実に行うことができる。

【図面の簡単な説明】

【図１】Thread_Get _Statusを説明するための構成図

【図２】図１の操作権リスト記憶部７の内部構成例を示す図

【図３】本実施例に係わる操作権管理装置において操作権を変更する動作を示すフローチャート

【図４】Thread_Add _Acl を説明するための構成図

【図５】本実施例に係わる操作権管理装置において操作権変更権を変更する動作を示すフロー図

【図６】図４の操作権リスト記憶部７の内部構成例を示す第１の図

【図７】図４の操作権リスト記憶部７の内部構成例を示す第２の図

【図８】電子会議開催時のスレッド操作権リストを示す図

【図９】電子会議開催直後のメモリ領域操作権リストを

20

示す図

【図１０】電子会議に新たなメンバが参加した際のスレッド操作権リストを示す図

【図１１】電子会議に新たなメンバが参加した際のメモリ領域操作権リストを示す図

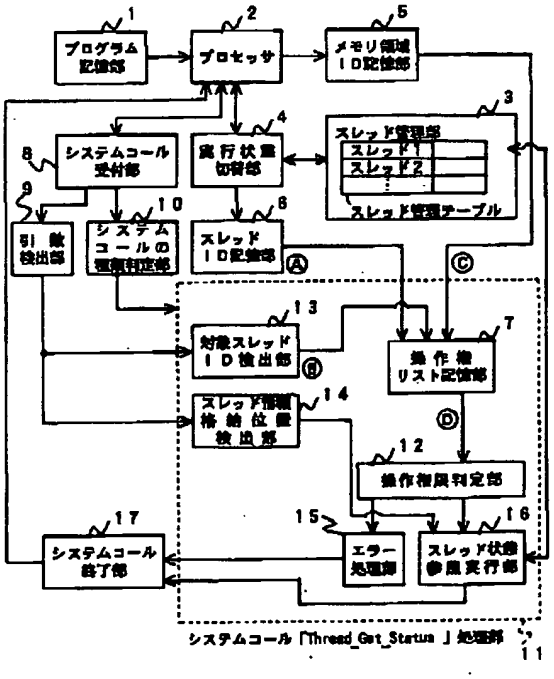
【図１２】ユーザーＡが電子会議の先導権をユーザーＢに委譲した際のスレッド操作権リストを示す図

【図１３】ユーザーＡが電子会議の先導権をユーザーＢに委譲した際のメモリ領域操作権リストを示す図

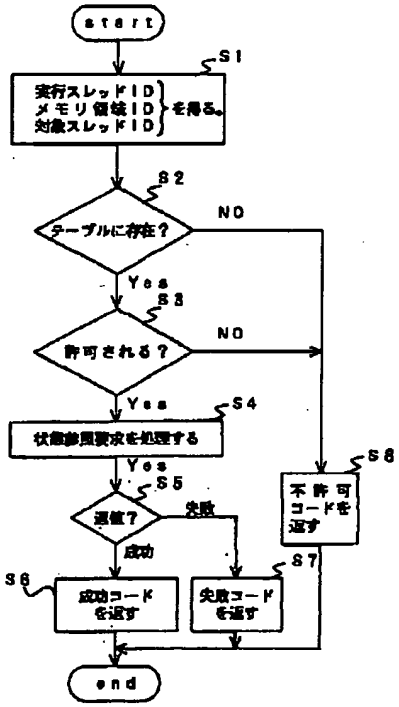
【符号の説明】

１…プログラム記憶部、２…プロセッサ、３…スレッド管理部、４…実行状態切替部、５…メモリ領域ＩＤ記憶部、６…スレッドＩＤ記憶部、７…操作権リスト記憶部、８…システムコール受付部、９…引数検出部、１０…システムコールの種類判定部、１１，２１…システムコール実行部、１２…操作権限判定部、１３…対象スレッドＩＤ検出部、１４…スレッド情報格納位置検出部、１５…エラー処理部、１６…スレッド状態参照実行部、１７…システムコール終了部、２２…指定操作スレッドＩＤ検出部、２３…指定対象スレッドＩＤ検出部、２４…指定メモリ領域ＩＤ検出部、２５…指定操作権限検出部、２６…操作権変更権判定部、２７…操作権リスト変更部

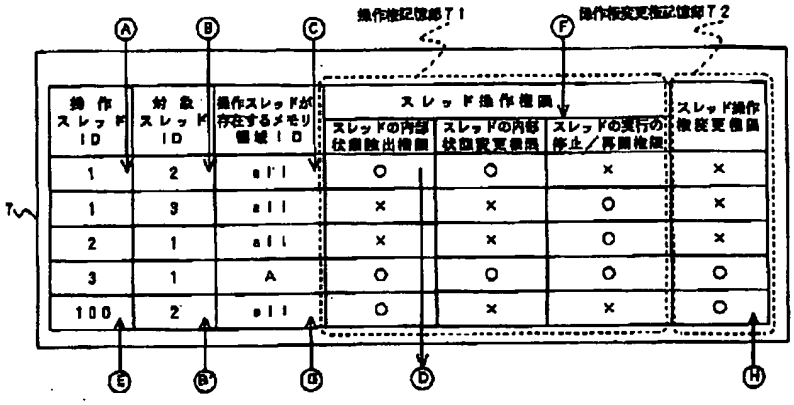
【図１】



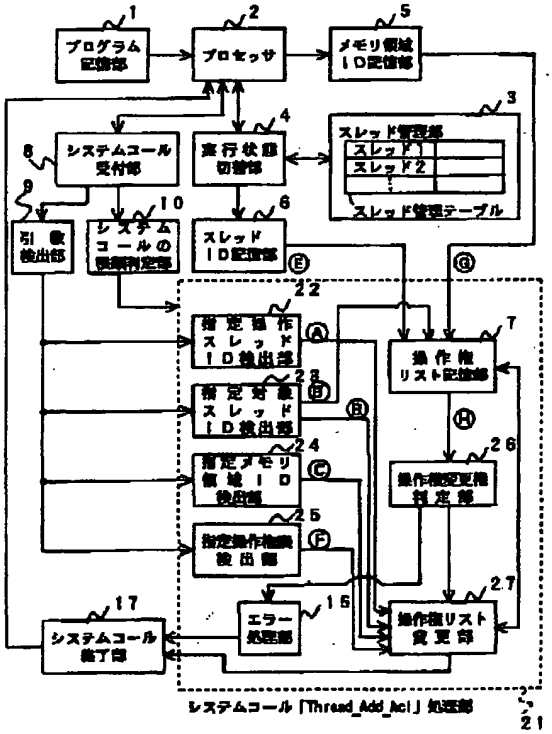
【図３】



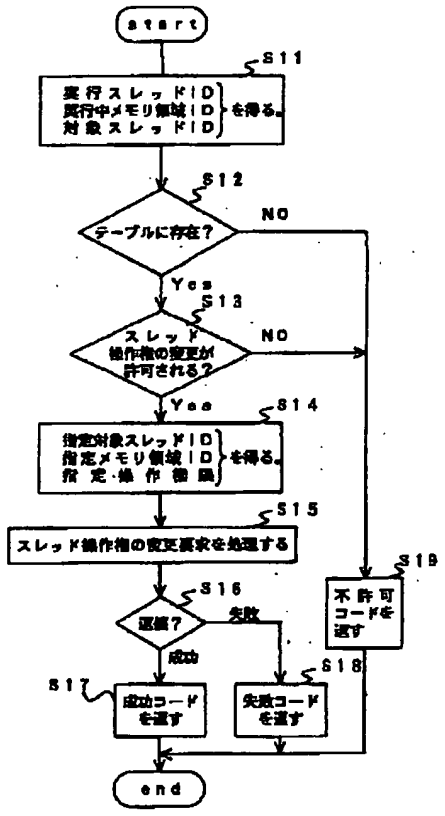
【図2】



【図4】



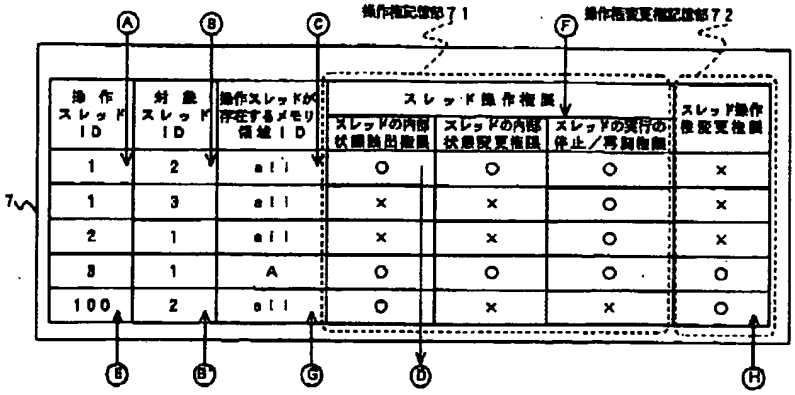
【図5】



【図9】

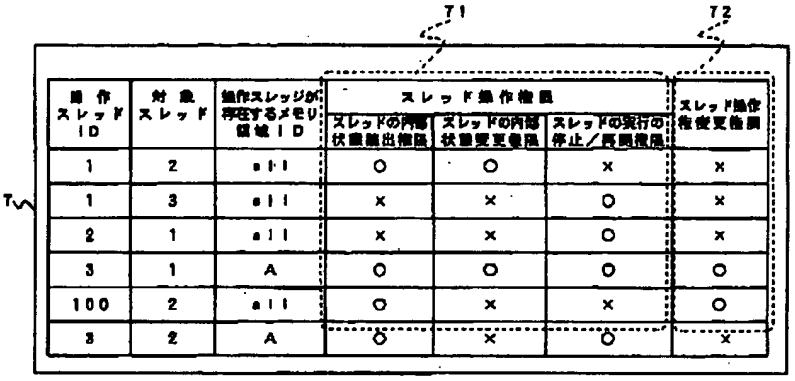
操作スレッドID	対象メモリ領域ID	操作スレッドが存在するメモリ領域ID	メモリ領域操作権限			メモリ領域操作権変更権限
			メモリ領域の内部データ読出権限	メモリ領域の内部データ変更権限	メモリ領域のプログラム実行権限	
10	30	all	○	○	×	○

【図6】



操作スレッドID	対象スレッドID	操作スレッドが存在するメモリ領域ID	スレッド操作権限			スレッド操作権変更権限
			スレッドの内部状態読出権限	スレッドの内部状態変更権限	スレッドの実行の停止／再開権限	
1	2	all	○	○	○	×
1	3	all	×	×	○	×
2	1	all	×	×	○	×
3	1	A	○	○	○	○
100	2	all	○	×	×	○

【図7】



操作スレッドID	対象スレッド	操作スレッドが存在するメモリ領域ID	スレッド操作権限			スレッド操作権変更権限
			スレッドの内部状態読出権限	スレッドの内部状態変更権限	スレッドの実行の停止／再開権限	
1	2	all	○	○	×	×
1	3	all	×	×	○	×
2	1	all	×	×	○	×
3	1	A	○	○	○	○
100	2	all	○	×	×	○
3	2	A	○	×	○	×

【図8】

操作スレッドID	対象スレッドID	操作スレッドが存在するメモリ領域ID	スレッド操作権限			スレッド操作権変更権限
			スレッドの内部状態読出権限	スレッドの内部状態変更権限	スレッドの実行の停止／再開権限	
11	10	all	○	×	×	○

【図10】

操作スレッドID	対象スレッドID	操作スレッドが存在するメモリ領域ID	スレッド操作権限			スレッド操作権変更権限
			スレッドの内部状態読出権限	スレッドの内部状態変更権限	スレッドの実行の停止／再開権限	
11	10	all	○	×	×	○
10	12	all	○	○	○	○
10	13	all	○	○	○	○

【図11】

操作スレッドID	対象メモリ領域ID	操作スレッドが存在するメモリ領域ID	メモリ領域操作権限			メモリ領域操作権変更権限
			メモリ領域の内部データ読出権限	メモリ領域の内部データ変更権限	メモリ領域のプログラム実行権限	
10	30	all	○	○	×	○
12	30	all	○	○	×	×
13	30	all	○	○	×	×

【図12】

操作スレッドID	対象スレッドID	操作スレッドが存在するメモリ領域ID	スレッド操作権限			スレッド操作権変更権限
			スレッドの内部状態読出権限	スレッドの内部状態変更権限	スレッドの実行の停止／再開権限	
12	13	all	○	○	○	○

【図13】

操作スレッドID	対象メモリ領域ID	操作スレッドが存在するメモリ領域ID	メモリ領域操作権限			メモリ領域操作権変更権限
			メモリ領域の内部データ読出権限	メモリ領域の内部データ変更権限	メモリ領域のプログラム実行権限	
12	30	all	○	○	×	○
13	30	all	○	○	×	×

フロントページの続き

(72)発明者　申　承昊
神奈川県川崎市幸区小向東芝町1番地　株式会社東芝研究開発センター内

(72)発明者　吉田　英樹
神奈川県川崎市幸区小向東芝町1番地　株式会社東芝研究開発センター内

Requested Patent: JP7244639A

Title: ACCESS RIGHT MANAGEMENT DEVICE ;

Abstracted Patent: JP7244639 ;

Publication Date: 1995-09-19 ;

Inventor(s): OURA MASAHIKO ;

Applicant(s): FUJITSU LTD ;

Application Number: JP19940032713 19940303 ;

Priority Number(s): ;

IPC Classification: G06F15/00 ;

Equivalents: ;

ABSTRACT:

PURPOSE:To flexibly change the right to access and facilitate its management as to access right management for managing the adequacy of a process request in an information system which offers plural information process services to plural users.

CONSTITUTION:The access right management device has a user qualification file 41 which holds records consisting of user IDs and qualification IDs as items, an access right files 42 consisting of service IDs, qualification IDS or user IDs, the kinds of access and whether or not the access is allowed, and the priority of records as items, a request acceptance means 1, an access right determination means 2, an access right holding means 3, and a service start means 5. The access right determination means 2 takes a qualification ID out of the user qualification file 41 based on the user ID as a key, retrieves the access right file 42 based on the qualification ID and the user ID as keys, and determines and holds whether or not final access is allowed with a high-priority record regarding the same service in the access right holding means 3.

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号

特開平7－244639

(43)公開日　平成７年(1995)９月19日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　15/00	３３０　Ａ	7459－5Ｌ		

審査請求　未請求　請求項の数３　ＯＬ　（全　７　頁）

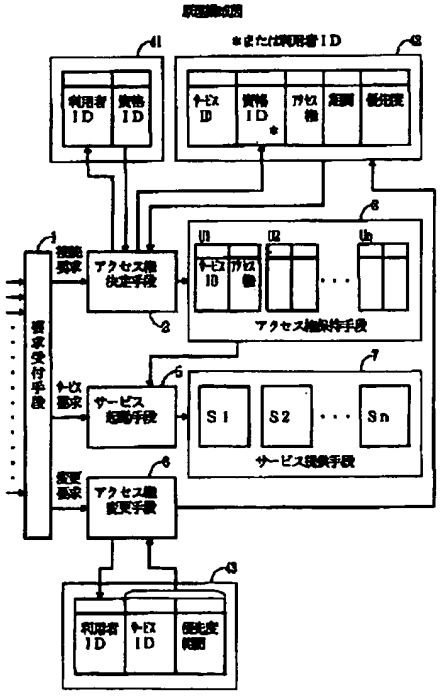
(21)出願番号　特願平6－32713

(22)出願日　平成６年(1994)３月３日

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中1015番地
(72)発明者　大浦　雅彦
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内
(74)代理人　弁理士　井桁　貞一

(54)【発明の名称】　アクセス権管理装置

(57)【要約】
【目的】多数の利用者を対象に複数の情報処理サービスを提供する情報システムでの処理要求の妥当性を管理するアクセス権管理に関し、アクセス権の変更を柔軟に、管理を容易にする。
【構成】利用者ＩＤと資格ＩＤとを項目とするレコードを保持する利用者資格ファイル41と、サービスＩＤと、資格ＩＤまたは利用者ＩＤと、アクセスの種類と可否と、レコードの優先度とを項目とするレコードを保持するアクセス権ファイル42と、要求受付手段１と、アクセス権決定手段２と、アクセス権保持手段３と、サービス起動手段５とを有する。アクセス権決定手段２は、利用者ＩＤをキーとして利用者資格ファイル41から資格ＩＤを取り出し、資格ＩＤと利用者ＩＤをキーとしてアクセス権ファイル42を検索して、同一のサービスについて優先度の高いレコードにより最終的なアクセスの可否を決定しアクセス権保持手段３に保持する。



1

【特許請求の範囲】

【請求項1】　複数の利用者に対し複数の処理サービスを提供する情報システムにおける利用者のサービスへのアクセス権を管理する装置であって、

利用者資格ファイル（41）と、アクセス権ファイル（42）と、要求受付手段（1）と、アクセス権決定手段（2）と、アクセス権保持手段（3）と、サービス起動手段（5）とを有し、

利用者資格ファイル（41）は、利用者ＩＤと、その利用者のサービスへのアクセスに関する資格を表す資格ＩＤとを項目とするレコードを保持し、

アクセス権ファイル（42）は、サービスＩＤと、資格ＩＤまたは利用者ＩＤと、アクセスの種類とその可否と、そのレコードの優先度とを項目とするレコードを保持し、

要求受付手段（1）は、利用者からの要求を受付けて、接続要求をアクセス権決定手段（2）へ、サービス要求をサービス起動手段（5）へ伝え、

アクセス権決定手段（2）は、利用者からの接続要求があると、利用者ＩＤをキーとして利用者資格ファイル（41）を検索してその利用者ＩＤの存在の確認と、対応する資格ＩＤの取り出しとを行い、資格ＩＤおよび利用者ＩＤをキーとしてアクセス権ファイル（42）を検索して、同一のサービスについて複数のレコードがある場合には、優先度の高いレコードのアクセス権の指定により最終的なアクセスの可否を決定して、利用者ＩＤごとにサービスＩＤとアクセスの可否とをアクセス権保持手段（3）に保持し、

サービス起動手段（5）は、受け付けた利用者からのサービスへのアクセス要求があると、利用者ＩＤとサービスＩＤとアクセス種類とを受け取り、その内容が、アクセス権保持手段の内容に合致する場合に指定されたサービス提供手段を起動し、合致しなければ拒絶するように構成したアクセス権管理装置。

【請求項2】　アクセス権ファイル（42）は、サービスＩＤと、資格ＩＤまたは利用者ＩＤと、アクセスの種類とその可否と、そのレコードの優先度と、そのレコードの有効期間とを項目とするレコードを保持し、アクセス権決定手段（2）は、アクセス権ファイル（42）のレコードにより最終的なアクセスの可否を決定する場合に、有効期間外のレコードは無視することを特徴とする請求項1に記載のアクセス権管理装置。

【請求項3】　管理者情報ファイル（43）とアクセス権変更手段（6）とを設け、

管理者情報ファイル（43）には、アクセス権ファイル（42）の更新を行なう権限をもつ利用者の利用者ＩＤとサービスＩＤと優先度範囲とを項目とするレコードを保持し、

アクセス権変更手段（6）は、要求受付手段（1）からアクセス権ファイル（42）の内容の変更要求を受ける

2

と、管理者情報ファイル（43）を検索し、その利用者が設定可能なサービスと優先度範囲をチェックし、それを許可するか否かを決定することを特徴とする請求項1または請求項2に記載のアクセス権管理装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は多数の利用者を対象に複数の情報処理サービスを提供する情報システムに関する。特にその処理要求の妥当性を管理するアクセス権管理装置に関する。

【０００２】多数の利用者を対象に複数の情報処理サービスを提供する情報システムにおいては、サービスに対する要求（アクセスの種類・アクセス権）の管理・規制を行なっている。運用中に、利用者のアクセス権の変更・追加、特に一時的な変更や、サービスの追加・変更、特に一時的変更・試験的提供等の処理を行なうことが必要であり、それをシステムの安全性を損なわずに、かつ利用者の不便をきたさないように行なうことが要求されている。

【０００３】

【従来の技術】サービスには、例えば、電子伝票、会議室予約、旅費精算等の全員がアクセスできるものや、人事情報・評価のように特定の資格者のみアクセスできるものがある。

【０００４】サービスに対する要求の管理・規制の方法として、利用者個々に対して利用できるサービスとその処理内容（参照・更新等）や期間等を定義しておく方法が考えられるが、管理情報の量が膨大になるので、利用者の資格によるグループ設定を行い、このグループに対してアクセス権の内容を設定するやり方がある。グループはサービス毎に設定することができ、さらに、一般利用者、管理職、サービスの管理者あるいは処理の開発者等により分けることができる。

【０００５】予算申請を受け付けて登録するサービスのように、ある期間を設け、申請期限を過ぎてからの新規申請や申請データの修正は特定者以外には禁止する場合、従来の技術でも利用者の属するグループによるアクセス権の管理は実現可能である。ところで、そのサービスの管理者が、申請データを処理（例えば集計）する際に、申請データに誤りを発見し、その申請を行なった利用者に再申請を指示する必要がある場合を考える。このとき、申請期限は過ぎているので、他の利用者に対しては受け付けないようにする必要がある。従来の技術では、このような場合、その特定の利用者をその属するグループから一時的に外し、別のグループに（申請可能なグループ）に入れることになるが、特定の利用者は一時的に元のグループから外されるため、そのグループに許されていた他のサービスへのアクセスが制限されたり、逆に、本来制限されるはずのアクセスができてしまったりする可能性ができる。

【0006】
【発明が解決しようとする課題】従って、このような副作用を起こさないようにするには、この処理はかなり面倒な管理を必要とするものとなる。すなわち、関連するアクセス権情報を矛盾のないようにすべて修正し、さらに一時的な処置が済んだ時点で早急にもとに戻す必要がある。

【0007】本発明は、個々のアクセス権情報に優先度表示を付けて複数のアクセス権情報を同時に存在させ、優先度の高い情報によって実際のアクセス権を決定することにより、一時的な変更を含めてアクセス権の変更を柔軟にでき、かつ管理が容易なアクセス権管理装置を実現することを目的としている。

【0008】
【課題を解決するための手段】図1は本発明の原理構成図である。複数の利用者に対し複数の処理サービスを提供する情報システムにおける利用者のサービスへのアクセス権を管理する装置であって、第1の発明は、利用者資格ファイル41と、アクセス権ファイル42と、要求受付手段1と、アクセス権決定手段2と、アクセス権保持手段3と、サービス起動手段5とを有する。

【0009】利用者資格ファイル41は、利用者ＩＤと、その利用者のサービスへのアクセスに関する資格を表す資格ＩＤとを項目とするレコードを保持する。アクセス権ファイル42は、サービスＩＤと、資格ＩＤまたは利用者ＩＤと、アクセスの種類とその可否と、そのレコードの優先度とを項目とするレコードを保持する。

【0010】要求受付手段1は、利用者からの要求を受付けて、接続要求をアクセス権決定手段2へ、サービス要求をサービス起動手段5へ伝える。アクセス権決定手段2は、利用者からの接続要求があると、利用者ＩＤをキーとして利用者資格ファイル41を検索してその利用者ＩＤの存在の確認と、対応する資格ＩＤの取り出しとを行い、資格ＩＤおよび利用者ＩＤをキーとしてアクセス権ファイル42を検索して、同一のサービスについて複数のレコードがある場合には、優先度の高いレコードのアクセス権の指定により最終的なアクセスの可否を決定して、利用者ＩＤごとに、サービスＩＤとアクセスの可否とをアクセス権保持テーブルとしてアクセス権保持手段3に保持する。

【0011】サービス起動手段5は、受け付けた利用者からのサービスへのアクセス要求があると、利用者ＩＤとサービスＩＤとアクセス種類とを受け取り、その内容が、アクセス権保持テーブルの内容に合致する場合に指定されたサービス提供手段を起動し、合致しなければ拒絶する。

【0012】第2の発明は、アクセス権ファイル42は、サービスＩＤと、資格ＩＤまたは利用者ＩＤと、アクセスの種類とその可否と、そのレコードの優先度と、そのレコードの有効期間とを項目とするレコードを保持する。そして、アクセス権決定手段2は、アクセス権ファイル42のレコードにより最終的なアクセスの可否を決定する場合に、有効期間外のレコードは無視する。

【0013】第3の発明は、管理者情報ファイル43とアクセス権変更手段6とを設け、管理者情報ファイル43は、アクセス権ファイル42の更新を行なう権限をもつ利用者の利用者ＩＤとサービスＩＤと優先度範囲とを項目とするレコードを保持する。

【0014】アクセス権変更手段6は、要求受付手段1からアクセス権ファイル42の内容の変更要求を受けると、管理者情報ファイル43を検索し、その利用者が設定可能なサービスと優先度範囲をチェックし、それを許可するか否かを決定する。

【0015】
【作用】利用者が情報システムにログインしてきたとき、要求受付手段1は接続要求として利用者ＩＤ（例えばＵ1）をアクセス権決定手段に渡す。アクセス権決定手段2は、利用者ＩＤをキーとして利用者資格ファイル41を検索し、資格ＩＤを得る。さらに、資格ＩＤと、利用者ＩＤとをキーとしてアクセス権ファイル42を検索して、どちらかを含むレコードを抽出する。ここで、もし同一サービスに対するレコードが複数ある場合は、優先度の値が一番大きなレコードのアクセス可否項目をアクセス権として採用する。このようにしてサービスＩＤとアクセス権とを対応させたアクセス権保持テーブルを生成してアクセス権保持手段3に保持する。この対応テーブルは利用者ＩＤごとに区別しておく。利用者がログアウトした場合は、その利用者ＩＤのアクセス権保持テーブルは削除することになる。

【0016】利用者からのサービスＳ1へのアクセス要求を受けて、要求受付手段1は利用者ＩＤとサービスＩＤとをサービス起動手段5へ渡す。サービス起動手段5は、アクセス権保持手段に保持されたアクセス権保持テーブルを参照してアクセス可能と判断したらサービス提供手段の所定のサービスを起動する。

【0017】このように構成することにより、アクセス権ファイルには、同じサービスに対する同じ利用者の異なるアクセス権を指定したレコードが複数存在することになるが、どちらの指定を採るかが優先度によって決定され一意に定まる。従って、部分的に、または一時的にアクセス権を変更する場合に、変更したい内容にした（優先度の値は大きい）レコードを追加することができる。もとにもどす場合にはそれを削除するだけでよい。

【0018】第2の発明では、アクセス権を指定したレコードの有効期間を項目の1つにして指定してあるので、その期間外であれば、そのレコードは無いのと同じであり、前もって追加しておいたり、削除を延ばしたりしても問題がなく、アクセス権管理が容易になる。

【0019】第3の発明では、アクセス権ファイル42の内容更新を要求された場合には、更新要求者の利用者Ｉ

5

Dをキーとして管理者情報43を検索し、該当する項目が存在しなければ、更新要求を拒絶し、存在するならば変更対象のサービスＩＤと設定可能な優先度の範囲を限度としてアクセス権の更新を許可する。従って、アクセス権の変更を適正に行なうことができる。

【００２０】

【実施例】以下、図面を参照して本発明の実施例を説明する。図２は本発明の一実施例の構成図である。図１と同一の機能のものは、同一の符号を付して示す。

【００２１】図２において、利用者用の端末装置91は回線90を通じて情報システム92に接続されている。情報システム92は、メモリ81、プロセサ82、ファイル装置83、通信制御装置84よりなる。メモリ81には、全体を制御するオペレーティングシステム70と、利用者から要求されたサービスを実行するサービスプログラム７と、図１の原理構成図に示したアクセス権管理のための手段を実現したプログラムとがある。アクセス権管理のための手段を実現したプログラムは、端末装置91から入力されたコマンドを受け付けたり、サービスの実行結果を端末装置91に表示する制御を行なう要求受付部１と、利用者毎のアクセス権を決定するアクセス権決定部２と、アクセス権のチェックを行なってサービスプログラムを起動するサービス起動部５と、アクセス権を変更するためのアクセス権変更部６とよりなる。

【００２２】利用者の情報を格納した利用者資格ファイル41と、各サービスのアクセス権情報を格納したアクセス権ファイル42と、アクセス権情報を管理する管理者の情報を格納した管理者情報ファイル43とはファイル装置83に保持され、利用者毎のアクセス権の内容を保持するアクセス権保持部３はメモリ81またはファイル装置に保持される。これらのファイルやテーブルの操作、端末装置91とのやり取り等はオペレーティングシステム70を通して行なうが、自明のこととして以下の説明では省略する。

【００２３】図３は本実施例のファイル構成図である。図３（１）は利用者資格ファイル41の構成を示す。各利用者には、利用者ＩＤ、パスワード、所属グループが定義してある。

【００２４】図３（２）は、アクセス権ファイル42の構成を示す。アクセス権情報はそれぞれ、対象となるサービス、グループ、処理（参照、更新毎の可否）、期間、優先度の各項目で構成されている。項目の値が‘ ALL ’の場合は、その項目についてはすべてが対象となることを示す。また、グループの項目には、グループＩＤ（資格ＩＤ）の他に、利用者ＩＤを設定することもできる。期間の項目は、そのアクセス権情報レコードが適用される期間を示している。優先度の項目は、同一サービス、同一利用者に対してレコードが複数あるとき数値が大きいレコードが優先して使用されることを示す。

【００２５】図３（３）は、管理者情報ファイル43の構

6

成を示す。管理者情報ファイル43は、管理対象となるサービスのＩＤと、管理者の利用者ＩＤと設定可能な優先度の範囲を示す値が定義してある。

【００２６】情報システムのサービスの例として、本実施例では電子伝票Ｓ１、会議室予約Ｓ２、旅費精算Ｓ３等が提供されている。上記サービスを提供するため、サービスプログラム７は、各サービスを実現するプログラムモジュール（Ｓ１，Ｓ２，Ｓ３，・・・）から成り、各サービスはサービス起動部５によって対応するプログラムモジュールが起動されることによって行なわれる。

【００２７】以下に、本実施例の動作について説明する。まず利用者（ＩＤ：Ｕ１）は、端末装置91を操作して、接続要求をする。具体的には、端末装置91から利用者ＩＤとパスワードを入力する。すると要求受付部１は、入力された利用者のＩＤとパスワードの組を渡してアクセス権決定部２を起動する。アクセス権決定部２は利用者資格ファイル41から抽出した情報と照合して、もし、照合の結果が一致すれば接続要求を受理し、そうでなければ拒絶する。この段階はいわゆるログイン処理である。接続要求が受理された利用者については、以下の手順でアクセス権の調査が行なわれ、その結果がアクセス権保持テーブルとしてアクセス権保持部３に書き込まれ、その利用者が情報システム92との接続を開放（ログアウト）するまで保持される。そして、利用者からのサービスへのアクセス要求があるたびに、サービス起動部５は、このアクセス権保持テーブルの内容をチェックし、利用者が所望するサービスと処理に関するアクセス権が‘可’であれば、対応するプログラムモジュールを起動し、‘否’であれば利用できない旨のメッセージを端末装置91に送る処理を行なう。

【００２８】アクセス権保持部３への利用者のアクセス権の格納の手順を、図３、図４を参照しながら以下に説明する。アクセス権決定部３は、要求受付部１からの要求を受けてまず利用者資格ファイル41から利用者Ｕ１が所属するグループ（資格ＩＤ）を抽出する（この例ではＧ１とＧ３に属している）。

【００２９】次に、情報システムで提供されている各サービスについて以下の処理を行なう。サービスＳ１に対するアクセス権を求めるために、先ずアクセス権ファイル42から、サービスの値が‘ ALL ’または‘Ｓ１’であり、かつグループの値が‘ ALL ’または利用者Ｕ１が所属するグループ（Ｇ１，Ｇ３）または利用者のＩＤ（この場合Ｕ１）と一致するレコードで、期間の指定があればその期間内であるものを抽出し、優先度の値が最大であるレコードの値を各処理（更新、参照）ごとに求め、アクセス権保持部３に格納する。この操作をすべてのサービスについて繰り返す。

【００３０】図４（１）は、アクセス権ファイル42に、図３（２）のａ～ｃのレコードが登録され有効である場合（すなわちレコードｄがない場合または指定期間外）

7

に、アクセス権保持部3に格納される利用者ＩＤ＝Ｕ1の利用者に関する情報すなわちアクセス権保持テーブルの内容の例である。

【００３１】図4（2）は、アクセス権ファイル42に図3（2）のａ～ｄのレコードが登録され有効である場合（すなわちレコードｄがあり指定期間内の場合）に、アクセス権保持部3に格納される利用者ＩＤ＝Ｕ1の利用者に関する情報すなわちアクセス権保持テーブルの内容の例である。

【００３２】このアクセス権ファイルのレコードｄは、従来技術の項で述べた電子伝票の誤りを訂正するような場合に、利用者Ｕ1のみ更新できるように一時的に追加した項目であり、利用者Ｕ1が更新を完了したらこの項目を削除することにより、本来のアクセス権設定状態に戻すことができる。なお、削除しなくても指定期間外になれば自動的に無効になる。

【００３３】なお、図3（2）のアクセス権ファイルの内容を説明する。レコードａは、すべてのサービスをすべての利用者に開放することを意味する。このレコードだけであれば、なんの制約もなくアクセスできる。レコードｂが追加されると、サービスＳ1についてはグループＧ1とＧ2に属する利用者にのみ更新アクセスが期間を限って許されなくなる。レコードｃが追加されると、指定された期間の間は、それまで全員がアクセスできたサービスＳ3がアクセスできなくなる。これは例えば、サービスＳ3の内容の変更のため一時的にサービスを中止する場合である。レコードｄが追加された場合は、先に説明した通りである。

【００３４】一方、利用者が端末装置91からアクセス権ファイル42の更新を要求した場合、要求受付部1はアクセス権変更部6を起動する。アクセス権変更部6は、管理者情報ファイル43に、更新を要求している利用者ＩＤと更新を要求しているサービスＩＤの組が登録されていれば、そこに登録されている優先度の範囲でそのサービスについてのアクセス権管理情報のレコードをアクセス

8

権ファイル42に追加・削除・更新を許可する。これにより、アクセス権の変更がみだりに行なわれたり、誤って他のサービスに影響するようなことを防ぐことができる。

【００３５】

【発明の効果】以上説明したように、情報システムの運用中にアクセス権の変更を柔軟にでき、かつ管理が容易なアクセス権管理装置を実現することができる。システムの安全性を損なわずに、かつ利用者の不便をきたさないように、利用者のアクセス権の変更・追加、特に一時的な変更や、サービスの追加・変更、特に一時的変更・試験的提供等の処理を行なうことができる。

【図面の簡単な説明】

【図1】　本発明の原理構成図

【図2】　本発明の実施例の構成図

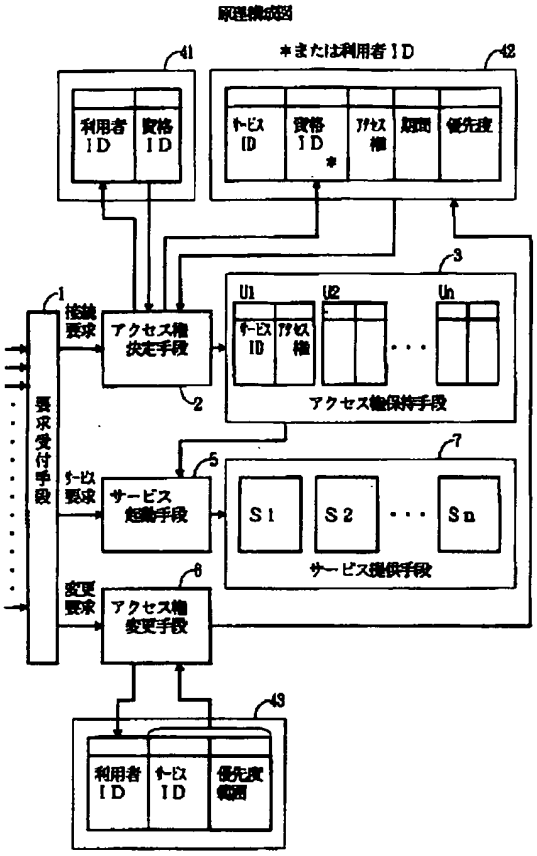
【図3】　実施例のファイル構成図

【図4】　アクセス権保持テーブルの内容の例

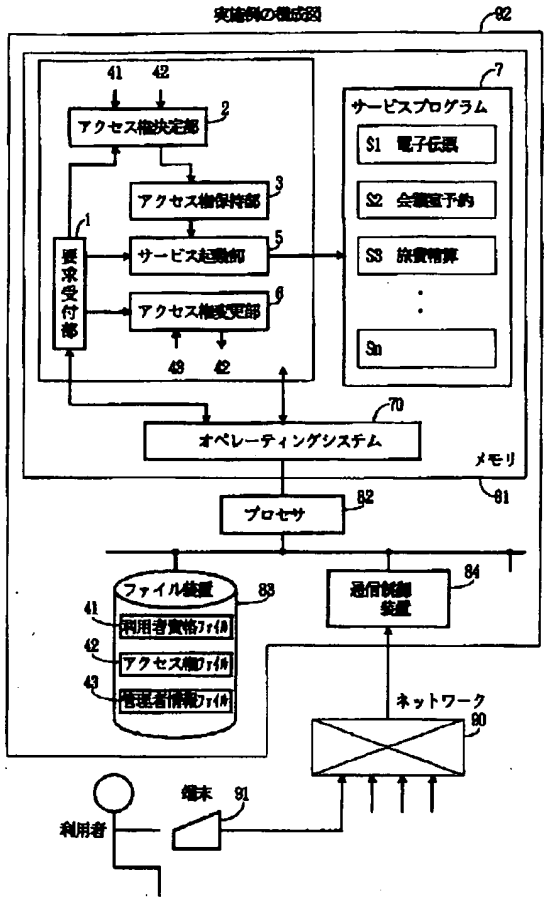
【符号の説明】

1　要求受付手段（要求受付部）
2　アクセス権決定手段（アクセス権決定部）
3　アクセス権保持手段（アクセス権保持部）
41　利用者資格ファイル
42　アクセス権ファイル
43　管理者情報ファイル
5　サービス起動手段（サービス起動部）
6　アクセス権変更手段（アクセス権変更部）
7　サービス提供手段（サービスプログラム）
70　オペレーティングシステム
81　メモリ
82　プロセサ
83　ファイル装置
84　通信制御装置
90　ネットワーク
91　端末装置
92　情報システム

【図１】



【図２】



【図４】

アクセス権保持テーブルの内容の例　（ＩＤ：Ｕ１の場合）

（１）レコードｄがない場合または指定期間外の場合

サービス	参照	更新
S1	可	否
S2	可	可
S3	否	否
：		

（２）レコードｄがあり指定期間内の場合

サービス	参照	更新
S1	可	可
S2	可	可
S3	否	否
：		

【図3】

実施例のファイル構成図

(1) 利用者資格ファイル

利用者ID	パスワード	グループ(資格)
U1	ABCD	G1, G3
U2	DEFG	G1, G2
U3	GHIJ	G1
M01	XYZA	G1, G10
・		

(2) アクセス権ファイル

	サービス	グループ	参照	更新	期間	優先度
a	ALL	ALL	可	可		10
b	S1	G1, G2		否	94/7/21 ～95/7/20	100
c	S3	ALL	否	否	94/7/20 ～94/7/22	50
d	S1	U1		可	94/7/21 ～94/7/21	200
	・					
	・					

(3) 管理者情報ファイル

サービス	利用者ID	優先度範囲
S1	M11	1～200
S2	M12, M13	1～100
S3	M31	1～100
S3	M32	1～500
・		
・		

Requested Patent: JP62241061A

Title: INFORMATION ACCESS MANAGEMENT SYSTEM ;

Abstracted Patent: JP62241061 ;

Publication Date: 1987-10-21 ;

Inventor(s): SHIMOSATO MASAO ;

Applicant(s): NEC CORP ;

Application Number: JP19860083842 19860411 ;

Priority Number(s): ;

IPC Classification: G06F15/16; G06F12/00 ;

Equivalents: ;

ABSTRACT:

PURPOSE:To eliminate the need for changing access rights for reference and updating required for each change in constituent members, by registering not only the names of the users who can refer to and update the information stored in a file area but also their hierarchical positions, etc.

CONSTITUTION:A register means 1 registers the access right management information including the accessible hierarchy information showing the hierarchical positions of the users who can refer to and update the information stored in a file and the file name and then stores the information in an access right memory means 2. A request accepting means 4 accepts file access requests for reference/updating of the information stored in the file from users and judges these request via a deciding means 3 based on user organization information informed via an organization information transmitting means 6 and the access right management information stored in the means 2. Based on the judgement result of the means 3, the permission or rejection of the file access request is informed in reply to the user who had file access through an answer means 5.

⑲日本国特許庁(JP)　⑪特許出願公開

⑫公開特許公報(A)　昭62-241061

⑤1Int.Cl.⁴	識別記号	庁内整理番号	⑬公開　昭和62年(1987)10月21日
G 06 F 15/16		Z-2116-5B	
12/00	302	6711-5B	

審査請求　未請求　発明の数 1　(全7頁)

⑤4発明の名称　情報アクセス管理方式

⑳特　願　昭61-83842

㉒出　願　昭61(1986)4月11日

⑦2発明者　下郷　昌夫　東京都港区芝5丁目33番1号　日本電気株式会社内

⑦1出願人　日本電気株式会社　東京都港区芝5丁目33番1号

⑦4代理人　弁理士　河原　純一

明　細　書

1．発明の名称

情報アクセス管理方式

2．特許請求の範囲

単一ないしはネットワークで結ばれた複数の電子計算機システムの主記憶装置ないしは補助記憶装置上に確保されたファイル領域に格納されている情報を参照、更新ないしは参照および更新するファイルアクセス権を管理するファイルアクセス管理システムにおいて、

前記電子計算機システムのファイルに格納されている情報を参照、更新ないしは参照および更新できる利用者の組織階層の階層位置ないしは上下位置関係を示すアクセス可能階層情報と前記情報が格納されているファイルを識別するファイル名とから構成されるアクセス権管理情報を登録する登録手段と、

この登録手段から供給される前記アクセス権管理情報を記憶するアクセス権記憶手段と、

前記電子計算機システムの利用者からのファイルに格納されている情報を参照ないしは更新するファイルアクセス要求を受け付ける要求受付け手段と、

前記電子計算機システムの利用者の利用者名と利用者の組織階層の階層位置情報とで構成される利用者組織情報を通知する組織情報通知手段と、

前記要求受付け手段から供給される前記ファイルアクセス要求の正当性を前記組織情報通知手段によって通知される前記利用者組織情報と前記アクセス権記憶手段によって記憶されている前記アクセス権管理情報とに基づいて判定する判定手段と、

この判定手段からの判定結果に基づき前記ファイルアクセス要求の許可ないしは拒否をファイルアクセスを要求した前記利用者に応答する応答手段と、

を含むことを特徴とする情報アクセス管理方式。

3．発明の詳細な説明

〔産業上の利用分野〕

本発明は情報アクセス管理方式に関し、特に単

一ないしはネットワークで結ばれた複数の電子計算機システムの主記憶装置ないしは補助記憶装置上のファイルに格納されている情報を参照、更新ないしは参照および更新するファイルアクセス権を管理するファイルアクセス管理システムにおける情報アクセス管理方式に関する。

〔従来の技術〕

不特定多数の利用者により利用される単一ないしはネットワークで結ばれた複数の電子計算機システムにおいては、電子計算機システム内の情報を第三者の参照ないしは更新から保護することが必要である。

従来、電子計算機システムにおいて、情報は主記憶装置ないしは補助記憶装置上に確保されたファイルと呼ばれる領域に格納され、第三者からの情報の参照ないしは更新の保護管理は情報が格納されているファイルへのアクセス権として管理され、各ファイルごとにアクセス権を持つ単一ないしはネットワークで結ばれた複数の電子計算機システムの利用者の一人ないしは複数の利用者名を指定せしめ、指定されていない利用者がファイルにアクセスすることを拒絶することにより、主記憶装置ないしは補助記憶装置上に格納されている情報を第三者の参照ないしは更新から保護している。

〔発明が解決しようとする問題点〕

不特定多数の利用者により利用される単一ないしはネットワークで結ばれた複数の電子計算機システムのファイル内に格納されている情報を複数の利用者間で参照ないしは更新する利用形態において、ファイルにアクセスできる利用者の範囲は、

① 範囲を限定しない、

② 特定個人に限定、

③ 複数人に限定

の3レベルに分類される。

①のレベルは、誰でもがアクセスできるレベルであるので、誰でもアクセス可能と指定せしめることにより従来技術によって実現されうる。

②のレベルは、ファイルごとにファイルをアクセスできる利用者名を設定せしめることにより従来技術によって実現されるが、情報を参照ないしは更新する利用者が少なく情報と情報を参照ないしは更新する利用者との関係が長期間固定しているような場合には余り問題にはならないが、情報を参照ないしは更新する利用者数が多いと情報が格納されているファイルに対応したファイルへのアクセス権を持つ利用者名数が非常に多くなり、また情報と情報を参照ないしは更新する利用者との関係が度々変わるような場合には、情報と情報を参照ないしは更新する利用者との関係が変わるごとに影響をうける各ファイルすべてのファイルへのアクセス権を持つ利用者名を変更しなければならないという欠点がある。

③のレベルは、複数の利用者名を指定せしめることにより従来技術でも実現されえないことはないが、利用者が多いと情報が格納されているファイルに対応したファイルへのアクセス権を持つ利用者名数が非常に多くなり、また情報と情報を参照ないしは更新する利用者との関係が度々変わるような場合には、情報と情報を参照ないしは更新する利用者との関係が変わるごとに影響をうける各ファイルすべてのファイルへのアクセス権を持つ利用者名を変更しなければならないとい欠点がある。

多数の利用者で情報を参照ないしは更新するのは、たとえば会社などの階層的組織がほとんどである。会社などの階層的組織では、組織内での階層位置ないしは階層の上下関係により情報へのアクセス権が決まり、各個人が誰であるかは問われない情報と個人のみで参照ないしは更新する情報とが大部分であり、特定の複数者のみ参照ないしは更新する情報は少ない。

本発明の目的は、上述の点に鑑み、階層的組織では組織内のかなりの情報が組織内での階層位置ないしは階層の上下関係により情報へのアクセス権が決まり各個人が誰であるかは問われない性格を持っていることに着目して、階層的組織の構成員の変動の度ごとに各情報が格納されているファイルの参照ないしは更新のためのアクセス権を変更する必要のない情報アクセス管理方式を提供す

ることにある。

〔問題点を解決するための手段〕

本発明の情報アクセス管理方式は、単一ないしはネットワークで結ばれた複数の電子計算機システムの主記憶装置ないしは補助記憶装置上に確保されたファイル領域に格納されている情報を参照、更新ないしは参照および更新するファイルアクセス権を管理するファイルアクセス管理システムにおいて、前記電子計算機システムのファイルに格納されている情報を参照、更新ないしは参照および更新できる利用者の組織階層の階層位置ないしは上下位置関係を示すアクセス可能階層情報と前記情報が格納されているファイルを識別するファイル名とから構成されるアクセス権管理情報を登録する登録手段と、この登録手段から供給される前記アクセス権管理情報を記憶するアクセス権記憶手段と、前記電子計算機システムの利用者からのファイルに格納されている情報を参照ないしは更新するファイルアクセス要求を受け付ける要求受付け手段と、前記電子計算機システムの利用者の利用者名と利用者の組織階層の階層位置情報とで構成される利用者組織情報を通知する組織情報通知手段と、前記要求受付け手段から供給される前記ファイルアクセス要求の正当性を前記組織情報通知手段によって通知される前記利用者組織情報と前記アクセス権記憶手段によって記憶されている前記アクセス権管理情報とに基づいて判定する判定手段と、この判定手段からの判定結果に基づき前記ファイルアクセス要求の許可ないしは拒否をファイルアクセスを要求した前記利用者に応答する応答手段とを含む。

〔実施例〕

次に、本発明について図面を参照して詳細に説明する。

第1図は参照すると、本発明の一実施例は、登録手段1、アクセス権記憶手段2、判定手段3、要求受付け手段4、応答手段5および組織情報通知手段6から構成されている。

第2図を参照すると、登録手段1における処理は、ファイルアクセス権情報登録要求受付けステップ21および登録情報通知ステップ22からなる。

第3図を参照すると、アクセス権記憶手段2における処理は、登録情報受付けステップ31および登録情報記憶ステップ32からなる。

第4図を参照すると、判定手段3における処理は、ファイルアクセス要求情報受付けステップ41、アクセス権情報取得ステップ42、組織情報取得ステップ43、正当性判定ステップ44、正当通知ステップ45および不当通知ステップ46からなる。

第5図を参照すると、要求受付け手段4における処理は、ファイルアクセス要求受付けステップ51およびファイルアクセス要求情報通知ステップ52からなる。

第6図を参照すると、応答手段5における処理は、判定結果取得ステップ61、判定結果比較ステップ62、要求許可通知ステップ63および要求拒否通知ステップ64からなる。

第7図を参照すると、組織情報通知手段6における処理は、取得要求受付けステップ71、組織情報取得ステップ72および組織情報通知ステップ73からなる。

第8図を参照すると、アクセス権管理情報80は、ファイルを識別するファイル名81と、そのファイルに格納されている情報を参照ないしは更新できる利用者の組織階層の階層位置ないしは上下位置関係を示すアクセス可能階層情報82とから構成されている。

第9図を参照すると、利用者組織情報90は、利用者の利用者名91と、利用者の組織階層の階層位置情報92とから構成されている。

ここで、組織の階層位置とは、会社の部とか課とか研究所のグループなどの階層的な組織の組織単位であり、上下位置関係とは、その階層の課長とか部長とかグループ長などの役職の上下関係のことである。

次に、このように構成された本実施例の情報アクセス管理方式の動作につい説明する。

まず、利用者は、情報を作成すると、作成した

情報を格納するファイルを電子計算機システムの主記憶ないしは補助記憶装置上に確保して情報を格納する。このとき、利用者は、情報を格納したファイル名８１に対応して格納情報を参照ないしは更新できる利用者の組織階層の階層位置ないしは上下位置関係を示すアクセス可能階層情報８２を指定して登録手段１を動作させる。

登録手段１は、ファイルアクセス権情報登録要求受付けステップ２１でこの登録要求を受け付け、登録情報通知ステップ２２でこの登録情報をアクセス権記憶手段２に通知する。

アクセス権記憶手段２は、この登録情報を登録情報受付けステップ３１で受け付け、登録情報記憶ステップ３２でこの登録情報であるアクセス可能階層情報８２と作成した情報を格納したファイルを識別するファイル名８１とがアクセス権管理情報として記憶されて動作が停止される。

次に、利用者は、ファイルに格納されている情報を参照ないしは更新するというファイルアクセス要求を行い、要求受付け手段４を動作させる。

要求受付け手段４は、このファイルアクセス要求をファイルアクセス要求受付けステップ５１で受け付け、ファイルアクセス要求情報通知ステップ５２で参照ないしは更新要求するファイル名と要求利用者名とで構成されるファイルアクセス要求情報を判定手段３に通知する。

判定手段３は、このファイルアクセス要求情報の通知をファイルアクセス要求情報受付けステップ４１で受け付け、アクセス権情報取得ステップ４２でこの受け付けたファイルアクセス要求情報のファイル名に対応するファイルのアクセス権記憶手段２によって記憶されたアクセス権管理情報８０が取得され、さらに組織情報取得ステップ４３でファイルアクセス要求情報の利用者名に対応する利用者の利用者組織情報９０を通知するように組織情報通知手段６に要求する。

組織情報通知手段６は、この要求を取得要求受付けステップ７１で受け付け、組織情報取得ステップ７２で要求された利用者名に対応する利用者組織情報９０を取得し、組織情報通知ステップ７３で判定手段３に通知する。

判定手段３は、この通知された利用者組織情報９０を組織情報取得ステップ４３で取得し、この利用者組織情報９０とアクセス権管理情報８０とから正当性判定ステップ４４で利用者のファイルアクセス要求の正当性が判定される。この判定結果が正当であれば、正当通知ステップ４５で正当という判定結果が応答手段５に通知される。判定結果が不当であれば、不当通知ステップ４６で不当という判定結果が応答手段５に通知される。

応答手段５は、この判定結果通知を判定結果取得ステップ６１で取得し、判定結果比較ステップ６２で取得された判定結果が正当であるか不当であるかを比較によって判定し、比較結果が正当であれば要求許可通知ステップ６３で利用者に許可が通知されて動作が停止される。比較結果が不当であれば、要求拒否通知ステップ６４で利用者に拒否が通知されて動作が停止される。

次に、判定手段３における判定処理について具体例を用いて説明する。ここでは、第１０図（ａ）で示される階層の組織において、第１０図（ｂ）で示されるアクセス可能階層情報８２が登録されているファイルに格納されている情報をアクセスする場合を例にとって説明する。

第１０図（ａ）は、組織階層の一例を示し、Ａ～Ａａａａｃｂはその組織の階層位置を示す。

第１０図（ｂ）は、ファイルに対応したアクセス可能階層情報８２の一例を示す。

第１０図（ｃ）は、第１０図（ａ）で示される組織において、第１０図（ｂ）で示されるアクセス可能階層情報が登録されているファイルにアクセスできる階層位置のリストを示す。ここで、参照可能階層位置リストのＡａａａａ，Ａａａｂ，ＡａａｃおよびＡａａａｄは、第１０図（ｂ）の参照アクセス可能階層情報の同階層という登録情報から、ＡａａａおよびＡａａは直上２階層までという登録情報から、ＡａａｂおよびＡａａｃは直上２階層下の直下１階層という登録情報からそれぞれ導かれる。また、更新可能階層位置リストのＡａａａは、第１０図（ｂ）の更新アクセス

可能階層情報の直上１階層という登録情報から導かれる。

例えば、Ａａａａａの階層位置の利用者のファイルに格納されている情報への参照アクセス要求は、第１０図（ｃ）に示される参照可能階層位置リストに利用者のＡａａａａという階層位置が含まれているので正当と判定されるが、更新アクセス要求は第１０図（ｃ）に示される更新可能階層位置リストにＡａａａａという階層位置が含まれていないので不当と判定される。また、Ａａａａの階層位置の利用者の参照アクセス要求は、第１０図（ｃ）に示される参照可能階層位置リストに利用者のＡａａａという階層位置が含まれているので正当と判定され、更新アクセス要求も第１０図（ｃ）に示される更新可能階層位置リストにＡａａａという階層位置が含まれているので正当と判定される。

しかしながら、ＡａａａｃａやＡａｂなどの階層位置の利用者の参照アクセス要求ないしは更新アクセス要求は、第１０図（ｃ）に示される参照可能階層位置リストにも更新可能階層位置リストにもＡａａａｃａおよびＡａｂという階層位置は含まれていないので、両方とも不当と判定される。

〔発明の効果〕

以上説明したように本発明には、主記憶装置ないしは補助記憶装置上に確保されたファイル領域に格納されている情報を参照、更新ないしは参照および更新できる１人ないしは複数の利用者名だけでなく、組織階層の階層位置ないしは上下位置関係を各ファイルに対して登録できるようにすることにより、単一ないしはネットワークで結ばれた複数の電子計算機システムの情報を複数の利用者で参照ないしは更新する利用形態において、組織構成員が絶えず変動しても、組織階層の階層位置ないしは上下位置関係を指定されたファイルにおいては、ファイルへのアクセス権はアクセス要求者の組織階層の階層位置ないしは上下位置関係により決まり、要求者個人の利用者名には依存しないので、構成員の変動の度ごとに各情報が格納されているファイルの参照ないしは更新のためのアクセス権を変更する必要がなくなるという効果がある。

４．図面の簡単な説明

第１図は本発明の一実施例を示す構成図、

第２図は第１図中に示した登録手段における処理を示す流れ図、

第３図は第１図中に示したアクセス権記憶手段における処理を示す流れ図、

第４図は第１図中に示した判定手段における処理を示す流れ図、

第５図は第１図中に示した要求受付け手段における処理を示す流れ図、

第６図は第１図中に示した応答手段における処理を示す流れ図、

第７図は第１図中に示した組織情報通知手段における処理を示す流れ図、

第８図はアクセス権管理情報の構成を示す図、

第９図は利用者組織情報の構成を示す図、

第１０図（ａ）～（ｃ）は階層組織の一例を示す図、アクセス可能階層情報を示す図およびアクセス可能階層位置のリストである。

図において、

１・・・・・登録手段、

２・・・・・アクセス権記憶手段、

３・・・・・判定手段、

４・・・・・要求受付け手段、

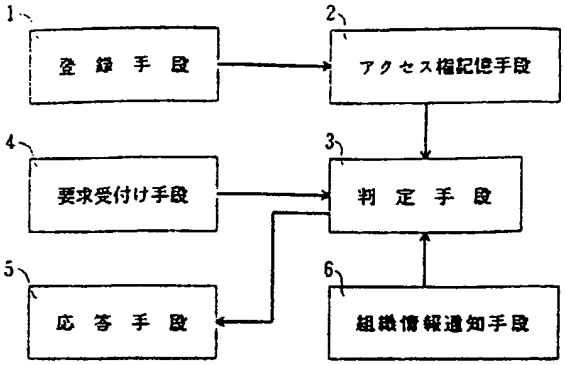
５・・・・・応答手段、

６・・・・・組織情報通知手段である。

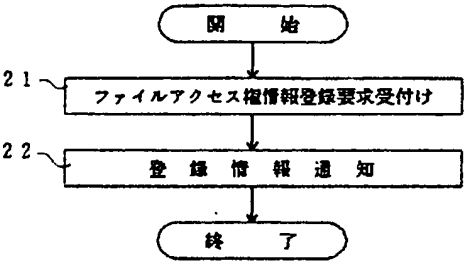
特許出願人　日本電気株式会社

代理人　弁理士　河原純一

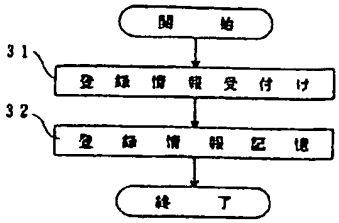
第 1 図



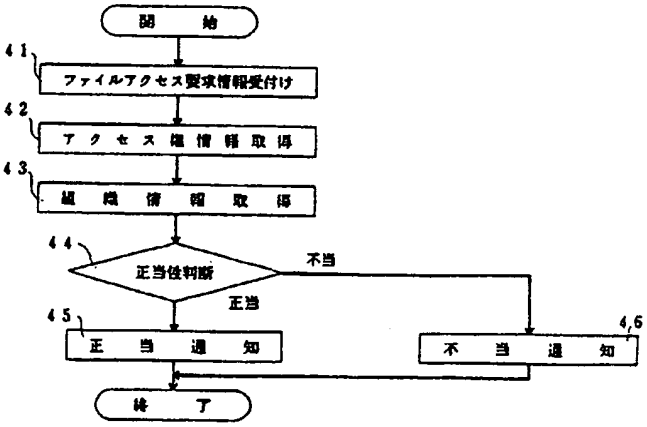
第 2 図



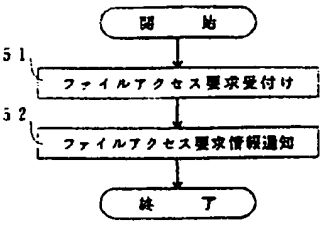
第 3 図



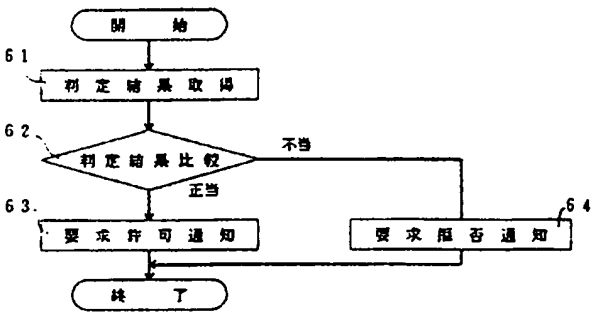
第 4 図



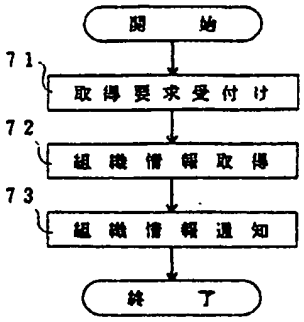
第 5 図



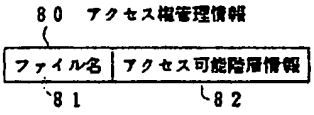
第 6 図



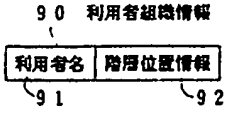
第 7 図



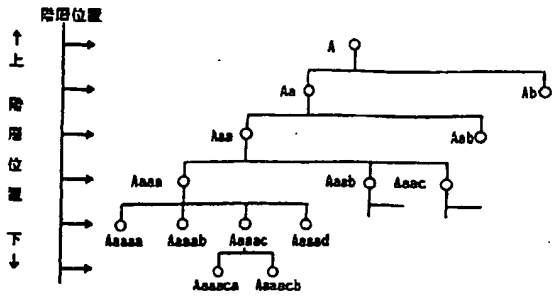
第 8 図



第 9 図



第 10 図 （a）



第 10 図 （b）

	登録情報
階層基準階層位置	A a a a b
参照アクセス可能階層	同階層，直上2階層まで，直上2階層下の直下1階層
更新アクセス可能階層	直上1階層

第 10 図 （c）

アクセス種類	アクセス可能階層位置リスト
参照	A a a a a，A a a a b，A a a a c，A a a a d，A a a a，A a a，A a a b，A a a c
更新	A a a a

Requested Patent: JP1068835A

Title: SOFTWARE RIGHT MANAGEMENT CONTROL METHOD ;

Abstracted Patent: JP1068835 ;

Publication Date: 1989-03-14 ;

Inventor(s): MORI RYOICHI; others: 01 ;

Applicant(s): RYOICHI MORI ;

Application Number: JP19870227106 19870910 ;

Priority Number(s): ;

IPC Classification: G06F9/06 ;

Equivalents: JP2723231B2 ;

ABSTRACT:

PURPOSE:To perform the right management superior in maintainability and flexibility in the environment of multiprogramming by individually assigning the key, with which ciphered instructions are deciphered, to a program or a program group as the right management object and switching the deciphering keys by a right management interrupt.

CONSTITUTION:A right management interrupt generating mechanism 13 which controls the change related to the right management state is provided independently of the supervisor interrupt which starts the processing function of an operating system or the like. When the right management interrupt is generated by the right management interrupt generating mechanism 13, the right management state is saved to a right management table 17 and a current right management ID 16 of a program status word 15 is changed by a right management state change processing 14, and the key with which instructions are deciphered is switched. Thus, right management in the environment of multiprogramming is possible.

⑲日本国特許庁（JP）　⑪特許出願公開

⑫公開特許公報（A）　昭64－68835

⑤1Int.Cl.⁴	識別記号	庁内整理番号	⑷3公開　昭和64年(1989)3月14日
G 06 F 9/06	330	A－7361－5B	

審査請求　未請求　発明の数 1　(全13頁)

⑤4発明の名称　ソフトウエア権利管理制御方法

⑵1特　願　昭62－227106
⑵2出　願　昭62(1987)9月10日

⑦2発明者　森　亮一　東京都文京区白山1－24－12
⑦2発明者　田代　秀一　千葉県柏市松葉町1－12－20－2
⑦1出願人　森　亮一　東京都文京区白山1－24－12
⑦4代理人　弁理士　小笠原　吉義　外2名

明　細　書

1．発明の名称　ソフトウェア権利管理制御方法

2．特許請求の範囲

（1）暗号化された命令を，命令フェッチ時に復号化して実行するデータ処理装置におけるソフトウェア権利管理制御方法であって，

権利管理状態の変更に関する割込みを制御する権利管理割込み発生機構(13)を設け，

権利管理対象となるプログラムまたはプログラム群に対し，暗号化された命令を復号化する鍵を個別に割り当て，

上記権利管理割込みにより，上記復号化する鍵を切り換える(14)ことを特徴とするソフトウェア権利管理制御方法。

（2）上記権利管理対象となるプログラムまたはプログラム群は，許諾条件プログラム，該許諾条件プログラムによって実行を制御されるソフトウェア本体，またはそれらの組み合わせであることを特徴とする特許請求の範囲第（1）項記載のソフトウェア権利管理制御方法。

3．発明の詳細な説明

〔概要〕

個々の権利対象ソフトウェア毎に，ソフトウェアの使用許諾，使用記録情報の収集などを，他のソフトウェアによる影響および干渉を受けずに行うことができるようにしたデータ処理装置におけるソフトウェア権利管理制御方法に関し，

マルチプログラミングの環境における権利管理の実現に適した制御方法を提供することを目的とし，

暗号化された命令を，命令フェッチ時に復号化して実行するデータ処理装置におけるソフトウェア権利管理制御方法であって，権利管理状態の変更に関する割込みを制御する権利管理割込み発生機構を設け，権利管理対象となるプログラムまたはプログラム群に対し，暗号化された命令を復号化する鍵を個別に割り当て，上記権利管理割込み

により，上記復号化する鍵を切り換えるように構成する。

〔産業上の利用分野〕

本発明は，個々の権利対象ソフトウェア毎に，ソフトウェアの使用許諾，使用記録情報の収集などを，他のソフトウェアによる影響および干渉を受けずに行うことができるようにしたデータ処理装置におけるソフトウェア権利管理制御方法に関する。

半導体技術の進歩によって，計算機ハードウェアの価格低下，小型化，高機能化が急速に進みつつある。ハードウェア機能の高度化は，ソフトウェアの高度化を要求し，結果として，ハードウェア開発者，ソフトウェア開発者，利用者の明確な分業化が促進されている。

しかしながら，ソフトウェア流通の環境は，現在，混乱した状況にあるといってよく，その環境を改善するために，ソフトウェア権利者の保護と利用者の便利さとを両立させるシステムであって，

ソフトウェアの自由な大量流通を支援するシステムを構築することが望まれている。

〔従来の技術〕

ソフトウェアの開発には，大量の人員および時間を要するにもかかわらず，利用者または第三者は，完成したソフトウェアを複製して，全く同じものを作成することが比較的容易にできる。そのため，ソフトウェアを開発した権利者（ソフトウェア権利者）は，開発コストの回収が不能になるケースが多く，これに対して，(1)ソフトウェア製品の価格を盗用による損失分を上乗せした値とする，(2)契約によって無断複製を禁止する，(3)ソフトウェアを複製不能な媒体に封入して流通させる，(4)計算機のシリアルナンバーなどにより使用可能な装置を限定する，などによって対処することが行われている。

しかしながら，上記(1)は，正当な利用者に過度の負担を与え，上記(2)は，繁雑な契約手続きが必要になると共に，それによる効果が充分でなく，

上記(3)は，バックアップがとれなくなるという問題や，複製不能にしている手段を調べ，それを回避することにより複製可能となるという問題があり，上記(4)は，特定計算機を識別するために，ソフトウェアの販売時などに特殊化処理が必要となり，自由なソフトウェアの流通が阻害されるという問題がある。

これらの問題を解決するものとして，本出願人は，ソフトウェア権利者の権利と，利用者の自由と，流通の容易性の３つを同時に満たすソフトウェアサービスシステム（ＳＳＳ）の基本構想を提案している。〔森亮一："ソフトウェアサービスシステム"，電子通信学会誌，Vol.67，No.4，pp.431-436（Apr.1984）〕

ここで提案されたソフトウェアサービスシステムの基本は，

(1) ソフトウェアは，いかなる条件が成立した場合に，その実行を許すかの許諾条件を内部に持つべきである。

(2) 計算機内部には，利用者の持つ権利を記述した権利記録があるべきである。これには，共通クレジット，買い取り記録，試用記録，特別許諾コード等が有り得る。

(3) 計算機は，上記(2)の権利記録が上記(1)の許諾条件を満足した場合にのみ，ソフトウェアの利用を許す実行管理機構を持つべきである。

(4) 権利者が，ソフトウェアの利用状況を把握できるように，計算機は，回収用作業記録を持つべきである。利用者に特別な面倒を与えることなく，この作業記録を回収する電子的手段が有り得る。

…というものである。

ところで，このソフトウェアサービスシステムの提案とは別に，暗号化された命令およびデータを主記憶にロードし，命令フェッチ時に，計算機のシリアルナンバーなどの復号鍵によって復号して実行する計算機アーキテクチャが知られている。

〔発明が解決しようとする問題点〕

上記ソフトウェアサービスシステムを，マルチプログラミングの環境下で実現する場合，複数の

ソフトウェア権利者に係る複数のプログラムが，時分割的に同時に計算機資源を利用して，走行することになる。この場合，第１に，権利の管理に係るハードウェア機構を制御する主体を，いかにして許諾条件をチェックするプログラム（以下，許諾条件プログラムという）に限定するか，第２に，マルチプログラミングの環境において，許諾条件プログラムと，それによって実行を制御されるソフトウェア本体との対応関係をいかにして保つか，第３に，オペレーティング・システムに対し，スワッピングやプロセス・スイッチを行うためのアクセス権を認めつつ，許諾条件プログラムの内容を改ざんしたり，権利管理のための流れを乱したりすることがないように，いかにしてアクセス権を制限するか，といったことに対する考慮が重要となる。従来，このような権利管理に対する適切な計算機の実行制御を行う機構はなかった。

本発明は上記問題点の解決を図り，マルチプログラミングの環境における権利管理の実現に適した制御方法を提供することを目的としている。

〔問題点を解決するための手段〕

第１図は本発明の原理説明図である。

第１図において，１０は権利管理の単位となる権利管理対象プログラム，１１は許諾条件をチェックしソフトウェア本体の実行を制御する許諾条件プログラム，１２はワープロソフト，コンパイラなどの利用者の使用目的に応じた処理を行うソフトウェア本体，１３は権利管理状態の変更に関する割込みを制御する権利管理割込み発生機構，１４は権利管理割込みによって権利管理状態を変更する権利管理状態の変更処理，１５はプログラムステータスワード（ＰＳＷ），１６は権利管理対象プログラム１０を識別する権利管理ＩＤ（ＰＭＩＤ），１７は権利管理対象プログラム１０の暗号化された命令を復号化する鍵を権利管理ＩＤ１６毎に記憶管理する権利管理テーブル，１８は暗号化された命令をその命令のフェッチ時に復号化する復号化回路を表す。

権利管理対象プログラム１０は，例えば許諾条

件プログラム１１とソフトウェア本体１２との対によって構成される。これらの全部または一部は，権利管理対象プログラム１０毎に定められた１または複数の暗号鍵によって予め暗号化されている。

権利管理対象プログラム１０のロード時に，暗号化された命令を復号化する鍵情報が，その権利管理対象プログラム１０に対してシステムでユニークに割り当てた権利管理ＩＤ１６と共に，権利管理テーブル１７に書き込まれる。

プログラムステータスワード１５は，カレントの権利管理ＩＤ１６が設定されるフィールドを持っている。プログラム実行時に，プログラムステータスワード１５に設定されている権利管理ＩＤ１６により，権利管理テーブル１７の該当エントリがアクセスされ，そのエントリ中の復号鍵（ＫＥＹ）が読み出されて，復号化回路１８へ送られる。復号化回路１８は，暗号化された命令をフェッチしたときに，権利管理テーブル１７から読みだした復号鍵によって復号し，その復号した命令を計算機の命令実行ユニットへ送る。

本発明では，オペレーティング・システム等の処理機能を起動するスーパバイザ割込みなどとは別に，権利管理状態に関する変更を制御する権利管理割込み発生機構１３を設ける。そして，権利管理割込み発生機構１３により，権利管理割込みが発生した場合に，権利管理状態の変更処理１４によって，権利管理テーブル１７への権利管理状態の退避やプログラムステータスワード１５のカレントの権利管理ＩＤ１６の変更などを行う。これにより，命令などを復号化する鍵を切り換える。

〔作用〕

本発明によれば，割込みが，オペレーティング・システムによる資源管理などに関連する従来の割込みと，権利管理状態を制御するための権利管理割込みの２系統に分離されることになる。

権利管理割込みが発生したときに，権利管理状態の変更処理１４によって，権利管理テーブル１７に基づく権利管理状態を変更するので，権利管理に係るハードウェア機構を制御することができ

る主体を，特定の許諾条件プログラム１１などに個別的に限定することができる。また，個々に復号鍵が管理されるので，ある許諾条件プログラム１１に対応した権利管理テーブル１７中のエントリ等，権利管理に関する情報を，他の権利管理対象プログラム１０などによって，改ざんされることを防止することができる。

また，割込みが，少なくとも従来の割込みと権利管理割込みの２系統になるので，オペレーティング・システムによる権利管理に対する望ましくない関与も，防止することが可能となる。

〔実施例〕

第２図は本発明を用いたソフトウェアの流通形態の例を説明するための図，第３図は本発明による権利管理の説明図，第４図は権利管理状態の状態遷移図，第５図は本発明に用いるハードウェア構成例，第６図は権利管理対象となるソフトウェアの構成例，第７図は権利管理テーブルの構成例，第８図はＰ割込み要求テーブルの構成例，第９図

はＳ割込み制御の例，第１０図はＳ割込みからの復帰制御の例，第１１図は本発明の一実施例におけるスタックの使用例，第１２図は本発明の一実施例における空間と鍵の関係説明図を示す。

本発明は，ソフトウェアの保護と利用促進のため，例えば第２図に示すようなソフトウェアの流通形態を採用する場合に用いることができる。

① ソフトウェアを開発したソフトウェア権利者は，そのソフトウェアを流通させる場合，ソフトウェアの利用者に対して要求する条件を記述した許諾条件プログラムを作成し，それを開発したソフトウェア本体と結合して，暗号化し，電波による放送も含めた自由な媒体により，ソフトウェアを配布する。

② ソフトウェアを利用したい利用者は，例えば予め共通クレジット（ＣＣ）を自動販売機などのベンディングマシンにより購入する。この共通クレジットの内容は，例えばＩＣカードによる媒体を介して，データ処理装置２０に入力できるようになっている。

③ 利用者は，共通クレジットの内容が記録されたＩＣカードを，本発明に係るソフトウェア権利管理制御機能を持つデータ処理装置２０に装着し，適当な流通路から入手したソフトウェアを，データ処理装置２０上で動作させる。これにより，そのソフトウェアの許諾条件プログラムが復号化されて動作し，ＩＣカードへのアクセスにより，共通クレジット情報に関連するチェックなどを行う。必要に応じて共通クレジットに記録されている料金の減算を行う。チェックに合格した場合にのみ，必要なソフトウェア本体の実行を可能とする制御を行う。そのとき，その利用に関する作業記録情報を蓄積しておき，その作業記録（ＡＲ）を適当な時期にＩＣカードへ転記する。

④ ＩＣカードに書き込まれた作業記録は，例えば共通クレジットを継続使用するために，新たな料金の支払いにより共通クレジットの内容を再設定するときに，ベンディングマシンに読み取られ，収集される。この作業記録情報を参照することにより，ソフトウェア利用状況の統計をとり，それ

に基づいて複数のソフトウェア権利者間で妥当な料金の分配を行うことができる。

第２図に示したソフトウェア流通形態は，一例であり，本発明は，共通クレジットではなく，各ソフトウェア個別の利用資格情報を管理する流通形態をとる場合にも採用することができる。また，ＩＣカード以外の媒体を用いることも可能である。

本発明による場合，データ処理装置２０は，複数の独立したソフトウェアを，マルチプログラミングによって並列動作させることができる。

本発明による権利管理は，概念的には，第３図に示すようになる。

マルチプログラミングの環境では，オペレーティング・システムが，入出力装置の割り当て，プロセッサの割り当てなどの計算機ハードウェア資源の管理を行っている。この場合，誤ったプログラムの実行などによって支障をきたさないようにするために，また計算機の資源を多く利用しようとする利用者によって，資源が恣意的に用いられることを防ぐために，計算機の実行状態をユーザ

状態と，スーパバイザ状態に分けることが広く行われている。

利用者のプログラムは，ユーザ状態で実行され，資源管理用のオペレーティング・システムは，通常，スーパバイザ状態で実行される。そして，利用者による資源の濫用を防止し，また利用者が勝手に実行状態を変更できないようにするために，ユーザ状態からスーパバイザ状態へと状態を変更する手段を，割込み（割出しを含む）に限ることによって，両状態の分離を図っている。

ここで，第２図で説明したような許諾条件プログラムの復号化や，共通クレジットの読み書きなどに関連する権利管理を，スーパバイザ状態のもとで，オペレーティング・システムにより制御するとすれば，利用者が自己の都合のよいように内部を改変したオペレーティング・システムをロードすることに関して，抵抗力を持たない。これに対し，オペレーティング・システムをファームウェア化して，内容の変更を困難にするという対応策も考えられるが，オペレーティング・システム

のバージョンアップが不可能になるなどの他の問題が発生する。

そこで本発明では，スーパバイザ状態とは異なる権利管理状態という特別な実行状態を新設している。権利管理状態は，ソフトウェアをできるだけ無料で実行したいと考える利用者があり得る環境の中で，利用者の故意または過失による介入を避けつつ，外部から供給されたソフトウェアの権利に関する管理を行うためのプログラムを実行可能とする状態である。

権利管理状態は，他の状態からそこへ状態を変更する手段を割込みに限っている点および他の状態では実行できない命令を持つ点で，スーパバイザ状態に類似しているが，暗号の応用と鍵の管理とによって不正なアクセスの防止を図る点が，スーパバイザ状態と大きく異なる。

第３図に示すように，オペレーティング・システムが，スーパバイザ状態のもとで，資源管理を行うのに対し，権利管理対象となっているプログラムの各許諾条件プログラム１１Ａ，１１Ｂ，…

は，個々の権利管理状態のもとで，ソフトウェア本体１２Ａ，１２Ｂ，…の実行を監視することによって，権利管理を行う。以下，許諾条件プログラムと，それによって制御されるソフトウェア本体とを合わせて権利管理対という。

権利管理状態を設けることにより，本発明に係るデータ処理装置は，例えば第４図に示すような状態遷移を行う。ユーザ（ＵＳ）状態，スーパバイザ（ＳＶ）状態，権利管理（ＰＭ）状態があり，スーパバイザ状態は，さらにＳＶｐ状態とＳＶｕ状態とに分かれる。ＳＶｐ状態とＳＶｕ状態とは，使用できる命令，アクセス権等に関しては全く同様であるが，割込みからの復帰命令を実行した場合に，ＰＭ状態に復帰するかＵＳ状態に復帰するかの点が異なる。

以上の４つの状態（ＳＶｐ，ＳＶｕ，ＰＭ，ＵＳ）は，独立した２つの状態ビットにより表現される。その１つは，プログラムステータスワード１５中に置くＳビットであり，他の１つは権利管理テーブル１７中に置くＰビットである。

Ｓビットの変化に伴う状態遷移は，スーパバイザ状態で実行するプログラム，即ち，オペレーティング・システムの制御に関係するものであり，Ｐビットの変化に伴う状態遷移は，ＰＭ状態で実行されるプログラム，即ち，許諾条件プログラムの制御に関係するものである。

これら２系統の状態遷移を直交した関係におき，互いに他の系統の状態遷移には直接影響を及ぼさないようにしているので，資源管理と権利管理との制御機構の分離が図られている。

以下，Ｓビットに関連する従来の割込みをＳ割込みといい，Ｐビットに関連する本発明に係る権利管理割込みをＰ割込みという。

本発明に用いるハードウェア構成は，例えば第５図に示すようになっている。

第５図において，第１図と同符号のものは，第１図に示すものに対応する。３０は命令実行ユニットである演算／制御部，３１はＩＣカードインタフェース，３２はＩＣカードから入力された共通クレジット情報を記憶する共通クレジットレジ

スタ（ＣＣＲ），３３はソフトウェア買い取り情報などの個々のソフトウェアに特有な権利管理情報を持つ個別権利メモリ（ＳＲＭ），３４はソフトウェアの利用実績情報が記録される作業記録メモリ（ＡＲＭ）である。

共通クレジットレジスタ３２，個別権利メモリ３３，作業記録メモリ３４は，不揮発メモリによって構成される。

３５は許諾条件プログラムまたはソフトウェア本体の全部または一部をＤＥＳ方式により復号化するＤＥＳ暗号機構，３６はＰ割込みの要因などの制御情報を記憶するＰ割込み要求テーブル（ＰＩＴ），３７は権利管理対象となるソフトウェアをロードする際に後述するキー格納部の復号化を行う公開鍵暗号機構である。

第５図に示す装置が扱うソフトウェアの構造は，例えば第６図に示すようになっている。キー格納部，許諾条件プログラム１１，ソフトウェア本体１２の３つの部分からなる。

本実施例では，許諾条件プログラム１１を，権

利者が任意に与える鍵（ＫＥＹ１）によって，いわゆるＤＥＳ方式で暗号化する。また，ソフトウェア本体１２の少なくとも一部を，やはり権利者が任意に与える鍵（ＫＥＹ２）によってＤＥＳ方式で暗号化する。なお，ＤＥＳ方式については，例えば「一松信監修：＂データ保護と暗号化の研究＂，日本経済新聞社〔1983〕」に記述されている。

キー格納部には，このソフトウェアを識別するユニークな権利番号（ＰＮ）と，個別権利記録（ＳＲ）に対するアクセスを管理するための非公開のＳＲアクセスキーと，許諾条件プログラム１１，ソフトウェア本体１２を復号化するためのＫＥＹ１，ＫＥＹ２とが設定されるようになっている。キー格納部は，例えば公開鍵暗号方式のひとつであるＲＳＡ法により暗号化される。なお，このＲＳＡ法も，例えば上記＂データ保護と暗号化の研究＂の著書などにより知られている。公開鍵暗号方式は，暗号化に用いる鍵を知っていても，秘密鍵である復号化鍵を知っていないと，暗号を

解くことができないという特徴がある。もちろん他の暗号方式を用いても，同様に本発明を実施することは可能である。

このキー格納部を復号化するための鍵は，ハードウェア製造時に，第５図に示す公開鍵暗号機構３７の内部に封入しておく。このソフトウェアを主記憶上にロードする際に，公開鍵暗号機構３７によってキー格納部を復号化し，その復号結果を権利管理テーブル１７に設定する。

権利管理テーブル１７は，例えば第７図に示すような権利管理に関する情報を持つ。

権利管理テーブル１７中のレコードは，このレコードに対応する権利管理対を識別するためのＰＭＩＤ，権利番号（ＰＮ），ＳＲアクセスキー，許諾条件プログラムを実行するために必要なＤＥＳ鍵（ＫＥＹＩ），ソフトウェア本体を実行するために必要なＤＥＳ鍵（ＫＥＹ２），ＫＥＹ２の有効／無効を示すビット（Ｋ２Ｆ），許諾条件プログラムによる権利管理状態か否かを表すＰビット，対応する権利管理対がロードされた直後であ

ることを表すイニシャルビット，およびスタックの正当性を確認するためのスタックチェックコードの９つのフィールドからなる。

システム中に同時に存在し得る権利管理対の数は，権利管理テーブル１７のレコード数に等しい。複数のレコードのうち，ＰＳＷ中のカレントＰＭＩＤの値と一致したＰＭＩＤを持つレコードを，カレントＰＭＴレコードと呼び，権利管理テーブル１７のレコードに対するアクセスは，通常の場合，このカレントＰＭＴレコードに対して行われる。

第５図に示すＰ割込み要求テーブル（ＰＩＴ）３６は，権利管理割込み（Ｐ割込み）発生機構１３が使用するテーブルであり，例えば第８図に示すような構成になっている。ＰＩＴ３６中の各レコードは，Ｐ割込みを要求する権利管理対のＰＭＩＤ，そのＳＲアクセスキーを記憶するフィールド，要求するＰ割込みの要因を記述するフィールド（Ｐ－Reason），Ｐ割込み発生時の飛び先アドレスを記述するフィールド（New-PC），Ｐ割込み

要求の要因に対するパラメータを記述するフィールドの５つのフィールドからなる。

ＰＩＴ３６中のレコードは，ＰＭ状態でしか使用できない特権命令によってのみ，生成／消去が可能にされ，また，ＳＲアクセスキーを用いたアクセス権のチェックにより，正当な権利者以外の許諾条件プログラムからは，既に存在するＰＩＴレコードの書き換えや消去ができないようになっている。

Ｐ割込みの要因には，例えばＵＳ状態におけるＰ割込み命令の実行，タイマー割込み，ＵＳ状態における特定命令の実行（命令トラップ）がある。タイマー割込みに対しては，パラメータとして割込み発生の時間間隔を与えることができ，命令トラップに対しては，任意の命令コードをパラメータとして与えることができる。これにより，許諾条件プログラムは，必要に応じてソフトウェア本体の実行制御を行い，またソフトウェア本体の実行に関する作業記録をとる契機を得ることができるようになっている。

第５図に示すＰ割込み発生機構１３は，許諾条件プログラムがＰＩＴ３６に指定した要因が発生したときに，自動的にＰ割込みを起こす機構であるが，その内部の回路構成等については，従来技術と同様な一般的な割込み技術を用いて実施可能であるので，これ以上の説明を省略する。

Ｓ割込みとＰ割込みとを独立させるために，Ｓ割込みが発生した場合におけるプロセッサの動作は，例えば第９図に示す(a)～(l)のようになる。

(a) Ｓ割込みが発生した場合，ファームウェアなどにより，カレントＰＭＴレコードのＰビットの０／１をチェックし，０であれば(d)へ制御を移す。

(b) Ｐビットが１であれば，現在のＰＳＷ及びプログラムカウンタの値から，１６bit のサイクリックコードを生成し，新スタックチェックコード１とする。

(c) １６bit の乱数を生成し，新スタックチェックコード２とする。

(d) 新スタックチェックコード１，２及びカレントＰＭＴレコードから取り出した旧スタックチェックコード１，２の４ワード（６４bit）をＫＥＹ１で暗号化し，スタックへストアする。

(e) 新スタックチェックコード１，２を，カレントＰＭＴレコードへストアする。

(f) ＰＩＴの飛び先アドレス（New-PC）を，プログラムカウンタにセットする。そして，(h)の制御へ移る。

(g) Ｐビットが０の場合には，割込みベクタをプログラムカウンタにセットする。

(h) ＰＳＷ，プログラムカウンタ，スタックフォーマットコードをスタックへストアする。

(i) ＰＳＷのＳビットを"１"にし，オペレーティング・システムの割込み処理ルーチンへ制御を移す。

なお，この例では，Ｓ割込み時における飛び先アドレス，即ち，プログラムカウンタに設定する新しい値を，いわゆる割込みベクタテーブルからロードする方式を用いている。

スーパバイザ状態において，ＲＳＢ（Return from S-Exception）命令を実行することにより，ＰＭ状態またはＵＳ状態のいずれかへ遷移する。そのときのプロセッサの動作は，例えば第１０図に示す(a)～(d)のようになる。

(a) カレントＰＭＴレコードのＰビットの０／１をチェックする。それが０であれば，(f)へ制御を移す。

(b) スタックから，スタックチェックコード１，２および旧スタックチェックコード１，２を取り出し，それらをＫＥＹ１で復号化してプロセッサ内に保持する。

(c) スタックのＰＳＷ及びプログラムカウンタの値から，１６bit のサイクリックコードを生成し，上記(b)で取り出したスタックチェックコード１と比較する。不一致であれば，エラーとして処理する。即ち，ＰＭ状態への遷移などを防止する。

(d) 次に，カレントＰＭＴレコードのスタックチェックコード２と，上記(b)で取り出したスタックチェックコード２とを比較する。不一致であ

ればエラーとして処理する。

(e) 旧スタックチェックコード1，2をカレントPMTレコードへストアする。

(f) PSWのSビットを"0"に戻す。

(g) スタックにストアされているプログラムカウンタの値を，プログラムカウンタにセットする。そして，割込み発生直前に実行されていたルーチンへ復帰する。

PM状態からS割込みによりSVp状態へ遷移する際に，例えば第11図（a）に示すようなスタック情報が，プロセッサによってスーパバイザ空間に退避される。スタックチェックコードは，スタックに退避された情報が，オペレーティング・システムによって書き換えられることを防止するために使用されるコードである。

スタックチェックコード1は，第9図で説明したように，スタックに退避されるPSW及びプログラムカウンタの値から生成するサイクリックコードであり，スタックチェックコード2は，16bitの乱数である。これらは，スタックに退避さ

れると同時に，カレントPMTレコードにも格納され，後にスタックの情報を取り出す際にチェックされる。これにより，スタック内容が書き換えられたことを検出できるようになっている。

旧スタックチェックコード1，2は，割込みがネストされた場合に，古いスタックチェックコードを記録するためのものである。

US状態からSVu状態へ遷移する場合には，第11図（b）に示すデータがスタックに退避される。また，SV状態時におけるS割込み発生時には，第11図（c）に示すデータがスタックに退避され，US状態時におけるP割込み発生時には，第11図（d）に示すデータがスタックに退避される。なお，SV状態においては，P割込みはマスクされる。

本実施例では，SV，PM，USの各状態毎に，プログラムのアクセスできる空間およびハードウェアによって自動的に使用される暗号の鍵が決められる。第12図は，その空間と鍵の関係を示している。

PM状態におけるプログラムの実行は，以下のとおりである。

PM状態において，命令フェッチ，データのロード／ストア，サブルーチン呼び出し，割込み発生時におけるプログラムカウンタのスタックへの退避など，主記憶に対するあらゆる書き込み，読み出しの際に，カレントPMTレコード中のKEY1を用いた自動的な暗号化および復号化が，第5図に示すDES暗号機構35によって行われる。PM状態で使用するアドレス空間は，US状態の空間と全く同一であり，区別されてない。これらは，鍵の管理によってのみ隔離されている。

オペレーティング・システムといえども，鍵を知らない限り，許諾条件プログラムを一時停止，強制終了させること以外，その動作に不正に介入することはできない。

PM状態からUS状態への戻りは，RPE（Return from P-Exception）命令の実行によって行われる。

US状態においては，命令フェッチ時に限り，

カレントPMTレコードのKEY2を用いた復号化を行う。ただし，復号化による命令実行速度の低下を軽減するために，プログラムのすべての部分を暗号化するのではなく，暗号化する範囲を任意に指定できるようにする方式をとることが望ましい。そのため，US状態で，カレントPMTレコードのK2Fビットをコントロールする命令が用意され，K2Fビットが"1"の間だけ，命令フェッチ時に，KEY2による復号化を自動的に行うようになっている。

オペレーティング・システムは，以下の処理を行う。

① 権利管理対象のソフトウェア全体を，暗号化されたまま主記憶へロードする。

② 次に，権利管理テーブル17内に新しいレコードを生成する命令を発行する。この命令では，これから実行しようとする権利管理対に与えるPMIDの値と，ロードしたソフトウェアのキー格納部の先頭論理アドレスとをパラメータとする。PMIDの値は，システム内部においてユニーク

であれば，任意でよい。この命令によって，ハードウェアは，ソフトウェアのキー格納部を復号化し，その情報を権利管理テーブル１７に格納して新しいＰＭＴレコードを作成する。同時に，ＰＳＷのカレントＰＭＩＤのフィールドを，これから実行しようとする権利管理対のＰＭＩＤの値にセットする。権利管理テーブル１７のイニシャルビットおよびＰビットを＂１＂にセットする。

③　ＲＳＥ命令を実行する。カレントＰＭＴレコードは，新しく生成されたレコードであり，このイニシャルビットが＂１＂である場合に，ＲＳＥ命令が実行されると，プロセッサの状態はＰＭ状態に遷移し，許諾条件プログラムに実行制御が渡される。同時に，イニシャルビットはクリアされ，以後，オペレーティング・システムから許諾条件プログラムへ制御を移す手段は，Ｓ割込みに対するＲＳＥ命令の実行以外に存在しなくなる。

④　上記③までで，権利管理対が生成され，許諾条件プログラムの先頭から実行が開始される。許諾条件プログラムの先頭部分には，利用者との会

話を行うプログラム，ソフトウェア本体の実行時間等の計量を開始するために，Ｐ割込み要求テーブルに割込み要求を書き込むプログラム等を置くことができる。その後，ソフトウェア本体に制御を移す命令を発行する。この命令では，Ｐビットのクリアが行われる。

以後，Ｐ割込みとＲＰＥ命令によって，ＵＳ状態とＰＭ状態とを間を行き来しながらプログラムの実行がなされる。権利管理対の実行を終了する場合，オペレーティング・システムは，消去したい権利管理対のＰＭＩＤをパラメータとするＰＭＴ消去命令を発行する。これにより，ハードウェアによって，権利管理テーブルの対応するレコードおよびＰＩＴの該当エントリが消去される。

権利管理に関する命令について，例えばいかなる命令を用意すればよいかについては，以上の実施例の説明で明らかであるが，各命令の機能変更，拡張は任意になし得る。

〔発明の効果〕

以上説明したように，本発明によれば，マルチプログラミングの環境において，保全性，柔軟性に優れた権利管理を行うことが可能になり，ソフトウェアを自由に大量に流通させ，かつソフトウェアの保護が充分であるシステムを構築できるようになる。

４．図面の簡単な説明

第１図は本発明の原理説明図，

第２図は本発明を用いたソフトウェアの流通形態の例を説明するための図，

第３図は本発明による権利管理の説明図，

第４図は権利管理状態の状態遷移図，

第５図は本発明に用いるハードウェア構成例，

第６図は権利管理対象となるソフトウェアの構成例，

第７図は権利管理テーブルの構成例，

第８図はＰ割込み要求テーブルの構成例，

第９図はＳ割込み制御の例，

第１０図はＳ割込みからの復帰制御の例，

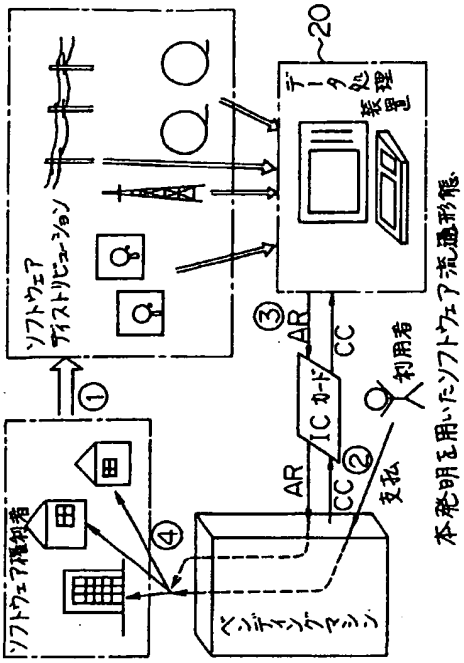
第１１図は本発明の一実施例におけるスタックの使用例，

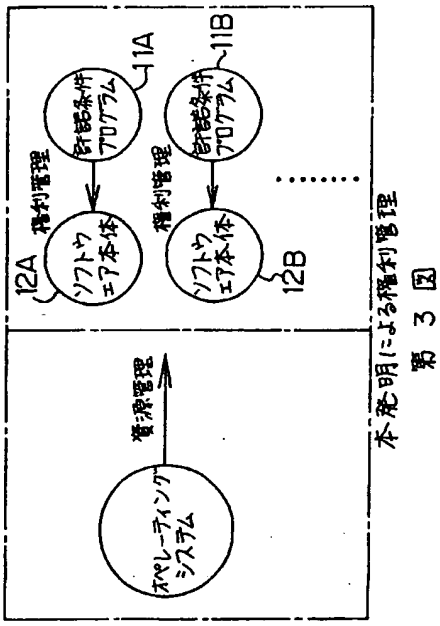
第１２図は本発明の一実施例における空間と鍵の関係説明図を示す。

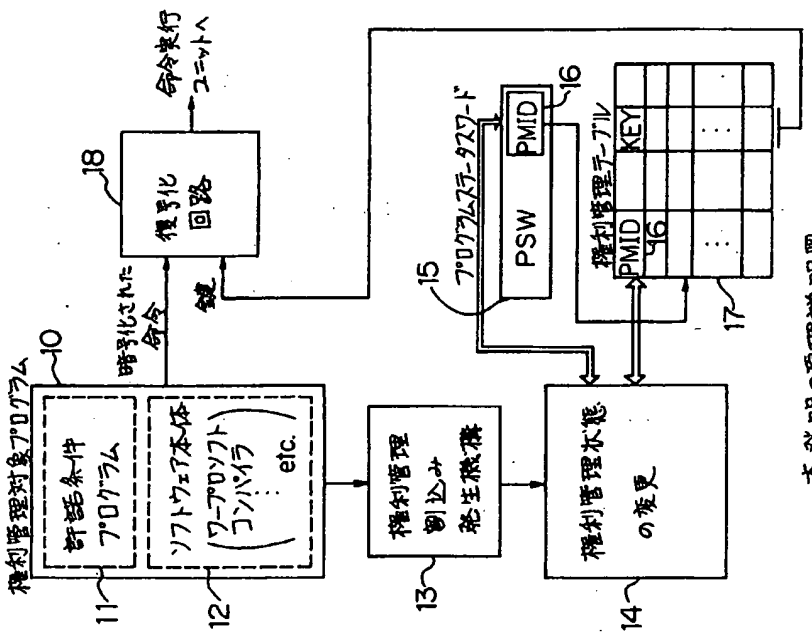
図中，１０は権利管理対象プログラム，１１は許諾条件プログラム，１２はソフトウェア本体，１３は権利管理割込み発生機構，１４は権利管理状態の変更処理，１５はプログラムステータスワード，１６は権利管理ＩＤ，１７は権利管理テーブル，１８は復号化回路を表す。

特許出願人　　森　　亮　一

代理人　弁理士　　小笠原　吉義（外２名）







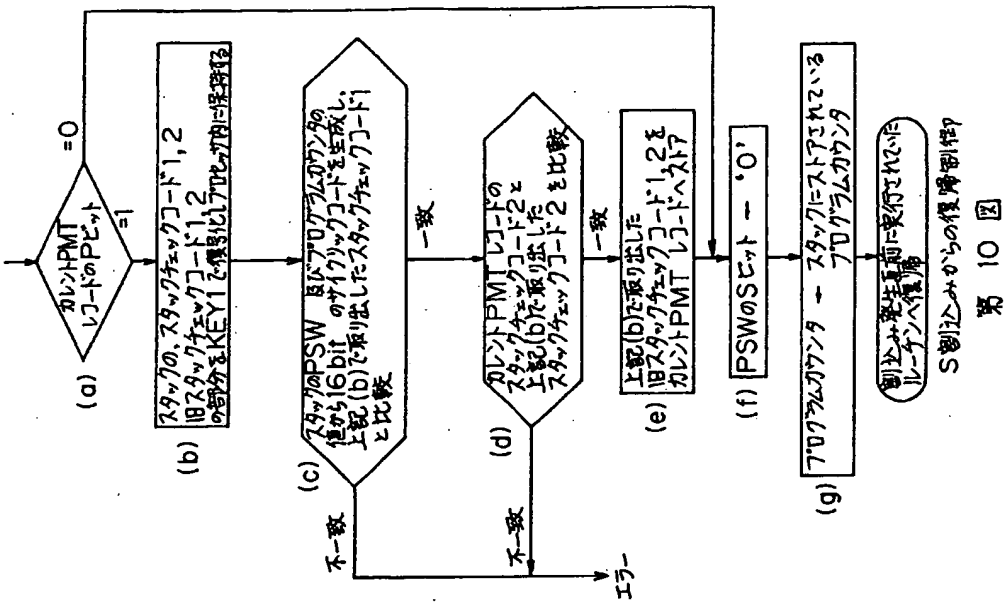
状態遷移図
第4図

本発明に用いるハードウェア構成例
第5図

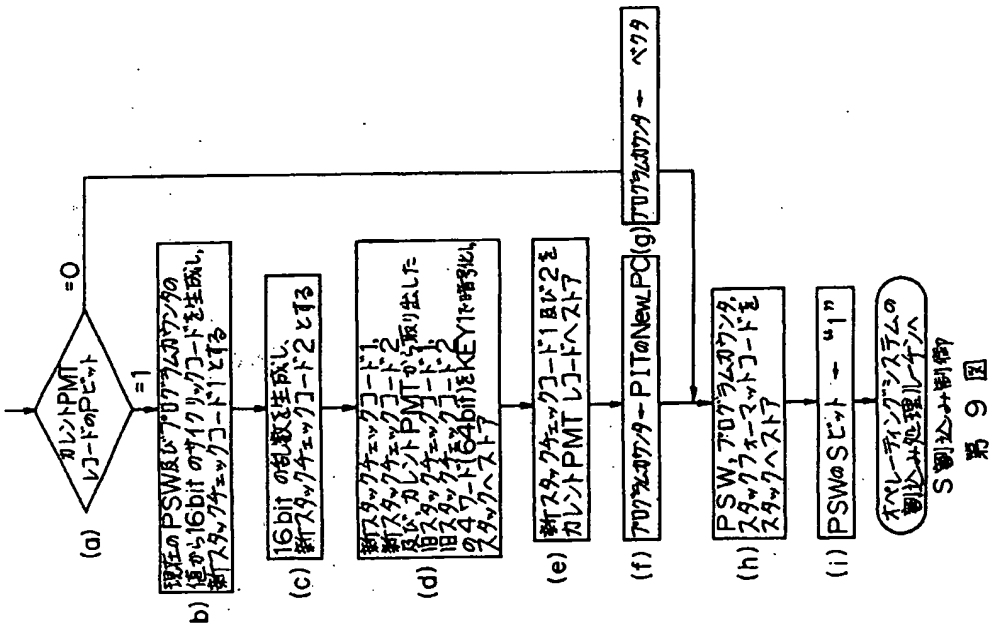
ソフトウェアの構成例
第6図

権利管理テーブルの構成例
第7図

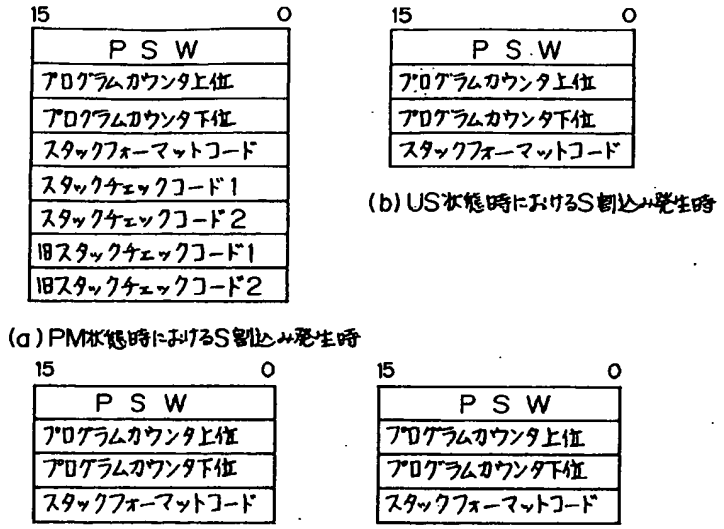
P割込み要求テーブルの構成例
第8図



第 10 図



第 9 図



スタックの使用例

第 11 図

	SV状態	PM状態	US状態
空間	スーパバイザ空間	ユーザ空間	
鍵	—	カレントPMTのKEY1	カレントPMTのKEY2

空間と鍵の関係

第 12 図

Requested Patent: WO9401821A1

Title:

SECURE COMPUTER NETWORK USING TRUSTED PATH SUBSYSTEM WHICH ENCRYPTS/DECRYPTS AND COMMUNICATES WITH USER THROUGH LOCAL WORKSTATION USER I/O DEVICES WITHOUT UTILIZING WORKSTATION PROCESSOR ;

Abstracted Patent: US5596718 ;

Publication Date: 1997-01-21 ;

Inventor(s):

BOEBERT WILLIAM E (US); HANSON MARK H (US); MARKHAM THOMAS R (US) ;

Applicant(s): SECURE COMPUTING CORP (US) ;

Application Number: US19920911900 19920710 ;

Priority Number(s): US19920911900 19920710 ;

IPC Classification: G06F15/17 ;

Equivalents: AU4672693, AU663406, EP0649546 (WO9401821), JP7509086T ;

ABSTRACT:

A method and apparatus for ensuring secure communication over an unsecured communications medium between a user working on an unsecured workstation or computer and a host computer. A secure user interface is created by inserting a trusted path subsystem between input/output devices to the workstation and the workstation itself. Data transferred from the input/output devices is intercepted, encrypted and transmitted in packets to the host computer. Packets of screen display data from the host computer are decrypted and presented within a user-defined screen overlay.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G06F 12/14, 1/00 — A1

(11) International Publication Number: WO 94/01821

(43) International Publication Date: 20 January 1994 (20.01.94)

(21) International Application Number: PCT/US93/06511

(22) International Filing Date: 9 July 1993 (09.07.93)

(30) Priority data:
07/911,900 10 July 1992 (10.07.92) US

(71) Applicant: SECURE COMPUTING CORPORATION [US/US]; 2675 Long Lake Road, Roseville, MN 55113-2536 (US).

(72) Inventors: BOEBERT, William, E. ; 4915 DuPont Avenue South, Minneapolis, MN 55409 (US). HANSON, Mark, H. ; 3560 Baltic Avenue, Eagan, MN 55122 (US). MARKHAM, Thomas, R. ; 709 River Lane, Anoka, MN 55303 (US).

(74) Agent: BRUESS, Steven, C.; Merchant, Gould, Smith, Edell, Welter & Schmidt, 3100 Norwest Center, 90 South Seventh Street, Minneapolis, MN 55402 (US).

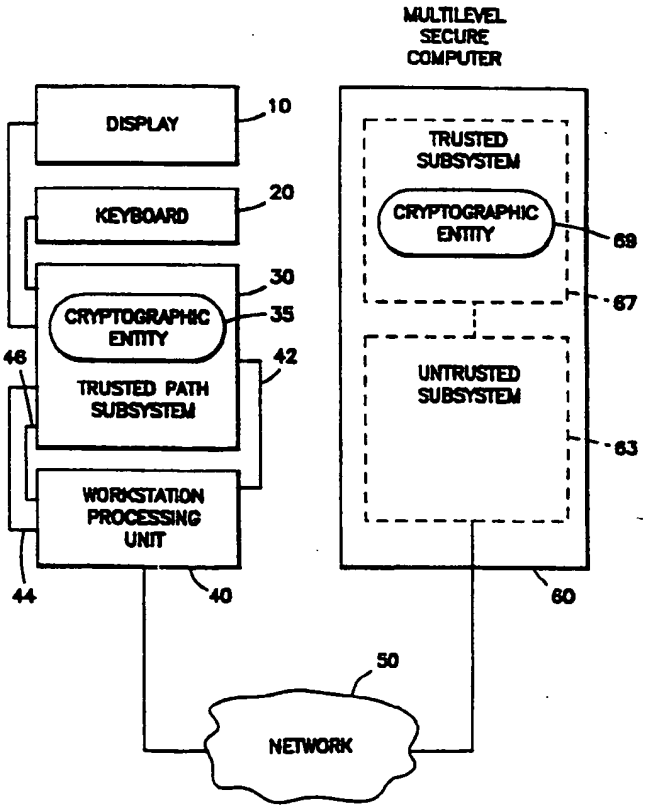
(81) Designated States: AT, AU, BB, BG, BR, BY, CA, CH, CZ, DE, DK, ES, FI, GB, HU, JP, KP, KR, KZ, LK, LU, MG, MN, MW, NL, NO, NZ, PL, PT, RO, RU, SD, SE, SK, UA, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: TRUSTED PATH SUBSYSTEM FOR WORKSTATIONS

(57) Abstract

A method and apparatus for ensuring secure communication over an unsecured communications medium between a user working on an unsecured workstation or computer and a host computer. A secure user interface is created by inserting a trusted path subsystem between input/output devices to the workstation and the workstation itself. Data transferred from the input/output devices is intercepted, encrypted and transmitted in packets to the host computer. Packets of screen display data from the host computer are decrypted and presented within a user-defined screen overlay.



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.

AT	Austria	FR	France	MR	Mauritania
AU	Australia	GA	Gabon	MW	Malawi
BB	Barbados	GB	United Kingdom	NE	Niger
BE	Belgium	GN	Guinea	NL	Netherlands
BF	Burkina Faso	GR	Greece	NO	Norway
BG	Bulgaria	HU	Hungary	NZ	New Zealand
BJ	Benin	IE	Ireland	PL	Poland
BR	Brazil	IT	Italy	PT	Portugal
BY	Belarus	JP	Japan	RO	Romania
CA	Canada	KP	Democratic People's Republic of Korea	RU	Russian Federation
CF	Central African Republic			SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovak Republic
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LU	Luxembourg	TD	Chad
CS	Czechoslovakia	LV	Latvia	TG	Togo
CZ	Czech Republic	MC	Monaco	UA	Ukraine
DE	Germany	MG	Madagascar	US	United States of America
DK	Denmark	ML	Mali	UZ	Uzbekistan
ES	Spain	MN	Mongolia	VN	Viet Nam
FI	Finland				

TRUSTED PATH SUBSYSTEM FOR WORKSTATIONS

Background of the Invention

Field of the Invention

The present invention relates to an apparatus and method for providing a trusted computer system based on untrusted computers, and more particularly to an apparatus and method for providing a trusted path mechanism between a user node based on an untrusted computer or workstation and a trusted subsystem.

Background Information

Advances in computer and communications technology have increased the free flow of information within networked computer systems. While a boon to many, such a free flow of information can be disastrous to those systems which process sensitive or classified information. In response to this threat, trusted computing systems have been proposed for limiting access to classified information to those who have a sufficient level of clearance. Such systems depend on identifying the user, authenticating (through password, biometrics, etc.) the user's identity and limiting that user's access to files to those files over which he or she has access rights. In addition, a trusted path mechanism is provided which guarantees that a communication path established between the Trusted Computer Base (TCB) and the user cannot be emulated or listened to by malicious hardware or software. Such a system is described in U.S. Patent Nos. 4,621,321; 4,713,753; and 4,701,840 granted to Boebert et al. and assigned to the present assignee, the entire disclosures of which are hereby incorporated by reference.

The last decade has marked a shift in the distributing of computational resources. Instead of connecting a large number of relatively "dumb" terminals to a mainframe computer, the automatic data processing

environment has gradually shifted to where a large number of current systems are file server systems. In a file server system, relatively low cost computers are placed at each user's desk while printers and high capacity data storage devices are located near the server or servers. Files stored in the high capacity data storage devices are transferred to the user's computer for processing and then either saved in local storage or transferred back to the storage devices. Documents to be printed are transferred as files to a print server; the print server then manages the printing of the document.

An even more loosely coupled distributed computing approach is based on the client-server paradigm. Under the client-server paradigm, one or more client processes operating on a user's workstation gain access to one or more server processes operating on the network. As in file server systems, the client processes handle the user interface while the server processes handle storage and printing of files. In contrast with file server systems, however, the client processes and the server processes share data processing responsibilities. A more complete discussion of distributed computing is contained in "Client-Server Computing" by Alok Sinha, published in the July 1992 issue of *Communications of the ACM*.

Both the file server and the client-server paradigms depend heavily upon the availability of low-cost computer systems which can be placed at each user's desk. The low-cost systems are then connected through a network such as a LAN or a WAN to the server systems. Such a networked system is illustrated in the block diagram shown in Fig. 1.

In Fig. 1, a workstation processing unit 40 is connected through a network 50 to a host computer 60. Workstation unit 40 is also connected through video port

44 and keyboard port 46 to display unit 10 and keyboard 20, respectively.

In a typical distributed computer system, the workstations 40, the host computers 60 and the connecting networks 50 are all at great risk of a security breach. Trusted computer systems based on host computers such as the Multilevel Secure (MLS) Computer 60 shown in Fig. 1 make security breaches at the host computer more difficult by partitioning the system to isolate security critical (trusted) subsystems from nonsecurity critical (untrusted) subsystems. Such computers do little, however, to prevent security breaches on network 50 or at user workstation 40.

A Multi-Level Secure (MLS) Computer such as is shown in Fig. 1 is capable of recognizing data of varying sensitivity and users of varying authorizations and ensuring that users gain access to only that data to which they are authorized. For example, an MLS computer can recognize the difference between company proprietary and public data. It can also distinguish between users who are company employees and those who are customers. The MLS computer can therefore be used to ensure that company proprietary data is available only to users who are company employees.

Designers of MLS computers assume that unauthorized individuals will use a variety of means, such as malicious code and active and passive wiretaps, to circumvent its controls. The trusted subsystem of an MLS computer must therefore be designed to withstand malicious software executing on the untrusted subsystem, to confine the actions of malicious software and render them harmless. One mechanism for avoiding malicious software is to invoke a trusted path, a secure communications path between the user and the trusted subsystem. A properly designed trusted path ensures that information viewed or sent to the trusted subsystem is not copied or modified along the way.

Extension of the trusted path through the network to the user is, however, difficult. As is described in a previously filed, commonly owned U.S. patent application entitled "Secure Computer Interface" (U.S. Patent Application No. 07/676,885 filed March 28, 1991 by William E. Boebert), "active" and "passive" network attacks can be used to breach network security. Active attacks are those in which masquerading "imposter" hardware or software is inserted into the network communications link. For example, hardware might be inserted that emulates a user with extensive access privileges in order to access sensitive information. "Passive" network attacks include those in which a device listens to data on the link, copies that data and sends it to another user. A system for ensuring secure data communications over an unsecured network is described in the above-identified patent application. That application is hereby incorporated by reference.

Active and passive attacks can also be used to breach computer security through software running on an untrusted user computer, an untrusted host or in the untrusted subsystem of a Multilevel Secure Computer. For example, malicious software running in the workstation could present itself to an authorized user as the trusted subsystem, and cause that user to enter highly sensitive data, such as a password. The data is then captured and given to the attacker. Under a passive software attack, data which is intended for one user could be copied and sent to a user who is not authorized to work with it.

Systems for ensuring secure communications over an unsecured network have been limited to date to scrambling devices which encrypt data written to the network and decrypt data received from the network. Such systems are limited in that they provide no assurance that the user's computer is secure or that the user has, in fact, established a trusted path to the

trusted subsystem. Therefore, despite the fact that the communications link is secure, it is possible for a user on the computer to be misled into believing that a program executing on his computer is actually running on the host computer.

What is needed is a mechanism for extending the trusted path from the trusted subsystem of the host computer to the user of an untrusted computer or workstation. Such a method should provide access to the workstation for normal workstation activities while shielding confidential data so that it cannot be read by software executing on the unsecured workstation.

Summary of the Invention

The present invention provides a method and apparatus for ensuring secure communication over an unsecured communications medium between a user working on an unsecured workstation or computer and a host computer. A secure user interface is created by inserting a trusted path subsystem between input/output devices to the workstation and the workstation itself. Data transferred from the input/output devices is intercepted, encrypted and transmitted in packets to the host computer. Packets of screen display data from the host computer are decrypted and presented within a user-defined screen overlay.

According to another aspect of the present invention, a method is disclosed for ensuring secure file transfers between an unsecured workstation and a host computer. A file to be transferred is downloaded to a trusted path subsystem inserted between the workstation and its keyboard and display device. The trusted path subsystem presents a representation of the file on the display device where the user can verify that the file is as expected. The verified file is then encrypted and transferred as packets to the host computer.

Brief Description of the Drawings

FIG. 1 is a system level block diagram representation of a networked computer system.

FIG. 2 is a system level block diagram representation of a secure networked computer system according to the present invention.

FIG. 3 is a block diagram representation of a user node including a trusted path subsystem according to the present invention.

FIG. 4 is a block diagram representation of a user node including a different embodiment of a trusted path subsystem according to the present invention.

FIG. 5 is an electrical block diagram representation of one embodiment of the trusted path subsystem according to the present invention.

FIG. 6 is a representation of a secure window overlay according to the present invention.

Detailed Description of the Preferred Embodiments

In the following Detailed Description of the Preferred Embodiments, reference is made to the accompanying Drawings which form a part hereof, and in which are shown by way of illustration specific embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

The present invention provides a method and apparatus for ensuring secure communication over an unsecured communications medium between a user working on an unsecured workstation or computer and a host

computer. A secure user interface is created by inserting a trusted path subsystem between input/output devices to the workstation and the workstation itself. Data transferred from the input/output devices is intercepted, encrypted and transmitted in packets through the workstation to the host computer. Packets of screen display data from the host computer are decrypted and presented within a user-defined screen overlay.

Cryptographic entities in the trusted path subsystem and the host computer apply end-to-end encryption to confidential data transferred to and from the network. End-to-end encryption is a technique whereby data is encrypted as close to its source as possible and decrypted only at its ultimate destination. This technique differs from link encryption, in which data is decrypted, then encrypted again as it moves from the sender to the receiver.

The present invention extends the notion of end-to-end encryption by performing the encryption/decryption closer to the originator and receiver than prior systems. In the present invention, the encryption/decryption is performed as the data enters and leaves the input/output device. The data is therefore protected from malicious software which might be operating on the workstation and from active or passive attacks on the network.

A secure networked computer system constructed according to the present invention is illustrated generally in Fig. 2. In Fig. 2, a workstation processing unit 40 is connected through a network 50 to a host computer 60. Workstation 40 can be any computer, workstation or X terminal which has a separate data path for communication between a trusted path subsystem 30 and the workstation. For instance, workstation 40 can be a commercially available workstation such as the UNIX workstations manufactured by Sun Microsystems, Mountain

View, California, an IBM PC compatible such as those available from Compaq, Houston, Texas or an X terminal such as Model NCD19g from Network Computing Devices, Inc, Mountain View, California.

Trusted path subsystem 30 is connected to workstation 40 (through auxiliary data port 42), keyboard 20 and display 10. Trusted path subsystem 30 includes cryptographic entity 35 for encrypting and decrypting information transferred between display 10, keyboard 20 and workstation 40.

Host computer 60 is a Multi-Level Secure computer which includes a trusted subsystem 67 and an untrusted subsystem 63. Trusted subsystem 67 includes a cryptographic entity 69 for encrypting and decrypting data transferred between trusted subsystem 67, untrusted subsystem 63, and network 50. In another embodiment of the present invention, host computer 60 is a computer running a trusted subsystem software package. In that embodiment, cryptographic entity 69 would be implemented in software.

In the embodiment shown in Fig. 2, all communication between trusted path subsystem 30 and host computer 60 is done via workstation 40. In one such embodiment, auxiliary data port 42 is an RS-232 line connecting workstation 40 and subsystem 30. Communications software running on workstation 40 receives encrypted packets from the trusted path subsystem and sends them to the host computer. In a like manner, encrypted packets from host computer 60 are received by workstation 40 and transferred to subsystem 30 for decrypting. This type of interface is advantageous since a standard communications protocol can be defined for transfers between subsystem 30 and host computer 60. Workstation 40 then implements the standard protocol for the communications media connecting it to host computer 60.

Network 50 can be implemented in a wide range of communications protocols, from FDDI to a simple telecommunications line between two modems. In a network implementation, subsystem 30 provides only the encrypted file; workstation 40 provides the layers of protocol needed for reliable communication on network 50.

Fig. 3 provides more detail of trusted path subsystem 30. Trusted path subsystem 30 consists of a processor 31 connected to a keyboard manager 37, a video manager 38 and cryptographic entity 35. Trusted path subsystem 30 operates in normal mode and in trusted path mode. When in normal mode, workstation trusted path subsystem 30 is transparent to workstation 40. Logical switches 37 and 38 are in the UP position, connecting workstation processor 40 directly to keyboard 20 and display 10. This permits the free transfer of information from keyboard 20 to workstation 40 and from workstation 40 to display 10. In normal mode, workstation processor 40 runs software and communicates with host computer 60 via network 50.

When the user invokes trusted path mode, however, workstation processor 40 is disconnected from keyboard 20 and display 10 by logical switches 37 and 38, respectively. Keyboard 20 and display 10 are then connected to their respective managers in workstation trusted path subsystem 30.

As is shown in Fig. 6, while in trusted path mode, video manager 34 creates a trusted window 82 which is overlaid on the screen display 80 generated by workstation 40 for display 10. Since window 82 is created outside of workstation 40, by trusted elements, it is not possible for malicious software in workstation 40 to control any of the video in trusted window 82. In the preferred embodiment the size of trusted window 82 can vary; if sufficient video RAM is present, window 82 may be as large as the entire display screen.

In a like manner, while in trusted path mode, keyboard manager 36 intercepts keyboard data intended for workstation 40. The data is then routed to cryptographic entity 35, where it is encrypted before being passed over auxiliary port 42 to workstation processing unit 40. Thus, keyboard inputs are protected from eavesdropping and undetected modification until they are decrypted by cryptographic entity 69 on host computer 60.

In one embodiment of the trusted path subsystem of Fig. 3, cryptographic entity 35 uses a pair-wise key to encrypt data to be transmitted from keyboard 20 to host computer 60. At the same time, cryptographic entity 35 decrypts data transmitted from host computer 60 to display 10. The encryption and integrity mechanisms protect the data from eavesdropping and undetected modification as it is passed through workstation processor 40, network 50 and host computer untrusted subsystem 63. Other types of symmetric encryption algorithms such as the Data Encryption Standard (DES) and asymmetric cryptographic techniques such as public key can also be used. Furthermore, the encryption algorithm can either be implemented in software, programmable hardware, or custom hardware.

Trusted path mode can be invoked in a number of ways. In one embodiment, a switch on trusted path subsystem 30 can be used to manually activate trusted path mode. A second method would be to invoke trusted path mode by a combination of keys pressed simultaneously on keyboard 20 (like the control/alt/delete key sequence on a PC-compatible computer). A third embodiment would require that the user insert some sort of token device into subsystem 30. A token device might range from a smart card to a cryptoignition key. In the preferred embodiment, subsystem 30 would also have a feedback mechanism such

as a light to notify the user that subsystem 30 was in trusted path mode.

The trusted path mode, used in conjunction with cryptographic entity 69 on host computer 60, provides security services such as user authentication, data confidentiality, data integrity and data origin authentication and confinement of malicious software. The user is authenticated to trusted path subsystem 30 and this authentication is securely passed to trusted subsystem 67 in MLS computer 60. Data passed between cryptographic entities 35 and 69 is protected from unauthorized disclosure and undetected modification. Cryptographic entities 35 and 69 also assure that the data was sent from one cryptographic entity to its peer cryptographic device. In addition, malicious software on workstation 40, network 50 or untrusted subsystem 63 is confined so that it cannot dupe the user or trusted subsystem 67 into performing an insecure action.

The user can be authenticated to the trusted computing system by either authenticating himself directly to trusted path subsystem 30 or by going through subsystem 30 to host computer 60. In the first method, the user can authenticate himself to subsystem 30 via such means as a personal identification number (PIN), a password, biometrics or a token device such as a smart card or a cryptographic ignition key. Once the user has authenticated himself to subsystem 30, subsystem 30 relays the authentication to trusted subsystem 65. The step of relaying authentication can be done by either automatically entering trusted path mode as part of the authentication process or by having subsystem 30 relay the authentication data at a later time.

A second method for authenticating a user would be to first enter trusted path mode and then authenticate the user directly to host computer 60.

This approach would reduce the processing power needed on subsystem 30.

In its simplest form, trusted path subsystem 30, in conjunction with workstation 40, display 10 and keyboard 20, forms an assured terminal. Data typed on keyboard 20 or extracted from a pointing device such as a mouse is encrypted and transferred over network 50 to host computer 60. Screen display data transferred from host computer 60 is decrypted and displayed within trusted window 82. Such a terminal might be implemented as a relatively dumb terminal such as a VT100, or it could be implemented as a X Windows terminal. The X Window embodiment would be useful since it would allow the creation of multiple trusted windows 82 and would permit the assigning of a different security level to each window. Such a mechanism would permit qualified users to cut information from a document of one sensitivity and paste it into a document of a different sensitivity.

An assured terminal is especially useful in an environment where you are trying to maintain a number of security levels despite having a workstation which will only operate at one level. An example is a trusted computing system mixing single level secure workstations with a multi-level computer with three security levels: unclassified (least sensitive), secret (much more sensitive), and top secret (most sensitive). Trusted path subsystem 30 can be used to expand the capabilities of the single level workstation since subsystem 30 allows the user to essentially disable subsystem 30, do all his work at the level permitted by the workstation (say, secret) using all the capabilities of his workstation and whatever facilities are available on the multilevel computer. Then, if the user has a small amount of work that he or she needs to do at top secret, the user can invoke trusted mode in subsystem 30, isolate their workstation, its processor memory and

storage devices, and he has, in effect, a keyboard and a terminal connected to a secure communications device through a multilevel host. The user can then do the operations required at top secret.

The cryptographic techniques applied in subsystem 30 will ensure that none of the top secret information going to or from the multilevel secure computer is linked to files within workstation 40 or is captured and copied on the network.

Likewise, if a user had to do a small amount of unclassified work, he could put the workstation into trusted path mode using subsystem 30. The user could, through a trusted path, invoke an unclassified level and again the cryptographic techniques applied at each end of the link would prevent secret information from being mixed in with the unclassified information. The system essentially provides a pipe to keep data from one security level from being mixed into data at a different security level.

Trusted subsystem 30 is not, however, limited to a role as an assured terminal. In a file server application, files stored at host computer 60 or within workstation 40 could be transferred to subsystem 30 for data processing tasks such as editing, reviewing the file or transferring it as electronic mail. In a client server application, processor 31 could execute one or more client processes such as an editor or a communications process. Software and firmware which could be implemented inside trusted path subsystem 30 would be limited only by the amount of storage within subsystem 30 and the review and approval process required to provide clean software.

Trusted path subsystem 30 has access not only to files on host computer 60 but also on workstation 40. Files transferred from either computer 60 or workstation 40 can be manipulated and transferred to other computers or workstations. For example, a secure electronic mail

system could be implemented in which trusted path subsystem 30 is used for reviewing, reclassifying, and electronically signing messages. A document file from computer 60 or workstation 40 can be displayed and reviewed. If appropriate, the user may downgrade its sensitivity level by attaching a different security level to the document. The finished file can then be sent via electronic mail to other users.

In one embodiment of such an electronic mail function, subsystem 30 would go out on the network to the directory server to retrieve the names, electronic mail addresses and public key information of the intended recipients. The directory server could be implemented as either a trusted or an untrusted process on host computer 60 or on another network computer. Subsystem 30 would then attach the addresses to the file, affix a digital signature, encrypt the final product and send it through host computer 60 to the designated addresses.

In another embodiment of such a function, in a system without a MLS computer, secure electronic mail is possible by first establishing a trusted path from the user to processor 31. The user then accesses files of workstation 40 (or on other network computers), displays and reviews the file, accesses an unsecured directory server to retrieve the names, electronic mail addresses and public key information and sends the encrypted message via electronic mail to its recipient.

Processor 31 can also be used to control video manager 34 in order to implement and control the user interface. Such an approach would permit the use of a graphical user interface (GUI) within trusted window 82 that would reduce the amount of screen information transferred by host computer 60. This approach also permits the user to implement, through processor 31, multiple trusted windows 82 at the user node in order to perform the cut-and-paste function referred to above.

In the preferred embodiment, subsystem 30 is a modular design in which processor 31 and cryptographic entity 35 are kept constant and video manager 34 and keyboard manager 36 are designed so that they can be replaced easily to handle different displays and keyboards. In one embodiment, subsystem 30 is designed to be portable. A portable subsystem 30 can be used to turn any modem equipped computer with the requisite auxiliary data port into a secure data terminal or computer.

Fig. 4 is a block diagram representation of an alternate embodiment of trusted path subsystem 30. In Fig. 4, processor 31 is connected through network interface 39 to network 50 and through communication port 48 to workstation 40. In the embodiment shown in Fig. 4, workstation processing unit 40 is isolated from the network. This approach allows the encryption of all network traffic associated with the user node. In the embodiment shown in Fig. 4, communication port 48 can be a communication medium ranging from RS0232 to an unsecured Ethernet.

A more detailed representation of one embodiment of trusted path subsystem 30 is shown in Fig. 5. In Fig. 5, keyboard logical switch 37 receives data from keyboard 20 and routes it to processor 31. During normal mode, processor 31 then sends the received keyboard data directly over keyboard port 46 to workstation 40.

In contrast, in trusted path mode, processor 31 captures the received keyboard data and sends it to cryptographic entity 35 for encrypting. No information is sent over keyboard port 46 to workstation 40. The resulting encrypted keyboard data is instead sent through auxiliary data port 42 to workstation 40 and from there to computer 60.

Video data from workstation 40 is transmitted from video port 44 to video manager 34. During normal

mode, the video data is sent through to display 10 without modification. During trusted path mode, however, the video data transferred from video port 44 is overlaid, at least in some part, by video data generated by video manager 34.

A representative video manager 34 is shown generally in Fig. 5. Video manager 34 consists of video synchronization hardware 72, video RAM 74, video driver 78 and video multiplexer 76. Video synchronization hardware 72 receives synchronization signals from video port 44 and uses the signals to coordinate the display of data from video RAM 74 with the display generated by workstation 40. During normal mode data from video RAM 74 is not used; video is transferred directly from workstation 40 through video multiplexer 76 to display 10. When, however, trusted path subsystem 30 is placed into trusted path mode, video data stored in video RAM 74 is used instead of the normal video stream to create trusted window 82.

In one embodiment synchronization hardware 72 uses the synchronization signals received from workstation 40 to control the reading of data from video RAM 74 and the conversion of that data into a video signal by video driver 78. The output of video driver 78 is then used to drive video multiplexer 76. Synchronization hardware 72 controls video multiplexer 76 in order to switch between the video generated by workstation 40 and the video being read from video RAM 74. The output of video multiplexer 76 is driven through video amplifiers to display 10.

The design of the video hardware needed to overlay one display on top of another is well known in the art. Window 82 can be synched up to the video going to display 10. Typically, if window 82 is not full screen, video synchronization hardware 72 counts the number of lines to the first line of window 82, counts in the number of pixels, and inserts the video at that

point. Trusted path video data is then written for the desired number of pixels and video multiplexer 76 is switched back to normal video for the remainder of the video line. This mechanism provides flexibility in placement and sizing of window 82 on screen 80.

Video multiplexer 76 can be built using a crosspoint video switch such as the MAX456 manufactured by Maxim Integrated Products. Video data to and from the crosspoint video switch can be buffered using the MAX457 by Maxim Integrated Products. Video RAM 74 can be any commercial video RAM. A typical video RAM is the MT42C8256 manufactured by Micron Technologies Inc. It should be obvious that the given design can be easily adapted for either a color or a black and white display or even for a black and white overlay of a color display.

In one embodiment, host computer 60 transmits, as encrypted packets, video data to be displayed within trusted window 82. The encrypted packets are passed to processor 31 by workstation 40 and then on to encryption device 35. Encryption entity 35 decrypts the video data and places it into video RAM 74. Synchronization hardware 72 then activates video multiplexer 76 and video RAM 74 in order to display the decrypted secure video data.

In a second embodiment (not shown), processor 31 creates the video overlay data and writes that data to video RAM 74. Display of the data is as above.

A trusted computing system based on unsecured, commercially available, workstations, trusted path subsystems and multilevel secure computers provides a powerful, highly secure computing environment. The ability of such a system to compensate for unsecured workstations allows the designers of such systems to use the latest versions of commercially available hardware and software without compromising the security of the system.

For instance, a user of a workstation may wish to edit a secret document and reclassify the edited document as unclassified. The document can be loaded into the workstation, edited with the user's favorite word processing software package, and saved. Then, in order to classify the document as unclassified, the user would invoke trusted path mode, the trusted window would be displayed and the user could review the revised document to verify that no additional information had been attached to the file. The reviewed document could then be released as an unclassified document and the user would then returns to normal mode.

The unique placement of cryptographic entity 35 relative to workstation 40 allows a single workstation to be used at different levels of security sensitivity. Therefore, instead of systems in which a workstation is required for each level of security sensitivity, in the present system a single commercial workstation may be used to protect and access a range of security levels.

Finally, the end-to-end characteristic of the encryption permits secure communication without the need to perform costly analysis of complex elements such as network controllers. The invention also allows use of commercial off-the-shelf workstations and network components and can be used with a variety of keyboards and displays.

Although the present invention has been described with reference to the preferred embodiments, those skilled in the art will recognize that changes may be made in form and detail without departing from the spirit and scope of the invention.

What is claimed is:

1. A secure computing network, comprising:

a network computer, wherein the computer comprises

a trusted subsystem; and

encryption means for encrypting and decrypting data transferred to and from the trusted subsystem;

communications means, connected to the network computer, for permitting data transfer between the network computer and other computers;

an input/output device;

a workstation comprising:

first communications interface means, connected to the communications means, for transferring data between the workstation and the network computer;

input/output device interface means for transferring data between the workstation and the input/output device; and

second communications means for transferring data between the workstation and another processor; and

trusted path means, inserted between the input/output device and the input/output device interface means and connected to the second communications means, for intercepting data transfers between the input/output device interface means and the input/output device, wherein the trusted path means comprises encryption means for encrypting and decrypting the data transfers and for routing such transfers over the second communications means to the trusted subsystem.

2. The secure computing network of claim 1 wherein the network computer is a multilevel secure computer capable of recognizing data of varying sensitivity and users of varying authorizations.

3. The secure computing network of claim 1 wherein the input/output device comprises a keyboard.

4. The secure computing network of claim 1 wherein the input/output device comprises a display device.

5. The secure computing network of claim 1 wherein the input/output device comprises a pointing device.

6. A secure computing network, comprising:

a network computer, wherein the computer comprises

a trusted subsystem; and

encryption means for encrypting and decrypting data transferred to and from the trusted subsystem;

communications means, connected to the network computer, for permitting data transfer between the network computer and other computers;

an input/output device;

a workstation comprising:

input/output device interface means for transferring data between the workstation and the input/output device; and

workstation communications means for transferring data between the workstation and another processor; and

trusted path means, inserted between the input/output device and the input/output device interface means and connected to the workstation communications means, for intercepting data transfers between the input/output device interface means and the input/output device, wherein the trusted path means comprises encryption means for encrypting and decrypting the data transfers and network interface means, connected to the communication means, for transferring

the encrypted data transfers between the trusted path means and the trusted subsystem.

7. The secure computing network of claim 6 wherein the network computer is a multilevel secure computer capable of recognizing data of varying sensitivity and users of varying authorizations.

8. The secure computing network of claim 6 wherein the input/output device comprises a keyboard.

9. The secure computing network of claim 6 wherein the input/output device comprises a display device.

10. The secure computing network of claim 6 wherein the input/output device comprises a pointing device.

11. A trusted path subsystem capable of being connected between an input/output device and a processor of a workstation in order to provide secure communication with a multilevel secure computer network server, the subsystem comprising:

input/output manager means for selectively intercepting, under user control, data transferred from the input/output device to the processor and from the processor to the input/output device;

encryption means for encrypting the intercepted data before transferring the encrypted data to the processor; and

decryption means for decrypting the intercepted data before transferring the decrypted data to the input/output device.

12. The trusted path subsystem according to claim 11 wherein the input/output manager means comprises keyboard manager logic, wherein the keyboard manager logic comprises:

a keyboard interface which captures information generated by a keyboard; and

processing means for transferring the captured information to a workstation processor, wherein the processing means transfers the captured information on a first path when in a first mode and on a second path when in a second mode.

13. The trusted path subsystem according to claim 11 wherein the input/output manager means comprises a video manager which can be used to generate a trusted window overlay on a video screen, wherein the video manager comprises:

a video multiplexer having first and second input ports and an output port, wherein the first input port can be connected to an external video signal and wherein the output port can be connected to a video display;

a video data memory;

converter means, connected to the video data memory and the second multiplexer input port, for converting data read from the video data memory into a trusted video signal representative of that data and for applying the trusted video signal to the second video multiplexer input port; and

video synchronization means, connected to the video data memory and the video multiplexer, for controlling the video data memory and the video multiplexer so as to insert the trusted video signal into the video signal generated at the video multiplexer output port.

14. A method of securely transferring data in a network comprising an unsecured workstation connected to a multilevel secure computer server, wherein the workstation comprises a processor and an input/output device and wherein the multilevel secure server comprises a trusted subsystem and encryption means for encrypting and decrypting data transferred to and from

the trusted subsystem, the method comprising the steps of:

providing trusted path means for providing a user selectable secure communications path between the input/output device and the trusted subsystem; and

inserting the trusted path means between the input/output device and the processor.

15. A method for providing secure file transfer capability on an unsecured workstation connected over a network to a second computer, wherein the workstation comprises a workstation processor and an input/output device and wherein the second computer comprises a trusted subsystem and encryption means for encrypting and decrypting data transferred to and from the trusted subsystem, the method comprising the steps of:

providing means for creating a trusted path between the input/output device and a trusted subsystem, said trusted path means including a trusted processor capable of executing a secure electronic mail program;

inserting the trusted path means between the input/output device and the workstation processor;

downloading from the workstation processor to the trusted processor a file to be transferred to the second computer;

displaying, on the input/output device, a representation of the file to be transferred;

if the file is as expected, transferring the file to the second computer; and

if the file is not as expected, generating an error message.

16. The method according to claim 15 wherein the step of generating an error includes allowing secured processing on the file.

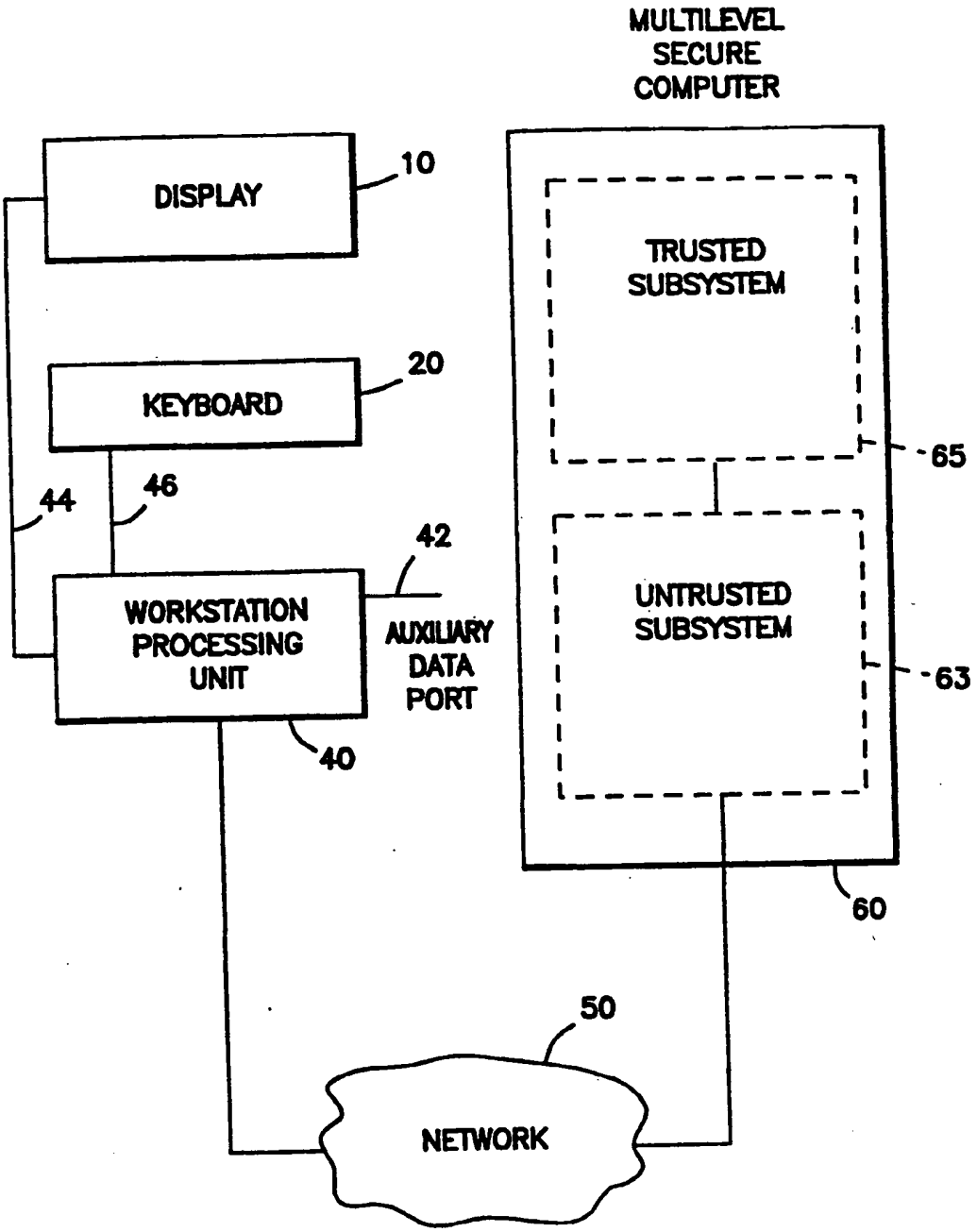


FIG. 1
PRIOR ART

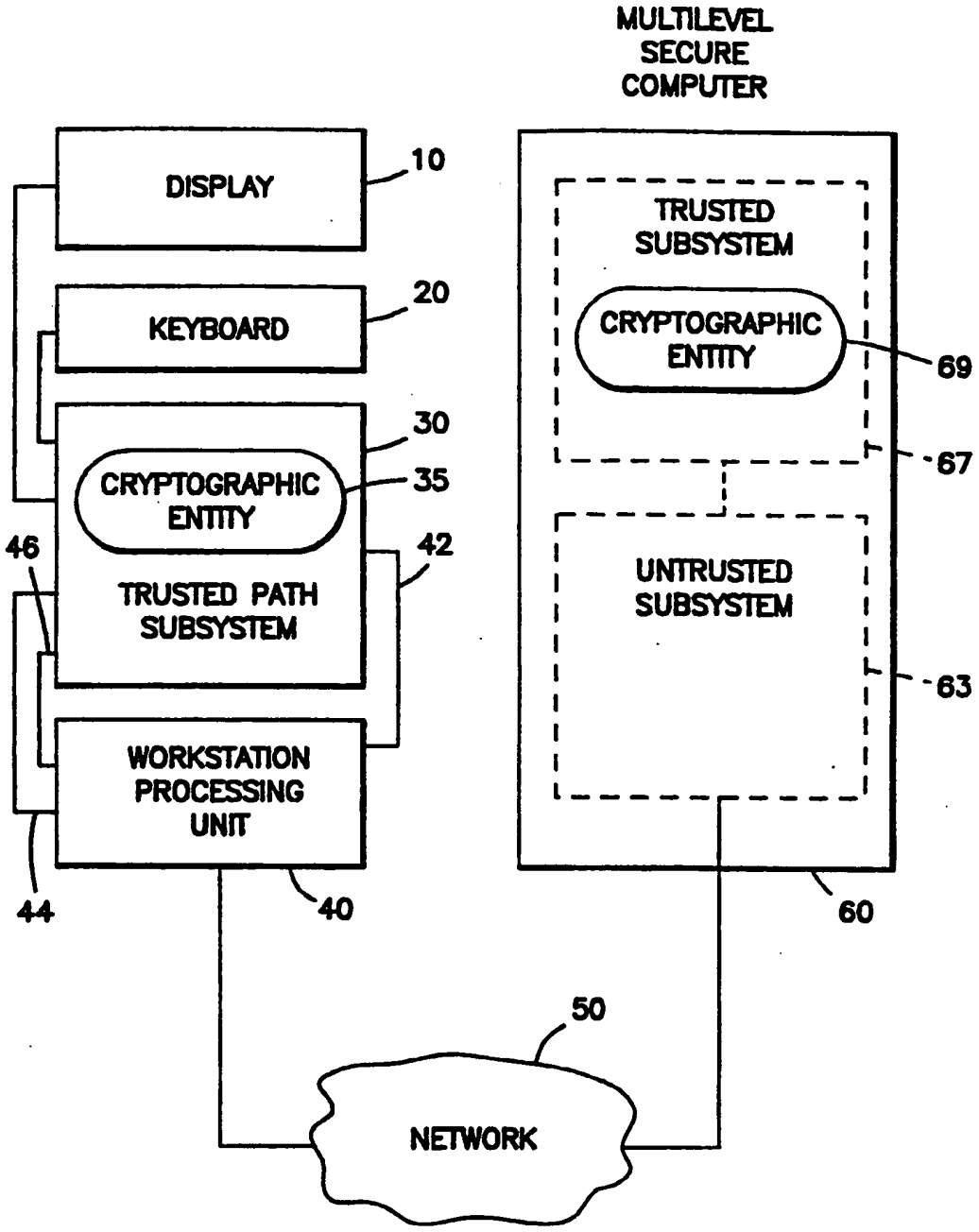

MULTILEVEL SECURE COMPUTER
DISPLAY
10
KEYBOARD
20
CRYPTOGRAPHIC ENTITY
30
35
TRUSTED PATH SUBSYSTEM
42
46
WORKSTATION PROCESSING UNIT
44
40
TRUSTED SUBSYSTEM
CRYPTOGRAPHIC ENTITY
69
67
UNTRUSTED SUBSYSTEM
63
60
50
NETWORK

FIG. 2

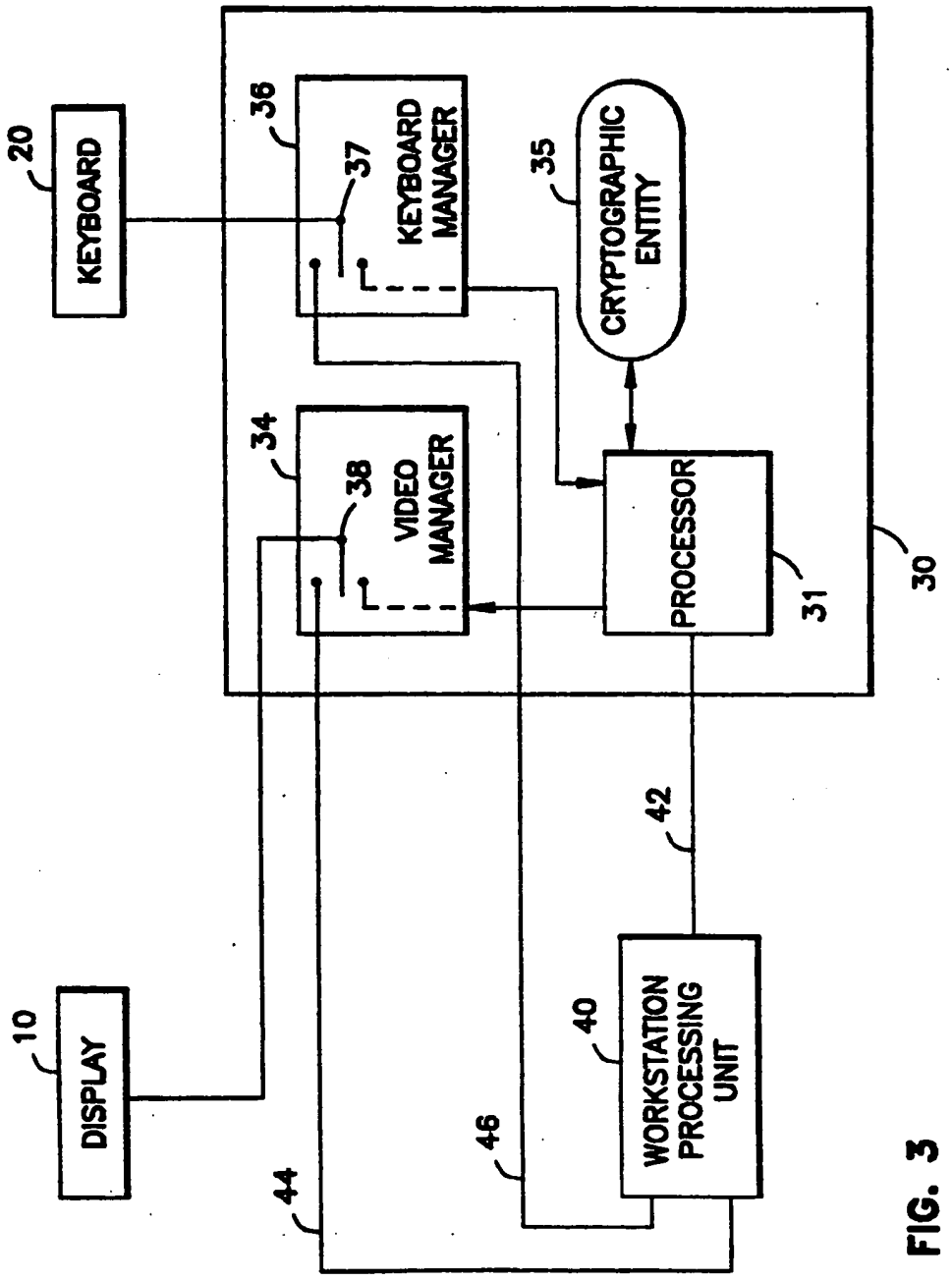

20
KEYBOARD
36
37
KEYBOARD MANAGER
35
CRYPTOGRAPHIC ENTITY
34
38
VIDEO MANAGER
PROCESSOR
31
30
42
10
DISPLAY
40
WORKSTATION PROCESSING UNIT
46
44

FIG. 3

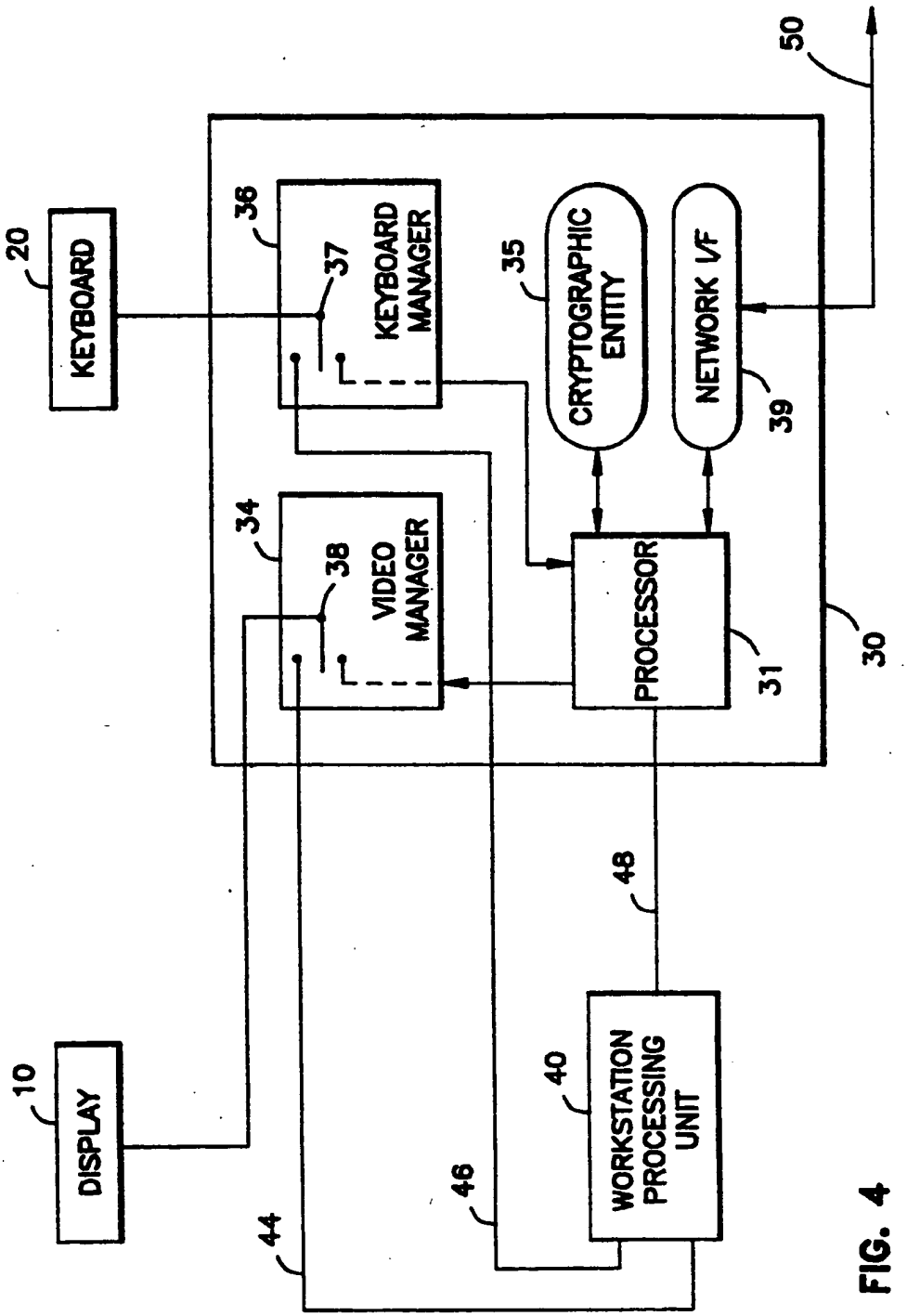

50
20
KEYBOARD
36
37
KEYBOARD MANAGER
35
CRYPTOGRAPHIC ENTITY
NETWORK I/F
39
34
38
VIDEO MANAGER
PROCESSOR
31
30
48
10
DISPLAY
40
WORKSTATION PROCESSING UNIT
46
44

FIG. 4

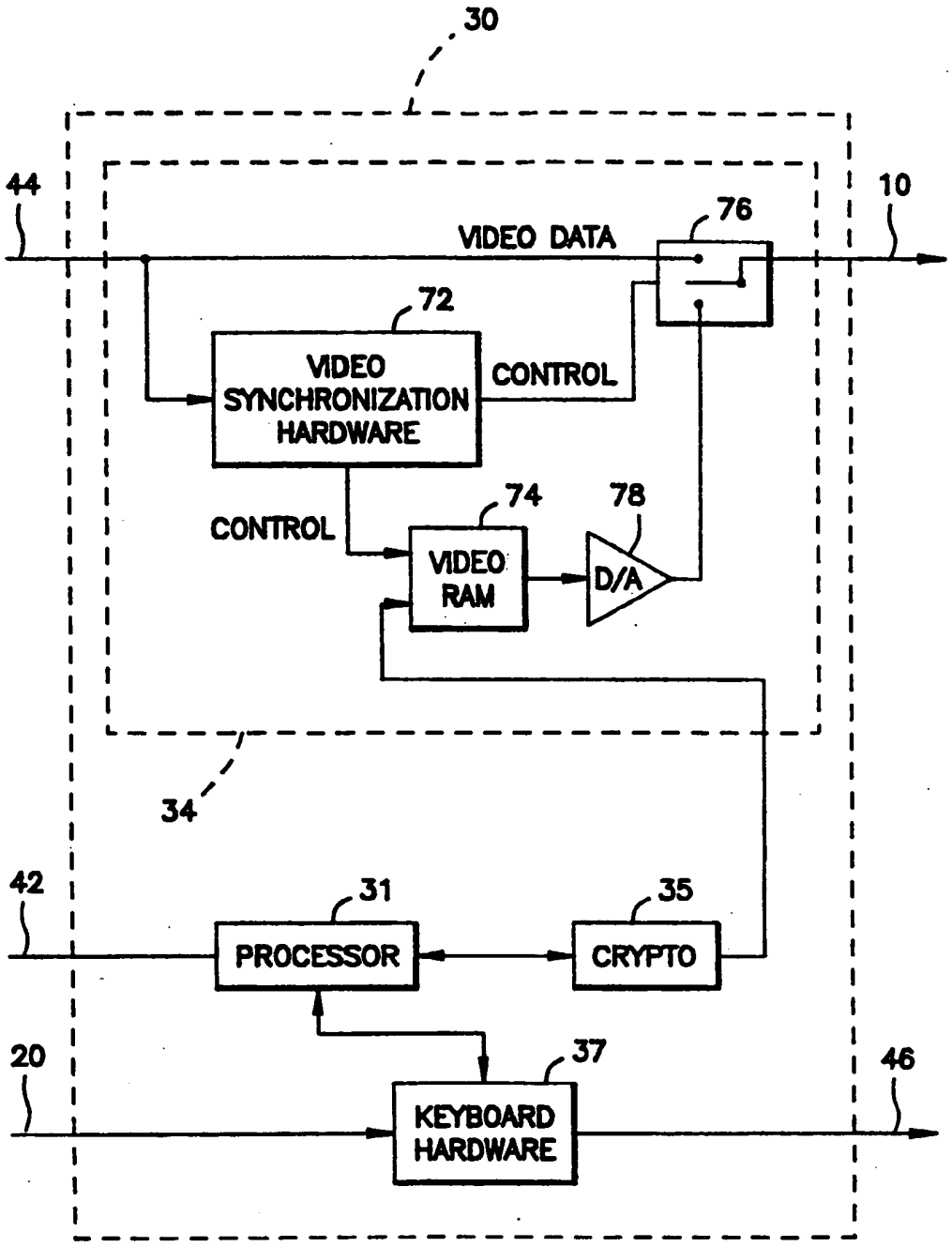

30
44
10
VIDEO DATA
76
72
VIDEO SYNCHRONIZATION HARDWARE
CONTROL
CONTROL
74
78
VIDEO RAM
D/A
34
42
31
35
PROCESSOR
CRYPTO
20
37
46
KEYBOARD HARDWARE

FIG. 5

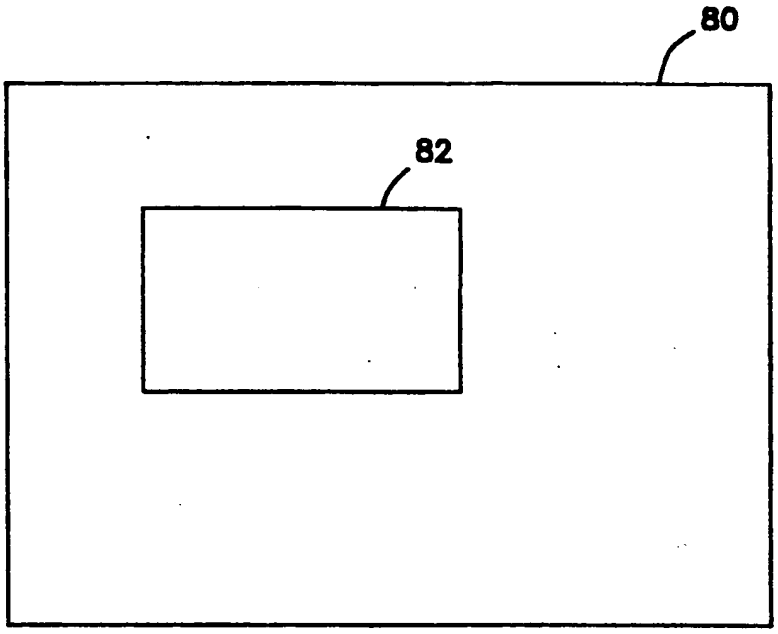


FIG. 6

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 93/06511

A. CLASSIFICATION OF SUBJECT MATTER
IPC 5 G06F12/14 G06F1/00

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)
IPC 5 G06F

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

	Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
11	Y	EP,A,0 192 243 (HONEYWELL) 27 August 1986 cited in the application see abstract; figures 3,4 see page 18, line 16 - page 21, line 14 see claims 1-10 ---	1-16
11	P,Y	WO,A,92 17958 (SECURE COMPUTING TECHNOLOGY) 15 October 1992 cited in the application see abstract; figure 1 see page 3, line 35 - page 6, line 16 see page 7, line 22 - page 10, line 35 --- -/--	1-16

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in 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 but 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

1

Date of the actual completion of the international search	Date of mailing of the international search report
23 November 1993	07. 12. 93

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl, Fax (+ 31-70) 340-3016	POWELL, D

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	IEEE SYMPOSIUM ON SECURITY AND PRIVACY, 4 May 1992 , OAKLAND, US; pages 226 - 239 J.EPSTEIN ET AL 'Evolution of a Trusted B3 Window Prototype' see figure 3 see page 229, left column, line 1 - page 230, right column, line 5 see page 231, right column, line 23 - page 232, left column, line 32 see page 233, left column, line 5 - page 234, left column, line 15 ---	3-5,8-13
Y	PROC. FALL JOINT COMPUTER CONF., 25 October 1987 , DALLAS, US; pages 411 - 420 J.PICCIOTTO ET AL 'Privileges and Their Use by Trusted Applications' see page 415, left column, line 23 - page 419, left column, line 18 ---	15,16
A	EP,A,0 096 628 (DIGITAL EQUIPMENT CORPORATION) 21 December 1983 see abstract; figure 1 ---	13
A	EP,A,0 443 423 (DIGITAL EQUIPMENT CORPORATION) 28 August 1991 see abstract; figures 4A,4B -----	15,16

INTERNATIONAL SEARCH REPORT

information on patent family members

International Application No
PCT/US 93/06511

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
EP-A-0192243	27-08-86	US-A-	4713753	15-12-87
		CA-A-	1252907	18-04-89
		JP-A-	61195443	29-08-86
WO-A-9217958	15-10-92	AU-A-	1576792	02-11-92
EP-A-0096628	21-12-83	US-A-	4498098	05-02-85
		AU-A-	1501683	08-12-83
		CA-A-	1185377	09-04-85
		JP-C-	1628356	20-12-91
		JP-B-	2052911	15-11-90
		JP-A-	59057279	02-04-84
EP-A-0443423	28-08-91	AU-A-	7103191	15-08-91

Requested Patent: WO9624092A2

Title:

A METHOD AND SYSTEM FOR MANAGING A DATA OBJECT SO AS TO COMPLY WITH PREDETERMINED CONDITIONS FOR USAGE ;

Abstracted Patent: WO9624092 ;

Publication Date: 1996-08-08 ;

Inventor(s): BENSON GREG (SE); URICH GREGORY H (SE) ;

Applicant(s): BENSON GREG (SE); URICH GREGORY H (SE) ;

Application Number: WO1996SE00115 19960201 ;

Priority Number(s): SE19950000355 19950201 ;

IPC Classification: G06F1/00; G06F12/14 ;

Equivalents:

AU4681496, EP0807283 (WO9624092), A3, JP10513289T, SE504085, SE9500355, US5845281 ;

ABSTRACT:

The present invention relates to a method and a system for managing a data object so as to comply with predetermined conditions for usage of the data object. To control the usage of the data object, a set of control data, defining usages of the data object which comply with the predetermined conditions, is created for the data object. The data object is concatenated with the user set of control data, encrypted and transferred to the user. When the user wants to use the data object, a special user program checks whether the usage complies with the control data. If so, the usage is enabled. Otherwise it is disabled.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 1/00, 12/14 — A2

(11) International Publication Number: **WO 96/24092**

(43) International Publication Date: 8 August 1996 (08.08.96)

(21) International Application Number: PCT/SE96/00115

(22) International Filing Date: 1 February 1996 (01.02.96)

(30) Priority Data:
9500355-4 1 February 1995 (01.02.95) SE

(71)(72) Applicant and Inventor: BENSON, Greg [US/SE]; Dalbackavägen 3, S-240 10 Dalby (SE).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: URICH, Gregory, H. [US/SE]; Warholmsvägen 8 B, S-224 65 Lund (SE).

(74) Agent: AWAPATENT AB; P.O. Box 5117, S-200 71 Malmö (SE).

(81) Designated States: AL, AM, AT, AT (Utility model), AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), EE, EE (Utility model), ES, FI, FI (Utility model), GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (Utility model), TJ, TM, TR, TT, UA, UG, US, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AZ, BY, KG, KZ, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

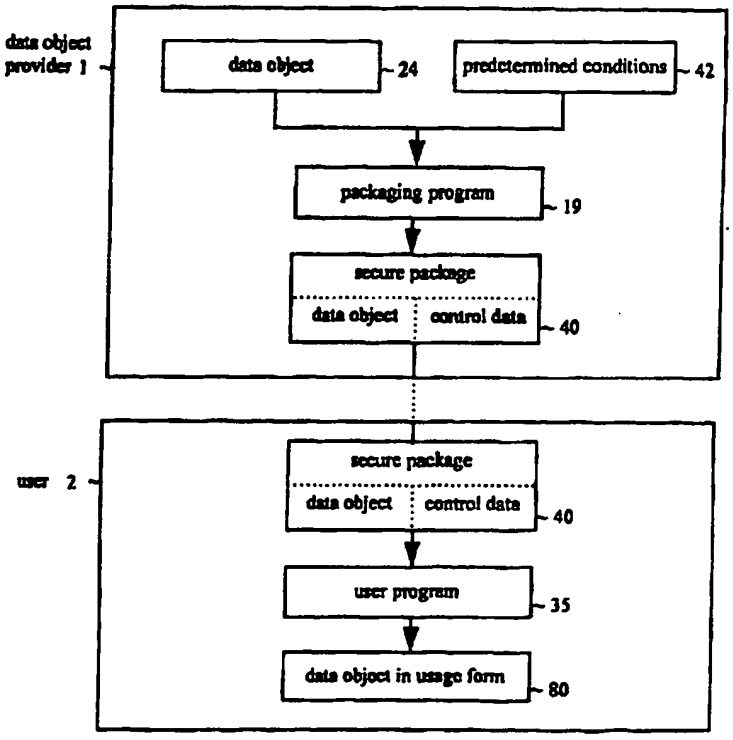
Published

Without international search report and to be republished upon receipt of that report.

(54) Title: A METHOD AND SYSTEM FOR MANAGING A DATA OBJECT SO AS TO COMPLY WITH PREDETERMINED CONDITIONS FOR USAGE

(57) Abstract

The present invention relates to a method and a system for managing a data object so as to comply with predetermined conditions for usage of the data object. To control the usage of the data object, a set of control data, defining usages of the data object which comply with the predetermined conditions, is created for the data object. The data object is concatenated with the user set of control data, encrypted and transferred to the user. When the user wants to use the data object, a special user program checks whether the usage complies with the control data. If so, the usage is enabled. Otherwise it is disabled.



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.

AM	Armenia	**GB**	United Kingdom	**MW**	Malawi
AT	Austria	**GE**	Georgia	**MX**	Mexico
AU	Australia	**GN**	Guinea	**NE**	Niger
BB	Barbados	**GR**	Greece	**NL**	Netherlands
BE	Belgium	**HU**	Hungary	**NO**	Norway
BF	Burkina Faso	**IE**	Ireland	**NZ**	New Zealand
BG	Bulgaria	**IT**	Italy	**PL**	Poland
BJ	Benin	**JP**	Japan	**PT**	Portugal
BR	Brazil	**KE**	Kenya	**RO**	Romania
BY	Belarus	**KG**	Kyrgystan	**RU**	Russian Federation
CA	Canada	**KP**	Democratic People's Republic of Korea	**SD**	Sudan
CF	Central African Republic			**SE**	Sweden
CG	Congo	**KR**	Republic of Korea	**SG**	Singapore
CH	Switzerland	**KZ**	Kazakhstan	**SI**	Slovenia
CI	Côte d'Ivoire	**LI**	Liechtenstein	**SK**	Slovakia
CM	Cameroon	**LK**	Sri Lanka	**SN**	Senegal
CN	China	**LR**	Liberia	**SZ**	Swaziland
CS	Czechoslovakia	**LT**	Lithuania	**TD**	Chad
CZ	Czech Republic	**LU**	Luxembourg	**TG**	Togo
DE	Germany	**LV**	Latvia	**TJ**	Tajikistan
DK	Denmark	**MC**	Monaco	**TT**	Trinidad and Tobago
EE	Estonia	**MD**	Republic of Moldova	**UA**	Ukraine
ES	Spain	**MG**	Madagascar	**UG**	Uganda
FI	Finland	**ML**	Mali	**US**	United States of America
FR	France	**MN**	Mongolia	**UZ**	Uzbekistan
GA	Gabon	**MR**	Mauritania	**VN**	Viet Nam

A METHOD AND SYSTEM FOR MANAGING A DATA OBJECT SO AS TO COMPLY WITH PREDETERMINED CONDITIONS FOR USAGE

Technical Field

The present invention relates to data processing and more particularly to a method and a system for managing data objects so as to comply with predetermined conditions for usage.

Background

Much has been written recently regarding the puzzle of universal connectivity. A typical vision of the data highway has long distance high speed data carriers interconnecting regional networks which provide telecommunications services and a wide range of interactive on-line services to consumers. Many of the pieces are already in place, others are in development or testing. In fact, even though the data highway is under construction it is currently open to limited traffic. On-line services are springing up daily and video on demand services are currently being tested.

The potential to benefit society is immense. The scope of information available to consumers will become truly global as the traditional barriers to entry for distribution of, and access to, information are lowered dramatically. This means that more diverse and specialized information will be made available just as conveniently as generic sources from major vendors used to be. The end result is that organizations and individuals will be empowered in ways heretofore only imagined.

However, a fully functioning data highway will only be as valuable as the actual services which it provides. Services envisioned for the data highway that involve the delivery of data objects (e.g. books, films, video, news, music, software, games, etc.) will be and are currently limited by the availability of such objects. Library and educational services are similarly affected. Before owners will allow their data objects to be offered they

must be assured of royalty payments and protection from piracy.

Encryption is a key component of any solution to provide copy protection. But encryption alone is not enough. During transmission and storage the data objects will be protected by encryption, but as soon as anyone is given the key to decipher the content he will have unlimited control over it. Since the digital domain permits data objects to be reproduced in unlimited quantities with no loss of quality, each object will need to be protected from unlimited use and unauthorized reproduction and resale.

The protection problem must not be solved by a separate solution for each particular data format, because then the progress will indeed be slow. It is important to consider the effect of standardization on an industry. Consider how the VHS, the CD and the DAT formats, and the IBM PC compatibility standards have encouraged growth in their respective industries. However, if there is to be any type of standardization, the standard must provide universal adaptability to the needs of both data providers and data users.

The data object owner may want to have permanent secure control over how, when, where, and by whom his property is used. Furthermore, he may want to define different rules of engagement for different types of users and different types of security depending on the value of particular objects. The rules defined by him shall govern the automated operations enabled by data services and networking. The owner may also want to sell composite objects with different rules governing each constituent object. Thus, it is necessary to be able to implement variable and extensible control.

The user on his part wants to be able to search for and purchase data objects in a convenient manner. If desired, the user should be able to combine or edit purchased objects (i.e. for creating a presentation).

Furthermore, the user may want to protect his children from inappropriate material. A complete solution must enable these needs as well.

What is needed is a universally adaptable system and method for managing the exchange and usage of data objects while protecting the interests of data object owners and users.

Prior Art

A method for enforcing payment of royalties when copying softcopy books is described in the European patent application EP 0 567 800. This method protects a formatted text stream of a structured document which includes a royalty payment element having a special tag. When the formatted text stream is inputted in the user's data processor, the text stream is searched to identify the royalty payment element and a flag is stored in the memory of the data processor. When the user for instance requests to print the document, the data processor requests authorization for this operation from a second data processor. The second data processor charges the user the amount indicated in the royalty payment element and then transmits the authorization to the first data processor.

One serious limitation of this method is that it can only be applied to structured documents. The description of the above-mentioned European patent application defines a structured document as: a document prepared in accordance with an SGML-compliant type definition. In other words it can not be applied to documents which are not SGML compliant and it cannot be applied to any other types of data objects.

Furthermore, this method does not provide for variable and extensible control. Anyone can purchase a softcopy book on a CD, a floppy disc or the like, and the same royalty amount is indicated in the royalty payment element of all softcopy books of the same title.

Thus, the method described in EP 0 567 800 does not satisfy the above-mentioned requirements for universally adaptable protection of data objects.

Summary of the Invention

Accordingly, it is a first object of the invention to provide a method and a data processing system for managing a data object in a manner that is independent of the format and the structure thereof, so as to comply with predetermined conditions for usage control and royalty payment.

It is a further object of the invention to provide such a method and system which is universally adaptable to the needs of both the owner and the user of the data object.

A further object of the invention is to provide such a method and system which enables a data object provider to distribute his data object while maintaining control of the usage thereof.

Yet another object of the invention is to provide a method and system which allows a data object provider to select the level of security for his data object in a flexible way.

Yet another object of the invention is to provide such a method and system which makes it possible to establish an audit trail for the data object.

Yet another object is to provide such a method and system which makes it possible to sell and buy data objects in a secure way.

The above-mentioned objects are achieved by a method and a system having the features of claims 1, 16, 21, 24 and 27.

Particular embodiments of the inventions are recited in the subclaims.

More particularly, a data object provider, e.g. the owner of a data object or his agent (broker), stores the data object in a memory device, e.g. a bulk storage device, where it is accessible by means of the data

provider's data processor. The data object can consist of digital data, analog data or a combination or hybrid of analog and digital data.

A general set of control data, which is based on the predetermined conditions for usage of the data object, is created and stored in the same memory device as the data object or another memory device where it is accessible by the data provider's data processor. The predetermined conditions for usage may be defined by the data object owner, by the broker or by anyone else. They may differ between different data objects.

The general set of control data comprises at least one or more usage control elements, which define usages of the data object which comply with the predetermined conditions. These usages may encompass for instance the kind of user, a time limit for usage, a geographical area for usage, allowed operations, such as making a hard copy of the data object or viewing it, and/or claim to royalty payment. The general set of control data may comprise other kinds of control elements besides the usage control element. In a preferred embodiment, the general set of control data comprises a security control element which defines a security procedure which has to be carried out before usage of the data object. It also comprises an identifier, which uniquely identifies the general set of control data.

The general set of control data is concatenated with a copy of the data object. Thus, the control data does not reside in the data object, but outside it, which makes the control data independent of the format of and the kind of data object and which allows for usage control independently of the data object format.

At least the usage control element(s) and the data object are encrypted, so that the user is unable to use the data object without a user program which performs the usage control and which decrypts the data object. Alter-

natively, the whole set of control data and the copy of the data object may be encrypted.

A user may request authorization for usage of a data object residing at a data provider's processor via a data network or in any other appropriate way. The authorization may or may not require payment. When a request for authorization for usage is received, a user set of control data is created by the data provider's processor. The user set of control data comprises the general set of control data or a subset thereof including at least one of said usage control elements which is relevant for the actual user. It typically also includes a new identifier which uniquely identifies this set of control data. If relevant, the user set of control data also comprises an indication of the number of usages authorized. If more than one kind of usage is authorized, the number of each kind of usage may be specified. Finally, the user set of control data is concatenated with a copy of the data object, and at least the usage control elements and the copy of the data object are encrypted to create a secure data package ready for transfer to the user.

Before the data package is transferred to the user, it should be confirmed that the request for authorization for usage has been granted. The check is preferably carried out before the user set of control data is created. However, it can also be carried out in parallel with or after the creation of the user control data. In the latter case, the number of usages requested by the user is tentatively authorized and included in the user set, but if the request is refused the user set is cancelled or changed.

The data package may be transferred to the user by electronic means or stored on bulk storage media and transferred to the user by mail or by any suitable transportation means.

Once the data object has been packaged in the above-described manner, it can only be accessed by a user

program which has built-in usage control and means for derypting the data package. The user program will only permit usages defined as acceptable in the control data. Moreover, if the control data comprises a security control element, the security procedure prescribed therein has to be complied with. In one embodiment, the usage control may be performed as follows. If the user decides to use a data object, the user program checks the control data to see if this action is authorized. More particularly, it checks that the number of authorized usages of this kind is one or more. If so, the action is enabled and the number of authorized usages decremented by one. Otherwise, the action is interrupted by the user program and the user may or may not be given the opportunity to purchase the right to complete the action.

After the usage, the user program repackages the data object in the same manner as it was packaged before.

When a data object is redistributed by a user or a broker, new control elements are added in the control data to reflect the relation between the old user/broker and the new user/broker. In this way, an audit trail for the data object may be created.

According to another aspect of the invention at least two data packages are stored on a user's data processor, which examines the usage control elements of the data packages in order to find a match. If a match is found, the user's data processor carries out an action which is specified in the user set of control data. This method can be used for selling and buying data objects.

Brief Description of Drawings

Fig. 1 is a flow diagram showing the general data flow according to the invention.

Fig. 2 is a system block diagram of a data object provider's data processor.

Fig. 3 is a block diagram showing the different modules of a data packaging program according to the invention.

Fig. 4 is a data flow diagram of a data packaging process.

Fig. 5 is an example of a header file.

Fig. 6 is an example of a usage data file.

Fig. 7 is a data flow diagram of loading an object to the data object provider's data processor.

Figs 8a and 8b are examples of control data for a data object on the data object provider's data processor and for an object ready to be transferred to a user, respectively.

Fig. 9 is a data flow diagram of data packaging on the data object provider's data processor.

Fig. 10 is a flow diagram of a data packaging procedure.

Fig. 11 is a memory image of a data object and its control data.

Fig. 12a is a memory image of the concatenated control data and data object.

Fig. 12b is a memory image of the concatenated and encrypted control data and data object.

Fig. 13 is a system block diagram of a user's data processor.

Fig. 14 is a block diagram showing the different modules of a user program according to the invention.

Fig. 15 is a flow diagram of using a data object on the user's data processor.

Fig. 16 is a flow diagram of how the user program operates in a specific application example.

Fig. 17 is an example of various data package structures for composite objects.

Description of the Best Mode for Carrying Out the Invention

General Overview

Fig. 1 is a flow diagram showing the general data flow according to the invention. The flow diagram is divided into a data object provider part 1 and a user part 2.

In the data object provider part 1, a data object 24 is created by an author. The data object can consist of digital data, analog data or a combination or hybrid of analog and digital data. The primary difference between analog data objects and digital data objects is the means for storage, transfer and usage.

The author also determines the conditions 42 for the usage of the data object 24 by a user. The data object 24 and the usage conditions 42 are input to a data packaging program 19, which creates a secure data package 40 of the data object and of control data which are based on the input usage conditions 42. Once packaged in this way, the data object can only be accessed by a user program 35.

The data object may be packaged together with a general set of control data, which is the same for all users of the data object. This may be the case when the data object is sent to a retailer or a bulletin board, wherefrom a user may obtain it. The data object may also be packaged as a consequence of a request from a user for usage of the data object. In that case, the package may include control data which is specifically adapted to that user. This control data is called a user set of control data. It may for example comprise the number of usages purchased by the user. Typically, the user set of control data will be created on the basis of the general set of control data and include at least a subset thereof. A user set of control data need not always be adapted for a specific user. All sets of control data which are created on the basis of a general set of control data will be called a user set of control data. Thus, a set of control data can be a general set in one phase and a user set in another phase.

The above-mentioned data packaging can be carried out by the author himself by means of the data packaging program 19. As an alternative, the author may send his data object to a broker, who inputs the data object and the usage conditions determined by the author to the data

packaging program 19 in order to create a secure package 3. The author may also sell his data object to the broker. In that case, the broker probably wants to apply his own usage conditions to the data packaging program. The author may also provide the data object in a secure package to the broker, who repackages the data object and adds further control data which is relevant to his business activities. Various combinations of the above alternatives are also conceivable.

In the user part 2 of the flow diagram, the secure package 40 is received by a user, who must use the user program 35 in order to unpackage the secure package 40 and obtain the data object in a final form 80 for usage. After usage, the data object is repackaged into the secure package 40.

The different parts of the system and the different steps of the method according to the invention will now be described in more detail.

The data provider's data processor:

Fig. 2 is a system block diagram of a data object provider's data processor. As mentioned above, the data object provider may be an author of a data object, an owner of a data object, a broker of a data object or anyone else who wants to distribute a data object, while retaining the control of its usage. The data processor is a general or special purpose processor, preferably with network capabilities. It comprises a CPU 10, a memory 11 and a network adapter 12, which are interconnected by a bus 13. As shown in Fig. 2, other conventional means, such as a display 14, a keyboard 15, a printer 16, a bulk storage device 17, and a ROM 18, may also be connected to the bus 13. The memory 11 stores network and telecommunications programs 21 and an operating system (OS) 23. All the above-mentioned elements are well-known to the skilled person and commercially available. For the purpose of the present invention, the memory 11 also stores a data packaging program 19 and, preferably, a database

20 intended for control data. Depending upon the current operation, one or more data objects 24 can be stored in the memory 11 as shown or in the bulk storage 17. The data provider's data processor is considered secure.

The Data Packaging Program:

The data packaging program 19 is used for creating control data for controlling the usage of a data object and for packaging the data object and the control data into a secure package.

As shown in Fig. 3, it comprises a program control module 301, a user interface module 302, a packaging module 303, a control data creation module 304, an encryption module 305, one or more format modules 306, and one or more security modules 307.

The control module 301 controls the execution of the other modules. The user interface module 302 handles interaction with the data object provider. The packaging module 303 packages the control data and the data object. It uses the control data creation module 304, the format modules 306, the security modules 307 and the encryption module 305 as will be described more in detail below.

The format modules 306 comprise program code, which is required to handle the data objects in their native format. They can fulfill functions such as data compression and data conversion. They can be implemented by any appropriate, commercially available program, such as by means of a routine from the PKWARE Inc. Data Compression Library for Windows and the Image Alchemy package from Handmade Software Incorporated, respectively. They can also be implemented by custom designed programs.

The security modules 307 comprise program code required to implement security, such as more sophisticated encryption than what is provided by the encryption module 305, authorization algorithms, access control and usage control, above and beyond the basic security inherent in the data package.

The data packaging program 19 can contain many different types of both format and security modules. The program control module 301 applies the format and security modules which are requested by the data provider.

The encryption module 305 may be any appropriate, commercially available module, such as "FileCrypt" Visual Basic subprogram found in Crescent Software's QuickPak Professional for Windows - FILECRPT.BAS, or a custom designed encryption program.

The control data creation module 304 creates the control data for controlling the usage of the data object. An example of a control data structure will be described more in detail below.

The Control Data:

The control data can be stored in a header file and a usage data file. In a preferred embodiment, the header file comprises fields to store an object identifier, which uniquely identifies the control data and/or its associated data object, a title, a format code, and a security code. The format code may represent the format or position of fields in the usage data file. Alternatively, the format code may designate one or more format modules to be used by the data packaging program or the user program. The security code may represent the encryption method used by the encryption module 305 or any security module to be used by the data packaging program and the user program. The header file fields will be referred to as header elements.

The usage data file comprises at least one field for storing data which controls usage of the data object. One or more usage data fields which represent one condition for the usage of the data object will be referred to as a usage element. In a preferred embodiment, each usage element is defined by an identifier field, e.g. a serial number, a size field, which specifies the size of the usage element in bytes or in any other appropriate way, and a data field.

The header elements and the usage elements are control elements which control all operations relating to the usage of the object. The number of control elements is unlimited. The data provider may define any number of control elements to represent his predetermined conditions of usage of the data object. The only restriction is that the data packaging program 19 and the user program 35 must have compatible program code to handle all the control elements. This program code resides in the packaging module and the usage manager module, to be described below.

Control elements can contain data, script or program code which is executed by the user program 35 to control usage of the related data object. Script and program code can contain conditional statements and the like which are processed with the relevant object and system parameters on the user's data processor. It would also be possible to use a control element to specify a specific proprietary user program which can only be obtained from a particular broker.

It is evident that the control data structure described above is but one example. The control data structure may be defined in many different ways with different control elements. For example, the partitioning of the control data in header data and usage data is not mandatory. Furthermore, the control elements mentioned above are but examples. The control data format may be unique, e.g. different for different data providers, or defined according to a standard.

The operation of the data packaging program

The operation of a first embodiment of the data packaging program will now be described with reference to the block diagram of Fig. 3 and the flow diagram of Fig. 4.

First a data provider creates a data object and saves it to a file, step 401. When the data packaging program is started, step 402, the user interface module

302 prompts the data object provider to input, step 403, the header information consisting of e.g. an object identifier, a title of the data object, a format code specifying any format module to be used for converting the format of the data object, and a security code specifying any security module to be used for adding further security to the data object. Furthermore, the user interface module 302 prompts the data object provider to input usage information, e.g. his conditions for the usage of the data object. The usage information may comprise the kind of user who is authorized to use the data object, the price for different usages of the object etc. The header information and the usage information, which may be entered in the form of predetermined codes, is then passed to the control module 301, which calls the packaging module 303 and passes the information to it.

The packaging module 303 calls the control data creation module 304, which first creates a header file, then creates header data on the basis of the header information entered by the data object provider and finally stores the header data, step 404-405. Then a usage data file is created, usage data created on the basis of the usage information entered by the data provider, and finally the usage data is stored in the usage data file, step 406-407.

The packaging module 303 then applies any format and security modules 306, 307 specified in the header file, steps 408-413, to the data object.

Next, the packaging module 303 concatenates the usage data file and the data object and stores the result as a temporary file, step 414. The packaging module 303 calls the encryption module 305, which encrypts the temporary file, step 415. The level of security will depend somewhat on the quality of the encryption and key methods used.

Finally, the packaging module 303 concatenates the header file and the encrypted temporary file and saves

the result as a single file, step 416. This final file is the data package which may now be distributed by file transfer over a network, or on storage media such as CD-ROM or diskette, or by some other means.

Example 1

An example of how the data packaging program 19 can be used will now be described with reference to Figs 5 and 6. In this example the data object provider is a computer graphics artist, who wants to distribute an image that can be used as clip art, but only in a document or file which is packaged according to the method of the invention and which has usage conditions which do not permit further cutting or pasting. The artist wants to provide a free preview of the image, but also wants to be paid on a per use basis unless the user is willing to pay a rather substantial fee for unlimited use. The artist will handle payment and usage authorization on a dial-up line to his data processor.

The artist uses some image creation application, such as Adobe's Photoshop to create his image. The artist then saves the image to file in an appropriate format for distribution, such as the Graphical Interchange Format (GIF). The artist then starts his data packaging program and enters an object identifier, a title, a format code and a security code, which in this example are "123456789", "image","a", and "b", respectively. In this example, the format code "a" indicates that no format code need be applied, and this code is selected since the GIF format is appropriate and already compressed. Furthermore, the security code "b" indicates that no security module need be applied and this code is selected since the security achieved by the encryption performed by means of the encryption module 305 is considered appropriate by the artist.

Then the artist enters his dial-up phone number, his price for a single use of the image and for unlimited use of the data object, a code for usage types approved, and

for number of usages approved. For this purpose, the user interface module 302 may display a data entry form.

The data packaging program 19 creates control data on the basis of the information entered by the artist and stores the data in the header file and in the usage data file as shown in Figs 5 and 6, respectively. This data consitutes a general set of control data which is not specifically adapted to a single user, but which indicates the conditions of usage determined by the artist for all future users.

Then the package program 19 concatenates the data object and the control data in accordance with steps 414-416 of Fig. 4 to achieve the secure package. No format module or security module is applied to the data object, since they are not needed according to the data in the header file.

When the secure package has been obtained, the artist sends it to a bulletin board, from where it can be retrieved by a user.

Example 2

Below, another embodiment of the data packaging program 19 will be described with reference to Figs 7-12b. In this example, the data object consists of a video film, which is created by a film company and sent to a broker together with the predetermined conditions 42 for usage of the video. The broker loads the video 24 to the bulk storage 17 of his data processor. Then, he uses his data packaging program 19 to create a general set of control data 50 based on the predetermined conditions 42 for usage indicated by the film company. Furthermore, the address to the video in the bulk storage 17 is stored in an address table in the control database 20 or somewhere else in the memory 11. It could also be stored in the general set of control data 50. Finally, the general set of control data 50 is stored in the control database 20. It could also be stored somewhere else in the memory 11.

After these operations, which correspond to steps 401-407 of Fig. 4, the data packaging program is exited.

Fig. 8a shows the general set of control data for the video according to this example. Here the control data includes an identifier, a format code, a security code, the number of usage elements, the size of the data object, the size of the usage elements and two usage elements, each comprising an identifier field, a size field and a data field. The identifier may be a unique number in a series registered for the particular broker. In this example, the identifier is "123456789", the format code "0010", which, in this example, indicates the format of a AVI video and the security code is "0010". Furthermore, the first usage element defines the acceptable users for the video and the second usage element data defines the number of viewings of the video purchased by a user. The first usage element data is 1 which, for the purposes of this example will signify that only education oriented users are acceptable to the film company. The data field of the second usage element data is empty, since at this stage no viewings of the video has been purchased.

Managing Object Transfer:

The broker wants to transfer data objects to users and enable controlled usage in return for payment of usage fees or royalties. Managing the broker-user business relationship and negotiating the transaction between the broker and the user can both be automated, and the control data structure can provide unlimited support to these operations. The payment can be handled by transmitting credit card information, or the user can have a debit or credit account with the broker which is password activated. Preferably, payment should be confirmed before the data object is transferred to the user.

Data packaging:

When a user wants to use a data object, he contacts the broker and requests authorization for usage of the data object. When the request for authorization is recei-

ved in the broker's data processor, a data program compares the usage for which authorization is requested with the usage control elements of the control data of the data object to see if it complies with the predetermined conditions for usage indicated therein. The comparison may include comparing the user type, the usage type, the number of usages, the price etc. If the requested usage complies with the predetermined conditions the authorization is granted, otherwise it is rejected.

Fig. 9 is a data flow diagram of the data packaging on the broker's data processor, which occurs in response to a granted request from a user for authorization for usage of the video, e.g. a granted request for the purchase of two viewings.

In response to a granted request, the broker again applies the data packaging program 19. The general set of control data 50 and the data object 24 are input to the program from the control database 20 and the bulk storage 17, respectively. The program creates a user set of control data 60 on the basis of the general set of control data 50 and concatenates the user set 60 and the data object 24 to create a secure data package 40, which may then be transferred to the user by any suitable means. A copy of the user set of control data is preferably stored in the broker's control database. This gives the broker a record with which to compare subsequent use, e.g. when a dial-up is required for usage.

Fig. 10 is a flow diagram of an exemplary procedure used for creating a user set of control data and for packaging the user set of control data and the video into a secure package. Here, the procedure will be described with reference to the general set of control data shown in Fig. 8a.

The user set of control data 60, i.e. a set of control data which is adapted to the specific user of this example, is created in steps 1001-1003 of Fig. 11. First, the general set of control data 50 stored in the control

database is copied to create new control data, step 1001. Second, a new identifier, here "123456790", which uniquely identifies the user set of control data, is stored in the identifier field of the new control data 60, step 1002. Third, the data field of the second usage element is updated with the usage purchased, i.e. in this example with two, since two viewings of the video were purchased, step 1003.

The thus-created user set of control data, which corresponds to the general set of control data of Fig. 8a is shown in Fig. 8b.

The user set of control data is stored in the control database 20, step 1004. Then, the video, which is stored in the bulk storage 17, is copied, step 1005. The copy of the video is concatenated with the user set of control data, step 1006. The security code 0010 specifies that the entire data package 40 is to be encrypted and that the user program 35 must contain a key which can be applied. Accordingly, the whole data package is encrypted, step 1007. Finally, the encrypted data package is stored on a storage media or passed to a network program, step 1008, for further transfer to the user.

Fig. 11 is a memory image of the video 24 and the user control data 60. The user control data and a copy of the video 24 are concatenated as shown in Fig. 12a. The encrypted data package 40 is shown in Fig. 12b.

The procedure of Fig. 10 can be implemented by the data packaging program of Fig. 3. As an alternative to the procedure of Fig. 10, the user set of control data can be created as in steps 1001-1003 and saved in a header file and in a usage data file, whereafter steps 408-416 of the data packaging program of Fig. 4 can be performed to create the secure package.

The above-described process for creating a user-adapted set of control data may also be used by a user who wants to redistribute a data object or by a broker who wants to distribute the data object to other brokers.

Obviously, redistribution of the data object requires that redistribution is a usage approved of in the control data of the data object. If so, the user or the broker creates a user set of control data by adding new control elements and possibly changing the data fields of old control element to reflect the relation between the author and the current user/broker and between the current user/broker and the future user/broker. In this way, an audit trail is created.

The user's data processor:

The user's data processor, which is shown in Fig. 13, is a general or special purpose processor, preferably with network capabilities. It comprises a CPU 25, a memory 26, and a network adapter 27, which are interconnected by a bus 28. As shown in Fig. 13, other conventional means, such as a display 29, a keyboard 30, a printer 31, a sound system 32, a ROM 33, and a bulk storage device 34, may also be connected to the bus 28. The memory 26 stores network and telecommunications programs 37 and an operating system (OS) 39. All the above-mentioned elements are well-known to the skilled person and commercially available. For the purpose of the present invention, the memory 26 also stores a user program 35 and, preferably, a database 36 intended for the control data. Depending upon the current operation, a data package 40 can be stored in the memory 26, as shown, or in the bulk storage 34.

The user program:

The user program 35 controls the usage of a data object in accordance with the control data, which is included in the data package together with the data object.

As shown in Fig. 14, the user program 35 comprises a program control module 1401 a user interface module 1402, a usage manager module 1403, a control data parser module 1404, a decryption module 1405, one or more format modules 1406, one or more security modules 1407, and a file transfer program 1409.

The control module 1401 controls the execution of the other modules. The user interface module 1402 handles interactions with the user. The usage manager module 1403 unpackages the secure package 40. It uses the control data parser module 1404, the decryption module 1405, the format modules 1406, and the security modules 1407.

The format modules 1406 comprise program code, which is necessary to handle the data objects in their native format, such as decompression and data format procedures. The security modules 1407 comprises program code required to implement security above the lowest level, such as access control, usage control and more sophisticated decryption than what is provided by the basic decryption module 1405.

The user program 35 can contain many different types of both format and security modules. However, they should be complementary with the format and security modules used in the corresponding data packaging program. The usage manager module 1401 applies the format and security modules which are necessary to use a data object and which are specified in its control data. If the proper format and security modules are not available for a particular data object, the usage manager module 1401 will not permit any usage.

The decryption module 1405 can be the above-mentioned FileCrypt Visual Basic subprogram or some other commercially available decryption program. It can also be a custom designed decryption module. The only restriction is that the decryption module used in the user program is complementary with the encryption module of the data packaging program.

The control data parser module 1403 performs the reverse process of the control data creation module 304 in Fig. 3.

The user program 35 can have code which controls use of the program by password or by any other suitable method. A password may be added in a password control

element during packaging of the data object. The password is tranferred to the user by registered mail or in any other appropriate way. In response to the presence of the password control element in the control data structure, the user program prompts the user to input the password. The input password is compared with the password in the control data, and if they match, the user program continues, otherwise it is disabled.

The user program 35 can also have procedures which alter the behavior of the program (e.g. provide filters for children) according to the control data of the user object 41. It is important to mention that the user program 35 never stores the object in native format in user accessible storage and that during display of the data object the print screen key is trapped.

The file transfer program 1409 can transfer and receive files via network to and from other data processor.

Since the data object is repackaged into the secure package after the usage, the user program should also include program code for repackaging the data object. The program code could be the same as that used in the corresponding data packaging program 19. It could also be a separate program which is called from the user program.

Operation of the user program:

The operation of an embodiment of the user program 35 will now be described with reference to the block diagram of Fig. 14 and the flow diagram of Fig. 15.

First the user receives a data package 40 via file transfer over a network, or on a storage media such as CD-ROM or diskette, or by any other appropriate means, step 1501. He then stores the data package as a file on his data processor, step 1502.

When the user wants to use the data object, he starts the user program 35, step 1503. Then he requests usage of the data object, step 1504. The request is received by the user interface module 1402, which noti-

fies the control module 1401 of the usage request. The control module 1401 calls the usage manager module 1403 and passes the usage request.

The usage manager module 1403 reads the format code from the data package to determine the control data format. Then it calls the decryption module 1405 to decrypt and extract the control data from the data package. The usage manager module 1403 applies the decryption module 1405 incrementally to decrypt only the control data. Finally, it stores the control data in memory, step 1505.

The usage manager module 1403 then calls the control data parser module 1404 to extract the data fields from the usage elements.

The usage manager module 1403 then compares the user request for usage with the corresponding control data, steps 1506-1507. If the requested usage is not permitted in the control data, the requested usage is disabled, step 1508. However, if the requested usage is approved of in the control data, the usage manager module 1403 applies any format and security modules 1406, 1407 specified in the header data or usage data, steps 1509-1514, to the data package.

Then the usage manager module 1403 calls the decryption module 1405, which decrypts the object data, step 1515, whereafter the requested usage is enabled, step 1516. In connection with the enabling of the usage, the control data may need to be updated, step 1517. The control data may for instance comprise a data field indicating a limited number of usages. If so, this data field is decremented by one in response to the enabling of the usage. When the user has finished usage of the data object, the user program 35 restores the data package in the secure form by repackaging it, step 1518. More particularly, the data object and the usage elements are reconcatenated and reencrypted. Then the header elements are added and the thus-created package is stored in the user's data processor.

Example 1 contd.

A specific example of how the user program operates will now be described with reference to Figs 6 and 15. The example is a continuation of Example 1 above, where an artist created an image and sent it to a bulletin board.

Assume that a user has found the image at an electronic bulletin board (BBS) and is intrested in using it. He then loads the data package 40 containing the image to his data processor and stores it as a file in the bulk storage. The user then executes the user program 35 and requests to preview the image. The user program then performs steps 1505-1507 of the flow diagram in Fig. 15. The request for a preview of the image is compared with the data field of the usage element "code for usage type approved". In this example, the code "9" designates that previews are permitted. Thus, the requested preview is OK. Then, the user program 35 performs step 1509-1515 of Fig. 15. Since the format code "a" and the security code "b" of the header data indicate that neither conversion, nor decompression, nor security treatment is required, the user program only decrypts the object data. The usage manager module 1403 then displays the preview on the user's data processor and passes control back to the user interface 1402.

When the user is finished previewing the image, the user interface module 1402 displays the costs for usage of the image in accordance with the price usage data of the control data ("price for single use" and "price for unlimited use" in Fig. 6) and prompts the user to enter a purchase request. The user decides to buy unlimited use of the image, and the user interface module 1402 inputs purchase information, such as an identification, billing, and address for that request and passes the request to the control module 1401. The control module calls the file transfer program 1409, which dials the artist's dial-up number as indicated in the usage data ("control

element for artist's phone number" in Fig. 6) and transfers the request and purchase information to a broker program on the artist's data processor. Upon approval of the purchase, the broker program returns a file containing an update for "usage type approved" control elements. The update is "10" for the usage type approved, which in this example indicates that unlimited use by that user is permitted. The file transfer program 1409 passes this update to the usage manager module 1403 which updates the control data with the "usage type approved" code. The user interface module 1402 then displays a confirmation message to the user.

Subsequently, the user interface module inputs a request to copy the image to a file packaged according to this invention, on the user's machine. The usage manager module then compares the user request control data. The usage manager module examines the data filed for "usage type approved", which now is "10". The usage manager module copies the image to the file.

When the user is finished with the image, the usage manager module 1403 repackages the image as before except with updated control data. This repackaging process is exactly like that shown in Fig. 4, except that the header and usage data already exist, so the process starts after step 406 where control data is created.

Improved security

If the data object provider wants to improve the security of a data package containing a data object, a security module 307 containing a sophisticated encryption algorithm, such as RSA, could be used. In that case the packaging module 303 calls the security module 307 in step 412 of the flow diagram of Fig. 4. The security module encrypts the image and passes a security algorithm code to the control data creation module 302, which adds a control element for the security module code, which will be detected by the user program 35. Then the data packaging continues with step 414. When the data package

is sent to the user, the public key is mailed to the user by registered mail. When the user program is executed in response to a request for usage of this data object, the usage manager module will detect the security module code in the control data and call the security module. This module passes control to the user interface module 1402, which requests the user to input the public key. If the key is correct, the user security module applies complementary decryption using that key and passes a usage approved message to the usage manager module, which enables the usage.

As another example of improved security, a security module may implement an authorization process, according to which each usage of the data object requires a dial-up to the data processor of the data object provider. When the corresponding security module code is detected by the user program 35, the relevant security module is called. This module passes a request for authorization to the control module 1401, which calls the file transfer program 1409, which dial the data object provider's dial-up number, which is indicated in a usage element and transfers the request for authorization of usage. Upon a granted authorization, the data provider's data processor returns a usage approved message to the user security module, which forwards the approval tl the usage control module, which enables one usage. If the user requests further usages of the data object, the authorization process is repeated. This procedures results in a permanent data object security.

Example 2 contd.

A further specific example of how the user program 35 operates will now be described with reference to Fig. 16. The example is a continuation of Example 2 above, where a user purchased two viewings of a video film from a broker.

The user wants to play the video which was purchased and transferred from the broker. The user applies the

user program 35, step 1601, and requests to play the video, step 1602. The user program 35 first examines the user set of control data 60, step 1603. In this example, the user program 35 contains only those format and security modules for objects with format code of 0010 and with a security code of 0010. Consequently, only those types of data objects may be used. If the program encounters other codes it will not enable the usage action, step 1604-1605.

Next, the user program 35 compares the first control element data which is 1, for educational users only, to user information entered by the user on request of the user program. Since the user type entered by the user is the same as that inicated in the first usage element the process continues, steps 1606-1607. Then the user program checks the second control element data which specifies that the number of plays purchased is 2. Consequently, the usage is enabled, step 1609. The user program applies the decryption module with the universal key and the AVI format video is displayed on the display unit 29. Then, the second control element data is decremented by one, step 1610. Finally, the video is repackaged, step 1611

Implementation of Variable and Extensible Object Control:

Object control is achieved through the interaction of the data packaging program 19 and the usage program 35 with the control data. Variation of object control can be applied to a particular object by creating a control data format with control elements defining the control variation and the circumstances in which the variation is applied. Program procedures should then be added to program modules to process the control elements. For example, suppose a broker wants to allow students to print a particular article for free but require business users to pay for it. He defines control elements to represent the user types student and business and the associated costs for each. He then adds program logic to examine the user type and calculate costs accordingly. Object control is

extensible in the sense that the control data format can have as many elements as there are parameters defining the rules for object control.

Implementation of Variable and Extensible Object Security:

Object security is also achieved through the interaction of the data packaging program 19 and the user program 35 with the control data. Security process and encryption/decryption algorithms can be added as program modules. Variation of object security can be applied to a particular object by creating a control data format with control elements defining the security variation and the circumstances in which the variation is applied. Program procedures should be added to program modules to process the control elements. For example, suppose a broker wants to apply minimal security to his collection of current news articles but to apply tight security to his encyclopedia and text books. He defines a control element for security type. He then adds program logic to apply the security algorithms accordingly. Object security is extensible in the sense that multiple levels of security can be applied. The level of security will of course be dependent on the encryption/key method which is implemented in the security modules. One level of security may be to require online confirmation when loading a data object to the user's data processor. This can be implemented in program code in a security module. This permits the broker to check that the object has not already been loaded as well as double check all other parameters.

It is also important to have version control with time stamping between the usage program and the user's control database. Otherwise the database can be duplicated and reapplied to the user program. The user program can place a time stamp in the control database and in a hidden system file each time the control database is accessed. If the time stamps are not identical, the control database has been tampered with and all usage is

disabled. Program code for handling time stamps can reside in a security module.

Handling Composite Objects:

A composite object can be handled by defining a control data format with control elements defining relationships between constituent objects and by defining a parent/child element and a related object id element. For example, suppose a broker wants to include a video and a text book in an educational package. He creates a parent object with control elements referring to the video and textbook objects. He also includes control elements in the control data for the video object and the textbook object referring to the parent object. Finally, he adds program procedures to program modules to process the control elements.

In other words, when the data object is a composite data object including at least two constituent data objects, a respective general set of control data is created for each of the constituent data object and the composite data object. In response to a request from a user, a respective user set of control data is created for each of the constituent data objects as well as for the composite data object.

Examples of various data package structures for composite objects are given in Fig. 17.

Another side of composite objects is when the user wants to combine data objects for some particular use. Combination is a usage action that must be permitted in each constituent data object. A new data object is created with control data linking the constituent data objects. Each constituent data object retains its original control data which continues to control its subsequent usage.

When a user requests authorization for usage of one constituent data object in a composite data object, a user set of control data is created only for that consti-

tuent data object and concatenated only with a copy of that constituent data object.

Scaleable Implementation:

The flexible control data structure and modular program structure permit almost boundless extensibility with regard to implementation of the owner's requirements for usage control and royalty payment. The control data structure can include control elements for complex user types, usage types, multiple billing schemes, artistic or ownership credit requirements and others. Security modules can be included which interact with any variation of the control data structure and the control data. Security modules could require a dial up to the brokers data processor to approve loading or usage actions and to implement approval authentication mechanisms.

User acting as a broker:

A limited or full implementation of the broker's data packaging program can be implemented on the user's machine to permit further distribution or reselling. However, only those data objects with control data permitting further distribution or reselling are enabled in that way.

Rebrokering

An author of a data object may want to allow his original broker to distribute his data objec to other brokers whom will also distribute his image. He then includes a control element which enables rebrokering in tne control data before distributing the data object with its associated control data to the origingal broker. Upon request for rebrokering, the original broker copies the general set of control data and updates the copy to create a user set of control data which will function as the general set of control data on the subsequent brokers data processor. The original broker packages the data object with the user set of control data and transfers the package to the subsequent broker. The subsequent broker then proceeds as if he were an original broker.

Automated transaction negotiation

This is an example of how the predetermined conditions for usage included in the control data can be used for achieving automated transaction negotiation.

Suppose some company wnats to provide a computer automated stock trading. Buy and sell orders could be implemented in the form of data packages and a user program could process the data packages and execute transactions. Data packages could carry digital cash and manage payment based on conditions defined in the control data.

In this example, the buy order is created using a data packaging program according to the invention on the buyer's data processor. The sell order is created using the data packaging program on the seller's data processor. Both orders are used by the the user program on the stock trader's data processor. The usages would take the form of using a sell order data package to sell stock and a buy order data package to buy stock. The rules or conditions for buying and selling stocks could be indicated in the control data of the packages. The data object consists of digital money. In this context it is important to remember that digital money is merely data which references real money or vitual money that is issued and maintained for the purpose of digital transactions.

In this example the buyer starts with a digital money data file. He uses the data packaging program to create control data, e.g. kind of stock, price, quantity, for the purchase, and he then packages the digital money data file and the control data into a secure package as described above.

The seller starts with an empty data file. This empty file is analogous to the digital money data file except it is empty. The seller creates control data, e.g. kind of stock, price, quantity, and packages the empty file and the control data into a secure package.

Both the sell order package and the buy order package are transferred to the data processor of the stock trading company, where they are received and stored in the memory. The user program of the stock trading company examines the control data of the buy and sell order packages in the same way as has been described above and looks for a match. Upon identifying matched buy and sell orders the user program executes a transaction, whereby the digital money is extracted from the buy order data package and transferred to the sell order package. Then the control data of the data packages is updated to provide an audit trail. Both packages are repackaged in the same manner as they were previously packaged and then transferred back to their authors.

The above described technique could be used for selling and buying any object as well as for automated negotiations. Payment may be carried out in other ways than by digital money.

In the general case, the data processor of the user decrypts the usage control elements of the user sets of control data and examines the usage control elements to find a match. In response to the finding of a match, the user's data processor carries out an action which is specified in the user set of control data.

CLAIMS

1. A method for managing a data object so as to comply with predetermined conditions for usage of the data object, comprising the steps of:

- storing the data object in a memory device, where it is accessible by means of a data object provider's data processor;

- creating, by said data processor, a general set of control data for the data object based on said predetermined conditions for usage, said general set of control data comprising at least one or more usage control elements defining usages of the data object which comply with said predetermined conditions;

- storing said general set of control data in a memory device, where it is accessible by said data processor;

- concatenating the general set of control data with a copy of the data object; and

- encrypting at least the copy of the data object and said one or more usage control elements to create a secure data package which is ready for transfer to a user.

2. A method as set forth in claim 1, wherein the step of encrypting comprises encrypting the data object and the general set of control data.

3. A method as set forth in claims 1 or 2, wherein the step of creating control data comprises creating an identifier which uniquely identifies the general set of control data.

4. A method as set forth in claims 1, 2 or 3, wherein the step of creating a general set of control data comprises creating a security control element which identifies a security process to be applied before usage of the data object is allowed.

5. A method as set forth in any of the preceding claims, wherein the step of creating a general set of

control data comprises creating a format control element which identifies the format of the control data.

6. A method as set forth in any of the preceding claims, comprising the further steps of:

- creating, in response to a request for authorization for usage of the data object by a user, a user set of control data, which comprises at least a subset of the general set of control data, including at least one of said usage control elements;
- using the user set of control data instead of the general set of control data in said concatenating step;
- using the at least one usage control element of the user set of control data instead of the one or more usage control elements of the general set of control data in the encrypting step;
- checking, before allowing transfer of the data package to the user, that said request for authorization for usage of the data object has been granted.

7. A method as set forth in any of the preceding claims, further comprising the steps of receiving in said data processor the request for authorization for usage by a user; comparing the usage for which authorization is requested with said one or more usage control elements of the general set of control data and granting the authorization if the usage for which authorization is requested complies with the usages defined by said one or more usage control elements.

8. A method as set forth in claim 7, further comprising the step of securing payment for the requested authorization for usage before granting the authorization.

9. A method as set forth in any one of claims 6-8, wherein the data object is composed of at least two constituent data objects and wherein the user set of control data, in response to a request for authorization for usage of one of said constituent data objects by a user, is created only for that constituent data object and

concatenated only with a copy of that constituent data object.

10. A method as set forth in any one of claims 6-9, wherein the data provider's data processor is connected to a data network and the request for authorization is received from a data processor of the user, which is also connected to the data network, further comprising the step of transferring the data package through the data network to the user's data processor.

11. A method as set forth in any one of claims 6-8 or 10, wherein the data object is a composite data object including at least two constituent data objects and wherein the step of creating a general set of control data comprises the step of creating a respective general set of control data for each of the constituent data objects and the composite data object and wherein the step of creating a user set of control data comprises the step of creating a respective user set of control data for each of the constituent data objects and the composite data object.

12. A method as set forth in any one of claims 6-11, comprising the further step of storing a copy of the user set of control data in the data object provider's processor.

13. A method as set forth in any of the preceding claims, comprising the further steps of:

- receiving the data package in a user's data processor;

- storing the data package in a memory device where it is accessible by means of the user's data processor;

- decrypting said one or more usage control elements;

- checking, in response to a request by the user for usage of the data object, whether the requested usage complies with the usage defined by the at least one usage control element of the general set of control data;

- decrypting, in response to the requested usage complying with the usage defined by the at least one usage control element of the general set of control data, the data object and enabling the requested usage, otherwise disabling it.

14. A method as set forth in any one of claims 6-12, comprising the further steps of:

- receiving the data package in a user's data processor;

- storing the data package in a memory device where it is accessible by means of the user's data processor;

- decrypting the at least one usage control element of the user set of control data;

- checking, in response to a request by the user for usage of the data object, whether the requested usage complies with the usage defined by the at least one usage control element of the user set of control data;

- decrypting, in response to the requested usage complying with the usage defined by the at least one usage control element of the user set of control data, the data object and enabling the requested usage, otherwise disabling it.

15. A method as set forth in claims 13 or 14, comprising the further steps of reconcatenating, after the usage of the data object, the data object and the one or more usage control elements, reencrypting at least the data object and the one or more usage control elements, and storing the thus-repackaged data package in the memory of the user's data processor.

16. A method for controlling the usage by a user of a data object so as to comply with predetermined conditions for usage of the data object, comprising the steps of:

- storing a data package in a memory device, where it is accessible by means of a data processor of the user, said data package comprising the data object and control data, which comprises at least one usage control

element defining a usage of the data object which complies with the predetermined conditions, the data object and said at least one usage control element being encrypted;

- receiving a request by the user for usage of the data object;

- decrypting the control data;

- checking, in response to the request by the user for usage of the data object, whether the requested usage complies with the usage defined by the at least one usage control element of the control data;

- decrypting, in response to the requested usage complying with the usage defined by the at least one usage control element of the control data, the data object and enabling the requested usage, otherwise disabling it.

17. A method as set forth in claim 16, wherein the usage control element is updated after the usage of the data object.

18. A method as set forth in claims 16 or 17, wherein said control data comprises an indication of the number of times the user is authorized to use the data object in accordance with said at least one user control element; wherein the requested usage of the data object is only enabled when said number of times is one or more; and wherein said number of times is decremented by one when the requested usage is enabled.

19. A method as set forth in any one of claims 16-18, wherein the control data comprise a security control element, and further comprising the step of carrying out, before each usage of the data object, a security procedure defined in the security control element.

20. A method as set forth in any one of claims 16-19, wherein the step of checking whether the requested usage complies with the usage defined by the at least one usage control element comprises the step of checking that the user's data processor is capable of carrying out the

security procedure specified in the security control element of the user set of control data, and if not, disabling the usage.

21. A method as set forth in any one of claims 16-20, comprising the further steps of reconcatenating, after the usage of the data object, the data object and the one or more usage control elements, reencrypting at least the data object and the one or more usage control elements, and storing the thus-repackaged data package in the memory of the user's data processor.

22. A system for managing a data object so as to comply with predetermined conditions for usage of the data object, comprising

- first means in the data object provider's data processor for creating a general set of control data for the data object based on the predetermined conditions for usage, said general set of control data comprising at least one or more usage control elements defining usages of the data object which comply with the predetermined conditions;

- storing means, which are accessible by means of said data processor, for storing the data object and the general set of control data;

- concatenating means for concatenating the general set of control data with a copy of the data object; and

- encrypting means for encrypting the copy of the data object and at least said one or more usage control elements to create a secure data package, which is ready for transfer to a user.

23. A system as set forth in claim 22, further comprising

- second means in said data processor for creating, in response to a request for authorization for usage of the data object by a user, a user set of control data, which comprises at least a subset of the general set of control data, which subset comprises at least one of said usage control elements; and

- checking means in said data processor for checking that said request for authorization for usage of the data object has been granted before allowing transfer of the data package to the user.

24. A system as set forth in claims 22 or 23, wherein the general set of control data comprises a control data element which defines the right to further distribution of the data object by the user.

25. A system for controlling the usage by a user of a data object so as to comply with predetermined conditions for usage of the data object, comprising

- storing means for storing a data package which comprises a data object and a control data comprising at least one usage control element defining a usage of the data object which complies with the predetermined conditions;

- means for decrypting the at least one usage control element and the data object;

- checking means for checking whether a usage requested by the user complies with the usage defined by said at least one usage control element;

- enabling means for enabling the usage requested by the user when the usage complies with the usage defined by said at least one usage control element; and

- disabling means for disabling the usage requested by the user when the usage does not comply with the usage defined by said at least one usage control element.

26. A system as set forth in claim 25, further comprising means for repackaging the data object after usage thereof.

27. A method for controlling the usage by a user of data objects so as to comply with predetermined conditions for usage of the data objects, comprising the steps of:

- storing at least two data packages in a memory device, where they are accessible by a data processor of the user, each said data package comprising a data object

and a user set of control data, which comprises at least one usage control element defining a usage of the data object which complies with the predetermined conditions, the data object and said at least one usage control elements being encrypted;

- decrypting the usage control elements of the user sets of control data;

- examining the usage control elements of said at least two data packages to find a match;

- using, in response to the finding of a match, the data processor to carry out an action, which is specified in the user sets of control data.

28. A method as set forth in claim 27, comprising the further steps of updating the usage control element of each data package, reconcatenating after the usage of the data objects, each of the data object and its usage control element, reencrypting each of the concatenated data objects and its usage control element and transferring the repackaged data objects to their creators.

Fig 1

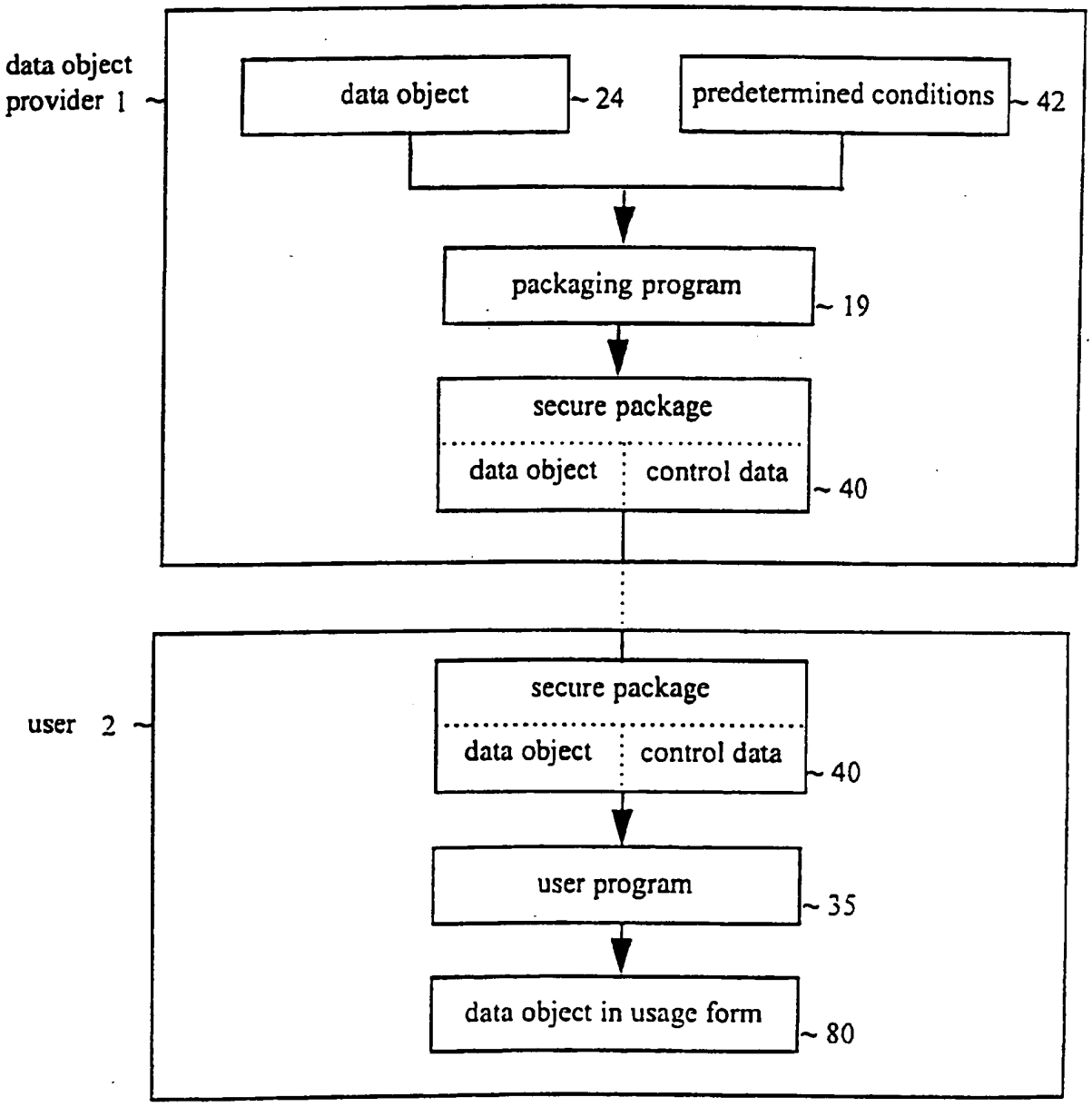


Fig 2

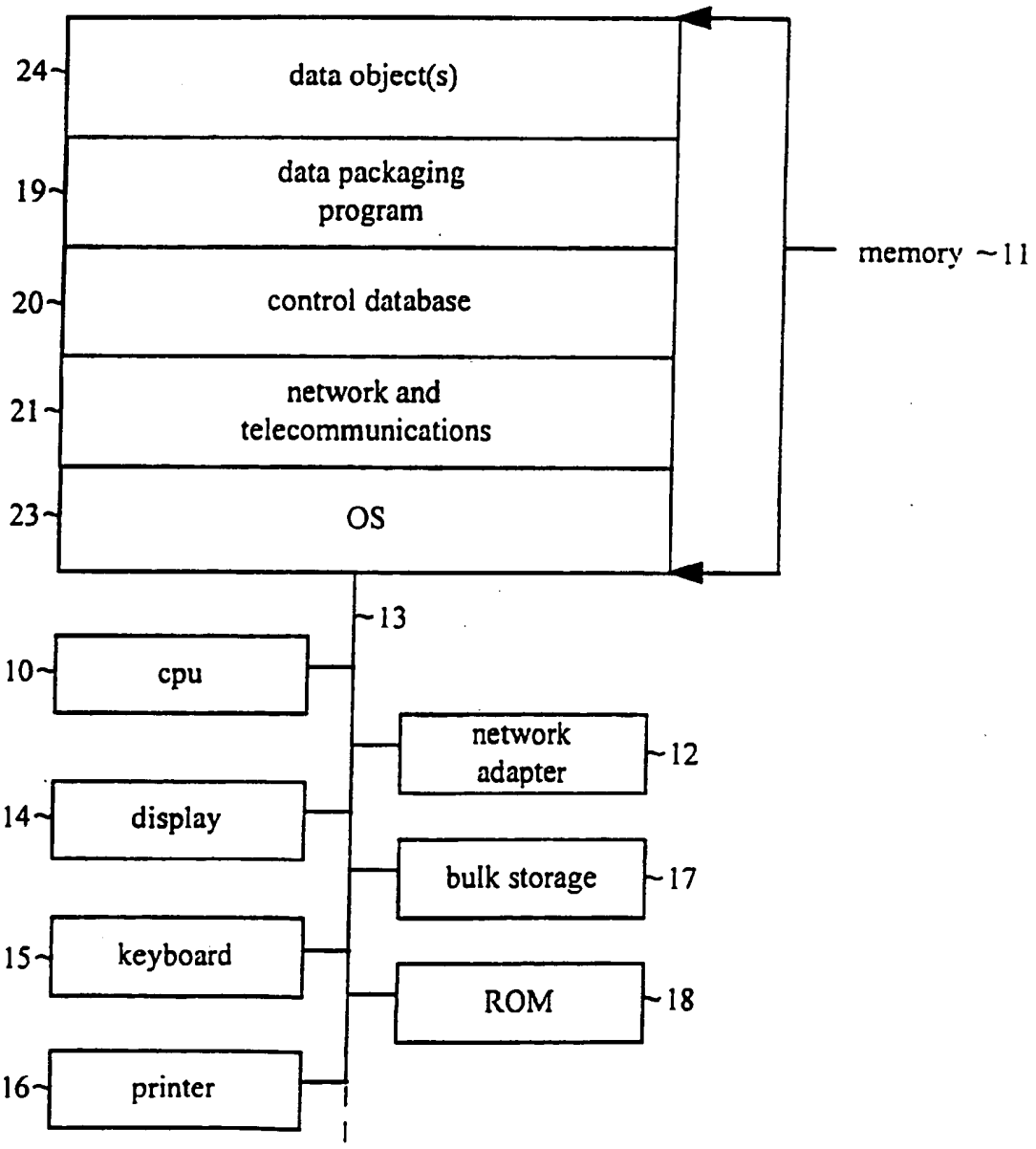


Fig 3

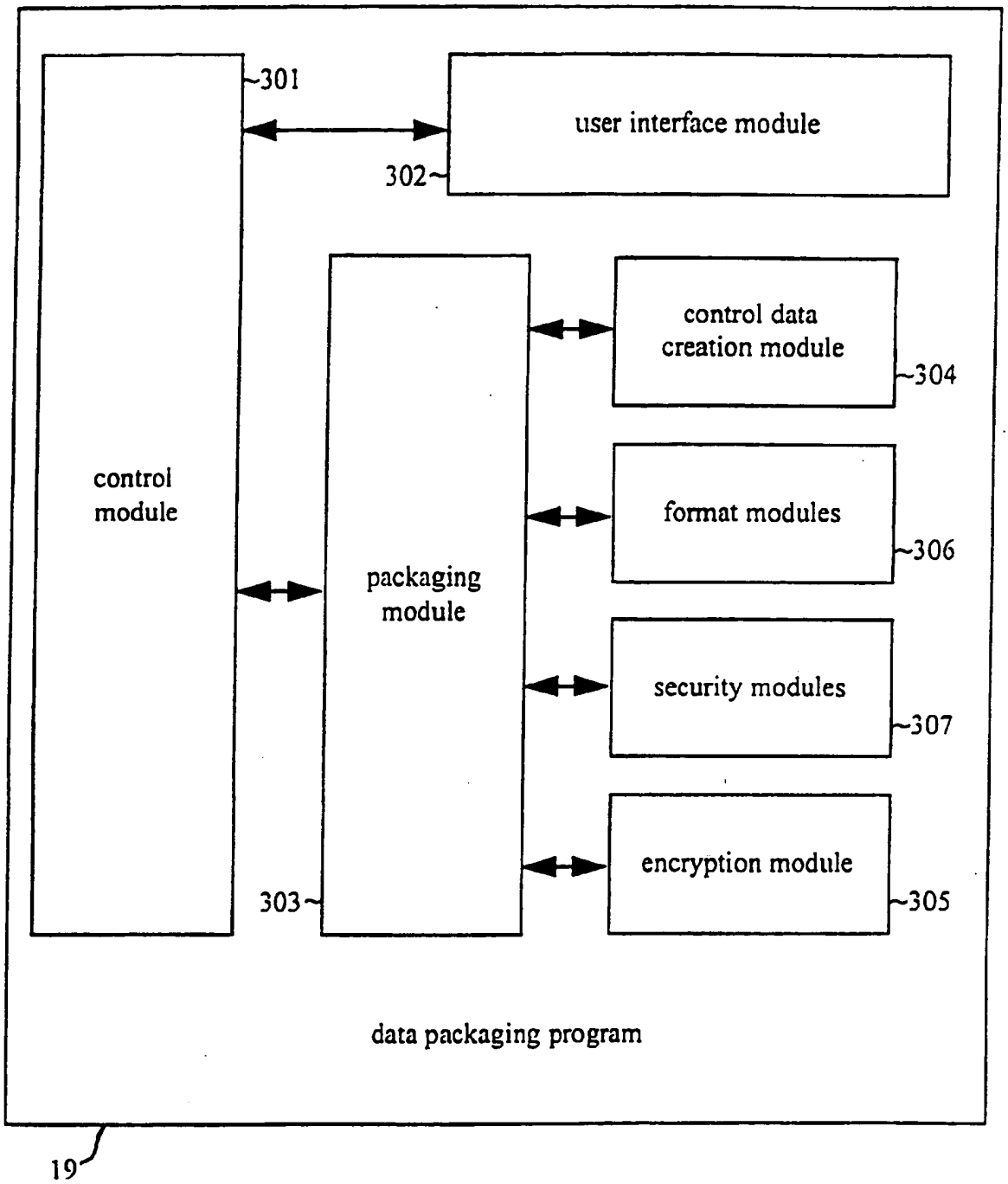


Fig 4

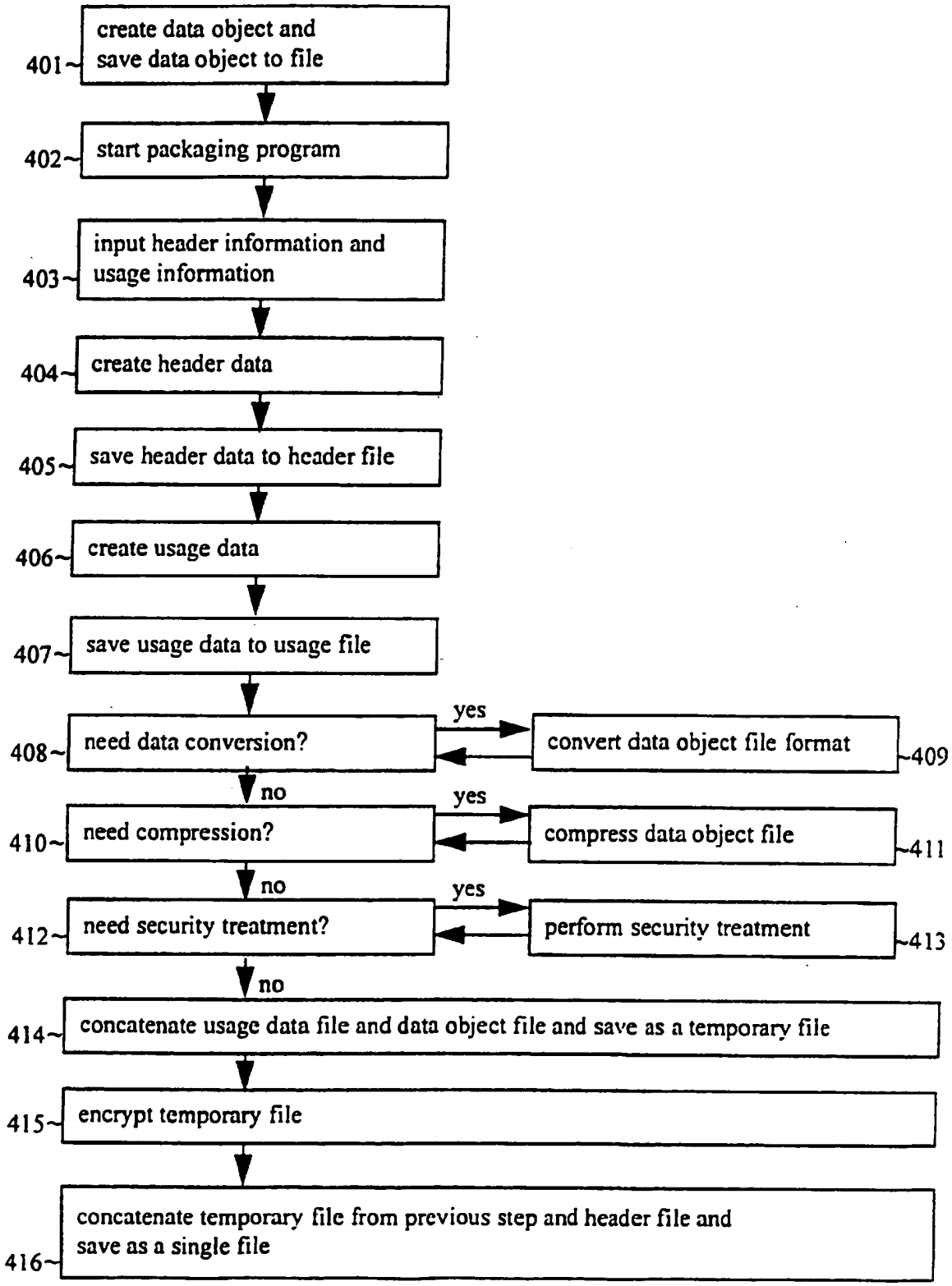


Fig 5

file identifier	123456789
title	image
format code	a
security code	b

Fig 6

usage element for author's phone number	identifer	1
	size	13
	data	716 381 5356
...price for single use	identifer	2
	size	4
	data	.50
...price for unlimited use	identifer	3
	size	4
	data	50.00
...code for usage type approved	identifer	4
	size	2
	data	9
...code for number of usages approved	identifer	5
	size	2
	data	1

Fig 7

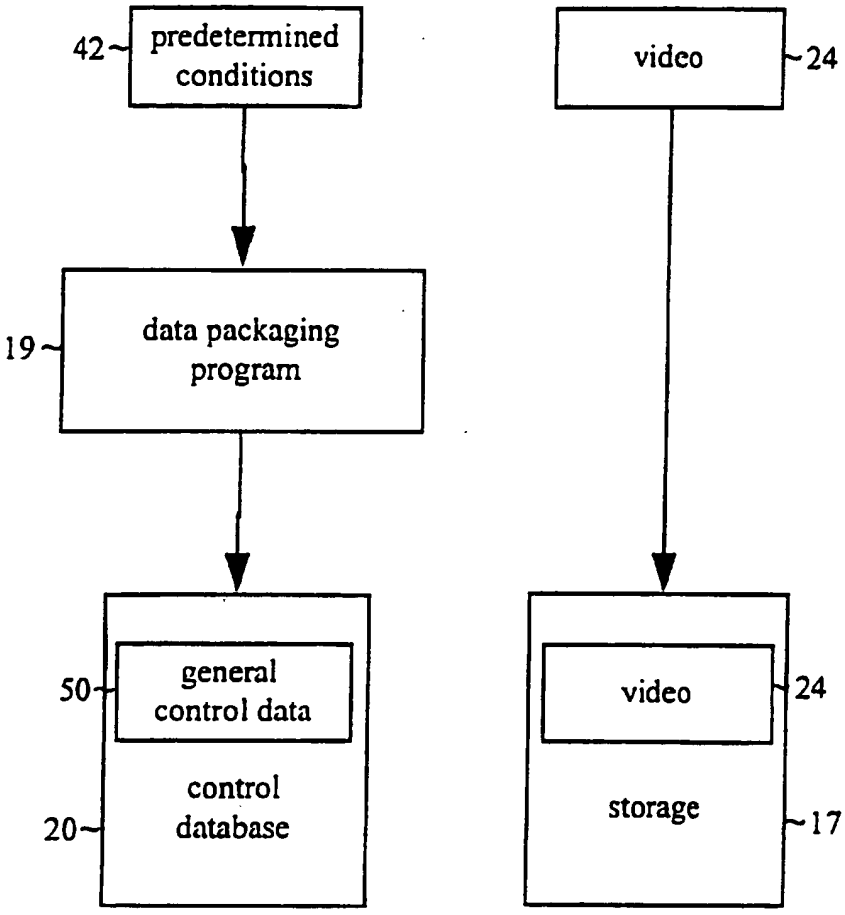


Fig 8a

header	object identifier	123456789
	format code	0010
	security code	0010
	number of usage elements	2
	size of usage data	17
	size of data object	273
	1st usage element id	001
	1st usage element size	6
	1st usage element data	1
	2nd usage element id	002
	2nd usage element size	3
	2nd usage element data	

Fig 8b

header	object identifier	123456790
	format code	0010
	security code	0010
	number of usage elements	2
	size of usage data	17
	size of data object	273
	1st usage element id	001
	1st usage element size	6
	1st usage element data	1
	2nd usage element id	002
	2nd usage element size	3
	2nd usage element data	2

Fig 9

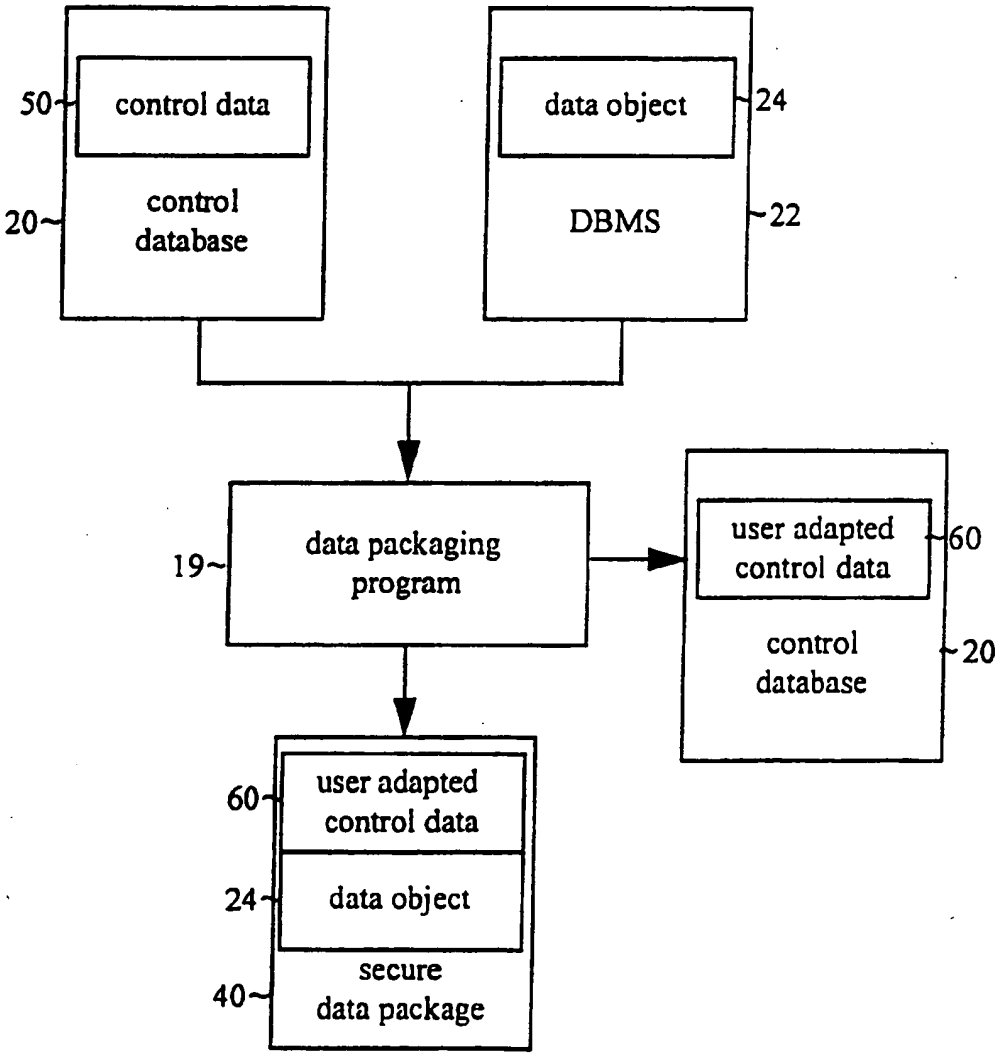


Fig 10

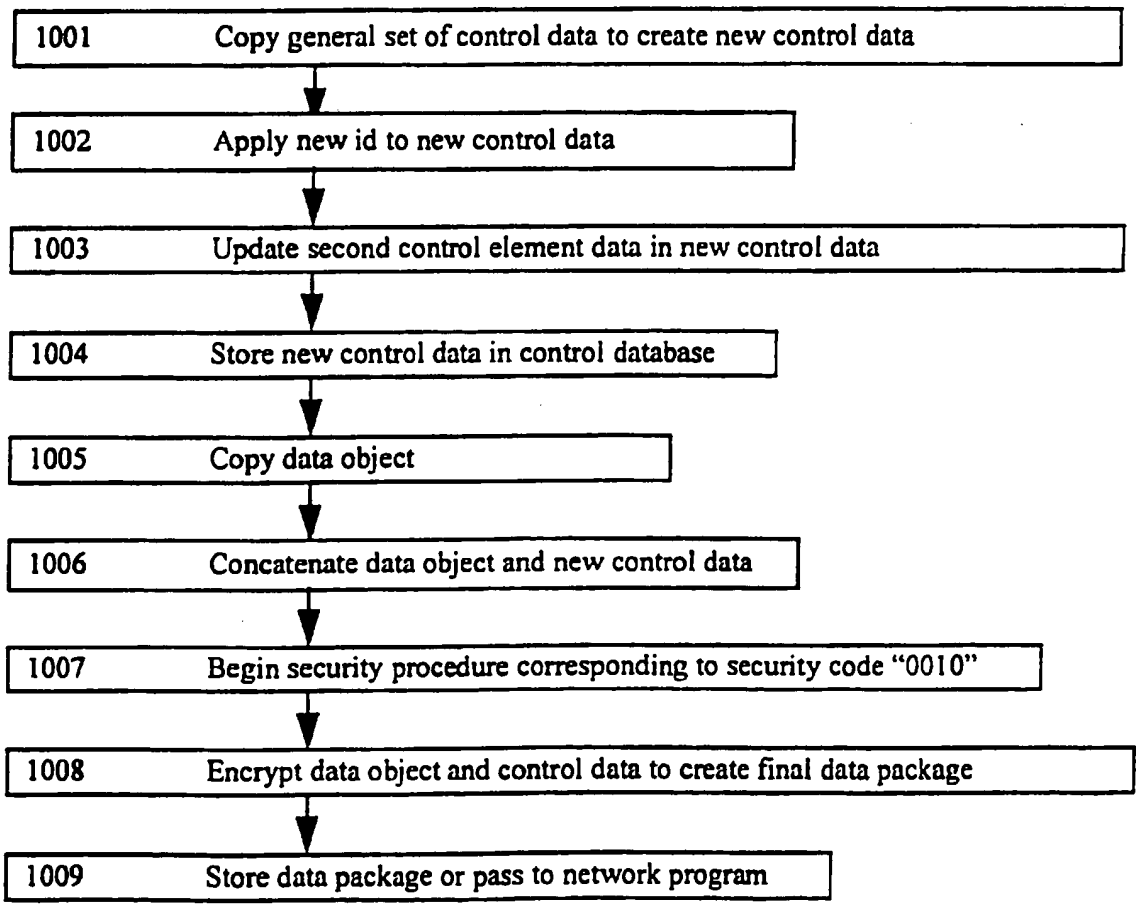
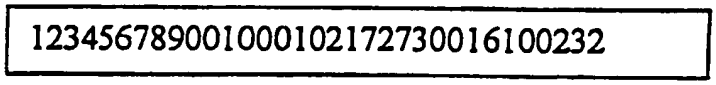


Fig 11

control data in memory



AVI file in memory

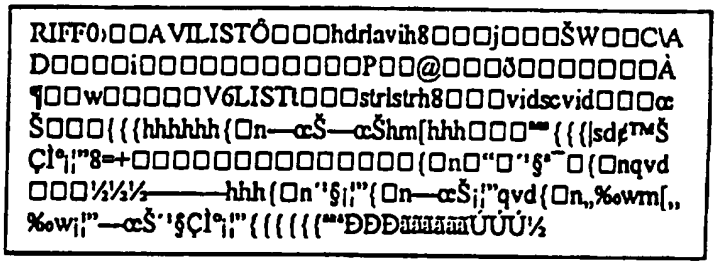


Fig 12a

concatenated control data and AVI file in memory

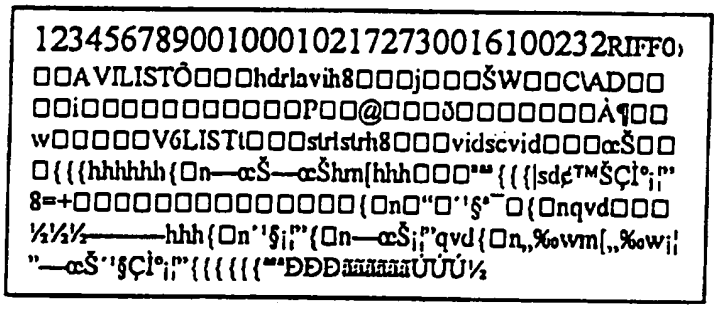


Fig 12b

concatenated and encrypted control data and AVI file in memory

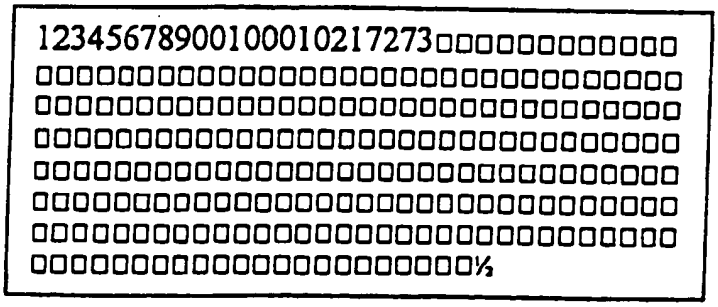


Fig 13

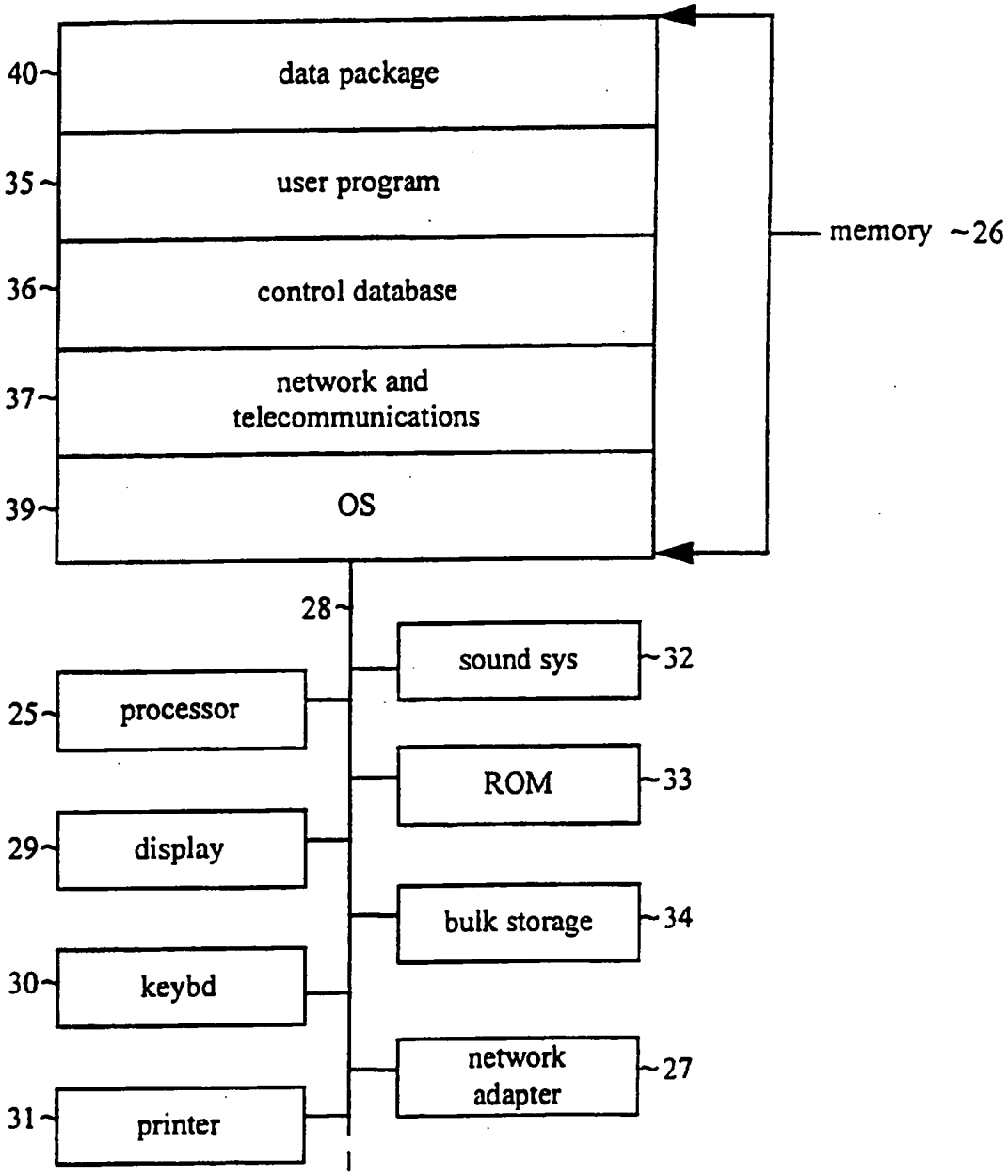


Fig 14

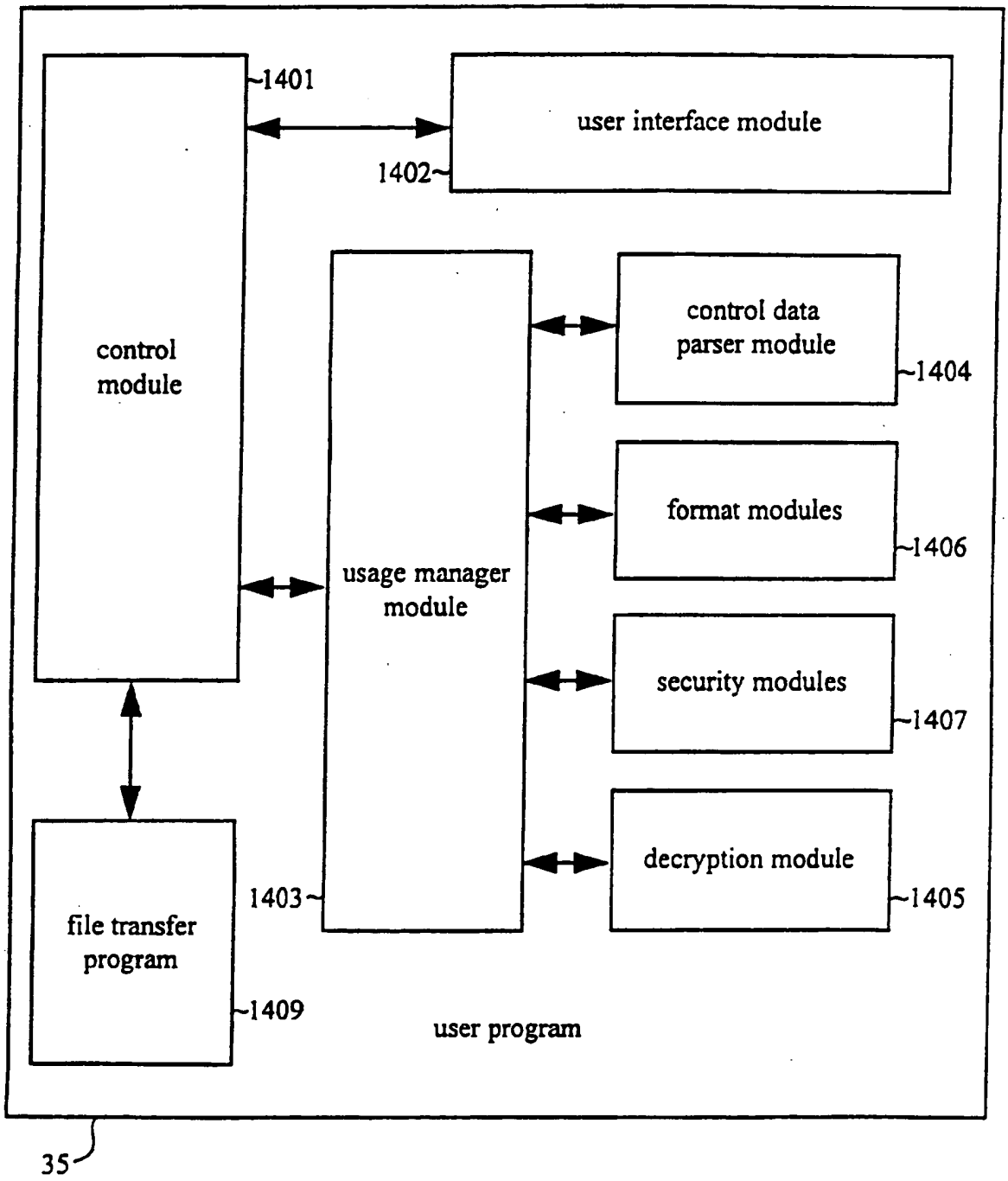


Fig 15

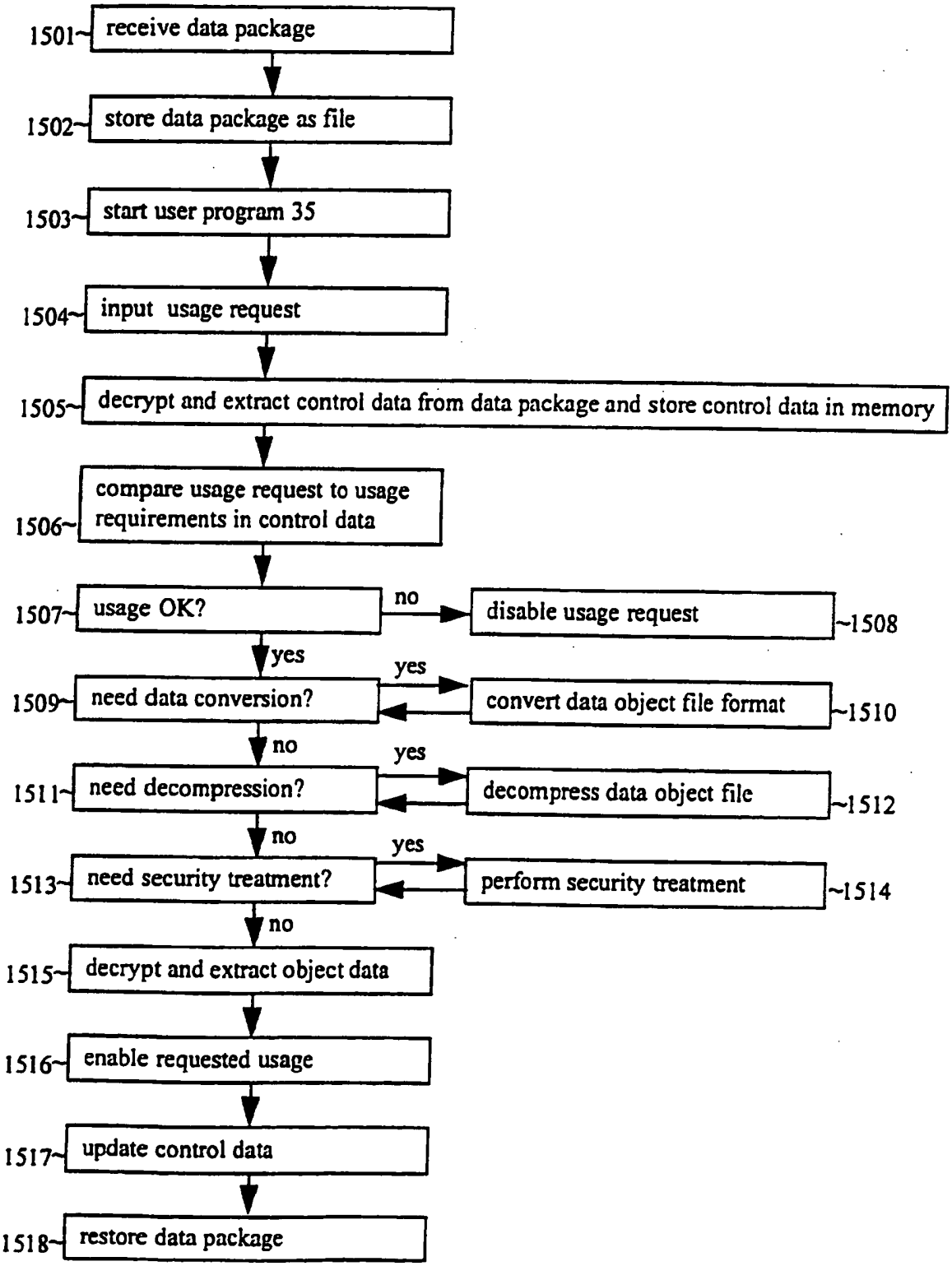


Fig 16

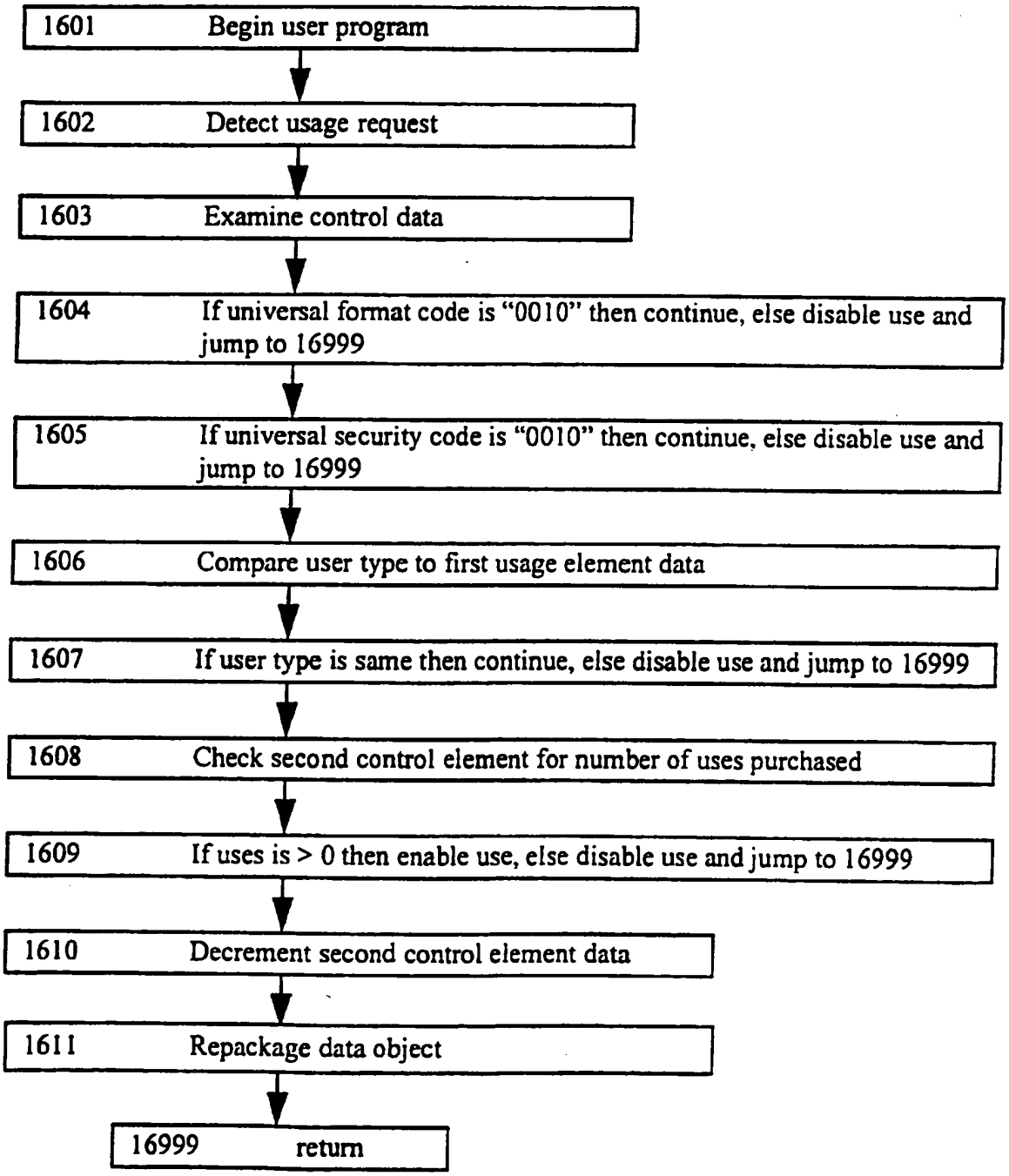
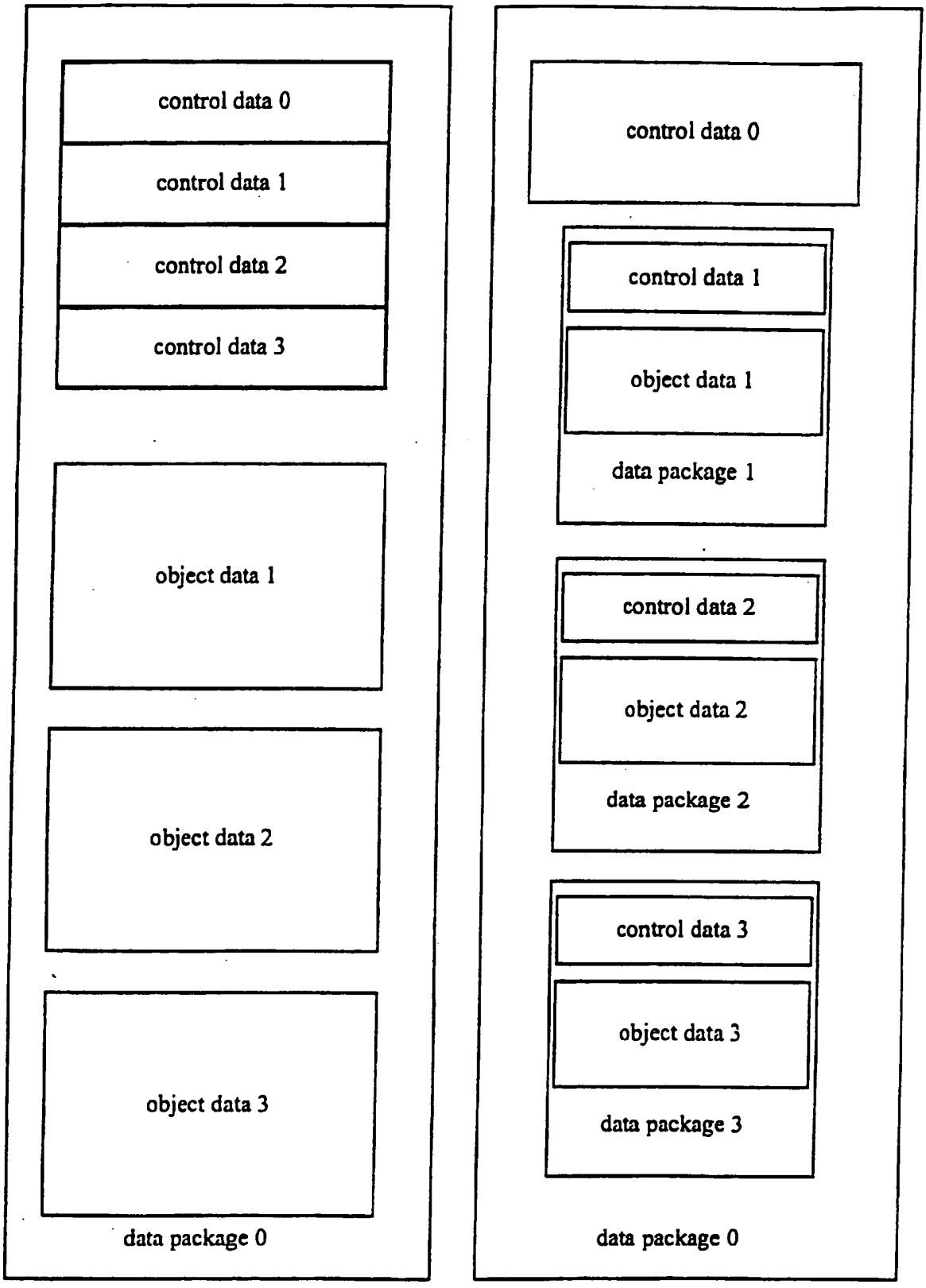


Fig 17



Requested Patent: WO9748203A1

Title: TAMPER RESISTANT METHODS AND APPARATUS ;

Abstracted Patent: US5892899 ;

Publication Date: 1999-04-06 ;

Inventor(s): AUCSMITH DAVID (US); GRAUNKE GARY (US) ;

Applicant(s): INTEL CORP (US) ;

Application Number: US19960662679 19960613 ;

Priority Number(s): US19960662679 19960613 ;

IPC Classification: H04L9/00 ;

Equivalents: AU3488397, AU723556, CA2258087, EP0900488 (WO9748203) ;

ABSTRACT:

In accordance with a first aspect of the present invention, a security sensitive program that operates with a secret is made tamper resistant by distributing the secret in space as well as in time. In accordance with a second aspect of the present invention, a security sensitive program is made tamper resistant by obfuscating the program. In accordance with a third aspect of the present invention, a security sensitive application is made tamper resistant by isolating its security sensitive functions, and making the isolated security sensitive functions tamper resistant by distributing the secrets of the security sensitive functions in time as well as in space, and/or obfuscating the security sensitive functions. In one embodiment where obfuscation is employed, the pseudo-randomly selected pattern(s) of mutations is (are) unique for each installation. In accordance with a fourth aspect of the present invention, a security sensitive system with security sensitive applications is made further tamper resistant by deploying an interlocking trust mechanism. In accordance with a fifth aspect of the present invention, a content industry association, in conjunction with content manufacturers, content reader manufacturers, and content player manufacturers of the industry jointly implement a coordinated encryption/decryption scheme, with the player apparatus manufactured by the content player manufacturers employing playing software that include tamper resistant decryption functions.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04K 1/00 — A1

(11) International Publication Number: WO 97/48203

(43) International Publication Date: 18 December 1997 (18.12.97)

(21) International Application Number: PCT/US97/10359

(22) International Filing Date: 12 June 1997 (12.06.97)

(30) Priority Data:
08/662,679 13 June 1996 (13.06.96) US

(71) Applicant: INTEL CORPORATION [US/US]; 2200 Mission College Boulevard, Santa Clara, CA 95052 (US).

(72) Inventors: AUCSMITH, David; 6995 S.W. Laber Road, Portland, OR 97225 (US). GRAUNKE, Gary; 12120 S.W. Trail Place, Beaverton, OR 97008 (US).

(74) Agents: TAYLOR, Edwin, H. et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

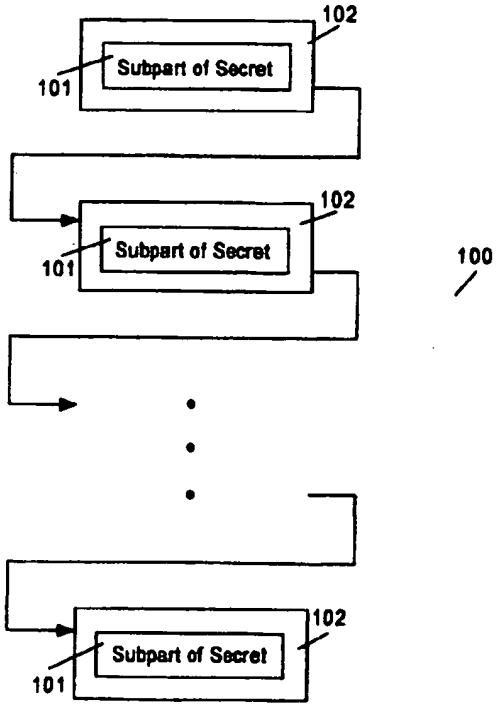
(81) Designated States: AL, AM, AT, AT (Utility model), AU (Petty patent), AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), EE, EE (Utility model), ES, FI, FI (Utility model), GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (Utility model), TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: TAMPER RESISTANT METHODS AND APPARATUS

(57) Abstract

In accordance with a first aspect of the present invention, a security sensitive program (100) that operates with a secret (101) is made tamper resistant by distributing the secret in space as well as in time. In accordance with a second aspect of the present invention, a security sensitive program is made tamper resistant by obfuscating the program. In accordance with a third aspect of the present invention, a security sensitive application is made tamper resistant by isolating its security sensitive functions, and making the isolated security sensitive functions tamper resistant by distributing the secrets of the security sensitive functions in time as well as in space, and/or obfuscating the security sensitive functions. In one embodiment where obfuscation is employed, the pseudo-randomly selected pattern(s) of mutations is (are) unique for each installation. In accordance with a fourth aspect of the present invention, a security sensitive system with security sensitive applications is made further tamper resistant by deploying an interlocking trust mechanism. In accordance with a fifth aspect of the present invention, a content industry association, in conjunction with content manufacturers, content reader manufacturers, and content player manufacturers of the industry jointly implement a coordinated encryption/decryption scheme, with the player apparatus manufactured by the content player manufacturers employing playing software that include tamper resistant decryption functions.



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	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

Tamper Resistant Methods And Apparatus

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to the field of system security. More specifically, the present invention relates to the tamper resistant methods and apparatus.

2. Background Information

Many applications, e.g. financial transactions, unattended authorizations and content management, require the basic integrity of their operations to be assumed, or at least verified. While a number of security approaches such as encryption and decryption techniques are known in the art, unfortunately, the security approaches can be readily compromised, because these applications and the security approaches are implemented on systems with an open and accessible architecture, that renders both hardware and software including the security approaches observable and modifiable by a malevolent user or a malicious program.

Thus, a system based on open and accessible architecture is a fundamentally insecure platform, notwithstanding the employment of security measures. However, openness and accessibility offer a number of advantages, contributing to these systems' successes. Therefore, what is required are techniques that will render software execution virtually unobservable or unmodifiable on these fundamentally insecure platforms, notwithstanding their openness and accessibility. As will be disclosed in more detail below, the present invention of tamper resistant methods and apparatus achieve these and other desirable results.

SUMMARY OF THE INVENTION

In accordance with a first aspect of the present invention, a security sensitive program that operates with a secret is made tamper resistant by distributing the secret in space as well as in time. The secret is partitioned into a number of subparts, and the security sensitive program is unrolled into a number of subprograms

that operate with the subparts, one subpart per subprogram. The subprograms are then executed over a period of time. In one embodiment, the subprograms are further interleaved with unrelated tasks. In one application, the security sensitive program is a decryption program and the secret is a private key.

In accordance with a second aspect of the present invention, a security sensitive program is made tamper resistant by obfuscating the program. The security sensitive program is divided into a number of subprograms, and a plaintext appearance location schedule is selected for the subprograms. An appropriate mutated initial state is determined for each of the subprograms, except for the subprogram where the program's entry point is located. The mutated initial states are determined based on one or more pseudo-randomly selected patterns of mutations that return the program to the initial state at the end of an execution pass. During execution, the subprograms are recovered when they are needed, one or more but not all at a time, following the pseudo-randomly selected pattern(s) of mutations. In one embodiment, each pseudo-randomly selected pattern of mutations is determined using a predetermined partnership function in conjunction with an ordered set of pseudo-random keys. In one application, the security sensitive program is a decryption program that operates with a secret private key. The decryption program may or may not have been made tamper resistant by distributing the secret private key in time as well as in space.

In accordance with a third aspect of the present invention, a security sensitive application is made tamper resistant by isolating its security sensitive functions, and making the isolated security sensitive functions tamper resistant by distributing the secrets of the security sensitive functions in time as well as in space, and/or obfuscating the security sensitive functions. In one embodiment where obfuscation is employed, the pseudo-randomly selected pattern(s) of mutations is (are) unique for each installation. In one application, the application is a content management application having a decryption function.

In accordance with a fourth aspect of the present invention, a security sensitive system with security sensitive applications is made further tamper resistant by providing a system integrity verification program having tamper resistant integrity verification kernels, that jointly deploy an interlocking trust mechanism with the tamper resistant security sensitive functions of the security sensitive applications. In one

application, the system is a content manipulation system, and the application is a content management application.

In accordance with a fifth aspect of the present invention, a content industry association, in conjunction with content manufacturers, content reader manufacturers, and content player manufacturers of the industry jointly implement a coordinated encryption/decryption scheme, with the player apparatus manufactured by the content player manufacturers employing playing software that include tamper resistant decryption functions.

BRIEF DESCRIPTION OF DRAWINGS

The present invention will be described by way of embodiments, but not limitations, illustrated in the accompanying drawings in which like references denote similar elements, and in which:

Figure 1 is a block diagram illustrating a first aspect of the present invention for making a security sensitive program tamper resistant by distributing the program's secret(s) in time and in space;

Figure 2 is a block diagram illustrating one embodiment of the first aspect of the present invention including a subprogram generator for generating the subprograms that operate with corresponding subparts of the distributed secret(s);

Figure 3 is a flow diagram illustrating one embodiment of the operational flow of the subprogram generator of **Figure 2**;

Figure 4 is a block diagram illustrating a second aspect of the present invention for making a security sensitive program tamper resistant by obfuscating the various subparts of the security sensitive program;

Figure 5 is a block diagram illustrating one embodiment of a subpart of the obfuscated program;

Figure 6 is a block diagram illustrating one embodiment of the second aspect of the present invention including an obfuscation processor for generating the obfuscated program;

Figure 7 is a graphical diagram illustrating distribution of key period for the second aspect of the present invention;

Figures 8a - 8b are flow diagrams illustrating one embodiment of the operational flow of the obfuscation processor of **Figure 6**;

Figure 9 is a flow diagram illustrating one embodiment of the operational logic of an obfuscated subprogram of the obfuscated program;

Figures 10 - 14 are diagrams illustrating a sample application of the second aspect of the present invention;

Figure 15 is a block diagram illustrating a third aspect of the present invention for making a security sensitive application tamper resistant;

Figure 16 is a block diagram illustrating a fourth aspect of the present invention for making a security sensitive system tamper resistant;

Figure 17 is a block diagram illustrating a fifth aspect of the present invention for making security sensitive industry tamper resistant; and

Figures 18 - 19 are block diagrams illustrating an example computer system and an embedded controller suitable for programming with the various aspects of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

In the following description, various aspects of the present invention will be described. However, it will be apparent to those skilled in the art that the present invention may be practiced with only some or all aspects of the present invention. For purposes of explanation, specific numbers, materials and configurations are set forth in order to provide a thorough understanding of the present invention. However, it will also be apparent to one skilled in the art that the present invention may be practiced without the specific details. In other instances, well known features are omitted or simplified in order not to obscure the present invention.

Parts of the description will be presented in terms of operations performed by a computer system, using terms such as data, flags, bits, values, characters, strings, numbers and the like, consistent with the manner commonly employed by those skilled in the art to convey the substance of their work to others skilled in the art. As well understood by those skilled in the art, these quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, and otherwise manipulated through mechanical and electrical components of the computer system; and the term computer system include general purpose as well as special purpose data processing machines, systems, and the like, that are standalone, adjunct or embedded.

Various operations will be described as multiple discrete steps in turn in a manner that is most helpful in understanding the present invention, however, the order of description should not be construed as to imply that these operations are necessarily order dependent, in particular, the order of presentation.

Referring now to **Figure 1**, a block diagram illustrating a first aspect of the present invention is shown. In accordance with this first aspect of the present invention, security sensitive program **100** is made tamper resistant by distributing its secret in space as well as in time. The secret (not shown in totality) is "partitioned" into subparts **101**, and program **100** is unrolled into a number of subprograms **102** that operate with subparts **101**; for the illustrated embodiment, one subpart **101** per subprogram **102**. Subprograms **102** are then executed over a period of time. As a result, the complete secret cannot be observed or modified in any single point in space nor in any single point in time.

For example, consider the artificially simple "security sensitive" program for computing the result of X multiply by S, where S is the secret. Assuming S equals to 8, S can be divided into 4 subparts, with each subpart equals 2, and the "security sensitive" program can be unrolled into 4 subprograms with each program computing A = A + (X multiply by 2). Thus, the complete secret 8 can never be observed or modified in any point in space nor time.

As a further example, consider the "security sensitive" program for computing the result of (X to the power of S) modulo Y, where S again is the secret. If S equals 16, S can be divided into 8 subparts, with each subpart equals 2, and the "security sensitive" program can be unrolled into 8 subprograms with each program computing A = (A multiply by ((X to the power of 2) modulo Y)) modulo Y. Thus, the complete secret 16 can never be observed or modified in any point in space nor time.

As will be appreciated by those skilled in the art, the function (X to the power of S) modulo Y is the basis function employed in many asymmetric key (private/public key) schemes for encryption and decryption. Thus, by practicing this first aspect of the present invention, an encryption/decryption function can be made tamper resistant.

In one embodiment, the subprograms are further interleaved with unrelated tasks to further obscure the true nature of the tasks being performed by the unrolled subprograms. The tasks may even have no purpose to them.

Figure 2 illustrates one embodiment of the first aspect of the present invention including a subprogram generator for generating the subprograms. For the illustrated embodiment, subprogram generator **104** is provided with the secret as input. Furthermore, subprogram generator **104** is provided with access to library **105** having entry, basis and prologue subprograms **106**, **108**, and **109** for used in generating subprograms **102** of a particular security sensitive program in view of the secret provided. In other words, entry and basis subprograms **106** and **108** employed are different for different security sensitive programs. For the above illustrated examples, in the first case, entry and basis subprograms **106** and **108** will initialize and compute A = A + (X multiply by a subpart of S), whereas in the second case, entry and basis subprograms **106** and **108** will initialize and compute A = (A multiply by ((X to the power of a subpart of S) modulo Y)) modulo Y. Prologue subprogram **109** is used to perform post processing, e.g. outputting the computed results as decrypted content.

For the illustrated embodiment, entry subprogram **106** is used in particular to initialize an appropriate runtime table **110** for looking up basis values by basis subprogram **108**, and basis subprogram **108** is used to perform the basis computation using runtime table **110**. For the modulo function example discussed above, runtime table **110** is used to return basis values for (X to the power of a subpart of secret) modulo Y for various subpart values, and basis subprogram **108** is used to perform the basis computation of A = (A multiply by (basis value of a subpart of secret)) modulo Y, where A equals the accumulated intermediate results. A's initial value is 1.

For example, entry subprogram **106** may initialize a runtime table **110** of size three for storing the basis values of bv1, bv2 and bv3, where bv1, bv2 and bv3 equal (X to the power of 1) modulo Y, (X to the power of 2) modulo Y, and (X to the power of 3) modulo Y respectively. For the modulo function (X to the power 5) modulo Y, subprogram generator **104** may partition the secret 5 into two subparts with subpart values 3 and 2, and generate two basis programs **108** computing A = (A * Lkup(3)) modulo Y and A = (A * Lkup(2)) modulo Y respectively.

Figure 3 illustrates one embodiment of the operational flow of subprogram generator **104** of **Figure 2**. For the illustrated embodiment, upon invocation, subprogram generator **104** first generates an instance of entry subprogram **106** for initializing at least an appropriate runtime lookup table **110** (Lkup) for returning the basis values of a modulo function for various subparts of a secret, and an accumulation variable (A) to an appropriate initial state, step **112**. Subprogram generator **104** then partitions the secret into subparts, step **114**. In one embodiment, the partition is performed to require the least number of basis programs, within the constraint of the basis values stored in runtime table **110**.

Next, subprogram generator **104** sets a subpart of the secret as the lookup index (LIDX), steps **116**. Then, subprogram generator **104** generates the current basis subprogram to compute A = [A multiply by Lkup (LIDX)] modulo Y, step **118**. Subprogram generator **104** repeats steps **116 - 118** for all subparts, until a basis program has been generated for each subpart of the secret, step **120**. Finally, subprogram generator **104** generates an instance of prologue subprogram **109** for performing post processing, as described earlier, step **122**.

Figure 4 illustrates a second aspect of the present invention. In accordance with this second aspect of the present invention, security sensitive program **203** is made tamper resistant by obfuscating the program. Security sensitive program **203** is divided and processed into a number of obfuscated subprograms **204**. A plaintext (i.e. unmutated) appearance location schedule (i.e. where in memory) is selected for obfuscated subprograms **204**. For the illustrated embodiment, the plaintext appearance location schedule is formulated in terms of the memory cells **202** of two memory segments, memory segment **201a** and memory segment **201b**. Initially, except for the obfuscated subprogram **204** where the program's entry point is located, all other obfuscated subprograms **204** are stored in mutated states. Obfuscated subprograms **204** are recovered or made to appear in plaintext form at the desired memory cells **202**, one or more at a time, when they are needed for execution, and mutated again, once executions are completed. As will be described in more detail below, the initial mutated states, and the process of recovery are determined or performed, in accordance with one or more pseudo-randomly selected pattern of mutations. The pseudo-randomly selected pattern(s) of mutations is (are) determined using a predetermined mutation partnership function in

conjunction with one or more ordered sets of pseudo-random keys. As a result, obfuscated subprograms **204** cyclically mutate back to their respective initial states after each execution pass. Actually, obfuscated subprograms **204** implementing the same loop also cyclically mutate back to the loop entry states after each pass through the loop.

For the illustrated embodiment, each obfuscated subprogram **204** and each cell **202** are of the same size, and first memory segment **201a** is located in high memory, whereas second memory segment **201b** is located in low memory. Furthermore, there are even number of obfuscated subprograms **204**, employing dummy subprogram if necessary.

Figure 5 illustrated one embodiment of subprogram **204**. In accordance with the present invention, for the illustrated embodiment, in addition to original subprogram **102**, obfuscated subprogram **204** is provided with mutation partner identification function **206**, mutation function **207**, partner key **208** and jump block **209**. Original subprogram **102** performs a portion of the functions performed by program **200**. Original subprogram **102** may be an entry/basis/prologue subprogram **106/108/109** in accordance with the first aspect of the present invention. Mutation partner identification function **206** is used to identify the partner memory cells **202** for all memory cell **202** at each mutation round. In one embodiment, the partner identification function **206** is the function: Partner Cell ID = Cell ID XOR Pseudo-Random Key. For a pseudo-random key, mutation partner identification function **206** will identify a memory cell **202** in the second memory segment **201b** as the partner memory cell for of a memory cell **202** in the first memory segment **201a**, and vice versa. Only ordered sets of pseudo-random keys that will provide the required periods for the program and its loops will be employed. The length of a period is a function of the pseudo-random keys' set size (also referred to as key length). Mutation function **207** is used to mutate the content of the various memory cells **202**. In one embodiment, mutation function **207** XORs the content of each memory cell **202** in first memory segment **201a** into the partner memory cell **202** in second memory segment **201b** in an odd mutation round, and XORS the content of each memory cell **202** in second memory segment **201b** into the partner memory cell **202** in first memory segment **201a** in an even mutation round. Partner key **208** is the pseudo-random key to be used by mutation partner identification function **206** to identify mutation partners of the various memory cells **202** for a

mutation round. Jump block **209** transfers execution control to the next obfuscated subprogram **204**, which at the time of transfer, has been recovered into plaintext through the pseudo-random pattern of mutations.

In one embodiment, an obfuscated subprogram **204** may also include other functions being performed for other purposes or simply unrelated functions being performed to further obscure the subpart functions being performed.

Figure 6 illustrates one embodiment of the second aspect of the present invention including an obfuscation processor for processing and transforming subprograms into obfuscated subprograms. For the illustrated embodiment, obfuscation processor **214** is provided with program **200** as inputs. Furthermore, obfuscation processor **214** is provided with access to pseudo-random keys' key length lookup table **212**, mutation partner identification function **206**, and mutation function **207**. For the illustrated embodiment, obfuscation processor **214** also uses two working matrices **213** during generation of obfuscated program **203**.

Key length lookup table **212** provides obfuscation processor **214** with key lengths that provide the required periods by the program and its loops. Key lengths that will provide the required periods is a function of the mutation technique and the partnership function. **Figure 7** illustrates various key lengths that will provide various periods for the first and second memory segment mutation technique and the partnership function described above.

Referring back to **Figure 6**, mutation partner identification function **206** identifies a mutation partner memory cell **202** for each memory cell **202**. In one embodiment, mutation partner identification function **206** identifies mutation partner memory cells in accordance with the "XOR" mutation partner identification function described earlier. Mutation function **207** mutates all memory cells **202**. In one embodiment, mutation function **207** mutates memory cells **202** in accordance with the two memory segments, odd and even round technique described earlier.

For the illustrated embodiment, working matrices **213** include two matrices M1 and M2. Working matrix M1 stores the Boolean functions of the current state of the various memory cells **202** in terms of the initial values of memory cells **202**. Working matrix M2 stores the Boolean functions for recovering the plaintext of

the various obfuscated subprograms **204** in terms of the initial values of memory cells **202**.

Referring now to **Figures 8a - 8b**, two block diagrams illustrating one embodiment of obfuscation processor **214** are shown. For the illustrated embodiment, as shown in **Fig. 8a** in response to a program input (in object form), obfuscation processor **214** analyzes the program, step **216**. In particular, obfuscation processor **214** analyzes branch flow of the program, identifying loops within the program, using conventional compiler optimization techniques known in the art. For the purpose of this application, any execution control transfer, such as a call and subsequent return, is also considered a "loop".

Next, obfuscation processor **214** may perform an optional step of peephole randomization, step **218**. During this step, a peephole randomization pass over the program and replaces code patterns with random equivalent patterns chosen from an optional dictionary of such patterns. Whether it is performed depends on whether the machine architecture of the instructions provide alternate ways of accomplishing the same task.

Then, obfuscation processor **214** restructures and partitions the program **200** into a number of equal size subprograms **204** organized by their loop levels, padding the subprograms **204** if necessary, based on the analysis results, step **220**. Except for very simple program with a single execution path, virtually all programs **200** will require some amount of restructuring. Restructuring includes e.g. removing as well as adding branches, and replicating instructions in different loop levels. Restructuring is also performed using conventional compiler optimization techniques.

Finally, obfuscation processor **214** determines the subprograms' plaintext appearance location schedule, and the initial state values for the various memory cells **202**, step **221**.

Fig. 8b illustrates step **221** in further detail. As shown, obfuscation processor **214** first initializes first working matrix M1, step **222**. Then, obfuscation processor **214** selects a memory cell for the program's entry subprogram to appear in plaintext, step **223**. In one embodiment, the memory cell **202** is arbitrarily selected

(within the proper memory segment **201a** or **201b**). Once selected, obfuscation processor **214** updates the second working matrix M2, step **224**.

Next, obfuscation processor **214** selects an appropriate key length based on the procedure's period requirement, accessing key length table **212**, step **226**. Obfuscation processor **214** then generates an ordered set of pseudo-random keys based on the selected key length, step **228**. For example, if key length equals 5 is selected among the key lengths that will provide a required period of 30, obfuscation processor **214** may randomly select 17, 18, 20, 24 and 16 as the ordered pseudo-random keys.

Next, obfuscation processor **214** determines the partner memory cells **202** for all memory cells **202** using the predetermined mutation partner identification function **206** and the next key in the selected set of ordered pseudo-random keys, step **230**. Upon making the determination, obfuscation processor **214** simulates a mutation, and updates M1 to reflect the results of the mutation, step **232**.

Once mutated, obfuscation processor **214** selects a memory cell for the next subprogram **204** to appear in plaintext, step **234**. Having done so, obfuscation processor **214** updates M2, and incrementally invert M2 using the Guassian Method, step **235**. In one embodiment, instead of incremental inversion, obfuscation processor **214** may just verify M2 remains invertable instead. If M2 is not invertable, obfuscation processor **214** cancels the memory cell selection, and restores M2 to its prior state, step **237**. Obfuscation processor **214** repeats steps **234 - 236** to select another memory cell **202**. Eventually, obfuscation processor **214** becomes successful.

Once succeeded, obfuscation processor **214** determines if there was a loop level change, step **238**. If there was a loop level change, obfuscation processor **214** further determines if the loop level change is down level or up level change, i.e. the subprogram is an entry subprogram of a new loop level or a return point of a higher loop level, step **239**. If the loop level change is "down", obfuscation processor **214** selects another appropriate key length based on the new loop's period requirement, accessing key length table **212**, step **241**. Obfuscation processor **214** then generates a new ordered set of pseudo-random keys based on the newly selected key length, step **242**. The newly generated ordered set of pseudo-random

keys becomes the "top" set of pseudo-random keys. On the other hand, if the loop level change id "up", obfuscation processor **214** restores an immediately "lower" set of pseudo random keys to be the "top" set of pseudo-random keys, step **240**.

Upon properly organizing the "top" set of pseudo-random keys or upon determining there's no loop level change, obfuscation processor **214** again determines the partner memory cells **202** for all memory cells **202** using the predetermined mutation partner identification function **206** and the next key in the "top" set of ordered pseudo-random keys, step **243**. Upon making the determination, obfuscation processor **214** simulates a mutation, and updates M1 to reflect the results of the mutation, step **244**.

Once mutated, obfuscation processor **214** determines if there are more subprograms **204** to process, step **245**. If there are more subprograms **204** to process, obfuscation processor **214** returns to step **234** and proceeds as described earlier. Otherwise, obfuscation processor **214** inserts the mutation partner identification function **206**, the partner key to be used to identify mutation partner memory cells, the mutation function, the jump block, and the address of the next subprogram **204** into each of the obfuscated subprograms **204**, step **246**. Finally, obfuscation processor **214** computes the initial values of the various obfuscated subprograms **204**, and outputs them, steps **247 - 248**.

Figure 9 illustrates one embodiment of the operational flow of an obfuscated subprogram **204**. For the illustrated embodiment, obfuscated subprogram **204** first executes the functions of the original subprogram, step **250**. For embodiments including additional and/or unrelated functions, they may be executed also. Then obfuscated subprogram **204** executes mutation partner identification function **206** to identify the mutation memory cell partners for all memory cells **202** using the stored partner key, step **252**. Having identified the mutation partners, obfuscated subprogram **204** executes mutation function **207** to mutate the memory cells based on the identified partnership.

Next, depending on whether obfuscated subprogram **204** is the last subprogram in an execution pass, obfuscated subprogram **204** either jumps to the next obfuscated subprogram (which should be in plaintext) or returns to the "caller".

Note that if obfuscated subprogram **204** returns to the "caller", all other obfuscated subprograms **204** are in their respective initial states.

Figures 10 - 14 illustrate a sample application of this second aspect of the present invention. **Figure 10** illustrates a sample security sensitive program **200** having six subprograms SPGM0 - SPGM5 implementing a simple single level logic, for ease of explanation, with contrived plaintext values of "000", "001", "010", "011", "100" and "111". Thus, the required period is 6. For ease of explanation, a keylength of one will be used, and the pseudo-random key selected is 3. Furthermore, the mutation partnership identification function is simply Partner Cell ID = Cell ID + 3, i.e. cell 0 always pairs with cell 3, cell 1 pairs with cell 4, and cell 2 pairs with cell 5.

Figure 10 further illustrates at invocation (mutation 0), memory cells (c0 - c5) contains initial values (iv0 - iv5), as reflected by M1. Assuming, cell c0 is chosen for SPGM0, M2 is updated to reflect that the Boolean function for recovering the plaintext of SPGM0 is simply iv0. **Figure 10** further illustrates the values stored in memory cells (c0 - c5) after the first mutation. Note that for the illustrated mutation technique, only the content of the memory cells (c3 - c5) have changed. M1 is updated to reflect the current state. Assuming, cell c3 is chosen for SPGM1, M2 is updated to reflect that the Boolean function for recovering the plaintext of SPGM1 is simply iv0 XOR iv3. Note that for convenience of manipulation, the columns of M2 have been swapped.

Figure 11 illustrates the values stored in memory cells (c0 - c5) after the second, third and fourth mutations. As shown, the content of half of the memory cells (c0 - c5) changed alternatingly after each mutation. In each case, M1 is updated to reflect the current state. Assuming, cells c1, c4 and c2 are chosen for SPGM2, SPGM3 and SPGM4 respectively after the second, third and fourth mutations respectively, in each case M2 is updated to reflect that the Boolean functions for recovering the plaintexts of SPGM2, SPGM3 and SPGM4, i.e. iv4, iv1, and iv2 XOR iv5.

Figure 12 illustrates the values stored in memory cells (c0 - c5) after the fifth mutation. As shown, the content of memory cells (c3 - c5) changed as in previous odd rounds of mutation. M1 is updated to reflect the current state.

Assuming, cell c5 is chosen for SPGM5, M2 is updated to reflect that the Boolean function for recovering the plaintext of SPGM5 is iv5.

Figure 13 illustrates how the initial values iv0 - iv5 are calculated from the inverse of M2, since M2 x ivs = SPGMs, ivs = M2-1 x SPGMs. Note that a "1" in M2-1 denotes the corresponding SPGM is selected, whereas a "0" in M2-1 denotes the corresponding SPGM is not selected, for computing the initial values (iv0 - iv5).

Figure 14 illustrates the content of the memory cells of the above example during execution. Note that at any point in time, at most only two of the subprograms are observable in their plaintext forms. Note that the pairing of mutation partners is fixed only because of the single pseudo-random key and the simple mutation partner function employed, for ease of explanation. Note also that with another mutation, the content of the memory cells are back to their initial states. In other words, after each execution pass, the subprograms are in their initial states, ready for another invocation.

As will be appreciated by those skilled in the art, the above example is unrealistically simple for the purpose of explanation. The plaintext of a subprogram contains many more "0" and "1" bits, making it virtually impossible to distinguish memory cell storing an obfuscated subprogram in a mutated state from a memory cell storing an obfuscated subprogram in plaintext form. Thus, it is virtually impossible to infer the plaintext appearance location schedule from observing the mutations during execution.

Figure 15 illustrates a third aspect of the present invention. In accordance with this aspect of the present invention, security sensitive application **300** may be made tamper resistant by isolating its security sensitive functions **302** and making them tamper proof by incorporating the first and/or second aspects of the present invention described above.

In employing the above described second aspect of the present invention, different sets of pseudo-random keys will produce a different pattern of mutations, even with the same mutation partner identification function. Thus, copies of the security sensitive application installed on different systems may be made unique by employing a different pattern of mutations through different sets of pseudo-random

keys. Thus, the security sensitive applications installed in different systems are further resistant from class attack, even if the obfuscation scheme is understood from observation on one system.

Figure 16 illustrates a fourth aspect of the present invention. In accordance with this aspect of the present invention, a security sensitive system **400** may be made tamper resistant by making its security sensitive applications **400a** and **400b** tamper resistant in accordance with the first, second and/or third aspects of the present invention described above. Furthermore, security of system **400** may be further strengthened by providing system integrity verification program (SIVP) **404** having a number of integrity verification kernels (IVKs). For the illustrated embodiment, a first and a second level IVK **406a** and **406b**. First level IVK **406a** has a published external interface for other tamper resistant security sensitive functions (SSFs) **402a - 402b** of the security sensitive applications **400a - 400b** to call. Both IVKs are made tamper resistant in accordance with the first and the second aspects of the present invention described earlier. Together, the tamper resistant SSFs **402a - 402b** and IVKs **406a - 406b** implement an interlocking trust mechanism.

In accordance with the interlocking trust mechanism, for the illustrated embodiment, tamper resistant SSF1 and SSF2 **402a - 402b** are responsible for the integrity of security sensitive applications **400a - 400b** respectively. IVK1 and IVK2 **406a - 406b** are responsible for the integrity of SIVP **404**. Upon verifying the integrity of security sensitive application **400a** or **400b** it is responsible for, SSF1/SSF2 **402a - 402b** will call IVK1 **406a**. In response, IVK1 **406a** will verify the integrity of SIVP **404**. Upon successfully doing so, IVK1 **406a** calls IVK2 **406b**, which in response, will also verify the integrity of SIVP **404**.

Thus, in order to tamper with security sensitive application **400a**, SSF1 **402a**, IVK1 **406a** and IVK2 **406b** must be tamper with at the same time. However, because IVK1 and IVK2 **406a - 406b** are also used by SSF2 and any other SSFs on the system, all other SSFs must be tamper with at the same time.

Figure 17 illustrates a fifth aspect of the present invention. In accordance with this aspect of the present invention, content industry association **500**, content manufacturers **502**, content reader manufacturers **510** and content

player manufacturer **506** may jointly implement a coordinated encryption/decryption scheme, with content players **508** manufactured by content player manufacturers **506** employing playing software that include content decryption function made tamper resistant in accordance with the above described various aspects of the present invention.

Content industry association **500** owns and holds secret private encryption key Kciapri. Content industry association **500** encrypts content manufacturer's secret content encryption key Kc and content player manufacturer's public encryption Kppub for the respective manufacturers **502** and **506** using Kciapri, i.e. Kciapri[Kc] and Kciapri[Kppub].

Content manufacturer **502** encrypts its content product Kc[ctnt] and includes with the content product Kciapri[Kc]. Content reader manufacturer **510** includes with its content reader product **512** the public key of content industry association Kciapub, whereas content player manufacturer **506** includes with its content player product **508** content player manufacturer's secret private play key Kppri, content industry association's public key Kciapub, and the encrypted content player public key Kciapri[Kppub].

During operation, content reader product **512** reads encrypted content Kc[ctnt] and the encrypted content encryption key Kciapri[Kc]. Content reader product **512** decrypts Kc using Kciapub. Concurrently, content player product **508** recovers its public key Kppub by decrypting Kciapri[Kppub] using content industry association's public key Kciapub. Content reader product **512** and content player product **508** are also in communication with each other. Upon recovering its own public key, content player product **508** provides it to content reader product **512**. Content reader product **512** uses the provided player public key Kppub to encrypt the recovered content encryption key Kc, generating Kppub[Kc], which is returned to content player product **508**. In response, content player product **508** recovers content encrypt key Kc by decrypting Kppub[Kc] using its own private key Kppri.

Thus, as content reader product **512** reads encrypted content Kc[ctnt], and forwards them to content player product **508**, content player product **508** decrypts them with the recovered Kc, generating the unencrypted content (ctnt). In accordance with the above described aspects of the present invention, the decryption

functions for recovering the content player's manufacturer's public key, and recovering the content encryption key Kc are made tamper resistant.

As will be appreciated by those skilled in the art, in addition to being made tamper resistant, by virtue of the interlocking trust, tampering with the content player product's decryption functions will require tampering of the content industry association, content manufacturer and content reader manufacturer's encryption/decryption functions, thus making it virtually impossible to compromise the various encryption/decryption functions' integrity.

As will be also appreciated by those skilled in the art, a manufacturer may play more than one role in the above described tamper resistant industry security scheme, e.g. manufacturing both the content reader and the content player products, as separate or combined products.

Figure 18 illustrates a sample computer system suitable to be programmed with security sensitive programs/applications with or without SIVP, including industry wise security mechanism, made tamper resistant in accordance with the first, second, third, fourth and/or fifth aspect of the present invention. Sample computer system **600** includes CPU **602** and cache memory **604** coupled to each other through processor bus **605**. Sample computer system **600** also includes high performance I/O bus **608** and standard I/O bus **618**. Processor bus **605** and high performance I/O bus **608** are bridged by host bridge **606**, whereas high performance I/O bus **608** and standard I/O bus **618** are bridged by bus bridge **610**. Coupled to high performance I/O bus **608** are main memory **612**, and video memory **614**. Coupled to video memory **614** is video display **616**. Coupled to standard I/O bus **618** are mass storage **620**, and keyboard and pointing devices **622**.

These elements perform their conventional functions. In particular, mass storage **620** is used to provide permanent storage for the executable instructions of the various tamper resistant programs/applications, whereas main memory **612** is used to temporarily store the executable instructions tamper resistant programs/applications during execution by CPU **602**.

Figure 19 illustrates a sample embedded controller suitable to be programmed with security sensitive programs for a security sensitive apparatus, made

tamper resistant in accordance with the first, second, third, fourth and/or fifth aspect of the present invention. Sample embedded system **700** includes CPU **702**, main memory **704**, ROM **706** and I/O controller **708** coupled to each other through system bus **710**. These elements also perform their conventional functions. In particular, ROM **706** may be used to provide permanent and execute-in-place storage for the executable instructions of the various tamper resistant programs, whereas main memory **704** may used to provide temporary storage for various working data during execution of the executable instructions of the tamper resistant programs by CPU **702**.

Thus, various tamper resistant methods and apparatus have been described. While the methods and apparatus of the present invention have been described in terms of the above illustrated embodiments, those skilled in the art will recognize that the invention is not limited to the embodiments described. The present invention can be practiced with modification and alteration within the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of restrictive on the present invention.

CLAIMS

What is claimed is:

1. An apparatus comprising:

an execution unit for executing programming instructions; and

a storage medium coupled to the execution unit, having stored therein a plurality of programming instruction blocks to be executed by the execution unit during operation, the programming instruction blocks operating on corresponding subparts of a secret distributed among them, and the execution being distributed over a period of time.

2. The apparatus as set forth in claim 1, wherein the programming instruction blocks jointly implement a decryption function, and the secret is a private key.

3. The apparatus as set forth in claim 1, wherein one or more of the programming instruction blocks further perform one or more unrelated tasks to further obscure the operations on the subparts of the secret.

4. A machine implemented method for executing a program that operates on a secret in a tamper resistant manner, the method comprises the steps of:

a) executing a first unrolled subprogram of the program at a first point a time, with the first unrolled subprogram operating on a first subpart of the secret; and

b) executing a second unrolled subprogram of the program at a second point a time, with the second unrolled subprogram operating on a second subpart of the secret.

5. The method as set forth in claim 3, wherein the first and second unrolled subprograms are unrolled subprograms of a decryption function; and the secret is a private key.

6. The method as set forth in claim 3, wherein

step (a) further includes the first unrolled subprogram performing at least a first unrelated task; and

step (b) further includes the second unrolled subprogram performing at least a second unrelated task;

said at least a first and a second unrelated task are performed to further obscure the first and second unrolled subprograms' operation on the first and second subparts of the secret.

7. An apparatus comprising:

an execution unit for executing programming instructions; and

a storage medium having stored therein a plurality of programming instructions to be executed by the execution unit during operation, wherein when executed, in response to a secret being provided, the programming instructions partition the secret into a plurality of subparts, and generate a plurality of programming instruction blocks that operate on the subparts.

8. The apparatus as set forth in claim 7, wherein the apparatus further includes a library having an entry programming instruction block, and a basis programming instruction block, to be accessed by the programming instructions in generating the programming instruction blocks.

9. The apparatus as set forth in claim 7, wherein during execution,

the entry programming instruction block initializes a table of values for use by the basis programming blocks to operate on their corresponding subparts of the secret; and

the basis programming blocks' operations on their corresponding subparts of the secret, include looking up values initialized in the table using the basis programming blocks' corresponding subparts of the secret.

10. A machine implemented method for generating a tamper resistant program to operate on a secret, the method comprising the steps of:

a) receiving the secret;

b) partitioning the secret into a plurality of subparts; and

c) generating a plurality of subprograms to correspondingly operate on the subparts of the secret.

11. The method as set forth in claim 10, wherein step (c) includes accessing a library having an entry subprogram, and a basis subprogram to generate the subprograms.

12. The method as set forth in claim 11, wherein during execution,

the entry subprogram initializes a table of values for use by the basis subprograms to operate on their corresponding subparts of the secret; and

the basis subprograms' operations on their corresponding subparts of the secret, include looking up values initialized in the table using the basis subprograms' corresponding subparts of the secret.

13. An apparatus comprising:

an execution unit for executing programming instructions; and

a storage medium having stored thereon a plurality of programming instruction blocks to be executed by the execution unit, the programming instruction blocks being stored in a mutated form, except for at least one, which is stored in a plaintext form, wherein the mutated programming instruction blocks are recovered into the plaintext form during execution on an as needed basis, one or more but not all at a time.

14. The apparatus as set forth in claim 13, wherein each programming instruction block includes a first programming instruction sub-block for performing a task, a second programming instruction sub-block for computing mutation partners for a plurality of memory cells, a key to be employed in said computation of mutation partners, a third programming instruction sub-block for mutating memory cells in accordance with the computed mutation partnering, and a fourth programming instruction sub-block for transferring execution control to another programming instruction block.

15. The apparatus as set forth in claim 14, wherein the first programming instruction sub-block operates on a subpart of a secret.

16. The apparatus as set forth in claim 14, wherein the second programming instruction sub-block computes the mutation partnering by performing a logical XOR operation on a memory cell's identifier and the key.

17. The apparatus as set forth in claim 14, wherein the key is a member of an ordered set of pseudo-randomly selected members, the ordered set having a set size that will provide a required period for a pattern of memory cell mutations, with the memory cells being partnered for mutation in accordance with the computed mutation partnering using the key.

18. The apparatus as set forth in claim 14, wherein the memory cells are divided into two memory cells groups, and pair-wise partnered by the second programming instruction sub-block, with the partnered memory cells being in different group; and the third programming instruction sub-block performs a logical XOR operation on the contents of each pair of partnered memory cells, and alternating between the two memory cell groups for odd and even mutation rounds, in storing the results of the logical XOR operations

19. A machine implemented method for executing a program, the method comprising:

a) executing a first of a plurality of subprograms generated to obfuscate the program;

b) computing mutation partners for a plurality of memory cells storing the plurality of subprograms, using a key, the subprograms being stored initially in the memory cells in a mutated form, except for at least one, which is stored initially in a plaintext form;

c) mutating the memory cells in accordance with the computed mutation partnering to recover a second of the plurality of subprograms for execution.

20. The method as set forth in claim 19, wherein the first and second subprograms operate on a first and a second subpart of a secret.

21. The method as set forth in claim 19, wherein step (b) comprises performing a logical XOR operation on a memory cell's identifier and the key for each memory cell.

22. The method as set forth in claim 19, wherein the key is a member of an ordered set of pseudo-randomly selected members, the ordered set having a set size that will provide a required period for a pattern of memory cell mutations, with the memory cells being partnered for mutation in accordance with the computed mutation partnering using the key.

23. The method as set forth in claim 19, wherein step (c) comprises performing logical XOR operations on the contents of memory cells of a first memory cell group and the contents of memory cells of a second memory cell group, and storing the results of the logical XOR operations into the first memory cell group if step (c) is being

performed for an odd number of times, and the second memory cell group if step (c) is being performed for an even number of times.

24 The method as set forth in claim 19, wherein the method further comprises the steps of:

d) executing the second of the plurality of subprograms;

e) computing mutation partners for the plurality of memory cells; and

f) mutating the memory cells in accordance with the computed mutation partnering to mutate the first of the plurality of subprograms, and recover a third of the plurality of subprograms for execution.

25. An apparatus comprising:

an execution unit for executing programming instructions; and

a storage medium having stored therein a first plurality of programming instructions to be executed by the execution unit, wherein when executed, in response to a program input, the first plurality of programming instructions generate a plurality of subprograms for the program to obfuscate the program, the subprograms being generated in a mutated form, except for at least one, which is generated in a plaintext form, the subprograms being further generated with logic to recover the subprograms in plaintext form on an as needed basis, one or more but not all at a time.

26. The apparatus as set forth in claim 25, wherein the storage medium further having stored therein a table of keylengths to be accessed by the first plurality of programming instructions in generating the subprograms, the keylengths denoting sizes of ordered sets of pseudo-randomly selected members that will provide various required mutation periods.

27. The apparatus as set forth in claim 25, wherein the storage medium further having stored therein a second plurality of programming instructions to be incorporated into each of the generated subprograms by the first plurality of programming instructions for identifying mutation partners for a plurality of memory cells storing the subprograms, for a mutation round, using a key, the key being a member of an ordered set of pseudo-randomly selected members that will provide a mutation period required by the generated subprograms.

28. The apparatus as set forth in claim 25, wherein the storage medium further having stored therein a second plurality of programming instructions to be incorporated into each of the generated subprograms by the first plurality of programming instructions for mutating memory cells storing the generated subprograms in accordance with computed mutation partnerings for a mutation round.

29. The apparatus as set forth in claim 25, wherein the first plurality of programming instructions include logic for analyzing the program for branch flow.

30. The apparatus as set forth in claim 25, wherein the first plurality of programming instructions include logic for performing peephole randomization on the program.

31. The apparatus as set forth in claim 25, wherein the first plurality of programming instructions include restructuring and partitioning the program into the subprograms.

32. The apparatus as set forth in claim 25, wherein the first plurality of programming instructions include logic for scheduling memory cells for the generated subprograms to be recovered in the plaintext form, and determining the appropriate initial values for the memory cells.

33. The apparatus as set forth in claim 32, wherein the first plurality of programming instructions include logic for determining a mutation period requirement for the program, a keylength for the required mutation period, the keylength denoting a set's set size, the set being an ordered set of pseudo-randomly selected members that will provide the required mutation period.

34. The apparatus as set forth in claim 32, wherein the first plurality of programming instructions include logic for selecting a memory cell for a generated subprogram to be recovered in the plaintext form, and determining a Boolean function for recovering the generated subprogram in the plaintext form in terms of initial state values of the memory cells used for storing the generated subprograms.

35. The apparatus as set forth in claim 32, wherein the first plurality of programming instructions include logic for determining mutation partners for a

plurality of memory cells storing the generated subprograms, using a key of an ordered set of pseudo-randomly selected keys, simulating memory cell mutations in accordance with the determined mutation partnering, and determining a plurality of Boolean functions for the memory cells, the Boolean functions expressing the post mutation states of the memory cells in terms of the memory cells' initial values.

36. A machine implemented method for generating a plurality of subprograms for a program to obfuscate the program, the method comprising the steps:

a) analyzing the program for branch flow;

b) restructuring and partitioning the program into a plurality of subprograms; and

c) determining a schedule in terms of a plurality of memory cells for recovering the subprograms in a plaintext form for execution, and initial state values for the memory cells to store the subprograms in the memory cells in a mutated form, except for at least, which is stored in one of the memory cells in the plaintext form.

37. The machine as set forth in claim 36, wherein step (a) further includes performing peephole randomization on the program.

38. The method as set forth in claim 36, wherein step (c) includes determining a mutation period requirement for the program, a keylength for the required mutation period, the keylength denoting a set's set size, the set being an ordered set of pseudo-randomly selected members that will provide the required mutation period.

39. The method as set forth in claim 36, wherein step (c) includes selecting a memory cell for a generated subprogram to be recovered in the plaintext form, and determining a Boolean function for recovering the generated subprogram in the plaintext form in terms of initial state values of the memory cells used for storing the generated subprograms.

40. The method as set forth in claim 36, wherein step (c) includes determining mutation partners for a plurality of memory cells storing the generated subprograms, using a key of an ordered set of pseudo-randomly selected keys, simulating memory cell mutations in accordance with the determined mutation partnering, and determining a plurality of Boolean functions for the memory cells, the Boolean

functions expressing the post mutation states of the memory cells in terms of the memory cells' initial values.

41. The method as set forth in claim 36, wherein the method further includes step (d) inserting a function and a key into each of the generated subprograms, the function being used for identifying mutation partners for a plurality of memory cells storing the subprograms, for a mutation round, using the key, the key being a member of an ordered set of pseudo-randomly selected members that will provide a mutation period required by the generated subprograms.

42. The method as set forth in claim 36, wherein the method further includes step (d) inserting a function into each of the generated subprograms for mutating memory cells storing the generated subprograms in accordance with computed mutation partnerings for a mutation round.

43. An apparatus comprising:
an execution unit for executing programming instructions;
a storage medium having stored therein a first and a second plurality of programming instructions to be executed by the execution unit, the first and second plurality of programming instructions implementing an application with the first plurality of programming instructions implementing a security sensitive function of the application and the second plurality of programming instructions implementing a non-security sensitive function of the application, the first plurality of programming instructions having incorporated a first defensive technique of distributing a secret in space and in time and/or a second defensive technique of obfuscation to render the first plurality of programming instructions virtually unobservable and unmodifiable during execution.

44. The apparatus as set forth in claim 43, wherein the first plurality of programming instructions incorporated the second defensive technique of obfuscation, including one or more unique ordered sets of pseudo-randomly selected members for generating one or more patterns of memory cell mutations, rendering the application unique from other copies of the application installed on other apparatus.

45. An apparatus comprising:
an execution unit for executing programming instructions;

a storage medium having stored therein a first, a second, a third, and a fourth, plurality of programming instructions to be executed by the execution unit, the first and second plurality of programming instructions implementing a first and a second integrity verification function for a first and a second application respectively, whereas the third and fourth programming instructions implementing a third and a fourth integrity verification function for a system integrity verification program, all four pluralities of programming instructions having incorporated defensive techniques rendering them tamper resistant, the four pluralities of programming instructions jointly implementing an interlocking trust mechanism, requiring the first and the second pluralities of programming instructions each to cooperate with both the third and fourth pluralities of programming instructions to complete any integrity verification on the apparatus.

46. A machine implemented method for verifying integrity on an apparatus, the method comprising the steps of:

a) a first and a second tamper resistant integrity verification function of a first and a second application of the apparatus individually calling a third tamper resistant integrity verification function of a system integrity verification program to jointly perform integrity verification with the first and second tamper resistant integrity verification functions respectively;

b) in response, the third tamper resistant integrity verification function calling a fourth tamper resistant Integrity verification function of the system integrity verification program to jointly perform the requested integrity verifications;

c) the fourth tamper resistant integrity verification function providing the first and the second tamper resistant integrity verification functions with respective results of the requested integrity verifications.

47. An apparatus comprising:

an execution unit for executing programming instructions;

a storage medium having stored therein a first and a second plurality of programming instructions to be executed by the execution unit, and a first secret private key, the first and second pluralities of programming instructions implementing a first and a second tamper resistant decryption function,

the first tamper resistant decryption function being used for recovering a first public key asymmetric to the first secret private key, using a second public

key, the first public key having been previously encrypted using a second secret private key asymmetric to the second public key,

the second tamper resistant decryption function being used for recovering a content encryption key using the first secret private key, the content encryption key having been previously encrypted using the first public key.

48. The apparatus as set forth in claim 47, wherein the storage medium further having stored therein a third plurality of programming instructions to be executed by the execution unit, the third plurality of programming instructions implementing a third decryption function for recovering content using the recovered content encryption key, the content having been previously encrypted using the content encryption key.

49. A machine implemented method for recovering content, the method comprising the steps of:

a) recovering a first public key using a second public key, the first and second public keys having a first and a second asymmetric private key respectively, the first public key having been previously encrypted by the second private key;

b) providing the recovered first public key to be used for encrypting a content encryption key;

c) receiving the encrypted content encryption key; and

d) recovering the content encryption key using the first private key.

50. The method as set forth in claim 47, wherein the method further comprises the steps of:

e) receiving encrypted content; and

f) recovering content using the recovered content encryption key.

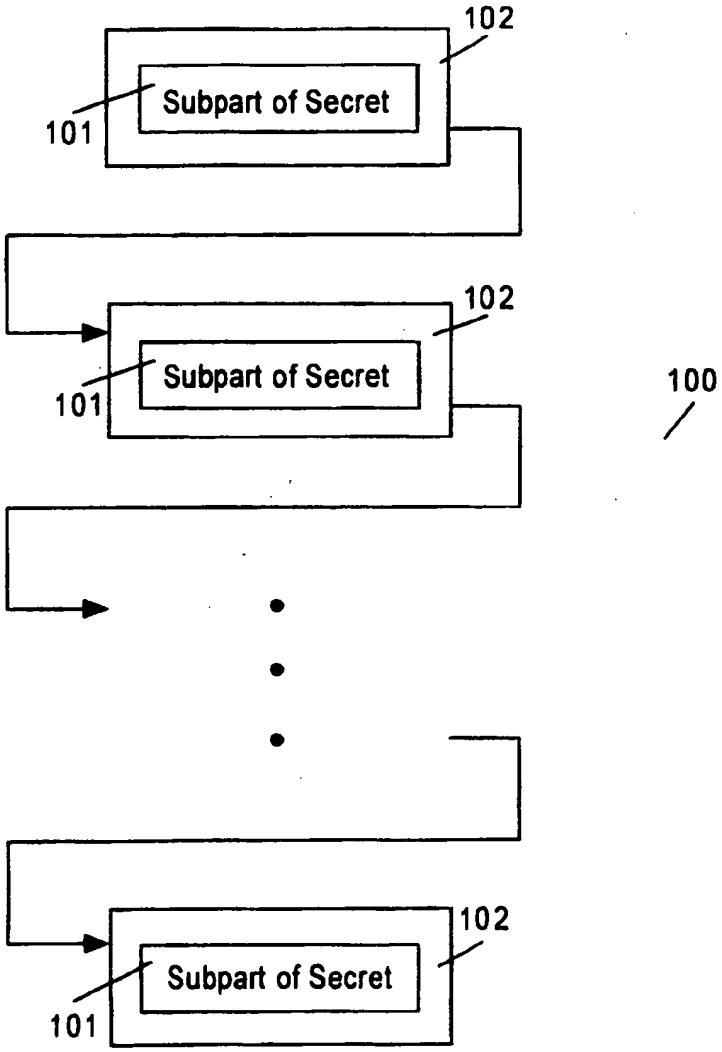

102
Subpart of Secret
101
102
Subpart of Secret
101
100
102
Subpart of Secret
101

Figure 1

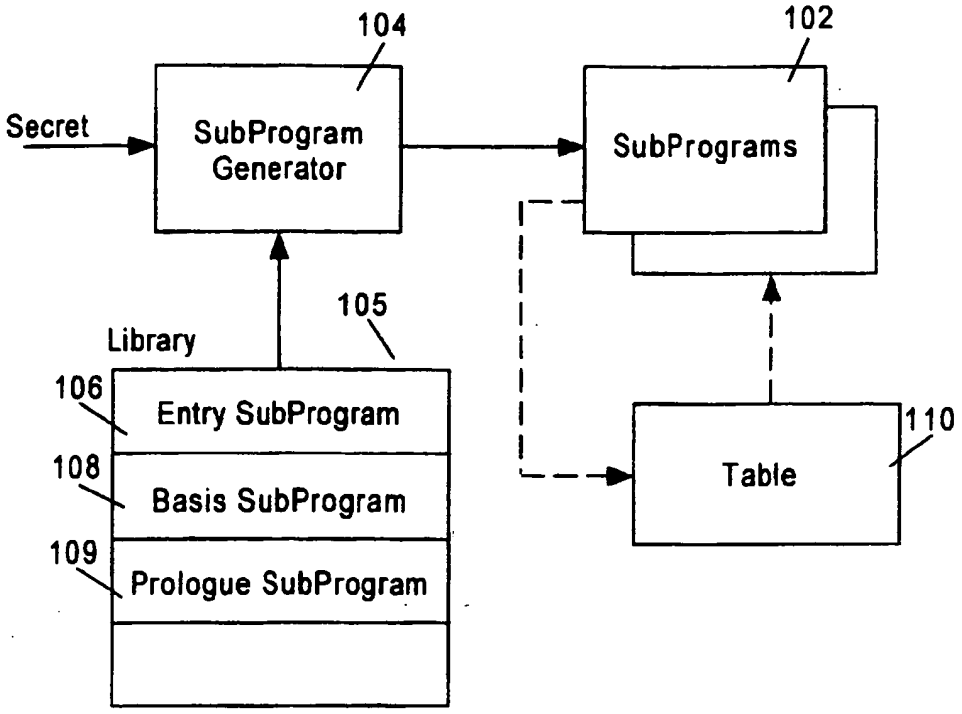

104
102
Secret
SubProgram Generator
SubPrograms
105
Library
106
Entry SubProgram
108
Basis SubProgram
109
Prologue SubProgram
110
Table

Figure 2

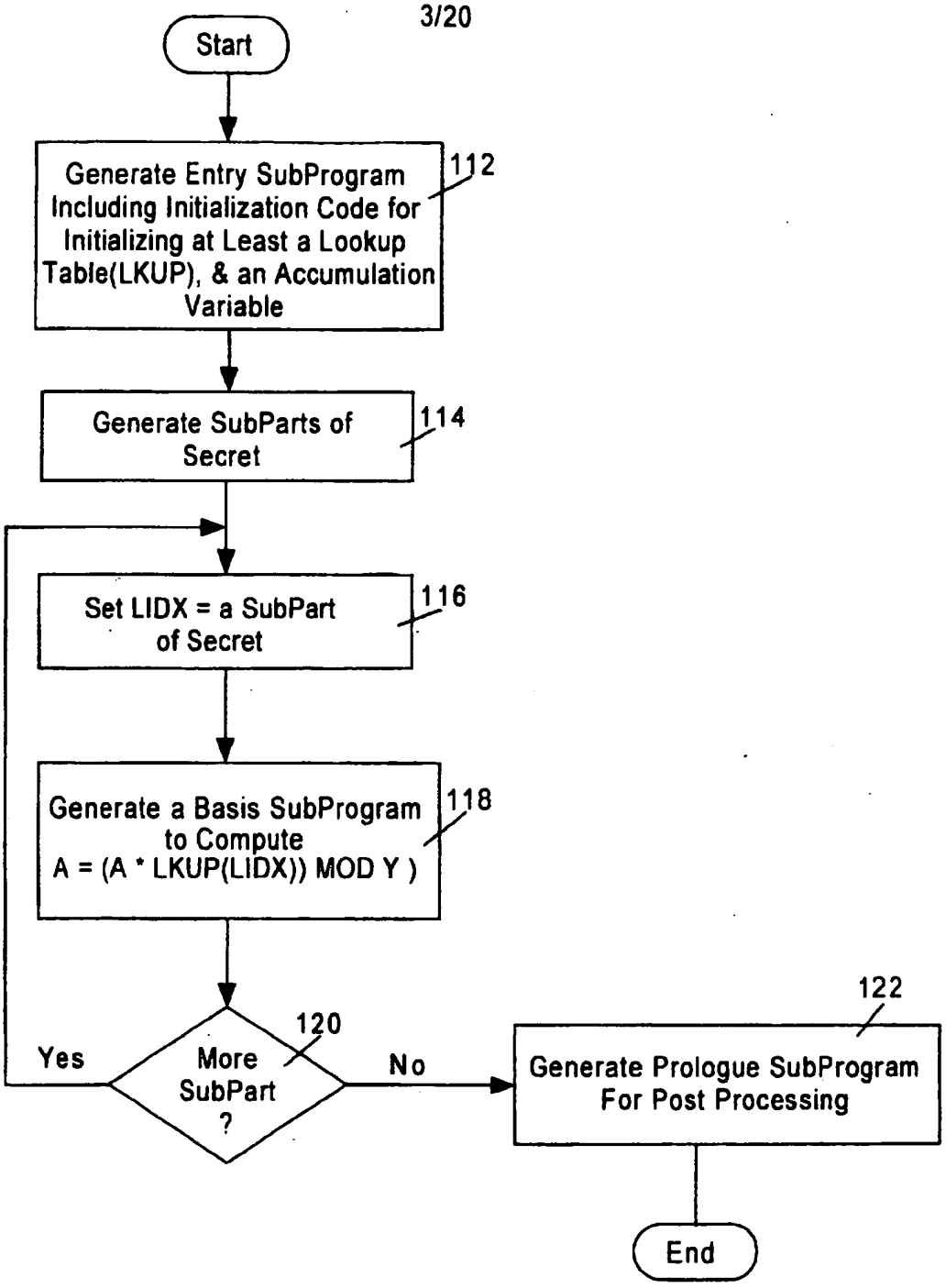

Start
Generate Entry SubProgram Including Initialization Code for Initializing at Least a Lookup Table(LKUP), & an Accumulation Variable
112
Generate SubParts of Secret
114
Set LIDX = a SubPart of Secret
116
Generate a Basis SubProgram to Compute A = (A * LKUP(LIDX)) MOD Y)
118
More SubPart ?
120
Yes
No
Generate Prologue SubProgram For Post Processing
122
End

Figure 3

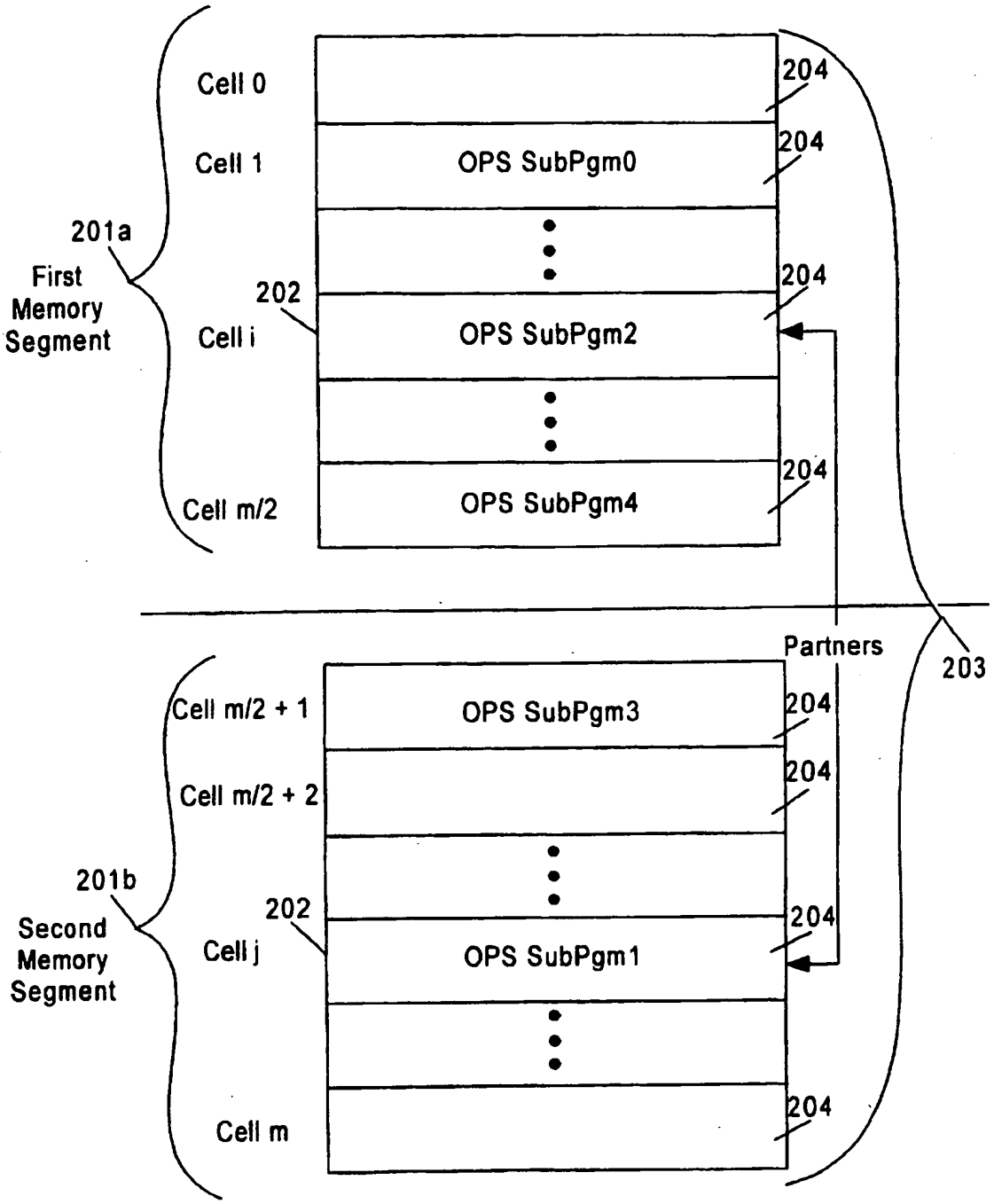

Cell 0
Cell 1
OPS SubPgm0
201a
First
Memory
Segment
202
Cell i
OPS SubPgm2
Cell m/2
OPS SubPgm4
204
Partners
203
Cell m/2 + 1
OPS SubPgm3
Cell m/2 + 2
201b
Second
Memory
Segment
Cell j
OPS SubPgm1
Cell m

Figure 4

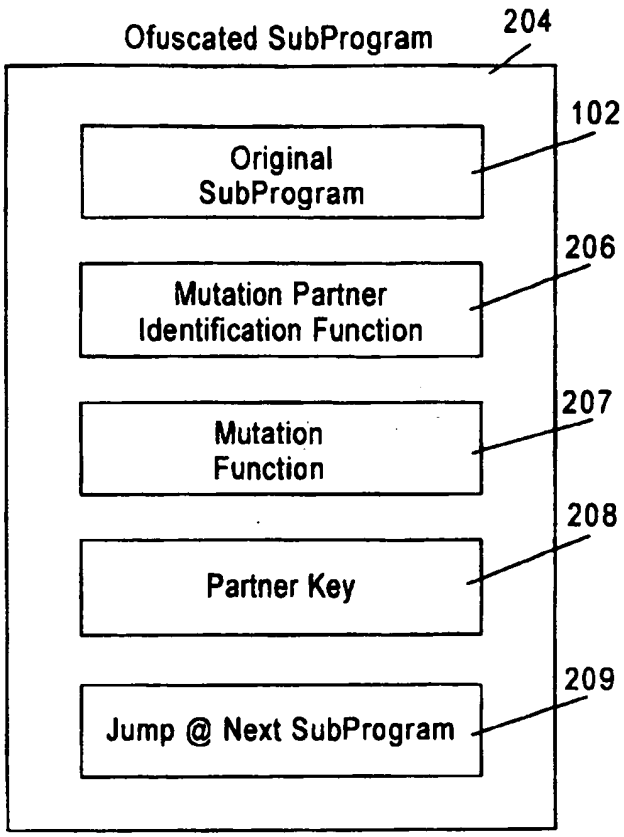

Ofuscated SubProgram
204
Original SubProgram
102
Mutation Partner Identification Function
206
Mutation Function
207
Partner Key
208
Jump @ Next SubProgram
209

Figure 5

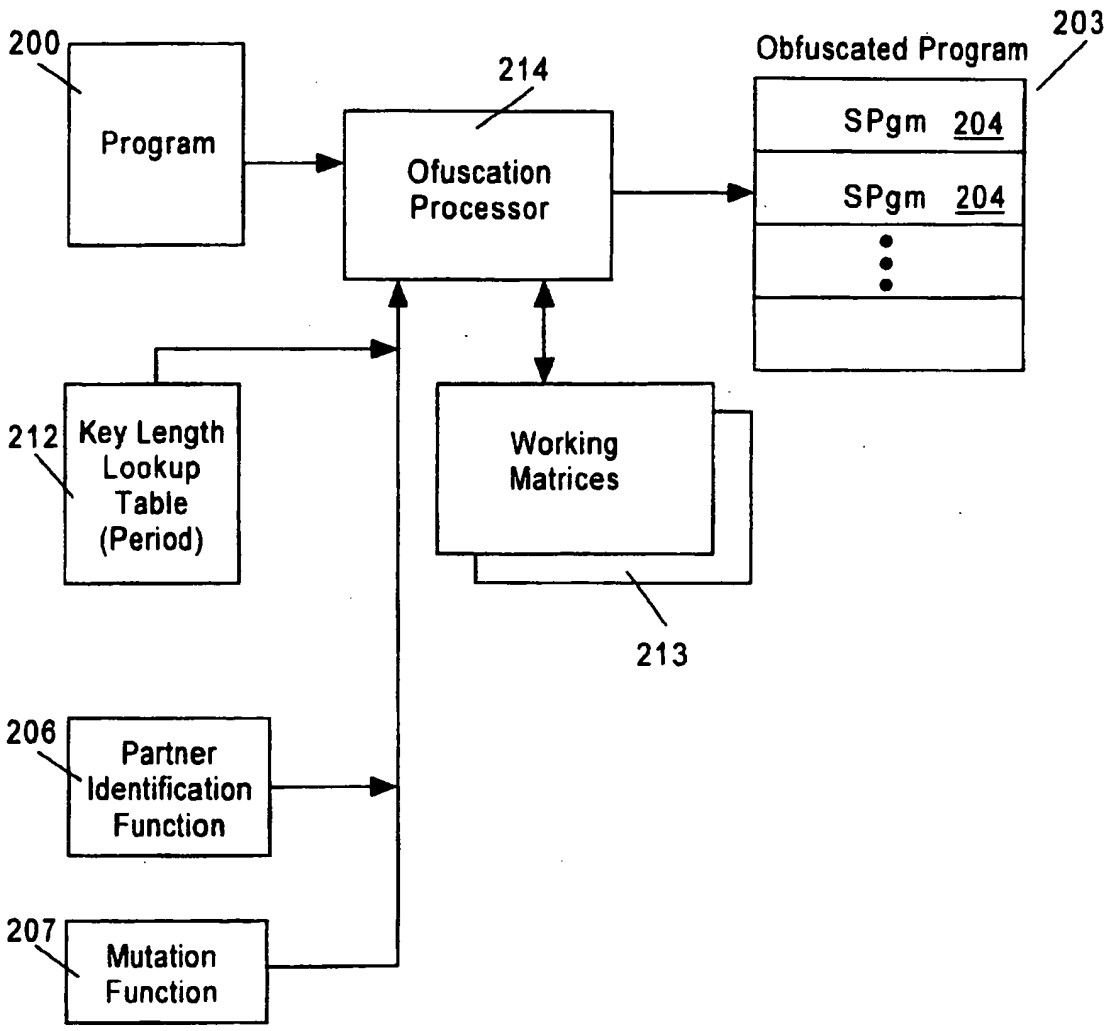

200
Program
214
Ofuscation Processor
Obfuscated Program
203
SPgm 204
SPgm 204
212
Key Length Lookup Table (Period)
Working Matrices
213
206
Partner Identification Function
207
Mutation Function

Figure 6

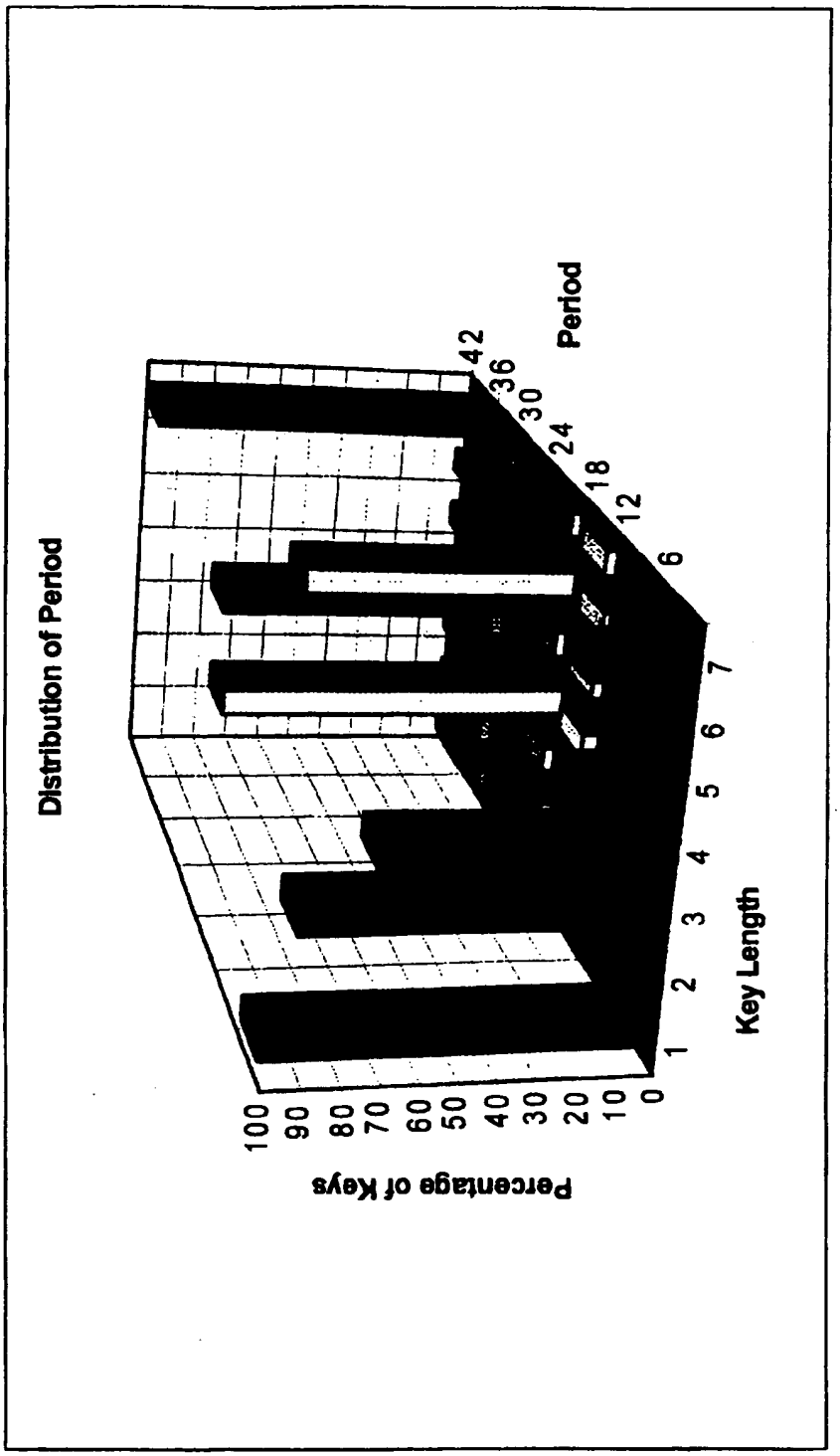

Distribution of Key Period
Distribution of Period
Period
42
36
30
24
18
12
6
Key Length
1
2
3
4
5
6
7
Percentage of Keys
100
90
80
70
60
50
40
30
20
10
0

Figure 7

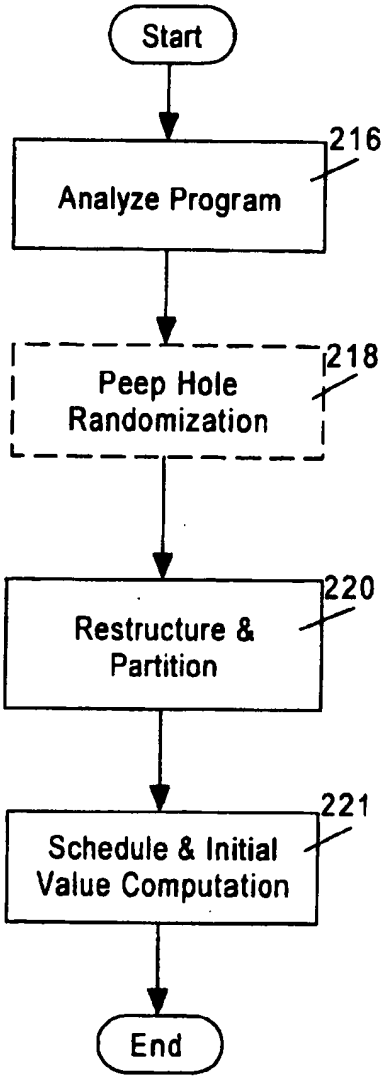

Start
Analyze Program
216
Peep Hole Randomization
218
Restructure & Partition
220
Schedule & Initial Value Computation
221
End

Figure 8a

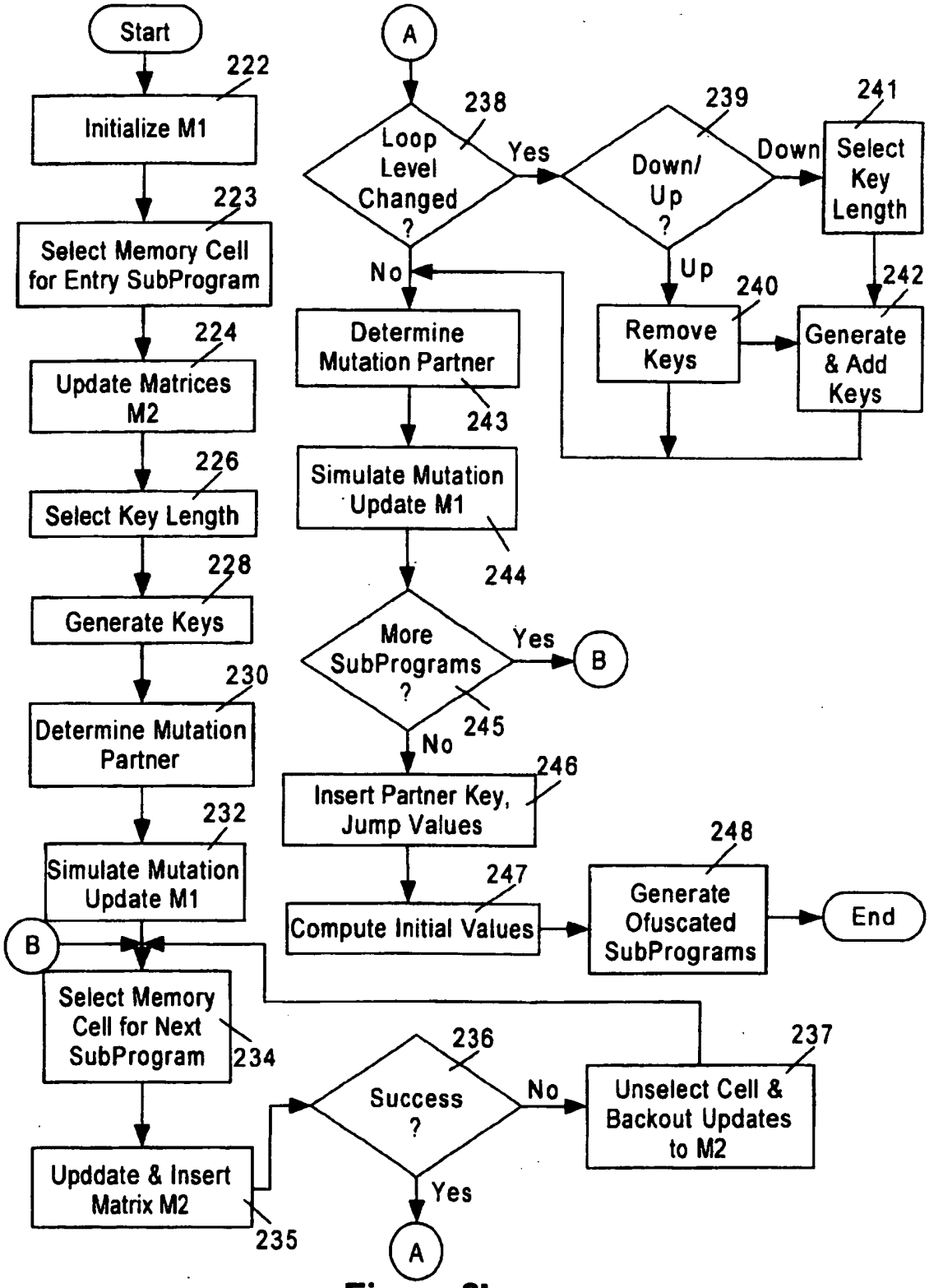

Start
222
Initialize M1
223
Select Memory Cell for Entry SubProgram
224
Update Matrices M2
226
Select Key Length
228
Generate Keys
230
Determine Mutation Partner
232
Simulate Mutation Update M1
B
Select Memory Cell for Next SubProgram
234
Upddate & Insert Matrix M2
235
A
238
Loop Level Changed ?
Yes
239
Down/ Up ?
Down
241
Select Key Length
242
Generate & Add Keys
Up
240
Remove Keys
No
Determine Mutation Partner
243
Simulate Mutation Update M1
244
More SubPrograms ?
Yes
B
245
No
246
Insert Partner Key, Jump Values
247
Compute Initial Values
248
Generate Ofuscated SubPrograms
End
236
Success ?
No
237
Unselect Cell & Backout Updates to M2
Yes
A

Figure 8b

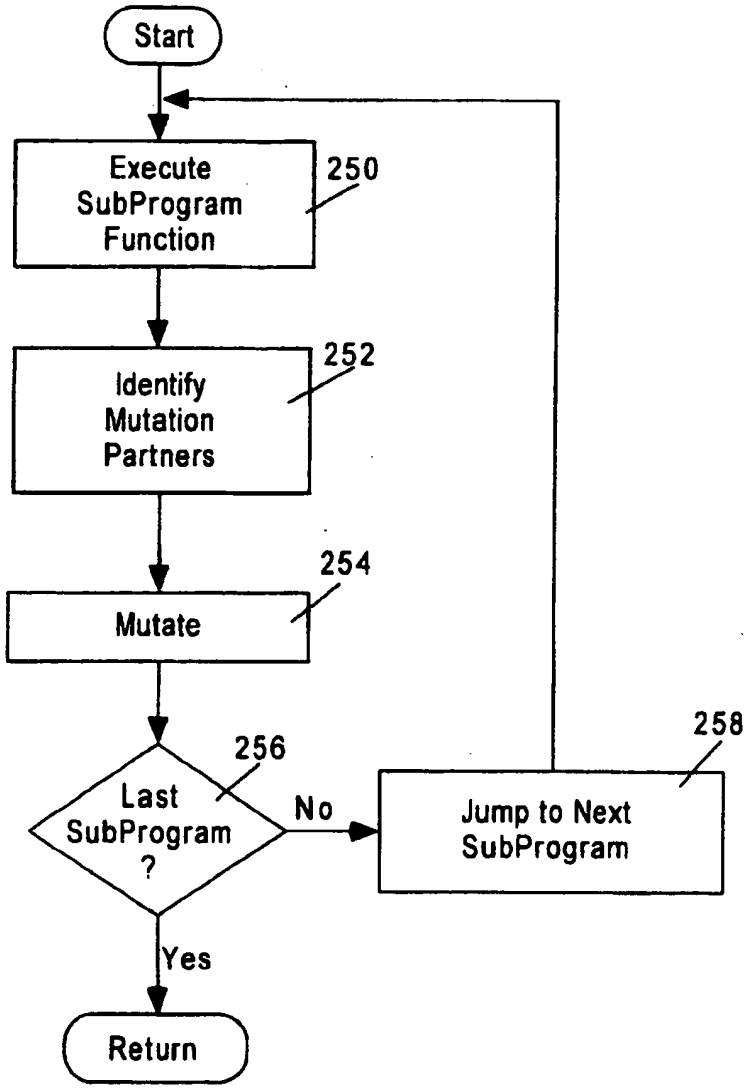

Start
Execute SubProgram Function
250
Identify Mutation Partners
252
Mutate
254
Last SubProgram ?
256
No
Jump to Next SubProgram
258
Yes
Return

Figure 9

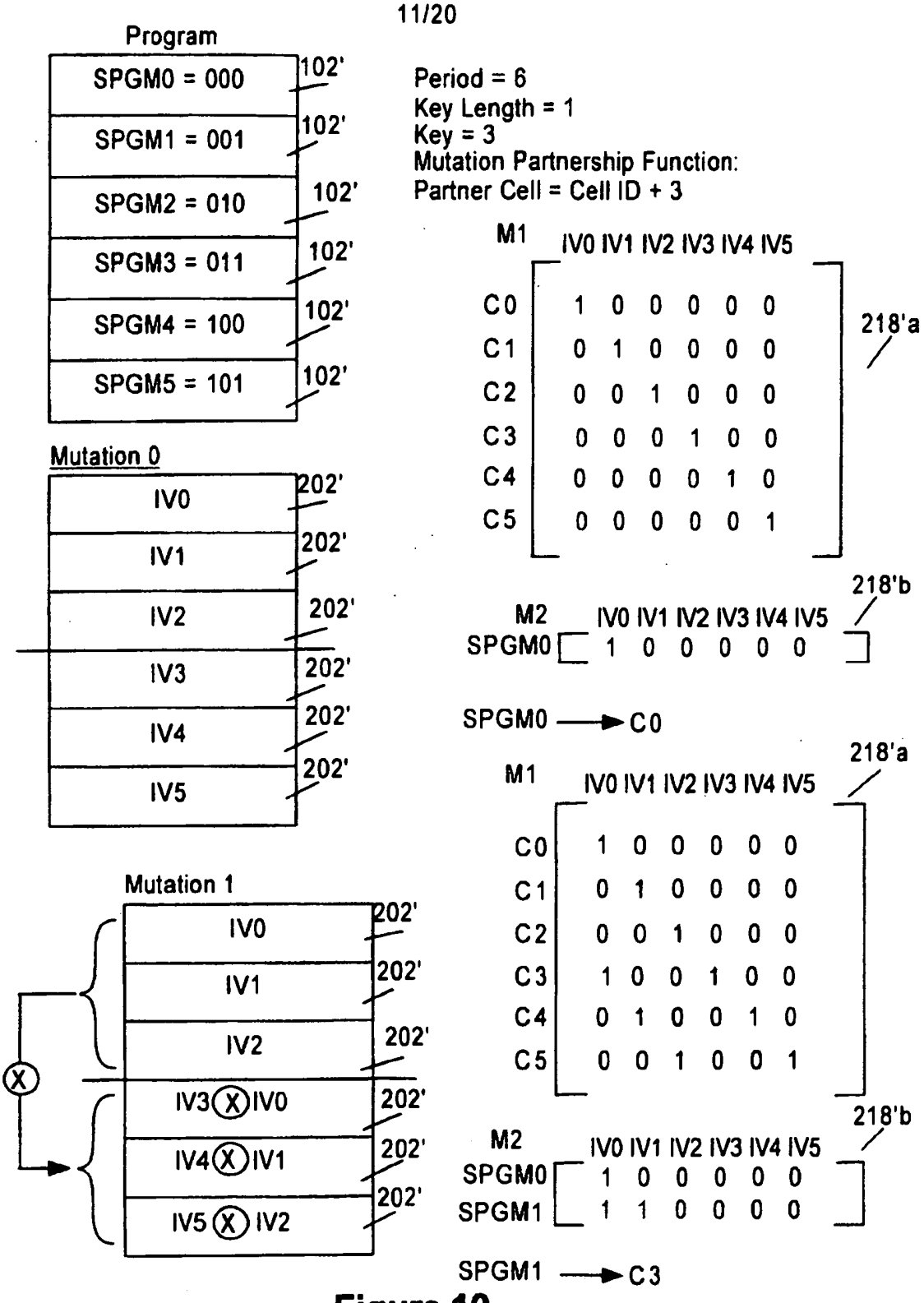

Program
SPGM0 = 000
SPGM1 = 001
SPGM2 = 010
SPGM3 = 011
SPGM4 = 100
SPGM5 = 101
102'
Mutation 0
IV0
IV1
IV2
IV3
IV4
IV5
202'
Mutation 1
IV3 X IV0
IV4 X IV1
IV5 X IV2
Period = 6
Key Length = 1
Key = 3
Mutation Partnership Function:
Partner Cell = Cell ID + 3
M1
IV0 IV1 IV2 IV3 IV4 IV5
C0 1 0 0 0 0 0
C1 0 1 0 0 0 0
C2 0 0 1 0 0 0
C3 0 0 0 1 0 0
C4 0 0 0 0 1 0
C5 0 0 0 0 0 1
218'a
M2
SPGM0 1 0 0 0 0 0
218'b
SPGM0 → C0
C0 1 0 0 0 0 0
C1 0 1 0 0 0 0
C2 0 0 1 0 0 0
C3 1 0 0 1 0 0
C4 0 1 0 0 1 0
C5 0 0 1 0 0 1
SPGM0 1 0 0 0 0 0
SPGM1 1 1 0 0 0 0
SPGM1 → C3

Figure 10

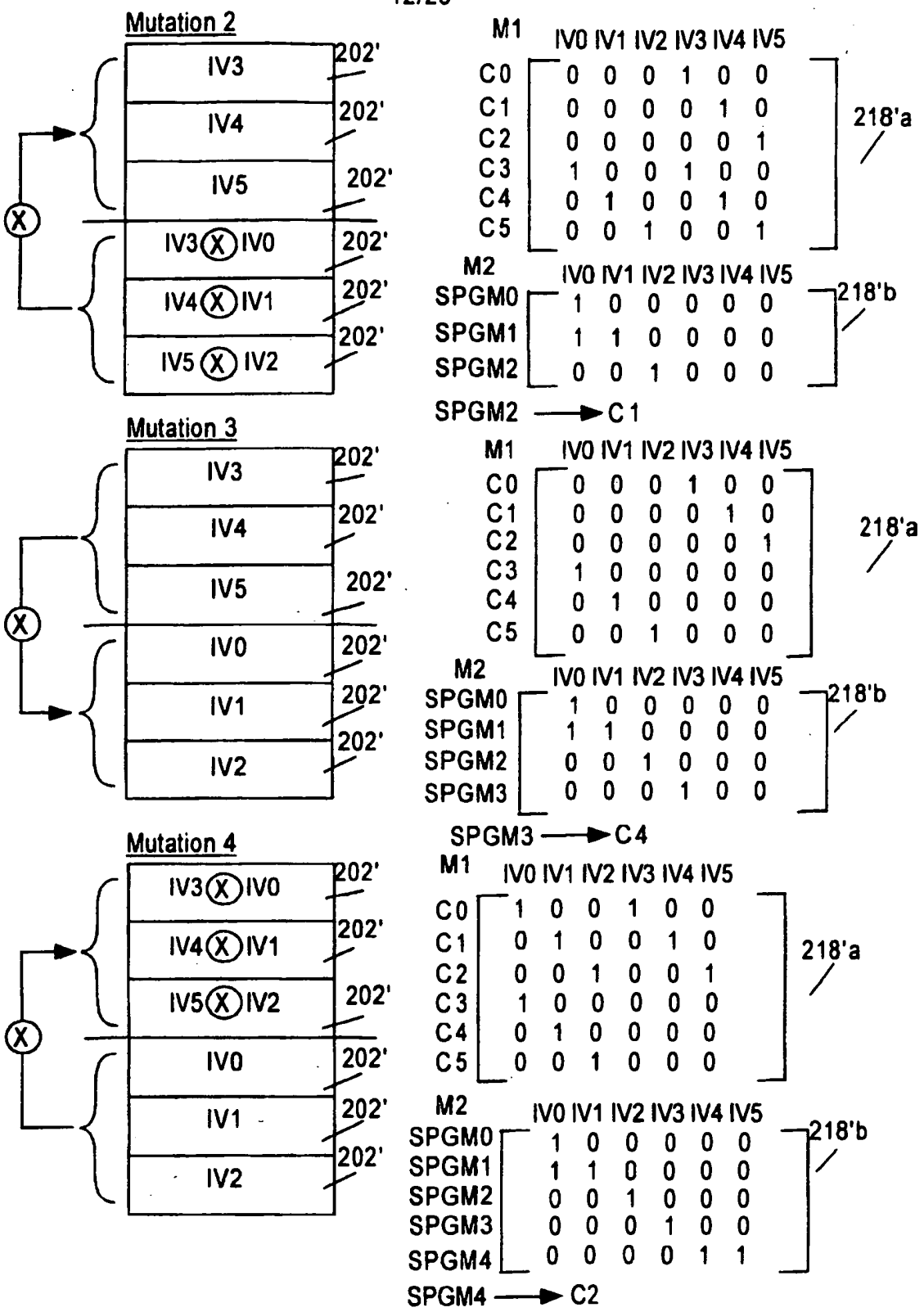

Mutation 2
IV3
IV4
IV5
IV3 X IV0
IV4 X IV1
IV5 X IV2
202'
M1
IV0 IV1 IV2 IV3 IV4 IV5
C0 0 0 0 1 0 0
C1 0 0 0 0 1 0
C2 0 0 0 0 0 1
C3 1 0 0 1 0 0
C4 0 1 0 0 1 0
C5 0 0 1 0 0 1
218'a
M2
IV0 IV1 IV2 IV3 IV4 IV5
SPGM0 1 0 0 0 0 0
SPGM1 1 1 0 0 0 0
SPGM2 0 0 1 0 0 0
218'b
SPGM2 → C1
Mutation 3
IV3
IV4
IV5
IV0
IV1
IV2
M1
IV0 IV1 IV2 IV3 IV4 IV5
C0 0 0 0 1 0 0
C1 0 0 0 0 1 0
C2 0 0 0 0 0 1
C3 1 0 0 0 0 0
C4 0 1 0 0 0 0
C5 0 0 1 0 0 0
218'a
M2
IV0 IV1 IV2 IV3 IV4 IV5
SPGM0 1 0 0 0 0 0
SPGM1 1 1 0 0 0 0
SPGM2 0 0 1 0 0 0
SPGM3 0 0 0 1 0 0
218'b
SPGM3 → C4
Mutation 4
IV3 X IV0
IV4 X IV1
IV5 X IV2
IV0
IV1
IV2
M1
IV0 IV1 IV2 IV3 IV4 IV5
C0 1 0 0 1 0 0
C1 0 1 0 0 1 0
C2 0 0 1 0 0 1
C3 1 0 0 0 0 0
C4 0 1 0 0 0 0
C5 0 0 1 0 0 0
218'a
M2
IV0 IV1 IV2 IV3 IV4 IV5
SPGM0 1 0 0 0 0 0
SPGM1 1 1 0 0 0 0
SPGM2 0 0 1 0 0 0
SPGM3 0 0 0 1 0 0
SPGM4 0 0 0 0 1 1
218'b
SPGM4 → C2

Figure 11

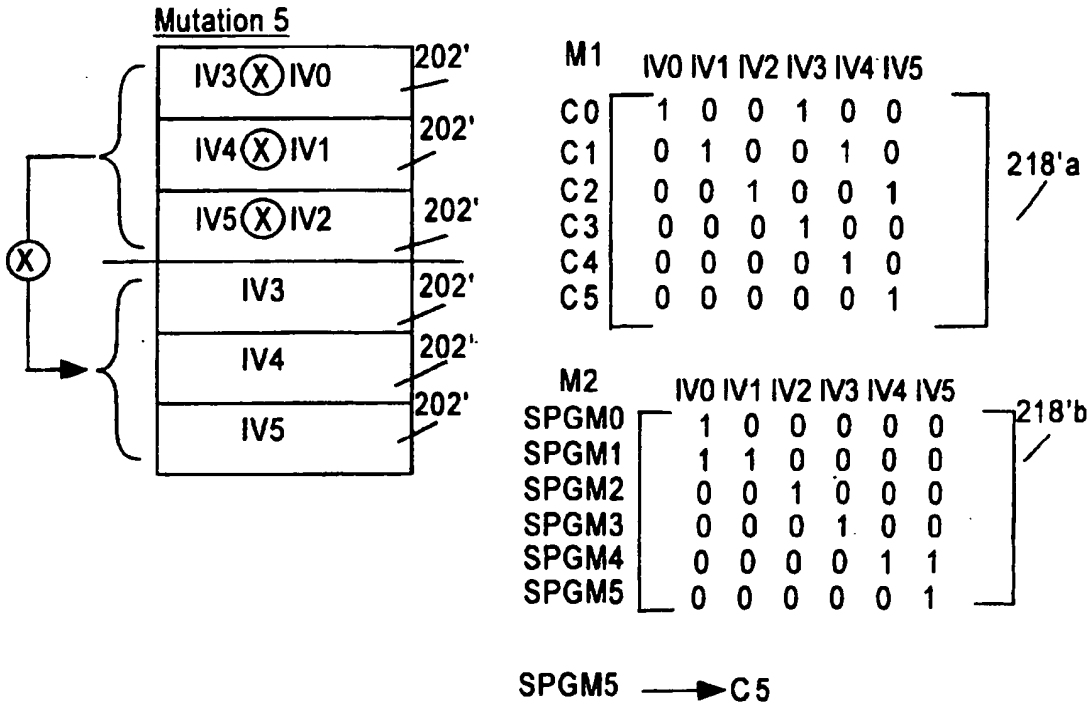

Mutation 5
IV3 X IV0
IV4 X IV1
IV5 X IV2
IV3
IV4
IV5
202'
M1
IV0 IV1 IV2 IV3 IV4 IV5
C0 1 0 0 1 0 0
C1 0 1 0 0 1 0
C2 0 0 1 0 0 1
C3 0 0 0 1 0 0
C4 0 0 0 0 1 0
C5 0 0 0 0 0 1
218'a
M2
IV0 IV1 IV2 IV3 IV4 IV5
SPGM0 1 0 0 0 0 0
SPGM1 1 1 0 0 0 0
SPGM2 0 0 1 0 0 0
SPGM3 0 0 0 1 0 0
SPGM4 0 0 0 0 1 1
SPGM5 0 0 0 0 0 1
218'b
SPGM5 → C5

Figure 12

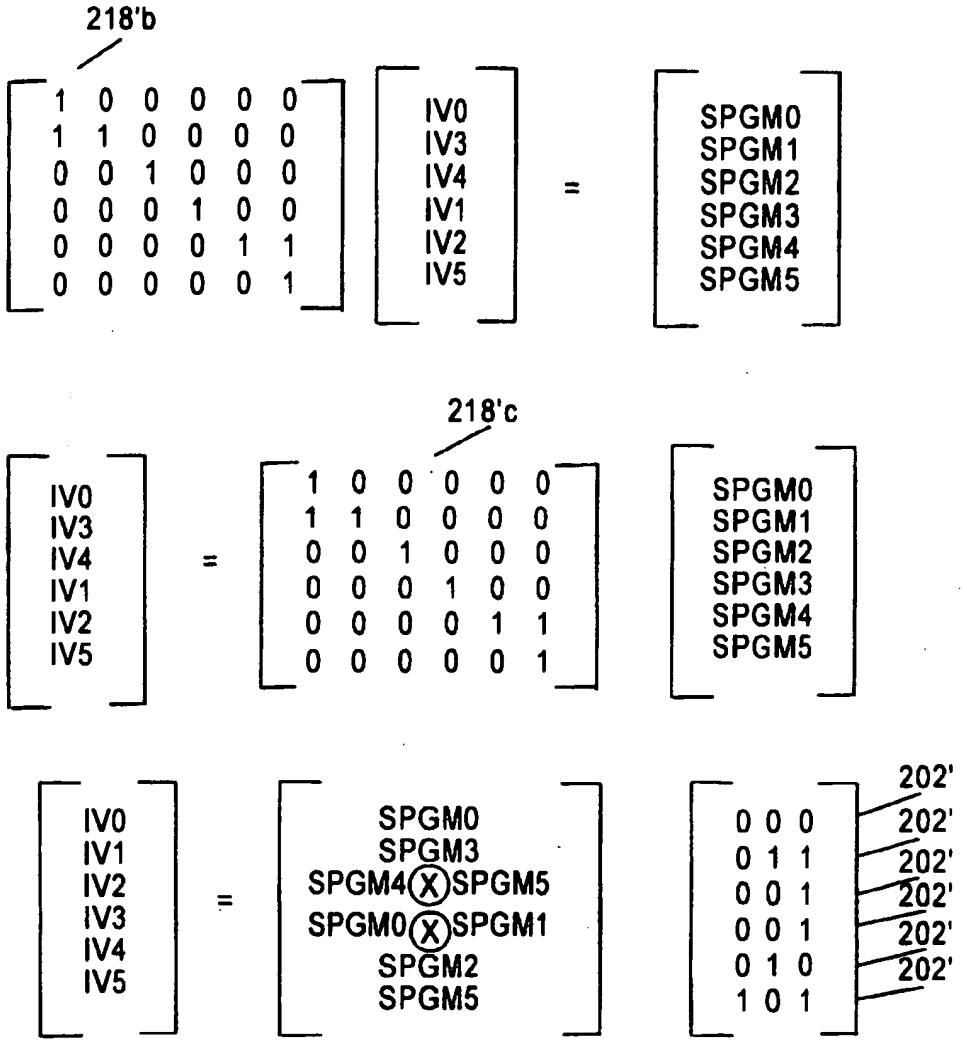

218'b
1 0 0 0 0 0
1 1 0 0 0 0
0 0 1 0 0 0
0 0 0 1 0 0
0 0 0 0 1 1
0 0 0 0 0 1
IV0 IV3 IV4 IV1 IV2 IV5
=
SPGM0 SPGM1 SPGM2 SPGM3 SPGM4 SPGM5
218'c
IV0 IV3 IV4 IV1 IV2 IV5
=
SPGM0 SPGM1 SPGM2 SPGM3 SPGM4 SPGM5
IV0 IV1 IV2 IV3 IV4 IV5
=
SPGM0
SPGM3
SPGM4 ⊗ SPGM5
SPGM0 ⊗ SPGM1
SPGM2
SPGM5
0 0 0
0 1 1
0 0 1
0 0 1
0 1 0
1 0 1
202'
202'
202'
202'
202'
202'

Figure 13

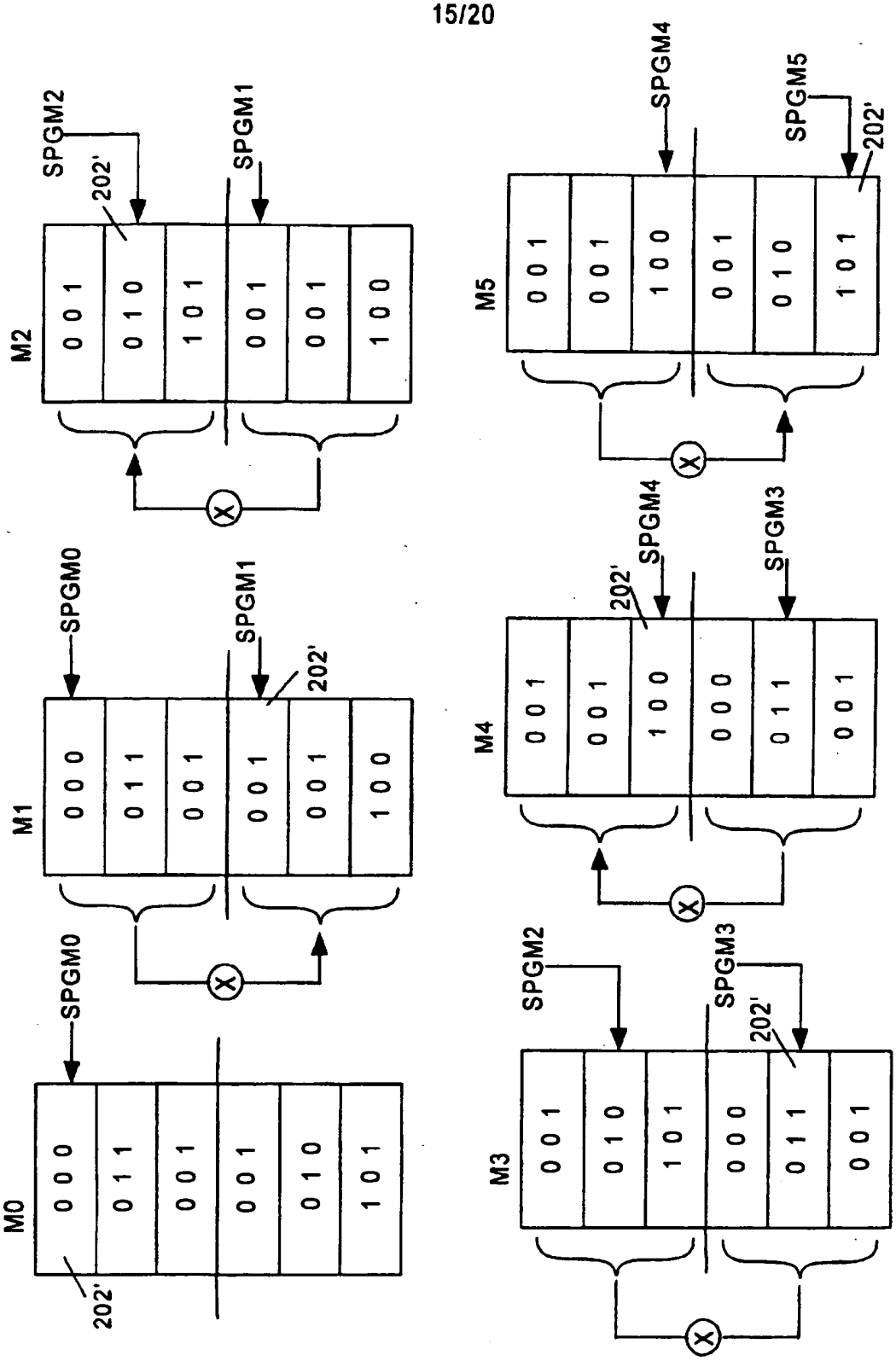

SPGM2
202'
SPGM1
M2
0 0 1
0 1 0
1 0 1
0 0 1
0 0 1
1 0 0
SPGM4
SPGM5
202'
M5
0 0 1
0 0 1
1 0 0
0 0 1
0 1 0
1 0 1
SPGM0
SPGM1
202'
M1
0 0 0
0 1 1
0 0 1
0 0 1
0 0 1
1 0 0
SPGM4
202'
SPGM3
M4
0 0 1
0 0 1
1 0 0
0 0 0
0 1 1
0 0 1
SPGM0
202'
M0
0 0 0
0 1 1
0 0 1
0 0 1
0 1 0
1 0 1
SPGM2
SPGM3
202'
M3
0 0 1
0 1 0
1 0 1
0 0 0
0 1 1
0 0 1

Figure 14

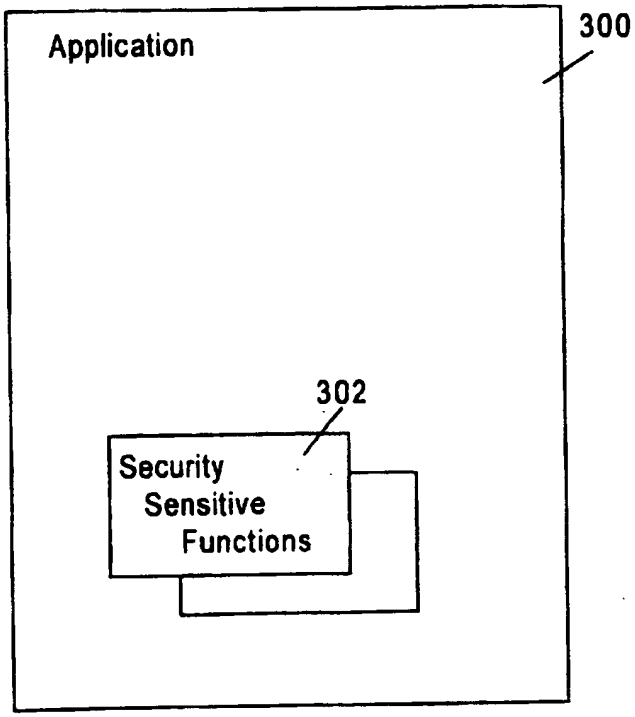


Figure 15

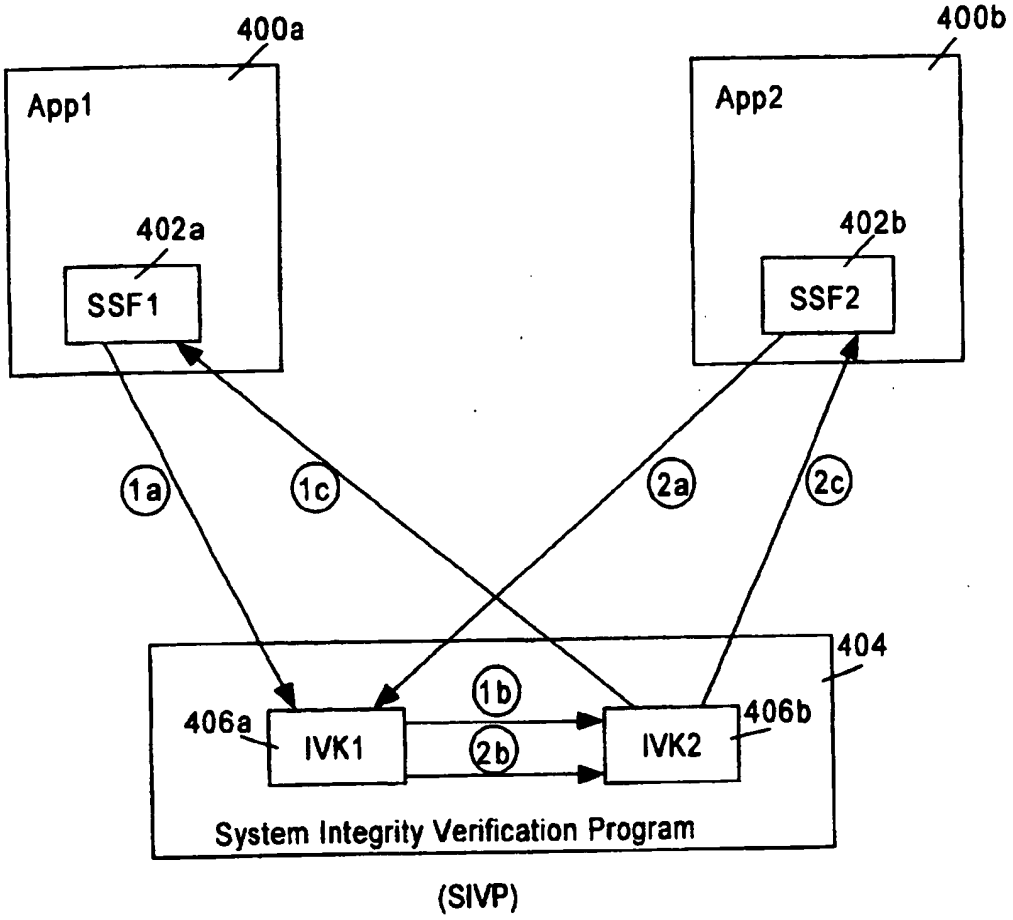

400a
400b
App1
App2
402a
402b
SSF1
SSF2
1a
1c
2a
2c
404
1b
406a
IVK1
2b
IVK2
406b
System Integrity Verification Program
(SIVP)

Figure 16

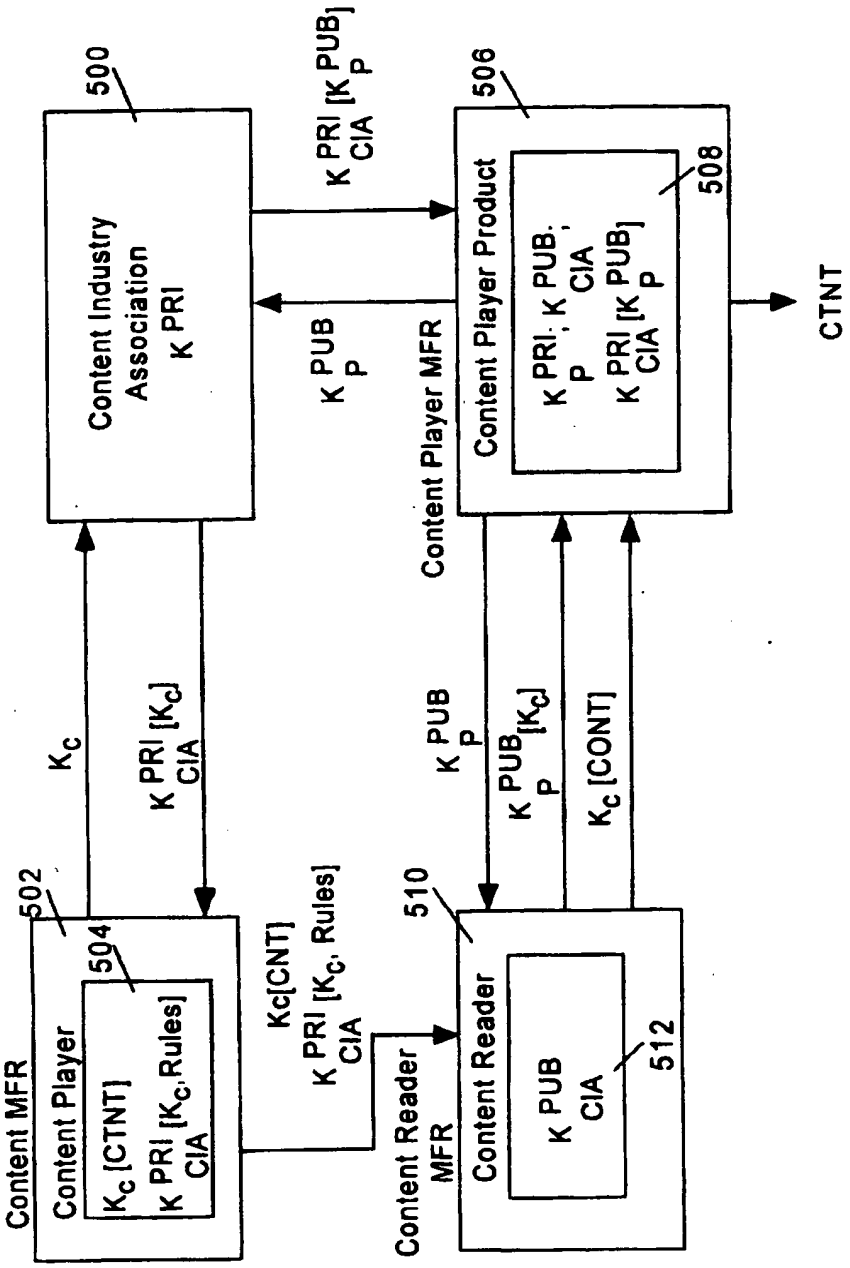

500
Content Industry Association
K PRI
K PRI CIA [K PUB P]
K PUB P
506
Content Player MFR
Content Player Product
508
K PRI P, K PUB CIA
K PRI CIA [K PUB P]
CTNT
Kc
K PRI CIA [Kc]
K PUB P
K PUB P [Kc]
Kc [CONT]
502
Content MFR
Content Player
504
Kc [CTNT]
K PRI CIA [Kc, Rules]
Kc[CNT]
K PRI CIA [Kc, Rules]
510
Content Reader MFR
Content Reader
K PUB CIA
512

Figure 17

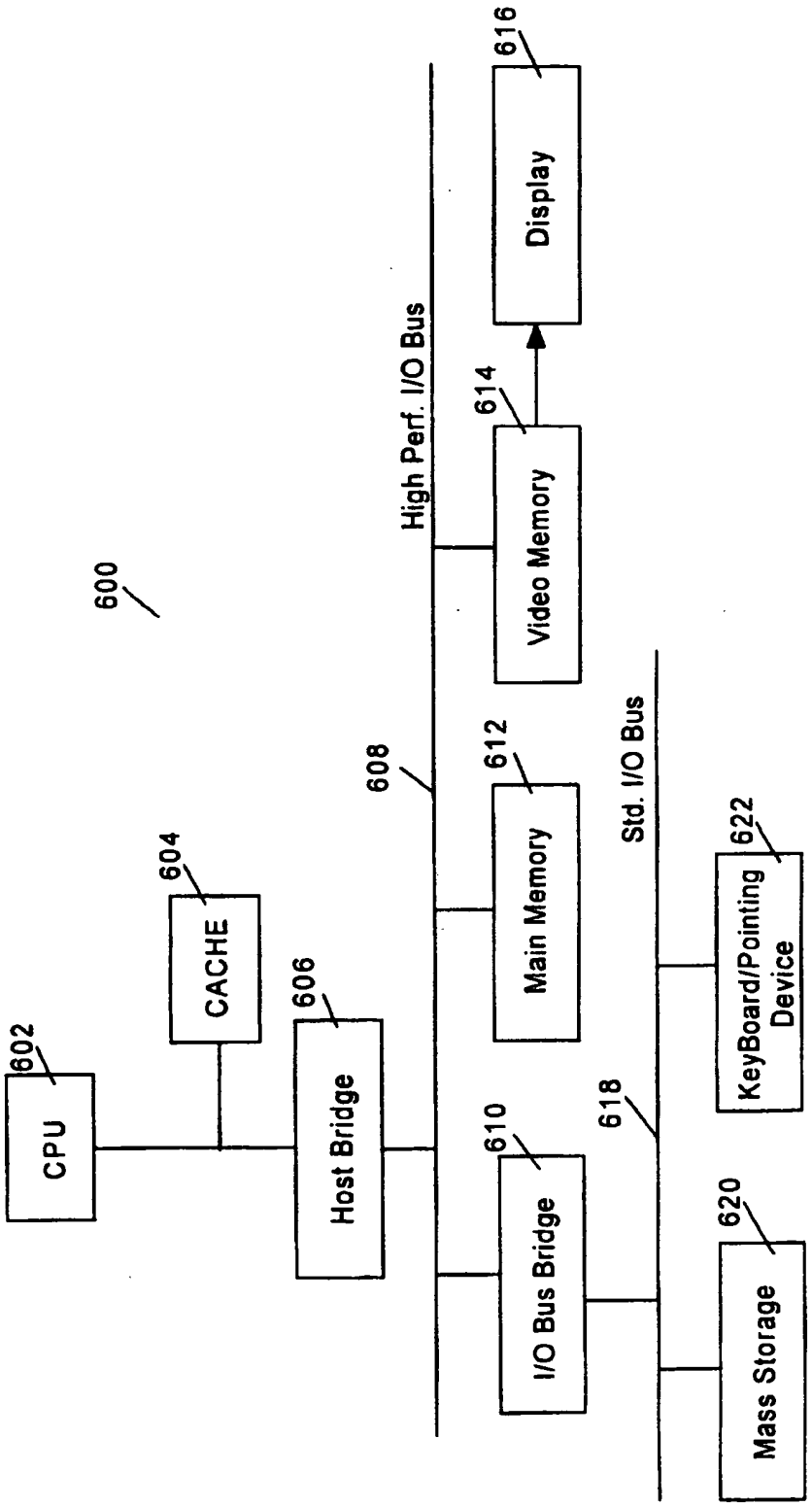

600
602
CPU
604
CACHE
606
Host Bridge
608
High Perf. I/O Bus
610
I/O Bus Bridge
612
Main Memory
614
Video Memory
616
Display
618
Std. I/O Bus
620
Mass Storage
622
KeyBoard/Pointing Device

Figure 18

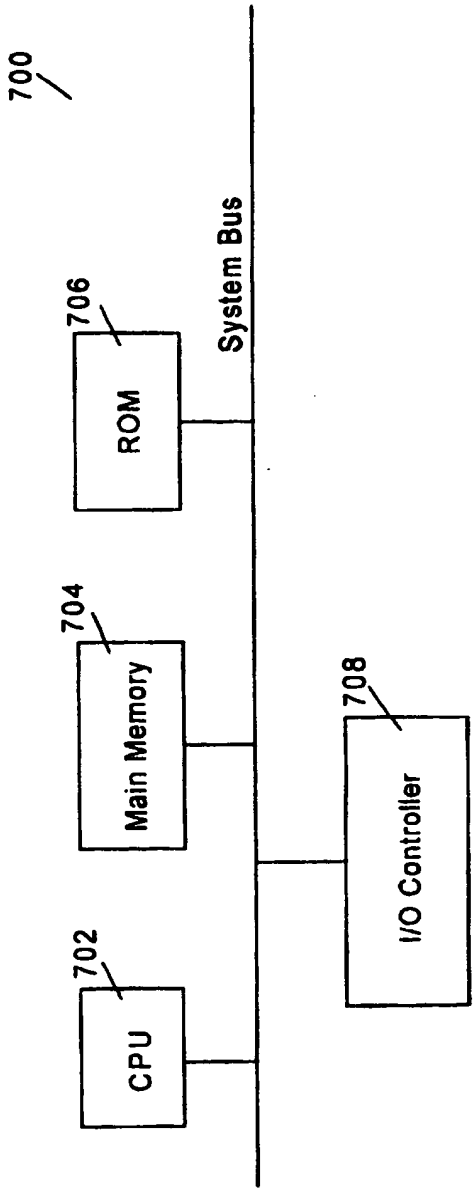

700
System Bus
706
ROM
704
Main Memory
708
I/O Controller
702
CPU

Figure 19

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/10359

A. CLASSIFICATION OF SUBJECT MATTER
IPC(6) :H04K 1/00
US CL :395/186
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. : 395/186, 187.01, 188.01; 380/ 4, 23, 24

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

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

APS, STN (WPIDS)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 4,786,790 (KRUSE et al.) 22 November 1988, see the abstract, see col. 2, lines 20-56.	10-12, 19-24, 36-42
Y	US 4,926,480 (CHAUM) 15 May 1990, see the abstract	1-50
Y	US 5,224,160 (PAULINI et al.) 29 June 1993, see the abstract and col. 6, lines 28-45.	1-50
Y	US 5,265,164 (MATYAS et al.) 23 November 1993, see fig. 10.	1-50

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents:	"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
"A"	document defining the general state of the art which is not considered to be of particular relevance	"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
"E"	earlier document published on or after the international filing date	"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
"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)	"&"	document member of the same patent family
"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		

Date of the actual completion of the international search	Date of mailing of the international search report
22 AUGUST 1997	05 NOV 1997
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 ALBERT DECADY Telephone No. (703) 308-3900

Form PCT/ISA/210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/10359

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 5,347,579 (BLANDFORD) 13 September 1994, col. 5, lines 24-56.	1-9, 25-35, 43-50 --- 10-12, 19-24, 36-42
Y	US 5,535,276 (GANESAN) 09 July 1996, see the abstract, col. 2, lines 55-62, col. 10, lines 33 et seq.	1-50

Form PCT/ISA/210 (continuation of second sheet)(July 1992)*

EP 0715241 A2

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

(11) EP 0 715 241 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
05.06.1996 Bulletin 1996/23

(51) Int. Cl.[6]: **G06F 1/00**

(21) Application number: **95116615.6**

(22) Date of filing: **21.10.1995**

(84) Designated Contracting States:
DE FR GB

(30) Priority: **27.10.1994 JP 264200/94**
02.12.1994 JP 299835/94

(71) Applicant: **MITSUBISHI CORPORATION**
Chiyoda-ku Tokyo 100 (JP)

(72) Inventors:
- **Saito, Makoto**
  Tama-shi, Tokyo (JP)
- **Momiki, Shunichi**
  Higashimur-ayama-shi, Tokyo (JP)

(74) Representative: **Neidl-Stippler & Partner**
Rauchstrasse 2
81679 München (DE)

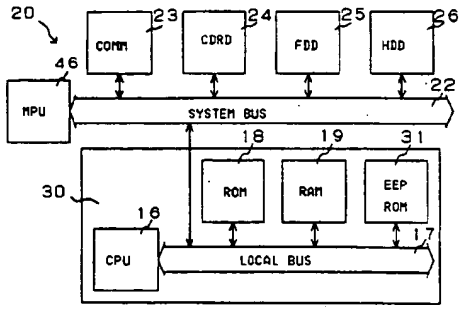
(54) Apparatus for data copyright management system

(57) A data copyright management apparatus is used with a user terminal and comprises a CPU, a CPU bus, ROM, EEPROM, and RAM.

The CPU, ROM, EPROM, and RAM are connected to the CPU bus, and a system bus of a device which utilizes the data can be connected to the CPU bus. A data copyright management system program, crypt algorithm, and user information are stored in the ROM, and a second private-key, a permit key, a second secret-key, and copyright information are stored in the EEPROM. A first public-key, a first private-key, a second public-key, and a first secret-key are transmitted to the RAM during the operation. The data copyright management apparatus may be configured in the form of a monolithic or hybrid IC, a thin IC card, PC card, and insertion board which have a unique terminal. If a copyright management program is supplied from the external, the it is stored in the EEPROM, otherwise it is stored in the ROM.

In addition to a microprocessor of user terminal which decrypts encrypted data for displaying and processing and re-encrypts the decrypted data for storing, copying, or transferring, at least one microprocessor, desirably two microprocessors, are added for decrypting and re-encrypting data which is encrypted and supplied. The microprocessors to be added may be connected to a system bus of the microprocessor of the user terminal, it is desirable that a multiprocessor configuration is implemented by using a SCSI bus, PCI bus, or SCI bus. Apparatus for decryption and re-encryption may be configured separately or as a unit. Device which is used to input and output encrypted data may be connected directly to the apparatus for decryption and re-encryption. The data copyright management apparatus may be implemented in the form of a monolithic IC, a hybrid IC, or a built-in subboard, and the apparatus in these forms is incorporated in a computer, television set, set-top box, digital video tape recorder, digital video disk recorder, digital audio tape apparatus, or personal digital assistants, and the like.

Fig. 3



Printed by Rank Xerox (UK) Business Services
2.11.9/3.4

Description

Field of the Invention

The present invention relates to an apparatus for displaying, storing, copying, editing or transmitting digital data in using data, and intends to protect digital data copyrights.

Background of the Invention

In information-oriented society of today, a database system has been spread in which various data values having independently been stored in each computer so far are mutually used by connecting computers by communication lines.

The information having been handled by the database system is classical type coded information which can be processed by a computer and has a small amount of information or monochrome binary data like facsimile data at most. Therefore, the database system has not been able to handle data with an extremely large amount of information such as a natural picture and a motion picture.

However, while the digital processing technique for various electric signals develops, development of the digital processing art for a picture signal other than binary data having been handled only as an analog signal is progressed.

By digitizing the above picture signal, a picture signal such as a television signal can be handled by a computer. Therefore, a "multimedia system" for handling various data handled by a computer and picture data obtained by digitizing a picture signal at the same time is noticed as a future technique.

Because picture data includes an overwhelmingly large amount of information compared to character data and audio data, it is difficult to directly store or transmit the picture data or apply various processings to the picture data by a computer.

Therefore, it has been considered to compress or expand the picture data and several standards for compressing or expanding picture data have been prepared. Among those standards, the following standards have been prepared so far as common standards: JPEG (Joint Photographic image coding Experts Group) standard for a still picture, H.261 standard for a video conference MPEG1 (Moving Picture image coding Experts Group 1) standard for storing pictures, and MPEG2 corresponding to the present telecast and tire high-definition telecast.

Real-time processing of digital picture data has been realized by these techniques.

Because hitherto widely-spread analog data is deteriorated in quality whenever storing, copying, editing, or transmitting it, copyrights produced due to the above operation has not been a large problem. However, because digital data is not deteriorated in quality after repeatedly storing, copying, editing, or transmitting it, the control of copyrights produced due to the above operation is a large problem.

Because there is not hitherto any exact method for dealing with a copyright for digital data, the copyright is handled by the Copyright Act or relevant contracts. Even in the Copyright Act, compensation money for a digital-type sound- or picture-recorder is only systematized.

Use of a database includes not only referring to the contents of the database but also normally effectively using the database by storing, copying, or editing obtained data. Moreover, it is possible to transmit edited data to another person via on-line by a communication line or a proper recording medium.

Furthermore, it is possible to transmit the edited data to the database to enter it as new data.

In an existing database system, only character data is handled. In a multimedia system, however, audio data and picture data which are originally analog data are digitized and formed into a database in addition to the data such as characters which have been formed into a database so far.

Under the above situation, how to deal with a copyright of data formed into a database is a large problem. However, there has not been adequate copyright management means for solving the problem so far, particularly copyright management means completed for secondary utilization of the data such as copying, editing, or transmitting of the data.

Although data of "Software with advertisement" or "free software" is, generally, available free of fee, it is copyrighted and its use may be restricted by the copyright depending on the way of use.

The inventor of the present invention et al. proposed a system for managing a copyright by obtaining a permit key from a key control center via a public telephone line through Japanese Patent Laid-Open No. 46419/1994 and Japanese Patent Laid-Open No. 141004/1994 and moreover, proposed an apparatus for managing the copyright through Japanese Patent Laid-Open No. 132916/1994.

Furthermore, they proposed a system for managing a copyright of digital data through Japanese Patent Application No. 64889/1994 and Japanese Patent Application No. 237673/1994.

In these systems and apparatus, one who wants to view and listen encrypted programs requests to a control center for viewing by using communication device via a communications line, and the control center sends a permit key to the requester, performs charging and collects a fee.

After receiving the permit key, the requester sends the permit key to a receiver by using an on-line or off-line means, the receiver then decrypts the encrypted programs using the permit key.

Moreover, the system disclosed in Japanese Patent Application No. 64889/1994 uses a program and copyright information for managing the coyright in addition to the permit key so that the copyright in display (including process to sound), storage, copying, editing, or transmit-

ting of the digital data in a database system including real-time transmission of a digital picture can be managed. The program for managing the copyright watches and manages to prevent users from using other than the conditions of user's request or permission.

The Japanese Patent Application No. 64889/1994 further discloses that data is supplied with encrypted from a database, decrypted by copyright management program when displayed or edited, and encrypted again when it is stored, copied or transmitted. Also the copyright management program itself being encrypted; decrypted by a permit key; the copyright management program thus decrypted performing encyption and decryption of copyright data; and when data is utilizedother than storage and displaying, copyright information including information of the person who has utilized, being stored as history in addition to original copyright information, are disclosed.

Though the present invention is described below, general description is made for cryptography at first.

The cryptography includes a secret-key cryptosystem and a public-key cryptosystem.

The secret-key cryptosystem is a cryptosystem using the same crypt key for encryption and decryption. While this cryptosystem requires only a short time for encryption or decryption, the secret-key is found, and thus, the cryption may be cryptanalized.

The public-key cryptosystem is a cryptosystem in which a key for encryption is open to the public as a public-key and a key for decryption is not open to the public. The key for encryption is referred to as a public-key and the key for decryption is referred to as a private-key. To use this cryptosystem, it is necessary that a party for transmitting information encrypts the information with a public-key of a party for receiving the information and the party for receiving the information decrypts the information with a private-key not open to the public. While this cryptosystem requires relatively a long time for encryption or decryption, the private-key can hardly be found and it is very difficult to cryptanalyze the cryption.

In the cryptography, a case of encrypting a plaintext M with a crypt key K to obtain a cryptogram C is expressed as

$$C = E(K, M)$$

and a case of decrypting the cryptogram C with the crypt key K to obtain the plaintext M is expressed as

$$M = D(K, C).$$

The cryptosystem used for the present invention uses a secret-key cryptosystem in which the same secret-key Ks is used for encryption and decryption, and a public-key cryptosystem in which a public-key Kb is used for encryption of a plaintext and a private-key Kv is used for decryption of a cryptogram.

Figure 1 shows a structure of the data copyright management system disclosed in the prior Japanese Patent Application No. 237673/1994 in which the apparatus for data copyright management system of the present invention is used.

In this system, encrypted data is two-way supplied in accordance with a request from the primary user 4.

This system rises the secret-key cryptosystem and the public-key cryptosystem as a cryptosystem.

It is matter of course that this system can be applied when using a satellite broadcast, ground wave broadcast, CATV broadcast or a recording medium other than a database as data supply means provided with advertisement requiring no charge or encryption.

In this system, reference numeral 1 represents a database, 4 represents a primary user terminal, 5 represents a secondary user terminal, 6 represents a tertiary user terminal, and 7 represents an n-order user terminal.

And 3 represents a copyright management center, 8, 9, and 10 represent a secondary copyright data, tertiary copyright data, and n-order copyright data stored at the copyright management center 3, and 2 represents a communication network such as a public telephone line offered by a communication enterprise or a CATV line offered by a cable television enterprise.

On the above arrangement, the database 1, primary user terminal 4, secondary user terminal 5, tertiary user terminal 6, n-order user terminal 7, and copyright management center 3 are connected to the communication network 2 and also they can be connected each other.

In this figure, a path shown by a broken line represents a path for encrypted data, a path shown by a solid line represents a path of requests from each user terminal, a path shown by a one-dot chain line represents a path through which authorization information corresponding to a utilization request in each data and a crypt key are transferred, and a path shown by a two-dot chain line represents a path through which copyright information is transferred from the database or from the data to a next-order data within copyright management center.

Each user who uses this system is previously entered in a database system and in this time, database utilization software is provided him. The database utilization software includes a program for decrypting an encrypted copyright management program in addition to normal communication software such as data communicating protocol.

To use the database 1, a primary user prepares primary-user authentication data Au1, a first public-key Kb1, a first private-key Kv1 corresponding to the first public-key Kb1, a second public-key Kb2, and a second private-key Kv2 corresponding to the second public-key Kb2, and accesses the database 1 from the primary user terminal 4 via the communication network 2.

The database 1 receiving the primary-user authentication data Au1, first public-key Kb1 and second public-key Kb2 from the primary user confirms the primary-user authentication data Au1 and transfers the confirmed primary-user authentication data Au1 to the secondary copyright management center 3 as the primary user information Iu1.

The database 1 prepares two secret-keys, that is, first secret-key Ks1 and second secret-key Ks2.

In the prepared first secret-key Ks1 and second secret-key Ks2, the second secret-key Ks2 is also previously transferred to the copyright management center 3.

As the result of the above transfer, a permit key corresponding to primary utilization, the primary user information Iu1, original copyright information Ic0 and the second secret-key Ks2 are stored in the copyright management center 3. In this case, the original copyright information Ic0 is used for copyright royalties distribution.

When a primary user who desires data utilization accesses the database 1 from the primary user terminal 4, a data menu is transferred to him. In this case, information for charges may be displayed together with the data menu.

When the data menu is transferred, the primary user retrieves in the data menu to select the data M. In this case, the original copyright information Ic0 of the selected data M is transmitted to the copyright management center 3. The primary user selects permit key Kp1 corresponding to the required form of the usage such as viewing, storing, copying, editing and transmitting of data. Permit key Kp1 is also transmitted to the copyright management center 3.

Because viewing and storing of data are the minimum required forms of use for the primary user, these forms of use may be excluded from the choices as the minimum usage, and offering only copying, editing and transmitting as the choices.

The original data M0 is read out of the database 1 in accordance with a request of the primary user. The read original data M0 is encrypted by the first secret-key Ks1:

$$Cm0ks1 = E(Ks1, M0).$$

The encrypted data Cm0ks1 is provided with the uncrypted original copyright information Ic0.

The first secret-key Ks1 is encrypted by the first public-key Kb1 and the second secret-key Ks2 is encrypted by the second public-key kb2:

$$Cks1kb1 = E(Kb1, Ks1)$$

$$Cks2kb2 = E(Kb2, Ks2).$$

While the copyright management program P is also encrypted by the second secret-key Ks2

$$CpKs2 = E(Ks2, P),$$

the copyright management program P must not always be encrypted by the second secret-key Ks2 but it may be encrypted by any other proper crypt key.

The encrypted original data Cm0ks1, encrypted copyright management program Cpks2, and two encrypted secret-keys Cks1kb1 and Cks2kb2 are transferred to the primary riser terminal 4 via the communication network 2, and charged, if necessary.

It is possible to store the encrypted copyright management program Cpks2 such as in a ROM in the user terminal 4 instead of being supplied from the database 1.

The primary user receiving the encrypted original data Cm0ks1, two encrypted secret-keys Cks1kb1 and Cks2kb2, and encrypted copyright management program Cpks2 from the database 1 decrypts the encrypted first secret-key Cks1kb1 by the database utilization software using the first private-key Kv1 corresponding to the first public-key Kb1:

$$Ks1 = D(Kv1, Cks1kb1),$$

and decrypts the encrypted second secret-key Cks2kb2 using the second private-key Kv2 corresponding to the second public-key Kb2:

$$Ks2 = D(Kv2, Cks2kb2).$$

And the primary user decrypts the encrypted copyright management program Cpks2 using the decrypted second secret-key Ks2:

$$P = D(Ks2, Cpks2).$$

Finally, the primary user decrypts the encrypted data Cm0ks1 by the decrypted copyright management program P using the decrypted first secret-key Ks1:

$$M0 = D(Ks1, Cm0ks1)$$

and uses the decrypted original data M0 directly or data M1 as edited.

As described above, the first private-key Kv1 and second private-key Kv2 are crypt keys prepared by the primary user but not opened to others. Therefore, even if a third party obtains tire data M, it is impossible to use the encrypted data M by decrypting it.

Thereafter, to store, copy, or transmit the data M as the original data M0 or the edited data M1, it is encrypted and decrypted by the second secret-key Ks2:

$$Cmks2 = E(Ks2, M)$$

$$M = D(Ks2, Cmks2).$$

The decrypted second secret-key Ks2 is thereafter used as a crypt key for encrypting/decrypting data when storing, copying, or transmitting the data.

The first private-key Kv1 and second private-key Kv2, the first secret-key Ks1 and second secret-key Ks2, the data M, the copyright management program P, the original copyright information Ic, and also the original copyright information Ic0 and also copyright information Ic1 for information of the primary user and edited date and time when edited the data by the primary user are stored in the primary user terminal 4.

Moreover, it is further protected by attaching the copyright information Ic1 to the data as copyright information label, and adding the digital signature.

The encrypted data Cmks2 is encrypted to be distributed. Since the copyright information label provides a clue to obtain the second secret-key Ks2 which is the key for decryption, the second secret key Ks2 cannot be obtained in the case where the copyright information label is removed from the encrypted data Cmks2.

When the encrypted data Cmks2 is stored in the primary user terminal 4, the second secret-key Ks2 is stored in the terminal 4. However, when the encrypted data Cmks2 is not stored in the primary user terminal 4 but is copied to the recording medium 11 or transmitted to the secondary user terminal 5 via the communication network 2, the second secret-key Ks2 is disused in order to disable subsequent utilization of the data in the primary user terminal 4.

In this case, it is possible to set a limitation for repetitions of copying or transmitting of the data so that the second secret-key Ks2 is not disused within limited repetions of copying and transmitting of the data.

The primary user who is going to copy the data M to the external recording medium 11 or transmit the data M via the communication network 2 must prepare the second secret-key Ks2 to encrypt the data M by this second secret-key Ks2 before copying or transmitting the data:

$$Cmks2 = E(Ks2, M).$$

The uncrypted original copyright information Ic0 and primary-user copyright information Ic1 are added to the encrypted data Cmks2.

Before using a database, a secondary user, similar to the primary user, prepares authentication data Au2 for authenticating the secondary user, a third public-key Kb3 and a third private-key Kv3 corresponding to the third public-key Kb3, a fourth public-key Kb4, and a fourth private-key Kv4 corresponding to the fourth public-key Kb4.

The secondary user who desires secondary utilization of the copied or transmitted encrypted data Cmks2 must designate original data name or number to the copyright management center 3 to request for secondary utilization to the center 3 from the secondary user terminal 5 via the communication network 2. In this time, the secondary user also transfers the third public-key Kb3 and the fourth public-key Kb4 as well as the secondary user authentication data Au2, original copyright information Ic0 and primary user copyright information Ic1.

The copyright management center 3 receiving the secondary utilization request from the secondary user confirms the secondary-user authentication data Au2, and transfers confirmed secondary-user authentication data Au2 to the tertiary copyright data 9 as secondary user information.

When the secondary copyright information Ic1 of the primary user is transferred, the secondary copyright information Ic1 is inquired to the secondary copyright data 8, and then, it recognizes the secondary copyright information Ic1 to be transferred to the tertiary copyright data 9.

The secondary user selects permit key Kp2 corresponding to the form of data usage such as viewing, storing, copying, editing and transmitting of data. Permit key Kp2 corresponding to the selected usage is sent to the tertiary copyright data 9.

Because viewing and storing of data are the minimum required forms of use for the secondary user, these forms of use may be excluded from the choices as the minimum usage, offering only copying, editing and transmitting as the choices.

The secondary copyright data 8 prepares a third secret-key Ks3.

The prepared third secret-key Ks3 is transferred to and stored in the tertiary copyright data 9.

As the result of the above transfer, the permit key Kp2, primary user copyright information Ic1, primary user information Iu1, original copyright information Ic0, secondary user information Iu2, and third secret-key Ks3 are stored in the tertiary copyright data 9. The permit key Kp2, primary user copyright information Ic1, and primary user information Iu1 are used for copyright royalties distribution.

Hereafter similarly, permit key Kpn corresponding to n-order usage, copyright information for secondary exploitation right Icn-1 of (n-1)-order user, primary user information Iu1, original copyright information Ic0, n-order user information Iun, and n-th secret-key Ksn are stored in n-order copyright data 10.

The permit key Kp2, primary user information Iu1, original copyright information Ic0 and second secret-key Ks2 are read out of the secondary copyright data 8. The original copyright information Ic0 is used for copyright royalties distribution.

The read second secret-key Ks2 and third secret-key Ks3 are encrypted by the third public-key Kb3 and fourth public-key Kb4 of the secondary user respectively:

$$Cks2kb3 = E(Kb3, Ks2)$$

$$Cks3kb4 = E(Kb4, Ks3).$$

The copyright management program P is encrypted by the third secret-key Ks3:

$$Cpks3 = E(Ks3, P).$$

The encrypted copyright management program Cpks3, encrypted second secret-key Cks2kb3, and encrypted third secret-key Cks3kb4 are transferred to the secondary user terminal 5 via the communication network 2. In this case, charging is performed, if necessary.

The secondary user receiving two encrypted secret-keys Cks2kb3 and Cks3kb4 and the encrypted copyright management program Cpks3 from the secondary copyright data 8 decrypts the encrypted second secret-key Cks2kb3 by the third private-key Kv3, and decrypts the

encrypted third secret-key Cks3kb4 by the fourth private-key Kv4 corresponding to the fourth public-key Kb4, using the database utilization software:

$$Ks2 = D(Kv3, Cks2kb3)$$

$$Ks3 = D(Kv4, Cks3kb4).$$

The encrypted copyright management program Cpks3 is decrypted by the decrypted third secret-key Ks3:

$$P = D(Ks3, Cpks3).$$

Then, the encrypted data Cmks2 is decrypted to use it by the decrypted second secret-key Ks2 using decrypted copyright management program P:

$$M = D(Ks2, Cmks2).$$

As described above, the third private-key Kv3 and the fourth private-key Kv4 are prepared by the secondary user but not opened to others. Therefore, even if a third party obtains the encrypted data Cmks2, it is impossible to use the data by decrypting it.

Each user who uses above-mentioned system must previously be entered in a database system, and when entered in the system, software for database is supplied to the user.

Because the software includes not only normal communication software such as a data communication protocol but also a program for decrypting a copyright management program by a first crypt-key, it is necessary to be protected.

A first crypt-key K1, a second crypt-key K2, and a copyright management program P are transferred to each user in order to use data M, and each user keeps these keys and the program.

Further, the copyright information label, user information, the public-key and private-key in the public-key cryptosystem and the program containing algorithm for generating the secret-key are kept when needed.

For keeping them, it is the simplest means to use a flexible disk. However, the flexible disk is easy in disappearance or alteration of data.

Moreover, a hard disk drive is also unstable for disappearance or alteration of data though it is more stable than the flexible disk.

Recently, an IC card is spread in which an IC element is sealed in a card-like package. Particularly, standardization of a PC card with a microprocessor sealed in it is progressed as a PCMCIA card or JEIDA card.

The data copyright management apparatus proposed by the inventor of the present invention et al. in the prior Japanese Patent application No. 237673/1994 is described in Figure 2.

The data copyright management unit 15 is configured as a computer system, comprising a microprocessor (CPU) 16, a local bus 17 of CPU 16, read only memory (ROM) 18 connected to local bus 17, and write/read memory (RAM) 19, wherein the local bus 17 being connected to system bus 22 of the microprocessor 21 of the user terminal 20.

Moreover, a communication unit (COMM) 23 which receives data from an external database and transfer data to the external database, a CD-ROM drive (CDRD) 24 which reads data provided by CD-ROM, a flexible disk drive (FDD) 25 which copies received or edited data to a flexible disk drive to provide outside with such data, and a hard disc drive (HDD) 26 which stores data are connected to the system bus 22 in the user terminal 20.

As a matter of course, ROM and RAM or the like are connected to the system bus 22 of the user terminal, however, it is not shown in the figure.

Fixed information, such as software and user data, for utilizing the database is stored in ROM 18 of the data copyright management unit 15.

A crypt-key and the copyright management program provided from the key control center or copyright management center are stored in RAM 19.

The process of decryption and re-encryption are performed by the data copyright management unit 15, only of which results are transferred to the user terminal 20 via the local bus 17 and the system bus 21 of the user terminal.

The data copyright management unit 15 is implemented as monolithic IC, hybrid IC, an expansion board, an IC card, or a PC card.

Summary of the Invention

In the present application, apparatus for data copyright management system, resulted from further implementation of the apparatus used in the user terminal proposed in the prior Japanese patent application No. 237673/1994, is proposed.

The apparatus for data copyright management in the present invention is attached to the user terminal, which comprises central processing unit, central processing unit bus, read only semiconductor memory, electrically erasable programmable memory, and read/write memory.

Central processing unit, read only semiconductor memory, electrically erasable programmable memory, and read/write memory are connected to the central processing unit bus, and also system bus of a unit which utilizes the data can be connected to it. Data copyright management system program, a crypt algorithm, and user information are stored in the read only semiconductor memory, and a second private-key, permit key, second secret-key, and copyright information are stored in the electrically erasable programmable memory, wherein first public-key, first private-key, second public-key, and first secret-key being transferred to the read/write memory at the operation of the unit. If the copyright management program is provided from the outside, it is stored in the EEPROM. Otherwise, it is stored in ROM.

As a form of the data copyright management apparatus, monolithic IC, hybrid IC, a thin IC card with special terminal, a PC card, and a board for insertion can be available.

In the data copyright management system described above as prior invention, while the obtained encrypted data is decrypted for utilization of displaying/editing, the obtained or edited data is re-encrypted to store/copy/transfer so that no unauthorized use of the data can be available.

Accordingly, in the apparatus used in the data copyright management system of the present invention, re-encryption of data, as well as decryption of data should be performed concurrently, however, those data copyright management apparatus described in the prior applications can perform only one process of either data decryption or data re-encryption.

Thus, in the present application, a data copyright management apparatus which, at the same time, can decrypt and re-encrypt data encrypted and supplied in order to manage copyright is proposed.

For the purpose of that, data which was encrypted and provided is decrypted and re-encrypted by adding at least one microprocessor, preferably 2 microprocessors, in addition to the microprocessor that controls the entire user terminal therein. When one microprocessor is added, one of the 2 microprocessors, one included in the user terminal or one added, will decrypt data and the other will re-encrypt data.

When 2 microprocessors are added, one of the added microprocessors will decrypt data, the other microprocessor will re-encrypt data, and the microprocessor of the user terminal will control the entire operation.

Although the added microprocessors may be connected to system bus of the microprocessor in the user terminal, this configuration may not allow a multiprocessor configuration to operate plural microprocessors concurrently.

Therefore, in the present application, a data copyright management apparatus as a multiprocessor configuration utilizing SCSI bus or PCI bus is proposed.

Other than character data, digital data includes graphic data, computer program, digital audio data, still picture data of JPEG standard, and motion-picture data of MPEG standard.

While the data works comprising these data are utilized by using various apparatus, it is necessary that these apparatus should also include the data copyright management function.

Thus, in the present application, it is proposed that, as a form of use, these data copyright management apparatus and the data copyright management apparatus described in the prior application are incorporated in various systems.

Brief Description of the Drawings

Figure 1 is a block diagram of the data copyright management system of the prior invention.

Figure 2 is a block diagram of the data copyright management apparatus of the prior invention.

Figure 3 is a block diagram of the data copyright management apparatus of embodiment 1 of the present invention.

Figure 4 is a specific block diagram of the data copyright management apparatus of the embodiment 1 of the present invention.

Figure 5 is a process flow chart of data copyright management system related to the present invention.

Figure 6 is a block diagram of the data copyright management system of the prior invention.

Figure 7 is a flow chart of a general edit process of digital data.

Figure 8 is a flow chart of encrypted data edit process of the present invention.

Figure 9 is a block diagram of the data copyright management apparatus of embodiment 2 of the present invention.

Figure 10 is a block diagram of the data copyright management apparatus of embodiment 3 of the present invention.

Figure 11 is a block diagram of the data copyright management apparatus of embodiment 4 of the present invention.

Figure 12 is a block diagram of the data copyright management apparatus of embodiment 5 of the present invention.

Figure 13 is a block diagram of the data copyright management apparatus of embodiment 6 of the present invention.

Figure 14 is a block diagram of the digital cash system as one example of use of the present invention.

Figure 15 is a block diagram of the video conference system as one example of use of the present invention.

Detailed Description of the Preferred Embodiments

The detailed embodiments of the present invention are described below with reference to the drawings.

The embodiment 1 of the data copyright management apparatus related to the present invention is shown in a block diagram of Figure 3.

The data copyright management unit 30 includes electrically erasable programmable memory (EEPROM) 31 in addition to the components of the data copyright management unit 15 described in the prior application No. 237673/1994.

The data copyright management unit 30 is a computer system having CPU 16, local bus 17 of CPU 16, ROM 18 connected to local bus 17, RAM 19, and EEPROM 31, wherein local bus 17 being connected to the system bus 22 of the microprocessor 21 in the user terminal 20.

Moreover, communication unit (COMM) 23 which receives data from external database and transfers data outside, CD-ROM drive (CDRD) 24 which read data provided by CD-ROM, a flexible disc drive (FDD) 25 which copies data received or edited in order to supply to the outside, and hard disk drive (HDD) 26 which stores data are connected to the system bus 22 of the user terminal 20.

Further, ROM and RAM are connected to the system bus 22 of the user terminal, however, it is not shown in the figure.

Fixed information such as a data copyright management program, a cryptography program based on crypt algorithm, and user data are stored in ROM 18.

A crypt-key and copyright information are stored in EEPROM 31. Further, when data copyright management program and cryptography program are supplied from outside such as from database, they are stored in EEPROM 31, rather than in ROM 18.

The data copyright management unit 30 performs the process of decryption or re-encryption, only the result of which are transferred to the user terminal 20 via local bus 17 and system bus 22.

The data copyright management unit 30 is implemented as a monolithic IC, a hybrid IC, an expansion board, an IC card, or a PC card.

Fixed data such as a data copyright management program, a cryptography program based on crypt algorithm, and user data are stored in ROM 18 of the data copyright management unit 30 in the embodiment 1.

Further, a program for generating secret-keys based on secret-key algorithm of not secret, a decryption program, and a re-encryption program may be stored in ROM 18.

A crypt-key and copyright information are stored in EEPROM 31. Moreover, when the copyright management program and the encryption program are supplied from the outside such as database, they are stored in EEPROM 31, rather than ROM 18. Still more, the EEPROM is not necessarily required and may be omitted.

Either one of the first crypt-key or the second crypt-key supplied from the key control center or copyright management center, and data copyright management system program are stored in RAM 19.

On the other hand, information such as software and the user data required by MPU 46 in the user terminal 20 are supplied to the user terminal 20 by the software, and stored in RAM of the user terminal 20.

Besides, either one of the first crypt-key or the second crypt-key supplied from the key control center or the copyright management center, and the data copyright management system program are stored in RAM of the user terminal unit 20.

The process of decryption and re-encryption are shared by MPU 46 of the main body of the user terminal 20 and CPU 16 of the data copyright management unit 30; one encrypts data and the other decrypts data, and only the processed results of the data copyright management unit 30 are transferred to the user terminal.

The specific internal structure of the data copyright management unit 30 in Figure 3 is shown in Figure 4.

A microcomputer (CPU) 16, read only semiconductor memory (ROM) 18, write/read memory (RAM) 19, and electrically erasable programmable memory (EEPROM) 31 are enclosed in the data copyright management unit 30, and are connected to microcomputer bus 17 of the microcomputer 16, the microcomputer bus 17 being further connected to system bus 22 of the user terminal 20 main body.

The data copyright management system program, crypt algorithm, and the user information are stored in the read only semiconductor memory 18.

Inside of the electrically erasable programmable memory 31 is divided into three areas.

In the first area 35, the first public-key Kb1, the first private-key Kv1, the second public-key Kb2, and the second private-key Kv2 are stored.

In the second area 36, the copyright management program P, the first secret-key Ks1 as a permit key in the primary use such as view permit/store permit/copy permit/edit permit/transfer permit, and the second secret key Ks2 as a permit key in the secondary use such as view permit/store permit/copy permit/edit permit/transfer permit are stored.

Further, in some case where the copyright management program is not supplied form the outside, but preset in the user side, the copyright management program is stored in the read only memory 18, rather than in the second area 36 of the electrically erasable programmable memory 31.

In the third area 37, copyright information such as the original copyright information and the secondary copyright information, and air access control key are stored.

As in the case of the electrically erasable programmable memory 31, inside of the write/read memory 19 is divided into three areas.

In the first area 32, the first public-key Kb1, the first private-key Kv1, and the second public-key Kb2 are stored during operation.

In the second area 33, the first secret-key Ks1 as a permit key in the primary utilization such as view permit/store permit/copy permit/edit permit/transfer permit is stored during operation.

In the third area 34, an access control key is stored during operation.

The user terminal attached with the data copyright management apparatus is reliable since it performs all the process for utilizing data within the data copyright management unit related to the present invention, so that only the results are transferred to the user terminal for various utilization.

When picture data containing large amount of information is transmitted/received, original data is transmitted after being compressed in order to reduce the amount of data and the compressed data is expanded after reception to utilize it. In this case, data copyright may be managed by encryption.

In Figure 5, an example of data copyright management flow when encrypted data is digital picture compressed in JPEG standard or MPEG standard. The flow is divided into transmitting side flow and receiving side flow with a transmit line in between, and the receiving side flow is further divided into display flow and storage flow.

The signal process in the transmitting side consists of process preparing digital picture and process processing the digital picture prepared. In this process, if an original picture is the digital picture 41, it proceeds to next process. If an original image is an analog picture 40, digitizing process 42 is performed.

The digital picture is compressed 43 first by given standard such as JPEG standard, or MPEG standard, then the compressed digital data is encrypted 44 using the first secret-key.

The picture data signal processed in transmitting side is transmitted through transmission line 45 such as satellite broadcasting wave, terrestrial broadcasting wave, CATV wave, or public telephone line/ISDN line.

Further, recording media such as a digital video tape, a digital video disk, or CD-ROM may be used as the transmission line.

Thus the picture data transmitted to the receiving side is decrypted 46 first using the first secret key, then the compressed picture data is expanded 47 to be displayed 49. When the display is a digital data display unit, it is directly displayed, however, when it is an analog data display unit, it is converted to analog data 48.

When data is stored in hard disk, flexible disk, optical magnetic disk, writable video disk or the like, it is stored after being re-encrypted 50 using the second secret key.

In displaying again the picture data re-encrypted and stored, it is re-decrypted 52 using the second secret key and displayed 49. If the display unit is a digital data display unit, it is directly displayed, however, if it is an analog data display unit, it is converted to analog data 48.

Moreover, for data compression/expansion means and transmission path, appropriate ones compatible with the data are used.

Figure 6 shows an example of the data copyright management system disclosed in the prior Japanese Patent Application No. 237673/1994. This system uses the secret-key system as a cryptosystem.

In the case of this system, reference numeral 1 represents a database in which text data, binary data serving as a computer graphic display or a computer program, digital audio data, and digital picture data are stored by being encrypted, 14 represents a space satellite such as a communications satellite or a broadcasting satellite, 15 represents a data recorder such as a CD-ROM or a flexible disk, 2 represents a communication network such as a public telephone line offered by a communication enterprise or a CATV line offered by a cable television enterprise, 4 represents a primary user terminal, and 16 represents a key control center for managing a secret-key, and 17 represents a copyright management center for managing a data copyright.

Reference numerals 5, 6, and 7 represent a secondary user terminal, a tertiary user terminal, and n-order user terminal respectively, and 11, 12, and 13 represent a secondary disk, tertiary disk, and n-order disk serving as a recording medium such as a flexible disk or CD-ROM respectively. The symbol "n" represents an optional integer. When "n" is larger than 4, a corresponding user terminal and a corresponding disk are arranged between the tertiary user terminal 6 and the n-order user terminal 7 and between the tertiary disk 12 and the n-order disk 13 respectively.

On the above arrangement, the database 1, key control center 16, copyright management center 17, primary user terminal 4, secondary user terminal 5, tertiary user terminal 6, and n-order user terminal 7 are connected to the communication network 2.

In this figure, the path shown by a broken line is a path of encrypted data, a path shown by a solid line is a path of requests from each user terminal, and a path shown by a one-dot chain line is a path through which authorization information corresponding to a utilization request and a secret-key are transferred.

Moreover, each user who uses this system is previously entered in the database system. When the user is entered in the system, a database utilization software is given to the user. The database utilization software includes not only normal communication software such as a data communication protocol but also a program for running a copyright management program.

Original data M0 of text data, binary data as a computer graphic display or computer program, digital audio data, or digital picture data stored in the database 1 or data recording medium 15 is one-way supplied to the primary user terminal 4 via the communication network 2, satellite 14 or recording medium 15.

In this case, the data is encrypted with a first secret-key Ks1:

$$Cm0ks1 = E(Ks1, M0).$$

Even if data provided with advertisement to be offered free of charge, it is necessary to be encrypted in order to protect the copyright.

It is disclosed in the Japanese Patent Application No. 64889/1994 which is the prior application that the data utilization includes not only displaying of data which is the most basic usage but also storing, editing, copying, and transmitting of the data, a use permit key is prepared which corresponds to one or several forms of usage, and its management is executed by the copyright management program.

Moreover, it is described there that data is encrypted again by the copyright management program for use such as storing, copying, editing and transmitting of the data other than displaying of the data and displaying for editing the data.

In other words, the data whose copyright is claimed is encrypted to be distributed, and only when the data is displayed or displayed for editing the data in a user ter-

minal having a copyright treatment function, the data is decrypted to a plaintext.

This system disclosed in Japanese Patent Application No. 237673/1994 uses the method described in the prior application No. 64889/1994.

A primary user who desires primary utilization of the supplied encrypted data Cm0ks1 requests for primary utilization of the encrypted original data Cm0ks1 by designating the original data name or the original data number to the key control center 16 via the communication network 2 from the primary user terminal 4. In this case, the primary user must present information Iu1 for primary user to the key control center 16.

The key control center 16 receiving the primary utilization request from the primary user terminal 4 transfers first secret-key Ks1 for decrypting the encrypted original data Cm0ks1 obtained from the database 1 by the primary user and second secret-key Ks2 for re-encrypting the decrypted original data M0 or edited data M1 from the original data, together with a copyright management program P via the communication network 2 to the primary user terminal 4.

In the primary user terminal 4 receiving the first secret-key Ks1 as a decryption key and the second secret-key Ks2 as an encryption/decryption key, the encrypted original data Cm0ks1 is decrypted by the first secret-key Ks1 using the copyright management program P

$$M0 = D(Ks1, Cm0ks1)$$

to use the decrypted original data M0 directly or data M1 as edited.

When the data M which is the original data M0 or edited data M1 is stored in a memory or a built-in hard disk drive of the primary user terminal 4, only the primary user can use the data. However, when the data M is copied to the external recording medium 11 such as a flexible disk or transmitted to the secondary user terminal 5 via the communication network 2, a problem of a copyright due to secondary utilization occurs.

When the original data M0 obtained by the primary user is directly copied and supplied to a secondary user, the copyright of the primary user is not effected on the data M0 because the original data M0 is not modified at all. However, when the primary user produces new data M1 by editing the obtained data M0 or by using means such as combination with other data, the copyright of the primary user, i. e., secondary exploitation right occurred from secondarily utilizing original data, is effected on the data M1.

Similarly, when a secondary user produces new data M2 by editing the original data M0 or edited data M1 obtained from the primary user or by means such as combination of other data, the copyright of the secondary user; i. e., secondary exploitation right on the secondary user is also effected.

In this system, to correspond to the problem of the copyright, the data M is encrypted by the second secret-key Ks2 using the copyright management program P when the data M is stored, copied, or transmitted. Thereafter, in the primary user terminal 4, the data M is decrypted and encrypted by the second secret-key Ks2:

$$Cmks2 = E(Ks2, M)$$

$$M = D(Ks2, Cmks2).$$

It is free in principle that the primary user displays and edits data to obtain edited data. In this case, however, it is possible to limit the repetitions of the operation by the copyright management program.

When the data M is copied to the external recording medium 11 or transmitted via the communication network 2, the first secret-key Ks1 and the second secret-key Ks2 in the primary user terminal 4 are disused by the copyright management program P. Therefore, when reusing the data M the primary user requests for utilization of the data M to the key control center 16 to again obtain the second secret-key Ks2.

The fact that the user receives the regrant of the second secret-key Ks2 represents secondary utilization of data in which the data M has been copied to the external recording medium 11 or transmitted to the secondary user terminal 5 via the communication network 2. Therefore, the fact is entered in the copyright management center 17 from the key control center 16 and subsequent secondary utilization comes possible.

The data M is moved from the primary user terminal 4 to the secondary user terminal 5 by the external recording medium 11 or the communication network 2. When the data M is copied to the external recording medium 11 or transmitted via the communication network 2, it is encrypted by the second secret-key Ks2.

When the data M is copied to the external recording medium 11 or transmitted via the communication network 2, the first secret-key Ks1 and the second secret-key Ks2 in the primary user terminal 4 are disused. In this time, uncrypted primary user information Iu1 is added to the encrypted data Cmks2 stored in the primary user terminal 4 and when the encrypted data Cmks2 is transmitted to the secondary user, the primary user information Iu1 is also transferred.

A secondary user who desires secondary utilization of the encrypted data Cmks2 copied or transmitted from the primary user must designate original data name or data number to the copyright management center 17 via the communication network 2 by the secondary user terminal 5 and also present the secondary user information Iu2 to request for secondary utilization of the data Cmks2 to the center 17. In this time, the secondary user further presents the uncrypted primary user information Iu1 added to the encrypted data Cmks2 in order to clarify the relationship with the primary user.

The copyright management center 17 confirms that the primary user has received a regrant of the second secret-key Ks2 for secondary-utilizing the data, in accordance with the presented primary user information

lu1 and then, transfers the second secret-key Ks2 serving as a decryption key and the third secret-key Ks3 serving as an encryption/decryption key to the secondary user terminal 5 via the communication network 2.

In the secondary user terminal 5 receiving the second secret-key Ks2 and the third secret-key Ks3, the encrypted data Cmks2 is decrypted using the second secret-key Ks2 by the copyright management program P

$$M = D(Ks2, Cmks2)$$

and is secondarily utilized such as being displayed or edited.

In this system, the key control center 16 processes a primary utilization requests and the copyright management center 17 processes a secondary utilization requests. While the data M supplied to a primary user is encrypted by the first secret-key Ks1, the data M supplied to a secondary user is encrypted by the second secret-key Ks2. Moreover, the first secret-key Ks1 and the second secret-key Ks2 are transferred to the primary user as crypt keys from the key control center 16.

Therefore, if the secondary user, instead of the primary user, falsely requests for primary utilization to the key control center 16, the first secret-key Ks1 for decryption and the second secret-key Ks2 for encryption/decryption are transferred to the secondary user. However, the secondary user cannot decrypt the encrypted data Cmks2 by using the first secret-key Ks1 transferred as a decryption key.

Therefore, it is impossible to falsely request for data utilization and resultingly, not only the original copyright of data but also the copyright of the primary user on the data are protected.

When storing, copying, or transmitting of the data M other than displaying and displaying for editing is performed in the secondary user terminal 5, the data M is encrypted using the third secret-key Ks3 by the copyright management program P and thereafter, the data is decrypted and encrypted by the third secret-key Ks3:

$$Cmks3 = E(Ks3, M)$$

$$M = D(Ks3, Cmks3).$$

Moreover, it is free in principle that the secondary user displays and edits data to obtain the edited data M2. In this case, it is possible to limit the repetitions of the operation by the copyright management program P.

When the data M is copied to the external recording medium 12 or transmitted via the communication network 2, the second secret-key Ks2 and the third secret-key Ks3 in the secondary user terminal 5 are disused by the copyright management program P. Therefore, when reusing the data M, the secondary user requests for the utilization of the data to the copyright management center 17 to again obtain the third secret-key Ks3.

The fact that the secondary user receives a regrant of the third secret-key Ks3 represents secondary utilization of data in which the data M has been copied to the external recording medium 12 or transmitted to the tertiary user terminal 6 via the communication network 2. Therefore, the fact is entered in the copyright management center 17 and allows subsequent data use.

The data M is moved from the secondary user terminal 5 to the tertiary user terminal 6 by the external recording medium 12 or by the communication network 2. When the data M is copied to the external recording medium 12 or transmitted via the communication network 2, it is encrypted by the third secret-key Ks3.

When the data M is copied to the external recording medium 12 or transmitted to the tertiary user terminal 6 via the communication network 2, the second secret-key Ks2 and the third secret-key Ks3 in the secondary user terminal 5 are disused. In this case, the uncrypted secondary user information lu2 is added to the encrypted data Cmks3 stored in the secondary user terminal 5, and when the encrypted data Cmks3 is transmitted to a tertiary user, the secondary user information lu2 is also transferred.

In adding each user information to data, there are two cases: a case in which every information is added to data whenever it is copied or transmitted; and another in which the history updated whenever the data is copied or transmitted is stored in the copyright management center.

A tertiary user who desires tertiary utilization of the encrypted data Cmks3 copied or transmitted from the secondary user must designate original data name or number to the copyright management center 17 from a tertiary user terminal 6 via the communication network 2 and also presents the tertiary user information lu3 to request for tertiary utilization of the data. In this time, the tertiary user further presents the uncrypted secondary user information lu2 added to the encrypted data Cmks3 in order to clarify the relationship with the secondary user.

The copyright management center 17 confirms that the secondary user has received a regrant of the third secret-key Ks3 for preparation of tertiary-utilizing the data, in accordance with the presented secondary user information lu2 and then, transfers the third secret-key Ks3 serving as a decryption key and fourth secret-keyKs4 serving as an encryption/decryption key to the tertiary user terminal 6 via the communication network 2.

In the tertiary user terminal 6 receiving the third secret-key Ks3 and the fourth secret-key Ks4, the encrypted data Cmks3 is decrypted using the third secret-key Ks3 by the copyright management program P

$$M = D(Ks3, Cmks3)$$

and is tertiarily utilized such as being displayed or edited.

In this system, the data M supplied to the primary user is encrypted by the first secret-key Ks1 and the data M supplied to the secondary user is encrypted by the second secret-key Ks2, and the data M supplied to the tertiary user is encrypted by the third secret-key Ks3.

Therefore, if the tertiary user, instead of the primary user, falsely requests for primary utilization to the key control center 16, the first secret-key Ks1 for decryption and the second secret-key Ks2 for encryption/decryption are transferred to the tertiary user. However, it is impossible to decrypt the encrypted data Cmks3 by the first secret-key Ks1 transferred as a decryption key. Moreover, if the tertiary user, instead of the secondary user, falsely requests for secondary utilization to the copyright management center 17, the second secret-key Ks2 and the third secret-key Ks3 are transferred to the tertiary user as a decryption key and an encryption/decryption key respectively. However, it is impossible to decrypt the encrypted data CmKs3 by the second secret-key Ks2 transferred as a decryption key.

Therefore, it is impossible to falsely request for data utilization. As a result, not only the original copyright of the data but also the copyrights of the primary and secondary users on the data are protected.

The same procedure is applied to quaternary and subsequent utilization.

In the above described system, the database 1, key control center 16, and copyright management center 17 are separately arranged. However, it is not always necessary to arrange them separately. It is also possible to set all of or proper two of them integrally.

Moreover, it is also possible to request for a regrant of the secondary secret-key from the primary user not to the key control center 16 but to the copyright management center 17.

In Figures 7(a) and 7(b), signal process flow in data edit method of digital video or digital audio is shown. An edit flow generally processed is shown in 7(a) and an edit flow 7(b) which can avoid deterioration of signals.

In the edit flow shown in 7(a), signals supplied as digital signals 61 are converted to analog signals 62, the analog signals are then edited while being displayed 64, and the analog signals completed editing are re-digitized 65 to be stored, copied, and transferred 66.

Though this process may be simple, it can not avoid deterioration of signals since signal is edited in analog and re-digitized after completion of editing.

The edit flow shown in 7(b), digital signals 61 are converted to analog signals 62 to be displayed. While the analog signals 62 are used in editing 63, the analog signals are used only for displaying 64 rather than for storing, copying, transferring.

Signals for storage, copy, and transfer are edited 67, copied, and transferred 66 in the form of digital signals 61 correspond to signals displayed in analog.

In the case of this edit flow, there is no deterioration of signals since digital signals which are stored, copied, and transferred are never converted to analog signals.

Figures 8(a) and 8(b) illustrate flow examples when editing encrypted data to which signal process in data editing method of digital video or digital audio shown in Figure is applied. 8(a) shows a simplified signal processing flow and 8(b) shows a signal processing flow which allows sufficient copyright management.

In the signal processing flow shown in (a), the original data 71 Cm0ks1, encrypted using the first secret-key Ks1 and supplied is initially decrypted 72 using the first secret key Ks1:

$$M0=D\ (Ks1,\ Cm0ks1),$$

and the decrypted data M0 is then edited 73 while being displayed 74. The data M1 completed editing is re-encrypted 75 using the second secret key Ks2:

$$Cm1ks2=E\ (Ks2,\ M1)$$

and stored, copied, and transferred 76.

Though the process may be simple, copyright can not be properly managed since there is possibility that the decrypted data might be stored, copied, or transferred due to the data editing process in decrypted form.

On the other hand, in the signal processing flow shown in 8(b), the original data 71 Cm0ks1, encrypted using the first secret key Ks1 is decrypted 72 using the first secret-key Ks1:

$$M0=D\ (Ks1,\ Cm0ks1)$$

the decrypted data M0 is displayed 74.

While, the encrypted data Cm0ks1 is edited 73, lead by the decrypted data M0, and the original data M0 for storage or the edited data M1 are re-encrypted using the second secret-key:

$$Cm0ks2=E\ (Ks2,\ M0)$$

$$Cm1ks2=E\ (Ks2,\ M1)$$

the encrypted data Cm0ks2 or Cm1ks2 is stored, copied, and transferred 76.

Without being decrypted corresponding to the decrypted and displayed data, it is edited 77 in the encrypted form, and the edition program and the data still encrypted are used for store, copy, transfer 76.

In the case of this signal processing flow, the decrypted data are never stored, copied, or transferred since the data for storage, copy, transfer remains encrypted.

In the data copyright management system which applies the data copyright management apparatus of the present invention, while data is decrypted for utilization when the obtained encrypted data are displayed/edited, data copyright is managed by encrypting data when obtained or edited data is stored/copied/transferred.

However, the data copyright management unit 15 of the prior invention shown in Figure 2 and the data copyright management unit 30 of the present invention described in Figure 3 can perform only one process of decryption of encrypted data or encryption of decrypted data. When decrypted or edited data is stored/copied/transferred, therefore, it is necessary to store data in the user terminal or RAM of the data copyright manage-

ment apparatus to re-encrypt the stored data afterwards. Thus, there is a possibility that decrypted or edited data might be lost due to accident or misoperation as well as posing limitation in volume to the data that can be processed.

With the exception of some high-class MPU, general MPU used in personal computers does not take into account the multiprocessor configuration which allows concurrent operation of plural microcomputers. Therefore, plural operations can not be performed at the same time, although accessory units are connected to the system bus of the personal computer.

Accordingly, to connect the data copyright management unit 15 shown in Figure 2 or the data copyright management unit 30 shown in Figure 3 to the system bus 22 of the user terminal 20 never provides multiprocessor function that enables concurrent operation of MPU 21 or 46 and CPU 16, and the processes of decryption of encrypted data and re-encryption of decrypted data are performed alternately, not concurrently. Thus, a large amount of data can not be processed since the data to be encrypted and decrypted is limited by the capacity of RAM. Further, it is impossible to increase the processing speed, even if the amount of data is not large.

On the other hand, in the data copyright management system described as the prior application, encrypted data obtained is decrypted to use for displaying or editing, and when the obtained or edited data is stored, copied, or transferred, it is re-encrypted in order to prevent unauthorized utilization of the data. Therefore, it is desirable that the apparatus in the data copyright management system of the present invention performs not only decryption but also re-encryption of data at the same time.

Recently, a PCI (Peripheral Component Interconnect) bus has attracted attention as means for implementing a multiprocessor configuration of typical personal computer.

The PCI bus is a bus for external connection connected to a system bus of personal computer via a PCI bridge, and allows to implement a multiprocessor configuration.

Figure 9 shows embodiment 2 of this invention, which is a configuration of data copyright management apparatus using a PCI bus and the same configuration of data copyright management unit 15 as shown in Figure 3, that is, a computer configuration having a CPU 16, a local bus 17 for the CPU 16, and ROM 18, RAM 19, and EEPROM 31 connected to the local bus 17.

In a user terminal 20, a PCI bus 81 is connected to a system bus 22 for a microprocessor 21 via a PCI bridge 82 and the local bus 17 for the CPU 16 of a data copyright management apparatus 80 is connected to the PCI bus 81. Also connected to the system bus 22 of the user terminal 20 are a communications device (COMM) 23 which receives data from external databases and transfers data to the external of tire terminal, a CD-ROM drive (CDRD) 24 which reads data supplied on CD-ROM a flexible disk drive (FDD) 25 which copies received or edited data to supply to the external of terminal, and hard disk drive (HDD) 26 used for storing data. COMM 23, CDRD 24, FDD 25, and HDD 26 may also be connected to the PCI bus 81.

While ROM, RAM etc., of course, are connected to the system bus 22 of the user terminal, these are not shown in Figure 9.

Configurations and operations of other parts are the same as embodiment 1 shown in Figure 3, and further explanation of them will be omitted.

A decryption task is performed by the MPU 21 of the user terminal 20 and an encryption task is performed by the CPU 16 of the data copyright management apparatus 80 at the same time, and vice versa. Since the configuration of the MPU 21 and CPU 16 in this embodiment is a multiprosessor configuration which performs parallel processing with a PCI bus, high processing speed can be achieved.

Other typical means for attaching external devices to a personal computer include SCSI (Small Computer System Interface), which is used for the connection of external storage medium such as hard disk drives and CD-ROM drives.

Up to eight devices, including the personal computer itself to which SCSI is attached, can be connected to SCSI, and a plurality of computers may be included in the eight devices. Each of these computers can play an equivalent role, in other words, SCSI function as not only an interface but also a multiprocessor bus.

Taking advantage of this function of SCSI, embodiment 3 connects a data copyright management apparatus 85 to the system bus 22 of a user terminal 20 via SCSI 86 (hereinafter called the "SCSI bus", for clear understanding) instead of the PCI bus 81 in embodiment 2.

Figure 10 shows a configuration block diagram of a data copyright management apparatus of embodiment 3 which uses and SCSI bus according to the present invention.

In embodiment 3, the configuration of the data copyright management apparatus 85 is the same as the data copyright management apparatus shown in Figure 3, that is, the apparatus has a CPU 16, a local bus 17 for the CPU 16, and ROM 18, RAM 19, and EEPROM 31 connected to the local bus 17.

On the other hand, an SCSI bus 86, which is controlled by an SCSI controller (SCSICONT) 87, is connected to a system bus 22 for a microprocessor 21 of a user terminal 20, and the local bus 17 for the CPU 16 of a data copyright management apparatus 85 is connected to this SCSI bus 86.

Also connected to the system bus 22 of the user terminal 20 are a communications device (COMM) 23 which receives data from external databases and transfers data to the external of the terminal, a CD-ROM drive (CDRD) 24 which reads data supplied on CD-ROM, a flexible disk drive (FDD) 25 which copies received or edited data to supply to the external of terminal, and hard disk drive (HDD) 26 used for storing data. COMM 23,

CDRD 24, FDD 25, and HDD 26 may also be connected to the SCSI bus 86.

While ROM, RAM etc., of course, are connected to the system bus 22 of the user terminal, these are not shown in Figure 10.

Configurations and operations of other parts are the same as embodimnt 1 shown in Figure 3, and further explanation of them will be omitted.

A decryption task is performed by the MPU 21 of the user terminal 20 and a encryption task is performed by the CPU 16 of the data copyright management apparatus 85 at the same time, and vice versa. Since the configuration of the MPU 21 and CPU 16 in this embodiment is a multiprosessor configuration which performs parallel processing with an SISI bus 86, high processing speed can be achieved.

Other means for implementing a multiprocessor configuration, such as SCI (Scalable Coherent Interface), may be used, and, if possible, the microprocessors may be connected with each other without using a bus.

Data to be managed by the data copyright management apparatus of the present invention includes, in addition to text data, graphic data, computer programs, digital audio data, JPEG-based still picture data, and MPEG-based moving picture.

The above-mentioned multiprocessor configuration of the data copyright management apparatus 80 of embodiment 2 and the data copyright management apparatus 85 of embodiment 3 is implemented by connecting the apparatus to the system bus 22 of the microprocessor 21 in the user terminal 20 via a PCI bus or a SCSI bus. In such multiprocessor configuration, the MPU 21 of the user terminal 20 must also control the overall system. For relatively slow and small data such as text data and graphic data, data copyright management with encryption and re-encryption can be performed by the multiprocessor configuration using the MPU 21 and CPU 16, for JPEG-still-picture-based moving picture data and MPEG1 or MPEG2-based moving picture data, however, data copyright management by such configuration is considerably difficult to perform because the data is fast and large.

To deal with this problem, a multiprocessor system is configured by connection a first data copyright management apparatus 80 and a second data copyright management apparatus 90 to a PCI bus 81 in embodiment 4 shown in Figure 11.

The configuration of the second data copyright management apparatus 90 is the same as that of the first data copyright management apparatus 80, that is, the apparatus comprises a CPU 91, a local bus 94 for the CPU 91, and ROM 92, RAM 93, and EEPROM 95 connected to the local bus 94.

In this embodiment, the first data copyright management apparatus 80 decrypts encrypted data and the second data copyright management apparatus 90 re-encrypts decrypted data.

Fixed information, such as software for utilizing databases and user data, are stored in the ROM 18 of the first data copyright management apparatus 80 decrypting encrypted data. A first crypt-key and data copyright management system program supplied by a key control center or copyright management center are stored in the RAM 19.

Similarly, fixed information, such as software for utilizing databases and user data, are stored in the ROM 92 of the second data copyright management apparatus 90 re-encrypting decrypted data, and a second crypt-key and data copyright management system program supplied by a key control center or copyright management center are stored in the RAM 93.

In this multiprocessor configuration, SCSI or SCI may be used, and, if possible, the microprocessors may be connected with each other without using a bus.

In the prior application shown in Figure 2 and in embodiment 1 of the present invention described with reference to Figure 3, the communications device (COMM) 23 to which encrypted data is supplied and the CD-ROM drive (CDRD) 24 are connected to the system bus of the user terminal 20. In order to decrypt encrypted data, therefore, the encrypted data must be transmitted by way of the system bus of the user terminal 20 and the local bus of the data copyright management apparatus, and consequently, the processing speed can be slowed. This is true for a configuration in which those attached devices are connected to a PCI bus or SCSI bus.

In embodiment 5 shown in Figure 12, a communications device 23 to which encrypted data is supplied and a CD-ROM drive 24 are connected to a local bus 17 of a data copyright management apparatus 97 for decryption, in order to prevent processing speed from being slowed.

The data copyright management apparatus 97 of embodiment 5 shown in Figure 12 is a data copyright management apparatus for decryption and its configuration is essentially the same as that of the data copyright management apparatus 30 of embodiment 1 shown in Figure 3, that is, the computer system has a CPU 16, a local bus 17 for CPU 16, and ROM 18, RAM 19 and EEPROM 31 connected to the local bus 17, and a communication device COMM 23 and a CD-ROM drive CDRD 24 are connected to the local bus 17.

Fixed information, such as a copyright management program, cryptography program based on crypt algorithm, and user data, are stored in the ROM 18.

Copyright information is stored in the EEPROM 31. If the copyright management program and cryptography program are supplied from the external such as databases, those programs are stored in the EEPROM 31, rather than in the ROM 18.

A crypt-key for decryption and a data copyright management system program supplied from a key control center or copyright management center are stored in the RAM 19.

Encrypted data supplied from the COMM 23 or CDRD 24 is decrypted by the data copyright management apparatus 97 and transferred to a user terminal 95.

While the above-mentioned data copyright management apparatus 80 and 90 of embodiment 4 are described as being configured separately, these apparatus, of course, can be configured as a unit.

Figure 13 shows a data copyright management apparatus of embodiment 6 which is extended from the data copyright management apparatus 97 of embodiment 5.

In the prior application shown in Figure 2 and the embodiment 1 described with reference to Figure 3, the storage medium, such as HDD 26, for storing re-encrypted data are connected to the system bus 22 of the user terminal 20. In order to store re-encrypted data, therefore, the encrypted data must be transmitted by way of the system bus 22 of the user terminal 20 and the local bus 17 of the data copyright management unit 15 or data copyright management unit 30, and consequently, processing speed can be slowed. This is true for a configuration in which those attached devices are connected to a PCI bus or SCSI bus.

In the data copyright management apparatus 100 of the embodiment 6 shown in Figure 13, in addition to the communications device COMM 23 and the CD-ROM drive CDRD 24 connected to the local bus 17 in the data copyright management apparatus 97 for decryption in the embodiment 5 shown in Figure 12, storage devices such as HDD 26 for storing re-encrypted data are connected to the local bus 94 of the data copyright management apparatus 101 for re-encryption.

The configuration of the data copyright management apparatus 101 for re-encryption in embodiment 6 is essentially the same as that of the data copyright management unit 30 shown in Figure 3, that is, the computer system has a CPU 91, a local bus 94 for the CPU 91, and ROM 92, RAM 93 and EEPROM 95 connected to the local bus 94, and HDD 26 is connected to the local bus 94.

Fixed information, such as a copyright management program, cryptography program based on crypt algorithm, and user data, are stored in the ROM 92.

Copyright information is stored in the EEPROM 95. If the copyright management program and cryptography program are supplied from the external such as databases, those programs are stored in the EEPROM 95 rather than the ROM 92.

A crypt-key for re-encryption and a data copyright management system program supplied from a key control center or copyright management center are stored in the RAM 93.

Data re-encrypted by the copyright management apparatus 101 for re-encryption is stored in HDD 26.

While the above-mentioned data copyright management apparatus 100 and 101 of embodiment 6 are described as being configured separately, these apparatus, of course, can be configured as a unit.

Digital data includes, in addition to text data, graphic data, computer programs, digital sound data, JPEG-based still picture data, and MPEG-based moving picture data.

A typical user terminal which utilizes copyrighted data is computer apparatus such as personal computers. Other apparatus which utilize such data are receivers such as television sets, set-top boxes used with those receivers, digital recording apparatus such as video tape recorders, digital video disk recorders, and digital audio tapes (DAT) which store digital data, and personal digital assistants (PDA).

The data copyright management apparatus shown in Figure 2 which is configured as an expansion board, IC card, or PC card and described in the prior patent application No. 237673/1994 or the data copyright management apparatus shown in Figure 6 may be used by attaching it to a user terminal which is a computer, receiver, set-top box, digital recording medium, or PDA. However, it is desirable that a data copyright management apparatus is factory-installed in the user terminal in order to eliminate labor and failure during the attachment of the apparatus.

To accomplish this, in each embodiment of the present invention, a data copyright management apparatus is implemented in the form of a monolithic IC, hybrid IC, or built-in subboard and is incorporated in a user terminal such as computer apparatus such as personal computers, receivers such as television sets, set-top boxes used with those receivers, digital recording medium such as digital video tape recorders, digital video disk recorders, and digital audio tape (DAT) which store digital signals, or personal digital assistants (PDA).

Further, the apparatus for managing data copyright described above can be applied not only to the data utilization but also to the handling of the digital cash and video conference systems.

The digital cash system which has been proposed so far is based on a secret-key cryptosystem. The encrypted digital cash data is transferred from a bank account or a cash service of a credit company, and is stored in the IC card so that a terminal device for input/output is used to make a payment. The digital cash system which uses this IC card as an electronic cash-box can be used at any place such as shops or the like as long as the input/ output terminal is installed. However, the system cannot be used at places such as homes or the like where no input/output terminal is installed.

Since the digital cash is an encrypted data, any device can be used as the electronic cash-box which stores digital cash data, in addition to the IC card, as long as the device can store encrypted data and transmit the data to the party to which the payment is made. As a terminal which can be specifically used as the electronic cash-box, there are personal computers, intelligent television sets, portable telephone sets such as personal information terminal, personal handyphone system (PHS), intelligent telephone sets, and PC cards or the like which has an input/ output function.

Trades in which such terminals are used as an electronic cash-box for a digital cash can be actualized by replacing in the constitution of the data copyright management system, the database with a customer's bank,

a first user terminal with a customer, the second user terminal with a retailer, the copyright control center with a retailer's bank and a third user terminal with a wholesaler or a maker.

An example of the trading system will be explained in which the digital cash is transferred via a communication network by using Figure 14.

The example uses the constitution of the data copyright management system shown in Figure 1. In Figure 14, reference numeral 111 represents a customer, 112 a bank of the customer 111, 113 a retail shop, 114 a bank of the retail shop 113, 115 a maker, 116 a bank of the maker 115, 2 a communication network such as a public line provided by a communication enterprise or CATV line provided by a cable television enterprise. Customer 111, the customer's bank 112, the retail shop 113, the retail shop's bank 114, the maker 115, the maker's bank 116 can be mutually connected with the communication network 2. In this system, the customer 111 can use a credit company offering cashing service other than banks and he can also interpose appropriate number of wholesalers between the retail shop and the maker.

In addition, 117 and 118 are either IC cards or PC cards in which digital cash data is stored. The cards are used when the communication network is not used.

Incidentally, in Figure 14, what is represented by a broken line is a path of encrypted digital cash data, what is represented by the solid line is a path of requests from the customer, the retail shop or the maker, and what is represented by a one-dot chain line is a path of the secret-key from each bank.

In this example, first secret-key prepared by the customer's bank 112, the second secret-key generated by the customer, the third secret-key generated by the retail shop, and the fourth secret-key prepared by the maker are used as crypt keys.

Further, while the customer's bank 112, the retail shop's bank 114, and the maker's bank 116 are explained as separate entities, these can be considered as a financial system as a whole.

Digital cash management program P for encrypting and decrypting the digital cash data is preliminarily distributed to the customer 111 and is stored in the user terminal. Further, it is possible to transfer the digital cash management program P together with data every time trade with the bank is executed. Further, it is desirable to install the common digital cash management program P in all banks.

The customer 111 uses the user terminal to designate the amount of money via the communication network 2 to request drawing out from the account of the customer's bank 112 to the bank. At this time, the terminal presents customer information Ic of the customer 111.

The customer's bank 112 which receives the customer's request of drawing out from the account selects or generates the first secret-key Ks1 so that the digital cash data MO of the amount is encrypted by the first secret-key Ks1:

$$CmOks1=E(Ks1, MO)$$

and the encrypted digital cash data CmOks1 and the first secret-key Ks1 for a decrypting key are transferred to the customer 111, and the customer information Ic and the first secret-key Ks1 are stored.

In this case, the first secret-key Ks1 can be selected from what is preliminarily prepared by the customer's bank 112, and also may be generated by presentation of the customer information Ic at the time of drawing by the customer using the digital cash management program P on the basis of the customer information Ic:

$$Ks1=P(Ic).$$

Through this means, the first secret-key Ks1 can be private for the customer 111. At the same time, it is not necessary to transfer the first secret-key Ks1 to the customer 111 so that the safety of the system can be heightened.

Further, the first secret-key Ks1 can be generated on the basis of the bank information Ibs of the customer's bank 112 or on the basis of the bank information Ibs and the date of key generation.

The customer 111 to which the encrypted digital cash data CmOks1 and the first secret-key Ks1 are transferred generates second secret-key Ks2 according to any one or both of the customer information Ic and the first secret-key Ks1 using the digital cash management program P, for example:

$$Ks2=P(Ic)$$

and the generated second secret-key Ks2 is stored in the user terminal.

Further, the customer 111 uses the first secret-key Ks1 to decrypt the encrypted digital cash data CmOks1 with the digital cash management program P:

$$MO=D(Ks1, CmOks1)$$

and the content is confirmed. When the decrypted digital cash data MO whose content is confirmed is stored in the user terminal as a cash-box, it is encrypted by the generated second secret-key Ks2 using the digital cash management program P:

$$CmOKs2=E(Ks2, MO).$$

The first secret-key Ks1 is disused at this time.

The customer 111 who wishes to buy an article from the retail shop 113 decrypts the encrypted digital cash data CmOks2 which is stored in the user terminal as a cash-box by the digital cash management program P using the second secret-key Ks2:

$$MO=D(Ks2, CmOks2)$$

and the digital cash data M1 which corresponds to the necessary amount of money is encrypted by the second secret-key ks2 using the digital cash management program P:

$$Cm1ks2=E(Ks2, M1)$$

and then, the payment is made by transmitting the encrypted digital cash data Cm1ks2 to the user terminal as a cash-box of retail shop 113 via the communication network 2.

At this time, the customer information Ic is also transmitted to the user terminal of the retail shop 113.

Further, the residual amount digital cash data M2 is encrypted by the second secret-key Ks2 using the digital cash management program P:

$$Cm2ks2=E(Ks2, M2)$$

and stored in the user terminal of the customer 111.

The retail shop 113 to which the encrypted digital cash data Cm1ks2 and the customer information Ic are transferred stores the transferred encrypted digital cash data Cm1ks2 and customer information Ic in the user terminal, and presents the customer information Ic to the retail shop's bank 114 via the communication network 2 for confirming the content to request the transmission of the second secret-key Ks2 for decription.

The retail shop's bank 114 which is requested by the retail shop 113 to transmit the second secret-key Ks2 transmits the request of the transmission of the second secret-key Ks2 and the customer information Ic to the customer's bank 112.

The customer's bank 112 which is requested to transmit the second secret-key Ks2 from the retail shop's bank 114 generates the second secret-key Ks2 according to the customer information Ic by the digital cash management program P in the case where the second secret-key Ks2 is based only on the customer information Ic, or generates the second secret-key Ks2 according to the customer information Ic and the first secret-key Ks1 by the digital cash management program P in the case where the second secret-key Ks2 is based on the customer information Ic and the first secret-key Ks1, and transmits the generated second secret-key Ks2 to the retail shop's bank 114.

The retail shop's bank 114 to which the second secret-key Ks2 is transmitted from the customer's bank 112 transmits the second secret-key Ks2 to the retail shop 113 via the communication network 2.

The retail shop 113 to which the second secret-key Ks2 is transferred decrypts the encrypted digital cash data Cm1ks2 by the second secret-key Ks2 using the digital cash management program P:

$$M1=D(Ks2, Cm1ks2)$$

and after confirming the amount of money, forwards the article to the customer 111.

Incidentally, in this case, the retail shop 111 can directly requests the transfer of the second secret-key Ks2 to the customer's bank 112 instead of the retail shop's bank 114.

In case where the digital cash received by the retail shop 113 is deposited in the account of the retail shop's bank 114, the customer information Ic is transferred to the retail shop's bank 114 together with the encrypted digital cash data Cm1ks2 via the communication network 2.

The retail shop's bank 114 to which the encrypted digital cash data Cm1ks2 and the customer information Ic are transferred requests the transfer of the second secret-key Ks2 to the customer's bank 112 by transmitting the customer information Ic.

The customer's bank 112, which is requested to transfer the second secret-key Ks2 from the retail shop's bank 114, generates the second secret-key Ks2 according to the customer's information Ic by the digital cash management program P when the second secret-key Ks2 is only based on the customer's information Ic, or generates the second secret-key Ks2 according to the customer's information Ic and the first secret-key Ks1 by the digital cash management program P when the second secret-key Ks2 is based on the customer's information Ic and the first secret-key Ks1, then the generated second secret-key Ks2 is transferred to the retail shop's bank 114.

The retail shop's bank 114, to which the second secret-key Ks2 is transferred from the customer's bank 112, decrypts the encrypted digital cash data Cm1ks2 by the second secret-key Ks2 using the digital cash management program P:

$$M1=D(Ks2, Cm1ks2)$$

and the decrypted digital cash data M1 is deposited in the bank account of the retail shop's bank 114.

In the general trade system, the retail shop 113 stocks products from the maker 115 or from the wholesaler which intervenes between the retail shop 113 and the maker 115. Then the retail shop 113 sells the products to the customer 111. Consequently, a trading form is present between the customer 111 and the retail shop 113 just as between the retail shop 113 and the maker 115.

The handling of the digital cash between the retail shop 113 and the maker 115 is not basically different from the handling of the digital cash which is carried out between the customer 111 and the retail shop 113. Therefore, the explanation there will be omitted for the sake of clarity.

In this digital cash system, the digital cash is handled through banks. As information such as the processed amount of the digital cash, date, and the secret-key demanding party information with respect to the handling of the digital cash is stored in the customer's bank, the residual amount of digital cash and usage history can be grasped.

Even in the case where the user terminal which is an electronic cash-box storing the digital cash data cannot be used owing to the loss or the breakage, it is possible to reissue the digital cash on the basis of the residual amount, and usage history kept in the customer's bank.

It is desirable to add a digital signature to the digital cash data for heighten the safety of the digital cash.

In this example, digital cash is added by the customer's information which may be accompanied by digital signature. Therefore, the digital cash in the example can also have a function of settlement system for checques drawn by customers.

Also this system can be applicable to various systems in the international trading such as payment settlement of import/export by a negotiation by a draft using a letter of credit and a bill of lading which have been executed by documents.

In the video conference system, a television picture has been added to the conventional voice telephone set. Recently the video conference system is advanced in which a computer system is incorporated in the video conference system so that the quality of the voice and the picture are improved, and data can be handled at the same time as well as the voice and the picture.

Under these circumstances, security against the violation of the user's privacy and the data leakage due to eavesdropping by persons other than the participants of the conference are protected by the cryptosystem using a secret-key.

However, since the conference content obtained by the participants themselves are decrypted, in the case where participants themselves store the content of the conference and sometimes edit the content, and further, use for secondary usage such as distribution to the persons other than the participants of the conference, the privacy of other participants of the video conference and data security remains unprotected.

In particular, the compression technology of the transmission data is advanced while the volume of the data storage medium is advanced with the result that the possibility is getting more and more realistic that all the content of the video conference is copied to the data storage medium or is transmitted via a network.

In view of the circumstances, the example is intended, when video conference participants perform secondary use, to secure the privacy of other participants and data security by using the aforementioned constitution of the data copyright management system.

This video conference data management system can be actualized, for example, by replacing the database in the data copyright management system constitution shown in Figure 1 with a participant of the video conference, the first user terminal with another participant of the video conference, and the second user terminal with non-participant of the video conference.

An example when utilizing will be explained by using Figure 15.

Referring to Figure 15, reference numeral 121 represents a participant as a host of the video conference, 122 a participant of the video conference as a guest, 123 a non-participant of the video conference as a user, 124 a non-participant of the video conference as another user, 2 a communication network such as a public telephone line provided by the communication enterprise and a CA television line provided by the cable television enterprise or the like. The participant 121 of the video conference is connected to the participant 122 of the video conference via the communication network 2. Further, the participant 122 of the video conference can be connected to the non-participant 123 of the video conference, and the non-participant 123 of the video conference to the non-participant 124 of the video conference, via the communication network 2. Reference numeral 125 and 126 represent a data recording medium.

Referring to Figure 15, what is represented by the broken line is a path of the encrypted video conference content, represented by the solid line is a path requesting the crypt key from the non-participants of the video conference 123 and 124 to the participant of the television conference 121, and represented by the one-dot chain line is a path of crypt keys from the participant of the video conference 121 to the participant of the video conference 122 and the non-participants of the video conference 123 and 124.

In this example, a video conference data management system is described here only the protection for data security and privacy in case of the video conference participant 121 to simplify the explanation, however, it is of course, possible to protect for data security and privacy of the video conference participant 122.

A video conference data management program P for encryption/decryption of the video conference data of the participant 121 including audio and picture is previously distributed to the video conference participant 122 and the video conference non-participants 123 and 124, and is stored in each terminal. This video conference data management program P may be transferred whenever a crypt-key is transferred.

In this example, further, a first secret-key prepared by the video conference participant 121, a second secret-key prepared by the video conference participant 122, a third secret-key prepared by the video conference non-participant 123 and subsequent secret-keys prepared similarly are used as a crypt key.

The video conference participant 121 and the video conference participant 122 perform the video conference by transmitting audio, picture and data (referred to as video conference data on the whole) each other, using each terminal via communication network 2. Before the video conference, the video conference participant 121 generates or selects the first secret-key Ks1 to transfer to the video conference participant 122 prior to the start of the video conference.

The video conference participant 122 receiving the first secret-key Ks1 generates the second secret-key

Ks2 by the first secret-key Ks1 using the video conference data management program P:

$$Ks2=P(Ks1).$$

The generated second secret-key Ks2 is stored in the terminal.

The video conference participant 121 encrypts the video conference data MO with the first secret-key Ks1, in the video conference through the communication network 2:

$$CmOks1=E(Ks1, MO)$$

and transfers the encrypted video conference data CmOks1 to the video conference participant 122.

The video conference participant 122 who receives the video conference data CmOks1 encrypted by the first secret-key Ks1 decrypts the video conference data CmOks1 by the first secret-key Ks1:

$$M0=D(ks1, CmOks1)$$

and uses decrypted video conference data MO.

Further, the second secret-key Ks2 is generated based on the first secret-key Ks1 with the video conference data management program P:

$$Ks2=P(Ks1).$$

In the case where the decrypted video conference data MO is stored in the terminal of the participant 122 of the video conference, copied to the data record medium 125, or transferred to the non-participant of the video conference via the communication network 2, the data M is encrypted by the second secret-key Ks2 using the video conference data management program P:

$$Cmks2=E(Ks2, M).$$

The encrypted data Cmks2 is copied to the record medium 125 or supplied to the non-participant of the video conference via the communication network 2, together with the video conference data name or the video conference data number.

The non-participant of the video conference 123 who obtains the encrypted data CmKs2 requests to the participant 121 for the secondary use of the video conference data M from the terminal by specifying the name or number of the video conference data.

The participant 121 of the video conference who receives the request for the second use of the data M finds out the first secret-key Ks1 according to the name or the number of the video conference data name or number to generate the second secret-key Ks2 based on the first secret-key Ks1:

$$Ks2=P(Ks1)$$

and supplies the generated second secret-key Ks2 to the non-participant of the video conference 123.

The non-participant of video conference 123 who receives the second secret-key Ks2 decrypts the encrypted data Cmks2 by the second secret-key Ks2 by using the television conference data management program P:

$$M=D(Ks2, Cmks2)$$

and then, uses decrypted video conference data M.

In the case where the video conference data M is stored in the terminal of the non-participant of the video conference 123, copied to the record medium 126, or transmitted to the non-participant of the video conference 124, the video conference data M is encrypted by the second secret-key Ks2 using the video conference data management program P:

$$Cmks2=E(Ks2, M).$$

Incidentally, the third secret-key Ks3 may be generated on the basis of the second secret-key Ks2 with the video conference data management program P:

$$Ks3=P(Ks2),$$

and the data M can be encrypted with the video conference data management program P by this generated third secret-key Ks3:

$$Cmks3=E(Ks3, M).$$

Claims

1. A data copyright management apparatus used with a user terminal for utilizing digital data,

said digital copyright management apparatus comprising a central processing unit, a central processing unit bus, read-only semiconductor memory, electrically erasable programmable memory, and read/write memory;

wherein,

said central processing unit, said read-only semiconductor memory, said electrically erasable programmable memory, and read/write memory are connected to said central processing unit bus, and a system bus of said user terminal is able to be connected to said central processing unit bus;

a data copyright management system program, a copyright management program, and user information are stored in said read-only semiconductor memory;

a second private-key, a permit key, a second secret-key, a copyright management program, and copyright information are stored in said electrically erasable programmable memory; and

a first public-key, a first private-key, a second

public-key, and a first crypt-key are transmitted to said read/write memory during operation.

2. A data copyright management apparatus used with a user terminal for utilizing digital data,

said data copyright management apparatus comprising a central processing unit, a central processing unit bus, read-only semiconductor memory, electrically erasable programmable memory, and read/write memory;

wherein,

said central processing unit, said read-only semiconductor memory, said electrically erasable programmable memory, and said read/write memory are connected to said central processing unit bus, and a system bus of said user terminal is able to be connected to said central processing unit bus;

a data copyright management system program, a copyright management program, crypt algorithm, and user information are stored in said read-only semiconductor memory;

a second private-key, a permit key, a second secret-key, and copyright information are stored in said electrically erasable programmable memory; and

a first public-key, a first private-key, a second public-key, and a first crypt-key are transmitted to said read/write memory during operation.

3. The data copyright management apparatus according to Claim 1 or 2, which is configured in the form of air IC.

4. The data copyright management apparatus according to Claim 1 or 2, which is configured in the form of an IC card.

5. The data copyright management apparatus according to Claim 1 or 2, which is configured in the form of a PC card.

6. The data copyright management apparatus according to Claim 1 or 2, which is configured in the form of an insertion board.

7. A data copyright management apparatus used in a user terminal for decrypting encrypted data to display or edit said data and for re-encrypting decrypted data to store, copy, or transfer said data;

wherein, a computer comprising a microprocessor, a local bus connected to said microprocessor, read-only semiconductor memory and read/write memory connected to said local bus is configured;

whereby, one of the microprocessor of said user terminal and the microprocessor of said data copyright management apparatus performs decryption and the other performs re-encryption.

8. A data copyright management apparatus used in a user terminal for decrypting encrypted data to display or edit said data and for re-encrypting decrypted data to store, copy, or transfer said data;

said data copyright management apparatus comprising a first microprocessor and a second microprocessor;

wherein, a first computer comprising a first local bus connected to said first microprocessor, and first read-only semiconductor memory arid first read/write memory connected to said first local bus; and,

a second computer comprising a second local bus connected to said second microprocessor, and second read-only semiconductor memory and second read/write memory connected said second local bus are configured;

whereby, said first microprocessor decrypts encrypted data, and

said second microprocessor re-encrypts decrypted data.

Fig. 1

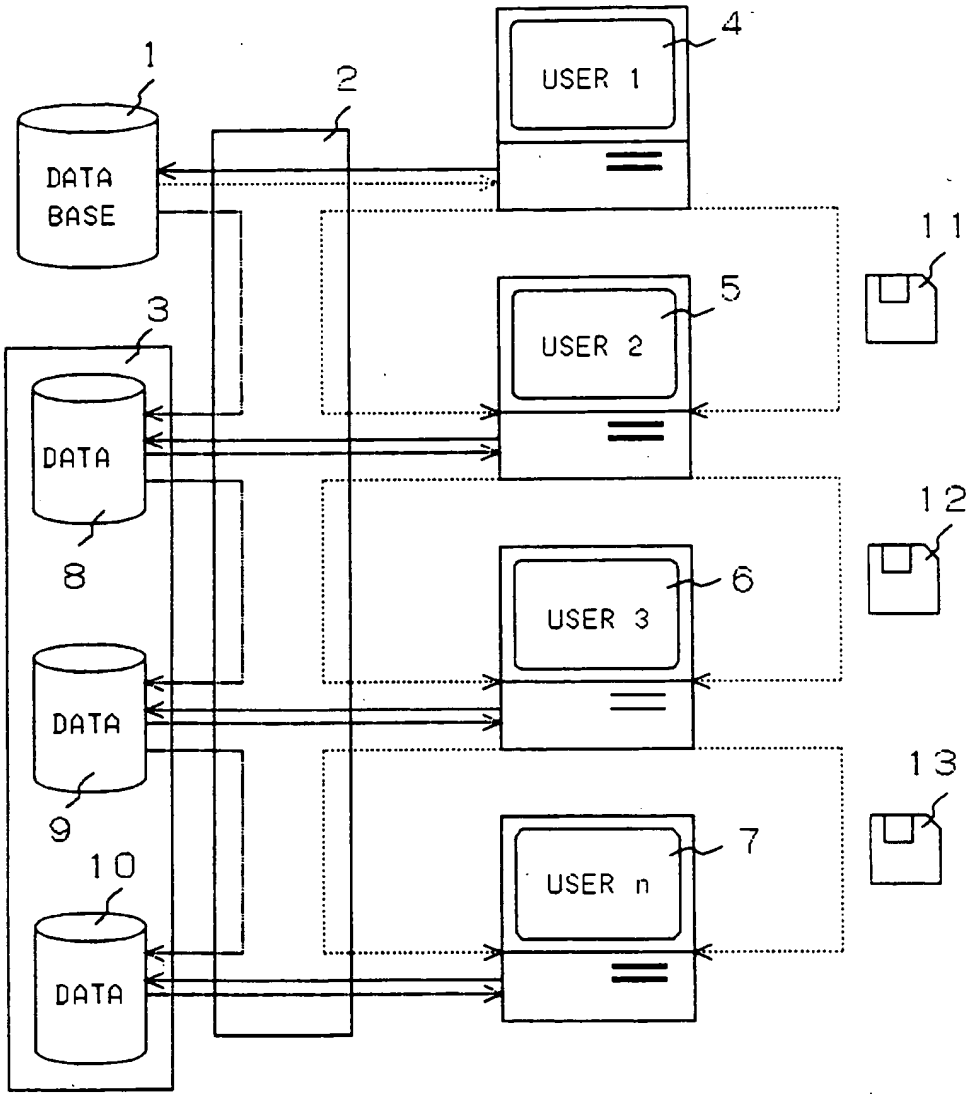


Fig. 2

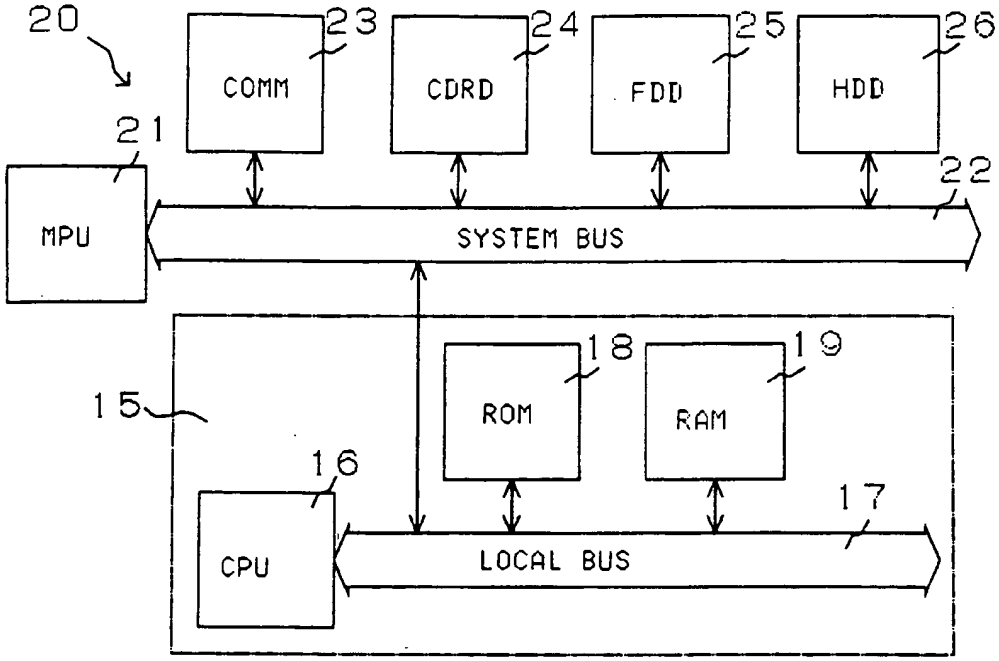


Fig. 3

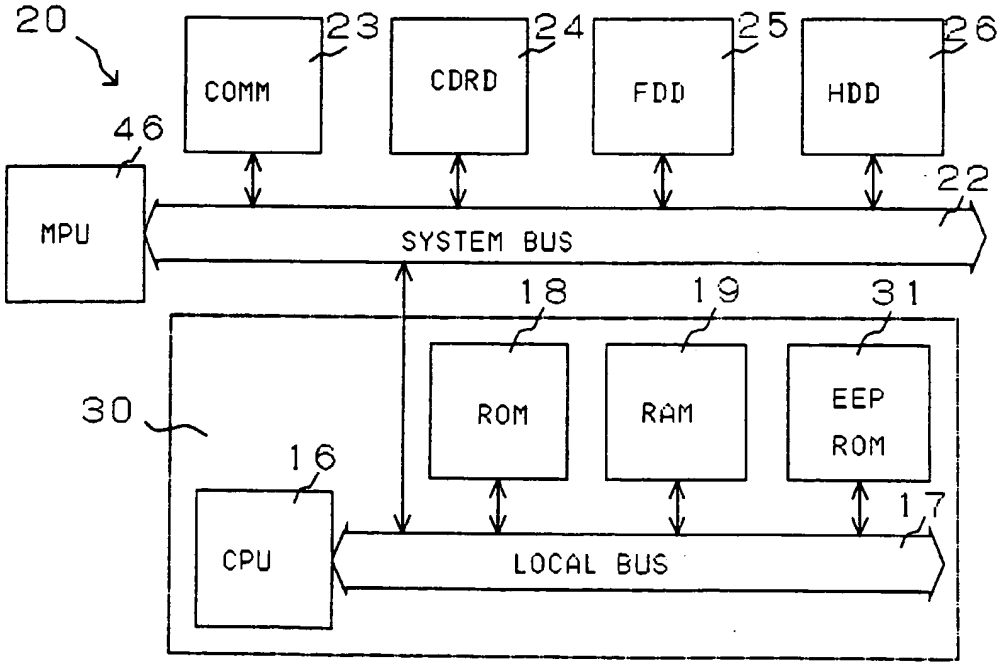


Fig. 4

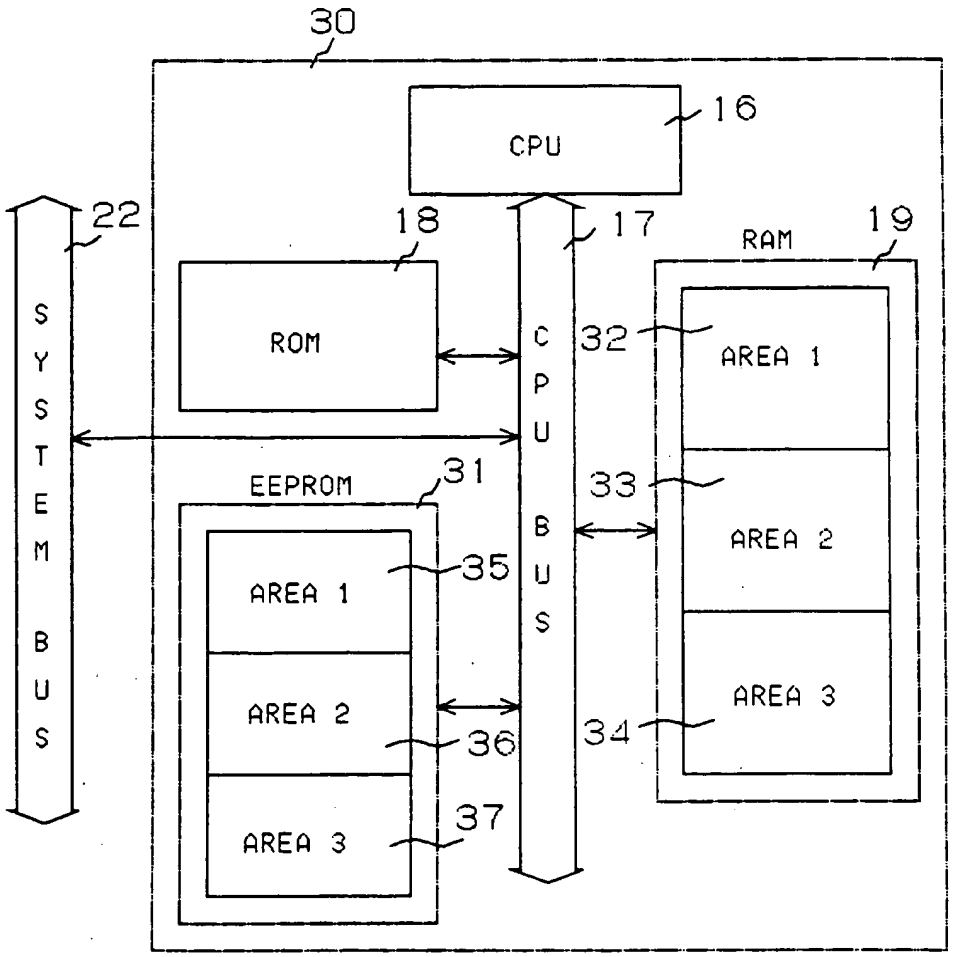


Fig. 5

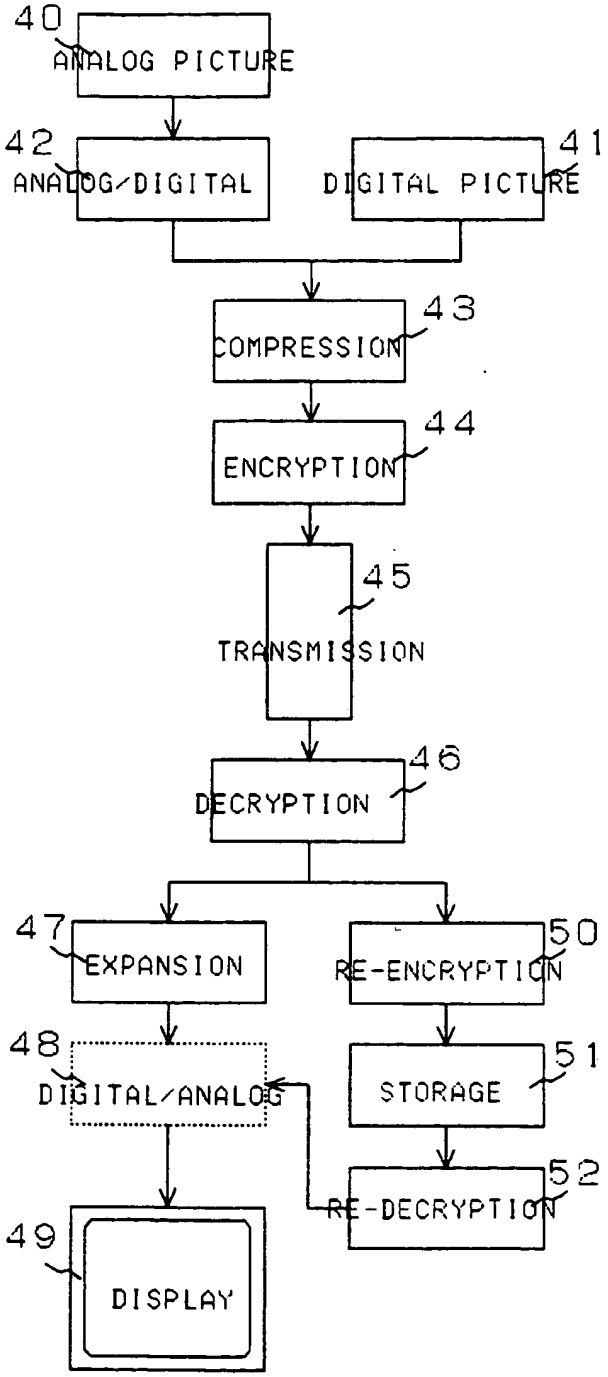


Fig. 6

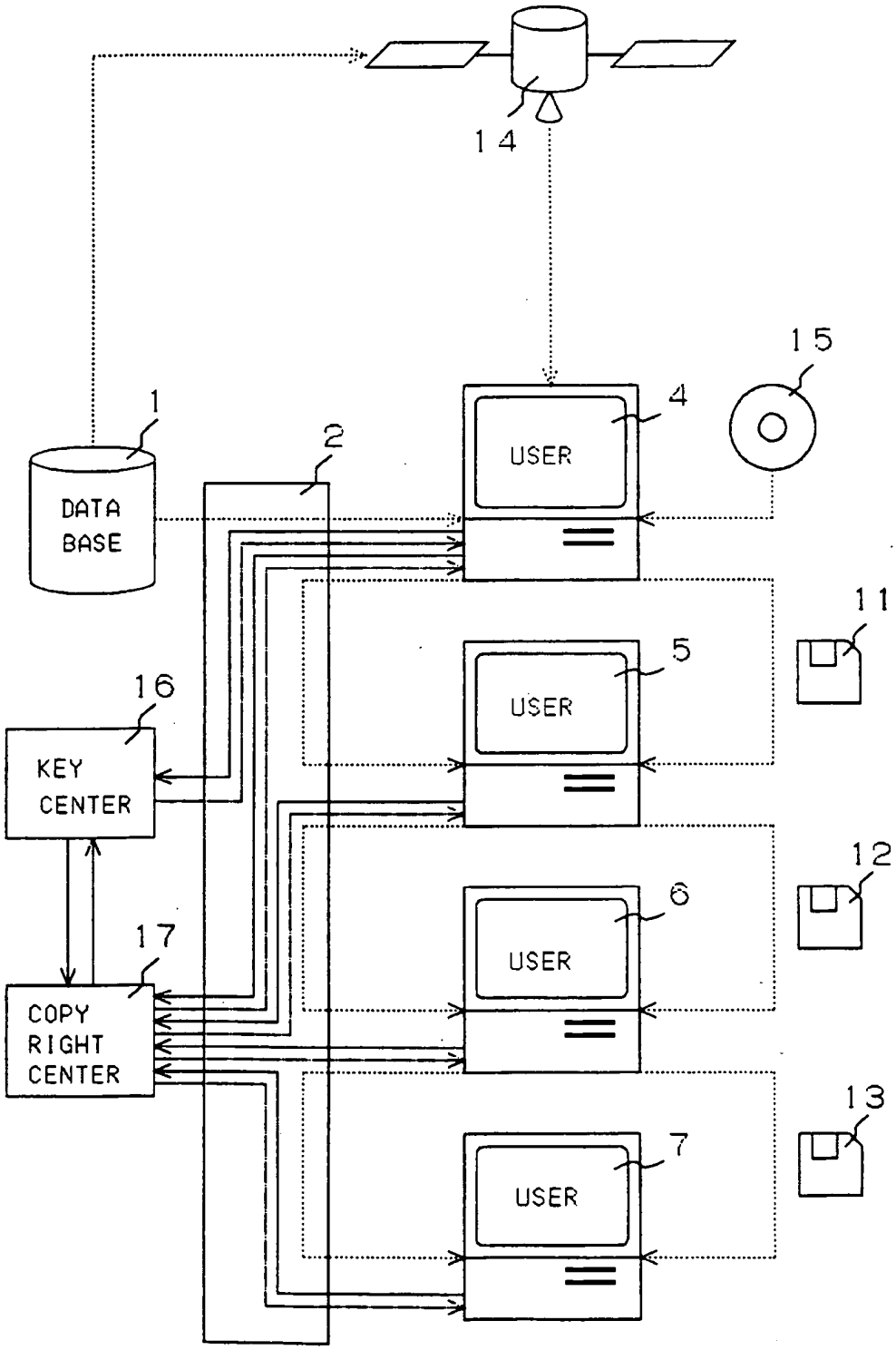


Fig. 7

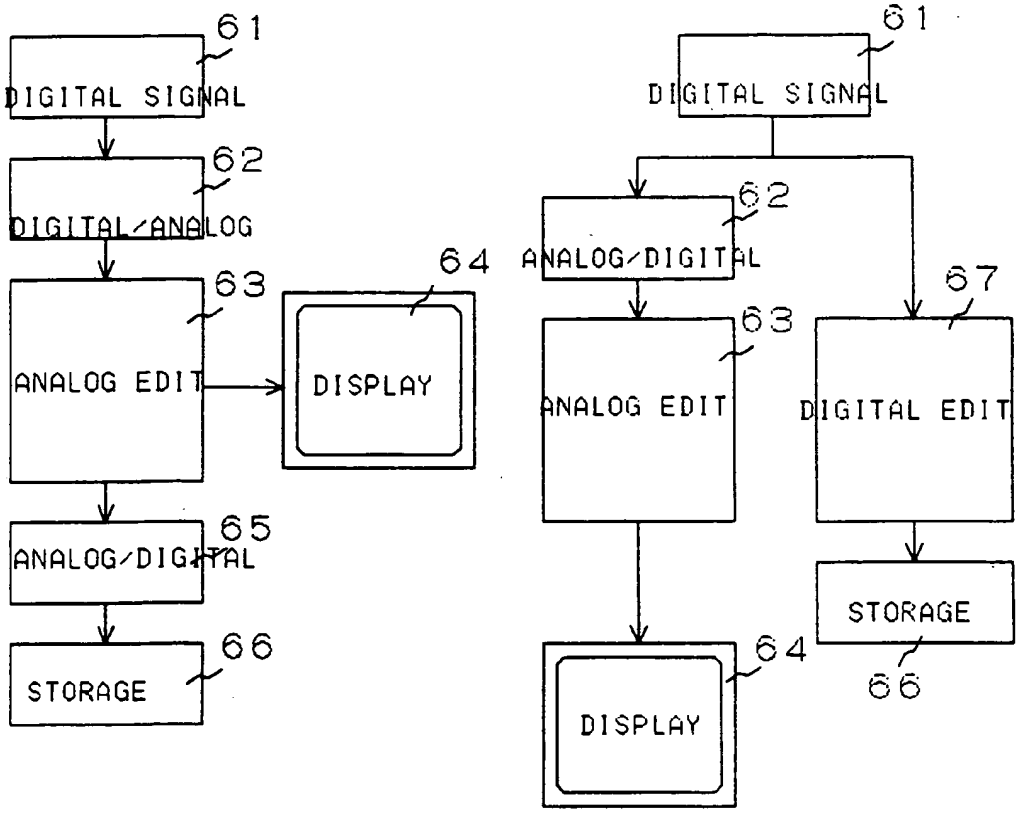


Fig. 7(a) Fig. 7(b)

Fig. 8

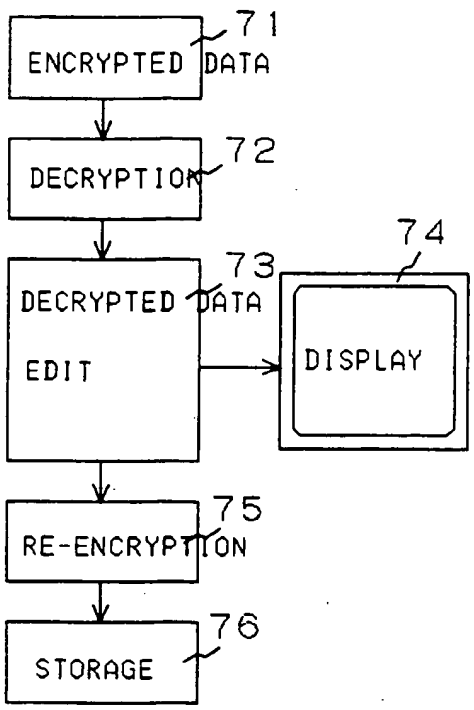


Fig. 8(a)

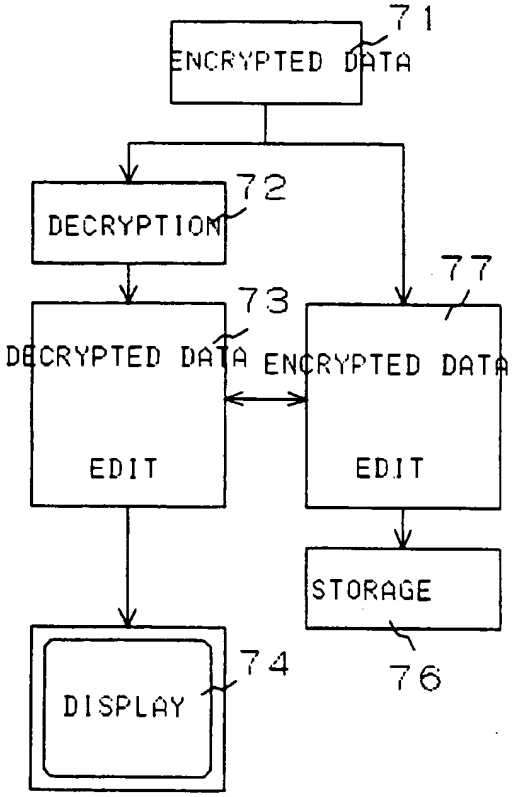


Fig. 8(b)

Fig. 9

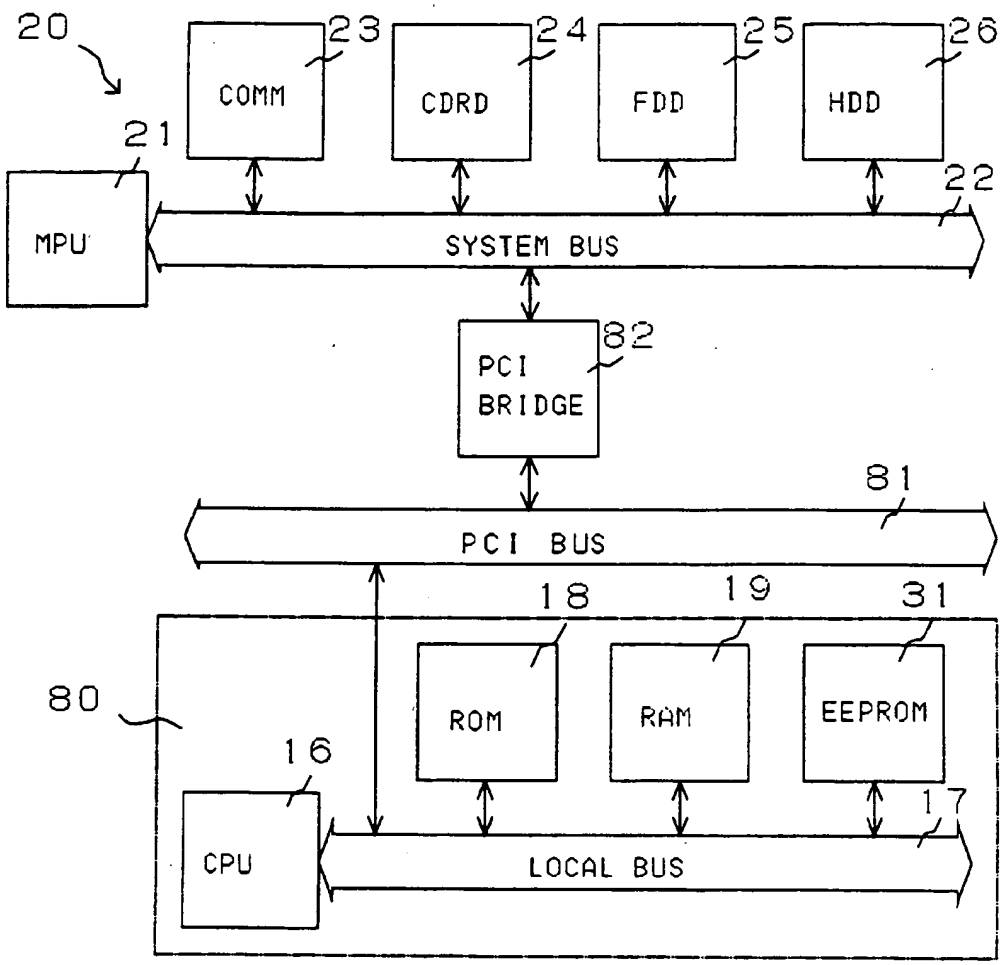


Fig. 10

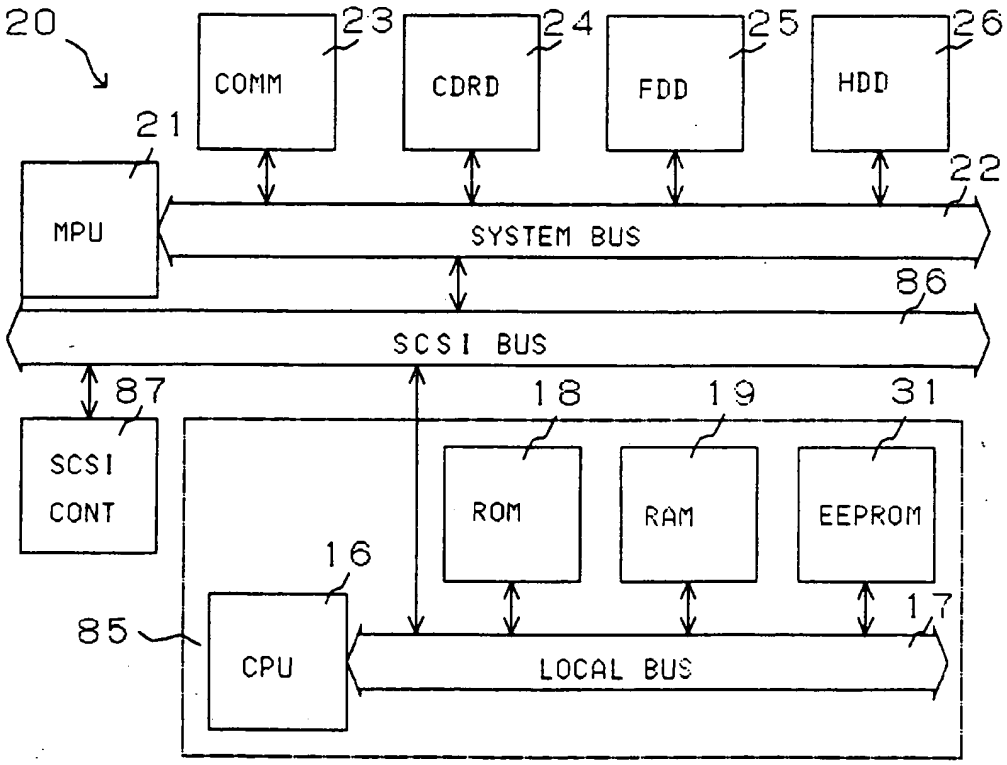


Fig. 11

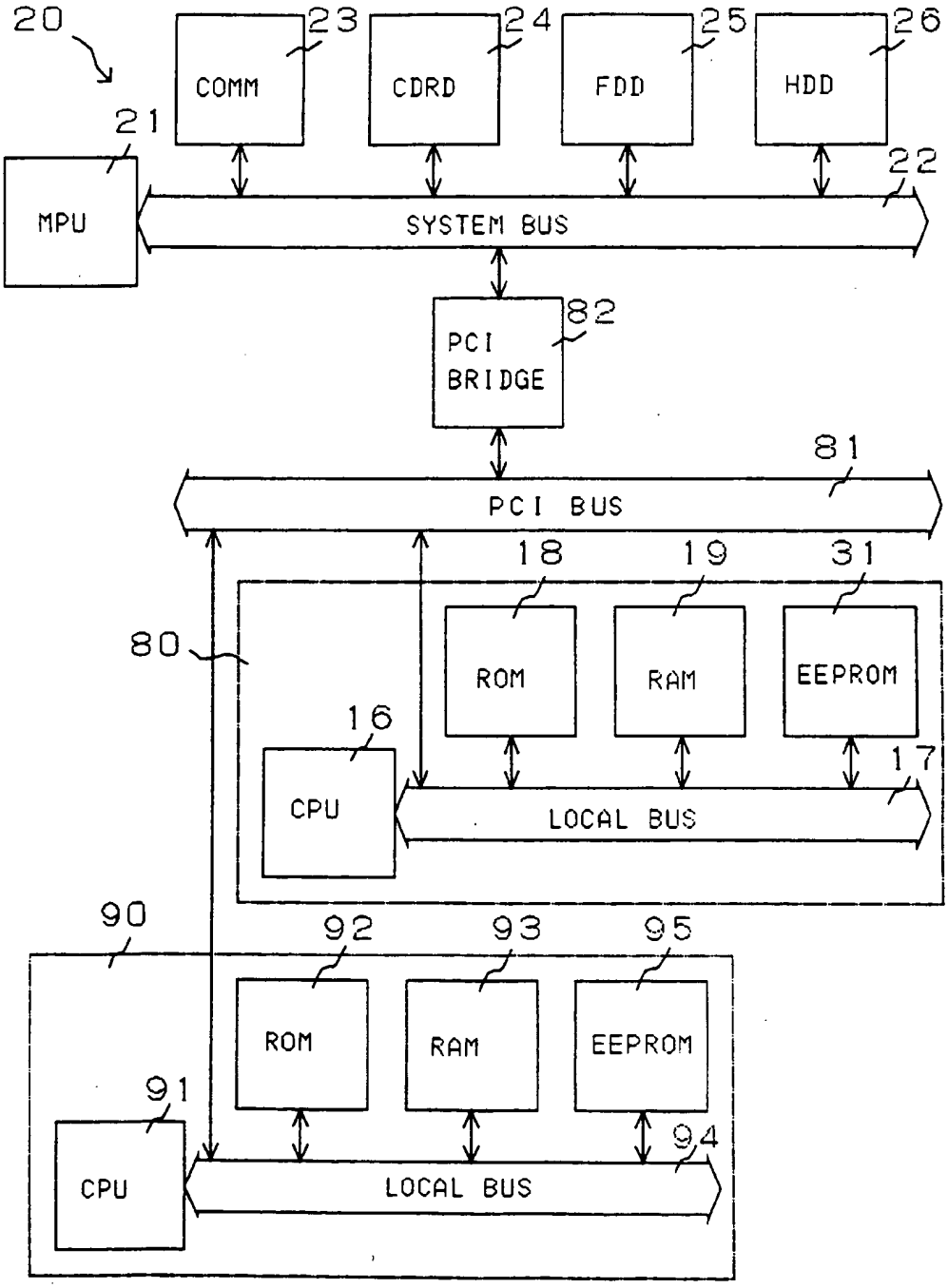


Fig. 12

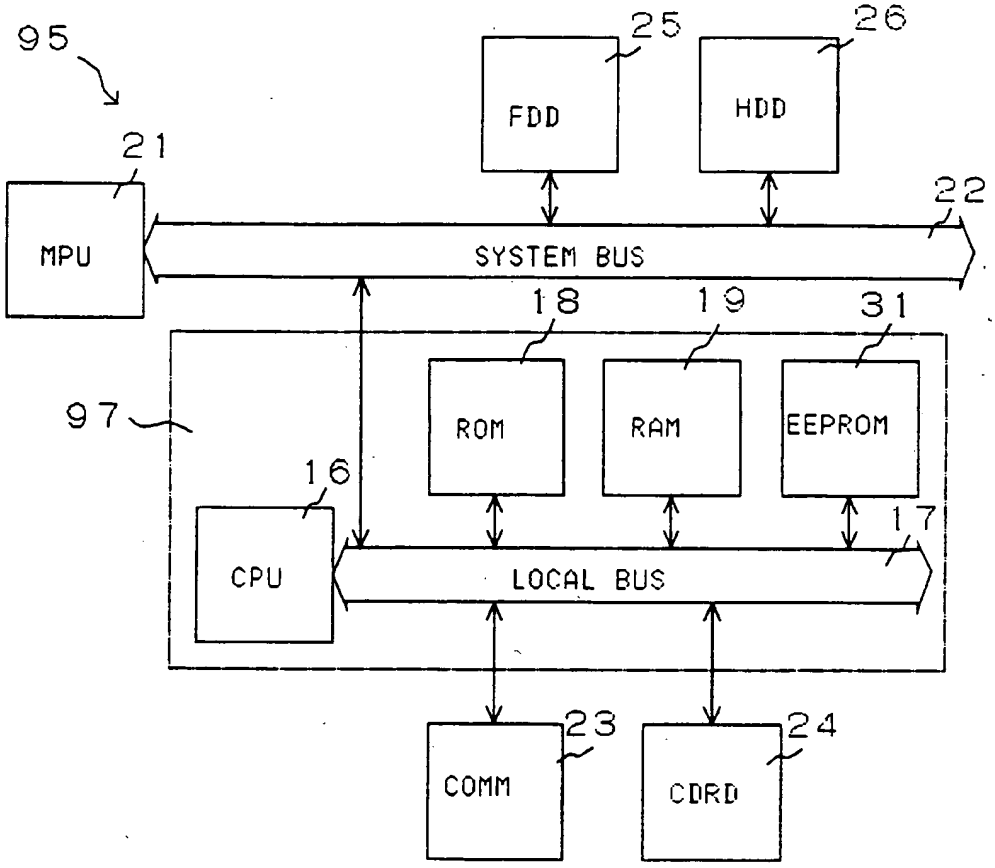


Fig. 13

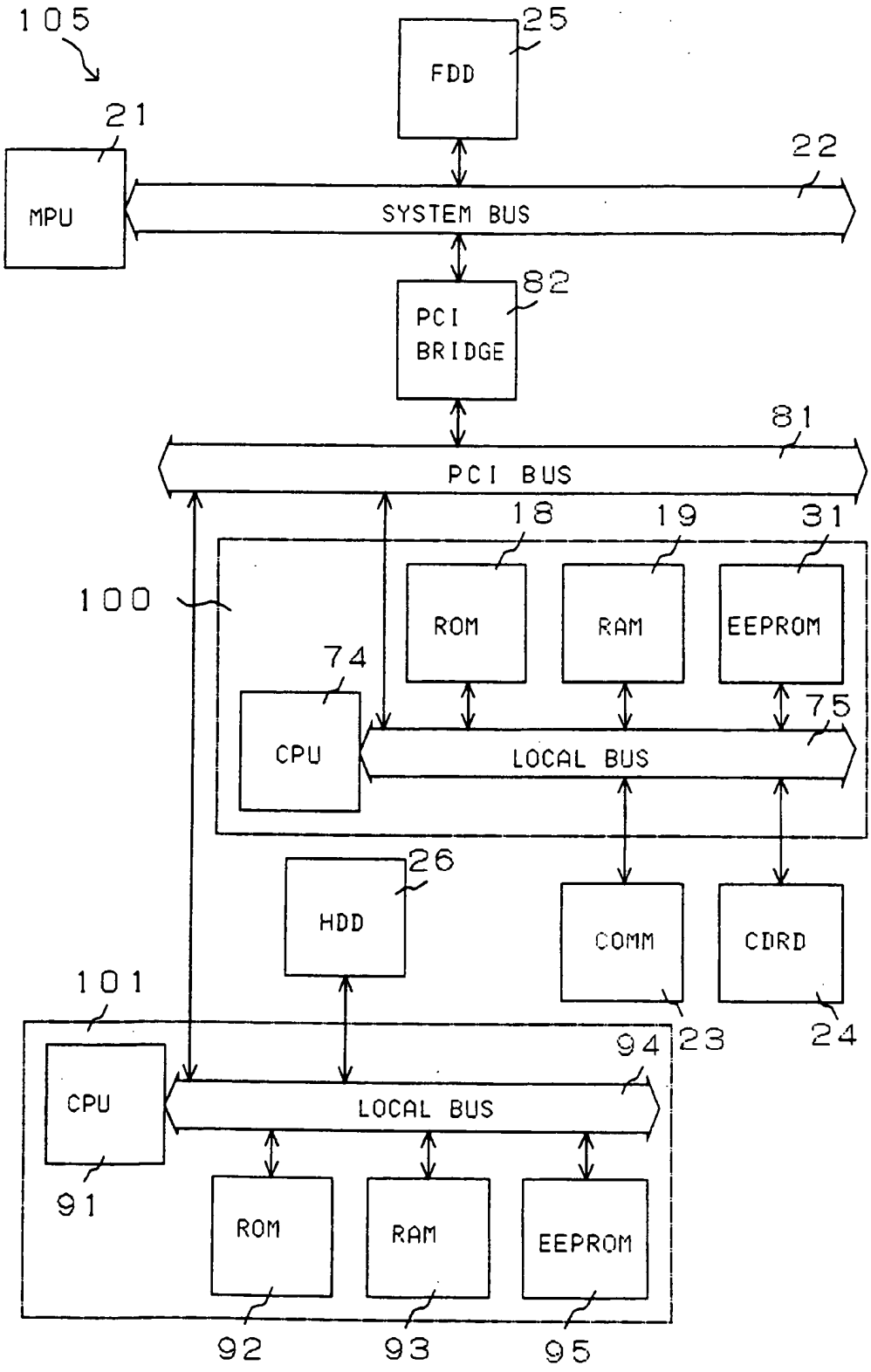


Fig. 14

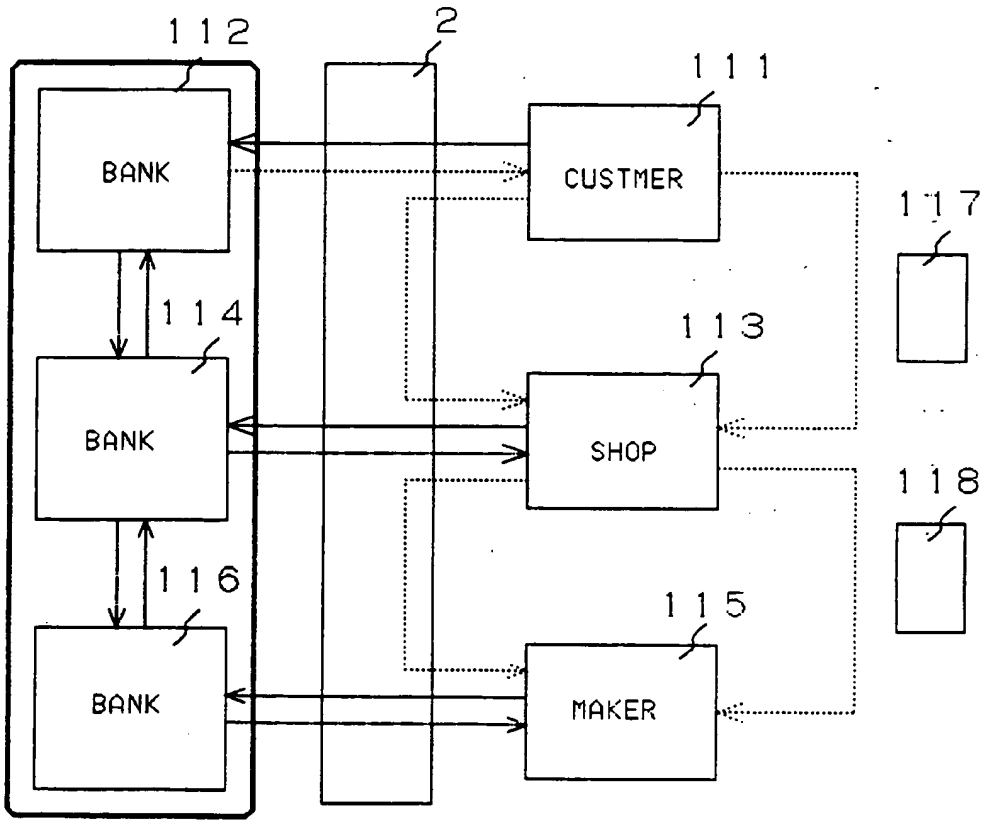
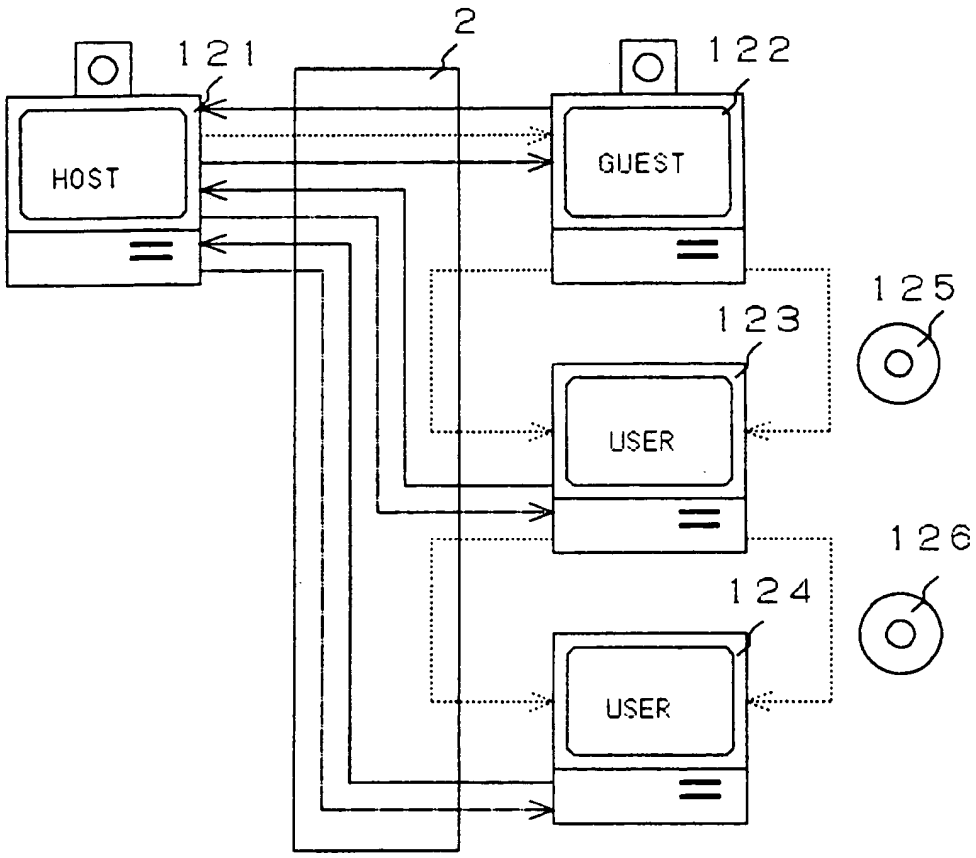


Fig. 15



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

(11) EP 0 715 241 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
03.02.1999 Bulletin 1999/05

(43) Date of publication A2:
05.06.1996 Bulletin 1996/23

(21) Application number: **95116615.6**

(22) Date of filing: **21.10.1995**

(51) Int. Cl.[6]: **G06F 1/00**, H04N 7/167

(84) Designated Contracting States:
DE FR GB

(30) Priority: **27.10.1994 JP 264200/94**
02.12.1994 JP 299835/94

(71) Applicant:
MITSUBISHI CORPORATION
Chiyoda-ku Tokyo 100 (JP)

(72) Inventors:
- **Saito, Makoto**
  Tama-shi, Tokyo (JP)
- **Momiki, Shunichi**
  Higashimur-ayama-shi, Tokyo (JP)

(74) Representative:
Neidl-Stippler & Partner
Rauchstrasse 2
81679 München (DE)

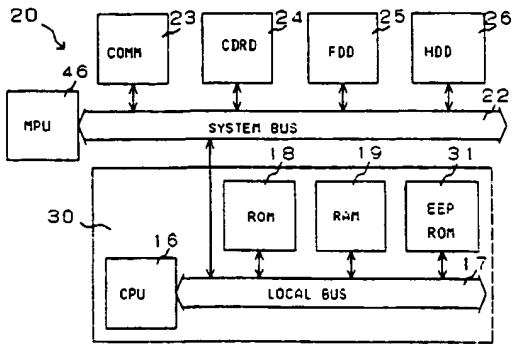
(54) Apparatus for data copyright management system

(57) A data copyright management apparatus is used with a user terminal and comprises a CPU, a CPU bus, ROM, EEPROM, and RAM.

The CPU, ROM, EPROM, and RAM are connected to the CPU bus, and a system bus of a device which utilizes the data can be connected to the CPU bus. A data copyright management system program, a crypt algorithm, and user information are stored in the ROM, and a second private-key, a permit key, a second secret-key, and copyright information are stored in the EEPROM. A first public-key, a first private-key, a second public-key, and a first secret-key are transmitted to the RAM during the operation. The data copyright management apparatus may be configured in the form of a monolithic or hybrid IC, a thin IC card, PC card, or an expansion board. If the copyright management program is provided from the outside, then it is stored in the EEPROM, otherwise it is stored in ROM.

In addition to a microprocessor in the user terminal which decrypts encrypted data for displaying and processing purposes and re-encrypts the decrypted data for storing, copying, or transferring purposes, at least one other microprocessor, desirably two other microprocessors, are added for decrypting and re-encrypting data. The microprocessors to be added may be connected to the system bus of the microprocessor of the user terminal. However, to allow concurrent microprocessor operation it is desirable that the multiprocessor configuration is implemented by using a SCSI bus, PCI bus, or SCI bus. The data copyright management apparatus may be implemented in the form of a monolithic IC, a hybrid IC, or a built-in subboard, and the apparatus in these forms is incorporated in a computer, television set, set-top box, digital video tape recorder, digital video disk recorder, digital audio tape apparatus, or personal digital assistants, and the like.

Fig. 3



EP 0 715 241 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 95 11 6615

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	EP 0 430 734 A (SCHLUMBERGER IND SA) 5 June 1991 * column 3, line 17 - line 37 * * column 4, line 4 - column 5, line 24; figures 1,2 *	1,2,4,7, 8	G06F1/00 H04N7/167
A	WO 90 02382 A (INDATA CORP) 8 March 1990 * page 35, paragraph 2 - page 38, paragraph 4; figures 10,12 *	1,2,7,8	
A	EP 0 121 853 A (BURROUGHS CORP) 17 October 1984 * page 3, line 30 - page 4, line 12; figure 1 *	1,2,7,8	
A	US 4 352 952 A (BOONE CHARLES A ET AL) 5 October 1982 * abstract; figures 1,2 *	7,8	

TECHNICAL FIELDS SEARCHED (Int.Cl.6)
G06F
H04N

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	1 December 1998	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)

CORRECTED VERSION*

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :		(11) International Publication Number:	WO 98/11690
H04L 9/00	A1	(43) International Publication Date:	19 March 1998 (19.03.98)

(21) International Application Number: PCT/US97/16223

(22) International Filing Date: 12 September 1997 (12.09.97)

(30) Priority Data:

60/025,991	12 September 1996 (12.09.96)	US
08/887,723	3 July 1997 (03.07.97)	US

(71)(72) Applicant and Inventor: GLOVER, John, J. [US/US]; 26 Amaranth Avenue, Medford, MA 02155 (US).

(74) Agent: GORDON, Peter, J.; Wolf, Greenfield & Sacks, P.C., 600 Atlantic Avenue, Boston, MA 02210 (US).

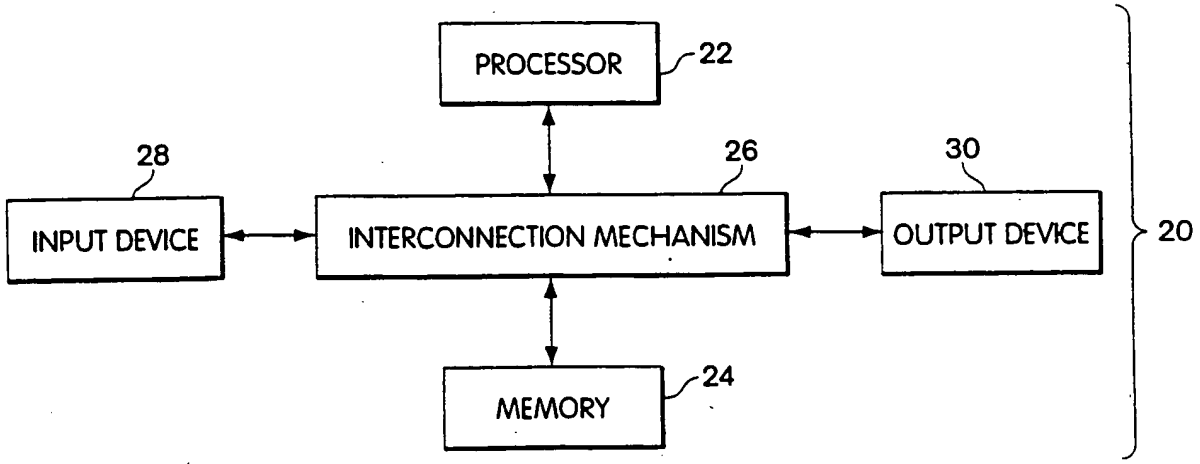
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent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

Published

With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: SELF-DECRYPTING DIGITAL INFORMATION SYSTEM AND METHOD



(57) Abstract

The claimed data protection device (20) includes a processor (22) connected to a memory system (24) through an interconnection mechanism (26). An input device (28) is also connected to the processor (22) and memory system (24) through the interconnection mechanism (26). The interconnection mechanism (26) is typically a combination of one or more buses and one or more switches. The output device (30) may be a display, and the input device (28) may be a keyboard and/or mouse or other cursor control device.

*(Referred to in PCT Gazette No. 32/1998, Section II)

BNSDOCID: <WO___9811690A1_IA>

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	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

SELF-DECRYPTING DIGITAL INFORMATION SYSTEM AND METHOD

Field of the Invention

The present invention is related to mechanisms for protecting digital information from being copied. In particular, the present invention is related to mechanisms which permit authorized execution of computer program code or access to other digital information which is encrypted or otherwise encoded.

Background of the Invention

A serious problem which faces the electronic publishing and software industries is the ease with which digital information can be copied without authorization from the publisher. Digital information also may be used or modified without authorization. For example, computer software may be reverse engineered or attacked by computer viruses.

There are many mechanisms available which may be used to limit or prevent access to digital information. Such mechanisms often either restrict the ability of the user to make back-up copies or involve the use of special purpose hardware to limit access to the digital information. For example, some mechanisms restrict the use of digital information to a particular machine. See, for example, U.S. Patent 4,817,140. Other mechanisms require the digital information to be stored on a particular recording medium in order to be used. See, for example, U.S. Patent 5,412,718. Yet other mechanisms allow only a certain number of uses of the digital information. See for example, U.S. Patent 4,888,798. Many of these access control mechanisms cause distribution to be more costly.

Several other patents describe a variety of systems for encryption, compression, licensing and royalty control and software distribution such as: U.S. Pat. No. 4,405,829, U.S. Pat. No. 4,864,616, U.S. Pat. No. 4,888,800, U.S. Pat. No. 4,999,806, U.S. Pat. No. 5,021,997, U.S. Patent No. 5,027,396, U.S. Pat. No. 5,033,084, U.S. Pat. No. 5,081,675, U.S. Pat. No. 5,155,847, U.S. Pat. No. 5,166,886, U.S. Pat. No. 5,191,611, U.S. Pat. No. 5,220,606, U.S. Pat. No. 5,222,133, U.S. Pat. No. 5,272,755, U.S. Pat. No. 5,287,407, U.S. Pat. No. 5,313,521, U.S. Pat. No. 5,325,433, U.S. Pat. No. 5,327,563, U.S. Pat. No. 5,337,357, U.S. Pat. No. 5,351,293, U.S. Pat. No. 5,341,429, U.S. Pat. No. 5,351,297, U.S. Pat. No. 5,361,359, U.S. Pat. No. 5,379,433, U.S. Pat. No. 5,392,351, U.S. Pat. No. 5,394,469, U.S. Pat. No. 5,414,850, U.S. Pat. No. 5,473,687, U.S. Pat. No. 5,490,216, U.S. Pat. No. 5,497,423, U.S. Pat. No. 5,509,074, U.S.

Pat. No. 5,,511,123, U.S. Pat. No. 5,524,072, U.S. Pat. No. 5,532,920, U.S. Pat. No. 5,555,304, U.S. Pat. No. 5,557,346, U.S. Pat. No. 5,557,765, U.S. Pat. No. 5,592,549, U.S. Pat. No. 5,615,264, U.S. Pat. No. 5,625,692, and U.S. Pat. No. 5,638,445.

Computer programs or other digital information also may be encrypted in order to prevent an individual from making a useful copy of the information or from reverse engineering a program. Even with such encryption, however, a computer program must be decrypted in order for a computer to load and execute the program. Similarly, other digital information must be decrypted before it can be accessed and used. Generally, digital information is decrypted to disk, and not to main memory of the computer which is more protected by the operating system, because decryption to main memory results in a significant loss of memory resources. If the purpose for using encryption is to prevent users from copying the digital information, then decryption of the information to accessible memory for use defeats this purpose.

One way to protect digital information using encryption has been made available by International Business Machines (IBM) and is called a "CRYPTOLOPE" information container. This technology is believed to be related to U.S. Patent Nos. 5,563,946 and 5,598,470 (to Cooper et al.), and published European patent applications 0679977, 0679978, 0679979 and 0681233. The CRYPTOLOPE system requires a user to have a "helper application" and a key. The CRYPTOLOPE information container is generated by IBM. The content provider submits data to IBM, which in turn encrypts and packages the data in a CRYPTOLOPE information container. The helper application is a form of memory resident program, called a terminate and stay resident (TSR) program, which is a form of input/output (I/O) device driver installed in the operating system and which monitors requests from the operating system for files on specified drives and directories. Because the TSR program must know the directory, and/or file name to be accessed, that information also is available to other programs. Other programs could use that information to manipulate the operation of the TSR program in order to have access to decrypted contents of the information container. The encrypted information container includes an executable stub which is executed whenever the application is run without the installed TSR program or from a drive not monitored by the TSR program to prevent unpredictable activity from executing encrypted code. This stub may be used to install decryption and cause the application be executed a second time, or to communicate with the TSR program to instruct the TSR program to monitor the drive. It may be preferable from the point of view of the content provider however to maintain an encryption process and keys independently of any third party.

Multimedia content, such as a movie or hypertext presentation also may be stored on a digital versatile disk (DVD), sometimes called a digital video disk, compact disk read-only memory (CD-ROM), rewriteable compact disks (CD-RW) or other medium in an encrypted digital format for use with special-purpose devices. For example, concern about illegal copying of content from digital video disks or other digital media has resulted in a limited amount of content being available for such devices. This problem has caused representatives of both multimedia providers and digital video disk manufacturers to negotiate an agreement on an encryption format for information stored on DVDs. This copy protection scheme is licensed through an organization called the CSS Interim Licensing organization. However, in this arrangement, the content provider is limited to using the agreed upon encryption format and a device manufacturer is limited to using a predetermined decryption system.

Encryption has also been used to protect and hide computer viruses. Such viruses are typically polymorphic, i.e., they change every time they infect a new program, and are encrypted. The virus includes a decryption program that executes to decrypt the virus every time the infected program is run. Such viruses are described, for example, in "Computer Virus-Antivirus Coevolution" by Carey Nachenberg, Communications of the ACM, Vol. 40, No. 1, (Jan. 1997), p. 46 et seq. Such viruses include decryption keys within them since, clearly, their execution is not carried out by the user and a user would not be asked for authorization keys to permit execution of the viruses. Additionally, such viruses are typically only executed once at the start of execution of an infected program and permanently return control to the infected program after execution.

Summary of the Invention

Some of these problems with digital information protection systems may be overcome by providing a mechanism which allows a content provider to encrypt digital information without requiring either a hardware or platform manufacturer or a content consumer to provide support for the specific form of corresponding decryption. This mechanism can be provided in a manner which allows the digital information to be copied easily for back-up purposes and to be transferred easily for distribution, but which should not permit copying of the digital information in decrypted form. In particular, the encrypted digital information is stored as an executable computer program which includes a decryption program that decrypts the encrypted information

to provide the desired digital information, upon successful completion of an authorization procedure by the user.

In one embodiment, the decryption program is executed as a process within a given operating system and decrypts the digital information within the memory area assigned to that process. This memory area is protected by the operating system from copying or access by other processes. Even if access to the memory area could be obtained, for example through the operating system, when the digital information is a very large application program or a large data file, a copy of the entire decrypted digital information is not likely to exist in the memory area in complete form.

By encrypting information in this manner, a platform provider merely provides a computer system with an operating system that has adequate security to define a protected memory area for a process and adequate functionality to execute a decryption program. The content provider in turn may use any desired encryption program. In addition, by having a process decrypt information within a protected memory area provided by the operating system, the decrypted information does not pass through any device driver, memory resident program or other known logical entity in the computer system whose behavior may be controlled to provide unauthorized access to the data. The ability to reverse engineer or attack a computer program with a computer virus also may be reduced.

In another embodiment, the decryption program is part of a dynamically loaded device driver that responds to requests for data from the file containing the encrypted data. When the digital information product is first executed, this device driver is extracted from the file and is loaded into the operating system. The executed digital information product then informs the loaded device driver of the location of the hidden information in the file, any keys or other passwords, and the name of a phantom directory and file to be called that only the digital information product and the device driver know about. The name of this directory may be generated randomly. Each segment of hidden information in the digital information product may be assigned its own unique file name in the phantom directory. The digital information product then makes a call to the operating system to execute one of the files in the phantom directory. The loaded driver traps these calls to the operating system, accesses the original file, decrypts the desired information and outputs the desired information to the operating system.

In combination with other mechanisms that track distribution, enforce royalty payments and control access to decryption keys, the present invention provides an improved method for

identifying and detecting sources of unauthorized copies. Suitable authorization procedures also enable the digital information to be distributed for a limited number of uses and/or users, thus enabling per-use fees to be charged for the digital information.

Accordingly, one aspect of the invention is a digital information product including a computer-readable medium with digital information stored thereon. The digital information includes computer program logic having a first portion of executable computer program logic and a second portion of digital information. The first portion of executable program logic, when executed, defines a mechanism for responding to requests for digital information from an operating system of a computer. This mechanism, when used to access the second portion of the encrypted digital information, decrypts the encrypted digital information, and provides the encrypted digital information to the operating system.

In the foregoing aspect of the invention, the digital information may be executable computer program logic. Hence, one aspect of the invention is a computer program product, including a computer readable medium with computer program logic stored thereon. The computer program logic includes a first portion of executable computer program logic and a second portion of encrypted computer program logic. The first portion of executable computer program logic, when executed, defines a mechanism for responding to requests for computer program logic from an operating system of a computer. This mechanism accesses the second portion of encrypted computer program logic, decrypts the encrypted computer program logic, and provides the decrypted computer program logic to the operating system.

Another aspect of the present invention is a computer program product, a computer system and a process which produce a computer program or digital information product in accordance with other aspects of the invention, using executable program code for the first and second portions of the desired computer program product.

Another aspect of the present invention is a computer program product including a self-decrypting encrypted executable computer program. The product includes a computer readable medium having computer program logic stored thereon. The computer program logic defines first, second and third modules, wherein the third module defines the encrypted executable computer program. The first module, when executed by a computer, defines a mechanism for loading the second module into memory of the computer. The second module, when executed by a computer, defines a mechanism for communicating with an operating system of the computer to receive requests for program code from the encrypted executable computer program from the

third module, and for processing the requests to access and decrypt the encrypted executable computer program and for providing the decrypted executable code from the third module to the operating system.

Another aspect of the invention is a process for executing encrypted executable computer programs on a computer system having a processor, memory and operating system. The process involves receiving computer program logic having a first module defining a start up routine, a second module, and a third module containing the encrypted executable computer program. The first module of the received computer program logic is executed using the processor. When the first module is executed, the second module is caused to be loaded into the memory of the computer system. Requests are generated from the operating system for data from the encrypted executable computer program and are received by the second module. The second module accesses and decrypts the encrypted executable computer program in response to these requests and returns the decrypted executable computer program to the operating system.

These and other aspects, advantages and features of the present invention and its embodiments will be more apparent given the following detailed description.

Brief Description of the Drawing

In the drawing,

Fig. 1 is a block diagram of a typical computer system with which the present invention may be implemented;

Fig. 2 is a block diagram of a memory system in the computer system of Fig. 1;

Fig. 3 is a diagram of a computer program or digital information product which may be recorded on a computer readable and writable medium, such as a magnetic disc;

Fig. 4 is a flowchart describing how the computer program or digital information product of Fig. 3 is used;

Fig. 5 is a flowchart describing operation of an example unwrap procedure as shown in Fig. 3 in one embodiment of the invention;

Fig. 6 is a flowchart describing operation of an example device driver as shown in Fig. 3 in one embodiment of the invention;

Fig. 7 is a block diagram of a computer system in the process of executing a computer program product in accordance with one embodiment of the invention;

Fig. 8 is a flowchart describing operation of an example unwrap procedure in another embodiment of the invention; and

Fig. 9 is a flowchart describing how a computer program product such as shown in Fig. 3 is constructed.

Detailed Description

The present invention will be more completely understood through the following detailed description which should be read in conjunction with the attached drawing in which similar reference numbers indicate similar structures.

Embodiments of the present invention may be implemented using a general purpose digital computer or may be implemented for use with a digital computer or digital processing circuit. A typical computer system 20 is shown in Fig. 1, and includes a processor 22 connected to a memory system 24 via an interconnection mechanism 26. An input device 28 also is connected to the processor and memory system via the interconnection mechanism, as is an output device 30. The interconnection mechanism 26 is typically a combination of one or more buses and one or more switches. The output device 30 may be a display and the input device may be a keyboard and/or a mouse or other cursor control device.

It should be understood that one or more output devices 30 may be connected to the computer system. Example output devices include a cathode ray tube (CRT) display, liquid crystal display (LCD), television signal encoder for connection to a television or video tape recorder, printers, communication devices, such as a modem, and audio output. It also should be understood that one or more input devices 28 may be connected to the computer system. Example input devices include a keyboard, keypad, trackball, mouse, pen and tablet, communication device, audio or video input and scanner. It should be understood that the invention is not limited to the particular input or output devices used in combination with the computer system or to those described herein.

The computer system 20 may be a general purpose computer system, which is programmable using a high level computer programming language, such as "C++," "Pascal," "VisualBasic." The computer system also may be implemented using specially programmed, special purpose hardware. In a general purpose computer system, the processor is typically a commercially available processor, such as the Pentium processor from Intel Corporation. Many other processors are also available. Such a processor executes a program called an operating

system, such as Windows 95 or Windows NT 4.0, both available from Microsoft Corporation, which controls the execution of other computer programs and provides scheduling, debugging, input output control, accounting compilation, storage assignment, data management and memory management, and communication control and related services. Other examples of operating systems include: MacOS System 7 from Apple Computer, OS/2 from IBM, VMS from Digital Equipment Corporation, MS-DOS from Microsoft Corporation, UNIX from AT&T, and IRIX from Silicon Graphics, Inc.

The computer system 20 also may be a special purpose computer system such as a digital versatile disk or digital video disk (DVD) player. In a DVD player, there is typically a decoder controlled by some general processor which decodes an incoming stream of data from a DVD. In some instances, the DVD player includes a highly integrated DVD decoder engine. Such devices generally have a simple operating system which may be modified to include the capabilities described and used herein in connection with the typical operating systems in a general purpose computer. In particular, some operating systems are designed to be small enough for installation in an embedded system such as a DVD player, including the WindowsCE operating system from Microsoft Corporation and the JavaOS operating system from SunSoft Corporation. The operating system allows a content provider to provide its own programs that define some of the content, which is particularly useful for interactive multimedia. This capability also can be used to provide encryption and decryption, in accordance with the invention.

The processor and operating system define a computer platform for which application programs in a programming language such as an assembly language or a high level programming language are written. It should be understood that the invention is not limited to a particular computer platform, operating system, processor, or programming language. Additionally, the computer system 20 may be a multi-processor computer system or may include multiple computers connected over a computer network.

An example memory system 24 will now be described in more detail in connection with Fig. 2. A memory system typically includes a computer readable and writable non-volatile recording medium 40, of which a magnetic disk, a flash memory, rewriteable compact disk (CD-RW) and tape are examples. The recording medium 40 also may be a read only medium such as a compact disc-read only memory (CD-ROM) or DVD. A magnetic disk may be removable, such as a "floppy disk" or "optical disk," and/or permanent, such as a "hard drive." The disk,

which is shown in Fig. 2, has a number of tracks, as indicated at 42, in which signals are stored, in binary form, i.e., a form interpreted as a sequence of 1's and 0's, as shown at 44. Such signals may define an application program to be executed by the microprocessor, or information stored on the disk to be processed by the application program. Typically, in the operation of a general purpose computer, the processor 22 causes data to be read from the non-volatile recording medium 40 into an integrated circuit memory element 46, which is typically a volatile random access memory, such as a dynamic random access memory (DRAM) or static random access memory (SRAM). The integrated circuit memory element 46 allows for faster access to the information by the processor than disk 40, and is typically called the system or host memory. The processor generally causes the data to be manipulated within the integrated circuit memory 46 and may copy the data to the disk 40, if modified, when processing is completed. A variety of mechanisms are known for managing data movement between the disk 40 and the integrated circuit memory 46, and the invention is not limited thereto. It should also be understood that the invention is not limited to a particular memory system.

The file system of a computer generally is the mechanism by which an operating system manages manipulation of data between primary and secondary storage, using files. A file is a named logical construct which is defined and implemented by the operating system to map the name and a sequence of logical records of data to physical storage media. An operating system may specifically support various record types or may leave them undefined to be interpreted or controlled by application programs. A file is referred to by its name by application programs and is accessed through the operating system using commands defined by the operating system. An operating system provides basic file operations provided by for creating a file, opening a file, writing a file, reading a file and closing a file.

In order to create a file, the operating system first identifies space in the storage media which is controlled by the file system. An entry for the new file is then made in a directory which includes entries indicating the names of the available files and their locations in the file system. Creation of a file may include allocating certain available space to the file. Opening a file returns a handle to the application program which it uses to access the file. Closing a file invalidates the handle.

In order to write data to a file, an application program issues a command to the operating system which specifies both an indicator of the file, such as a file name, handle or other descriptor, and the information to be written to the file. Given the indicator of the file, the

operating system searches the directory to find the location of the file. The directory entry stores a pointer, called the write pointer, to the current end of the file. Using this pointer, the physical location of the next available block of storage is computed and the information is written to that block. The write pointer is updated in the directory to indicate the new end of the file.

In order to read data from a file, an application program issues a command to the operating system specifying the indicator of the file and the memory locations assigned to the application where the next block of data should be placed. The operating system searches its directory for the associated entry given the indicator of the file. The directory may provide a pointer to a next block of data to be read, or the application may program or specify some offset from the beginning of the file to be used.

A primary advantage of using a file system is that, for an application program, the file is a logical construct which can be created, opened, written to, read from and closed without any concern for the physical storage used by the operating system.

The operating system also allows for the definition of another logical construct called a process. A process is a program in execution. Each process, depending on the operating system, generally has a process identifier and is represented in an operating system by a data structure which includes information associated with the process, such as the state of the process, a program counter indicating the address of the next instruction to be executed for the process, other registers used by process and memory management information including base and bounds registers. Other information also may be provided. The base and bounds registers specified for a process contain values representing the largest and smallest addresses that can be generated and accessed by an individual program. Where an operating system is the sole entity able to modify these memory management registers, adequate protection from access to the memory locations of one process from another process is provided. As a result, this memory management information is used by the operating system to provide a protected memory area for the process. A process generally uses the file system of the operating system to access files.

The present invention involves storing encrypted digital information, such an audio, video, text or an executable computer program, on a computer readable medium such that it can be copied easily for back-up purposes and transferred easily for distribution, but also such that it cannot be copied readily in decrypted form during use. In particular, the digital information is stored as a computer program that decrypts itself while it is used to provide the digital information, e.g., to provide executable operation code to the operating system of a computer, as

the digital information is needed. Any kind of encryption or decryption may be used and also may include authorization mechanisms and data compression and decompression. In one embodiment of the present invention, decrypted digital information exists only in memory accessible to the operating system and processes authorized by the operating system. When the digital information is a large application program, a copy of the entire decrypted application program is not likely to exist in the main memory at any given time, further reducing the likelihood that a useful copy of decrypted code could be made. The decryption operation also is performed only if some predetermined authorization procedure is completed successfully.

One embodiment of the invention, in which the decryption program is a form of dynamically loaded device driver, will first be described. Fig. 3 illustrates the structure of digital information as stored in accordance with one embodiment of the present invention, which may be stored on a computer readable medium such as a magnetic disc or compact disc read only memory (CD-ROM) to form a computer program product. The digital information includes a first portion 50, herein called an unwrap procedure or application, which is generally unencrypted executable program code. The purpose of the unwrap procedure is to identify the locations of the other portions of the digital information, and may perform other operations such as verification. In particular, the unwrap procedure identifies and extracts a program which will communicate with the operating system, herein called a virtual device driver 52. The unwrap procedure may include decryption and decompression procedures to enable it to decrypt/decompress the driver, and/or other content of this file. The program 52 need not be a device driver. The virtual device driver 52 typically follows the unwrap procedure 50 in the file container, the digital information. The virtual device driver, when executed, decrypts and decodes the desired digital information such as an executable computer program code from hidden information 54, which may be either encrypted and/or encoded (compressed). It is the decrypted hidden information which is the desired digital information to be accessed. This hidden information may be any kind of digital data, such as audio, video, text, and computer program code including linked libraries or other device drivers.

In this embodiment of the computer program product, labels delineate the boundaries between the device driver and the hidden files. These labels may or may not be encrypted. A first label 56 indicates the beginning of the code for the virtual device driver 52. A second label 58 indicates the end of the virtual device driver code. Another label 60 indicates the beginning of the hidden information and a label 62 indicates the end of that application. There may be one

or more blocks of such hidden information, each of which can be given a different name. It may be advantageous to use the name of the block of information in its begin and end tags. This computer program product thus contains and is both executable computer program code and one or more blocks of digital information. A table of locations specifying the location of each portion of the product could be used instead of labels. Such a table could be stored in a predetermined location and also may be encrypted.

The overall process performed using this computer program product in one embodiment of the invention will now be described in connection with Fig. 4. This embodiment may be implemented for use with the Windows95 operating system and is described in more detail in connection with Figs. 5-7. An embodiment which may be implemented for use on the WindowsNT 4.0 operating system is described in more detail below in connection with Fig. 8. In both of these described embodiments, the digital information is an executable computer program which is read by the operating system as data from this file and is executed. The same principle of operation would apply if the data were merely audio, video, text or other information to be conveyed by a user. In the embodiment of Fig. 4, the computer program is first loaded into memory in step 70, and the unwrap procedure 50 is executed by the operating system, as any typical executable computer program is executed. The unwrap procedure may perform authorization, for example by checking for a required password or authentication code, and may receive any data needed for decryption or decompression, for example keys or passwords, in step 72. Suitable authorization procedures may provide the ability to distribute software for single use. The unwrap procedure locates the virtual device driver 52 within the computer program in step 74, and then locates the hidden application in step 76. The virtual device driver 52 is then extracted by the unwrap procedure from the computer program, copied to another memory location and loaded for use by the operating system in step 78. An advantage of an operating system like Windows95 is that it allows such device drivers to be loaded dynamically without restarting the computer.

The executed unwrap procedure 50, in step 80, informs the loaded virtual device driver 52 of the location of the hidden information in the file, any keys or other passwords, and a name of a phantom directory and file to be called that only the unwrap procedure and the virtual device driver know about. The name of this phantom directory may be generated randomly. Each segment information hidden in the digital information product may be assigned its own unique file name in the phantom directory.

After the loaded virtual device driver 52 receives all communications from the unwrap procedure, it opens the original application file for read only access in step 82. The unwrap procedure then makes a call to the operating system in step 84 to execute the file in the phantom directory for which the name was transmitted to the loaded virtual device driver. One function of the loaded virtual device driver 52 is to trap all calls from the operating system to access files in step 86. Any calls made by the operating system to access files in the phantom directory are processed by the virtual device driver, whereas calls to access files in other directories are allowed to proceed to their original destination. In response to each call from the operating system, the virtual device driver obtains the bytes of data requested by the operating system from the original computer program file in step 88. These bytes of data are then decrypted or decompressed in step 90 and returned to the operating system. When processing is complete, the phantom application is unloaded from the operating system in step 92, and may be deleted from the memory.

A more detailed description of the process of Fig. 4 will now be described in connection with Figs. 5-7. Fig. 5 is a flowchart describing the operation of one embodiment of the unwrap procedure in more detail. The first step performed by this procedure is identifying the operating system being used, in step 100. This step is useful because different methods may be used with different operating systems. All code that may be used to run in various operating systems may be placed in this unwrap procedure. This procedure also may contain the decompression/decryption code, for example or any other computer program code to be executed.

The executed application then opens the original executable file as a data file and searches for the begin and end tags of the device driver and hidden files in step 102. The device driver code is copied into memory and loaded into the operating system in step 104. The unwrap procedure then informs the device driver of the name of the original application file, offsets of the hidden files and the name of a phantom directory, which is typically randomly generated (step 106). This communication may be performed using a "DeviceIOControl" function call in the Windows95 operating system. The unwrap procedure then makes a call to the operating system to execute the hidden file in the phantom directory, in step 108.

The operation of one embodiment of a device driver will now be described in connection with Fig. 6. After the device driver is loaded into the operating system, it hooks into a position between the operating system and a file system driver (FSD), in step 110, to intercept

calls made by the operating system to the FSD for data from files in the phantom directory. The FSD is the code within the operating system that performs physical reading and writing of data to disk drives. The operating system makes requests to the FSD for data from files in directories on the disk drives. The driver then receives information from the unwrap procedure including the name of the original file, the location of hidden files within the original file, and the name of the phantom directory created by the unwrap procedure (step 112). The device driver opens the original file as a read only data file. The device driver now traps calls, in step 114, made from the operating system for files in the phantom directory. Calls to other directories are ignored and passed on to the original destination. The device driver then reads the data from the original data file, decrypts and decompresses it, and returns the decrypted/decompressed data to the operating system in step 116.

For example, if the offset for the hidden application in the original data file is 266,270 bytes and the operating system asks for 64 bytes starting at offset 0 of the hidden application in the phantom directory, the device driver reads 64 bytes from the original file starting at offset 266,270, decrypts/decompresses those 64 bytes, and returns the first 64 decrypted/decompressed bytes back to the operating system. From the point of view of the operating system, the 64 bytes appear to have come from the file in the phantom directory. Steps 114 and 116 are performed on demand in response to the operating system.

A block diagram of the computer system in this embodiment, with a device driver loaded and in operation, will now be described in more detail in connection with Fig. 7. Fig. 7 illustrates the operating system 120, the loaded device driver 122, a file system driver 124, the original executable file 126 as it may appear on disk and the unwrap procedure 128. The executable file may in fact be on a remote computer and accessed through a network by the device driver. The unwrap procedure causes the operating system to begin execution of the hidden file by issuing an instruction to execute the file in the phantom directory, as indicated at 130. This command is issued after the device driver 122 is informed of the file name of the original executable file 126, offsets of the hidden files within that file and the name of the phantom directory, as indicated at 132. The operating system then starts making calls to the phantom directory as indicated at 134. The device driver 122 traps these calls and turns them into requests 136 to the file system driver to access the original executable file 126. Such requests actually are made to the operating system 120, through the device driver 122 to the file system driver 124. The file system driver 124 returns encrypted code 138 to the device driver

122. The encrypted code 138 actually passes back through the device driver 122 to the operating system 120 which in turn provides the encrypted code 138 to the device driver 122 as the reply to the request 136 for the original file. The device driver 122 then decrypts the code to provide decrypted code 140 to the operating system 120.

Another embodiment of the invention will now be described in connection with Fig. 8. This embodiment may be implemented using the WindowsNT 4.0 operating system, for example. In this embodiment, the device driver portion 52 of the computer program product is not used. The unwrap procedure for this embodiment begins by identifying the operating system being used similar, which is step 100 in Fig. 5. If the operating system is Windows NT 4.0, for example, a different unwrap procedure for this embodiment is performed. Before describing this unwrap procedure, a brief description of some of the available operating system commands will be provided.

Currently, under all versions of the Window operating system or operating environment from Microsoft Corporation (such as Windows 3.1, Windows 95 and Windows NT 3.51 and 4.0) all executable files (.exe) or dynamic link library (.dll and .ocx) files, which are executable files with different header and loading requirements than .exe files, that are loaded into memory by the operating system must reside as a file either locally, e.g., on a disk drive or remotely, e.g., over a network or communications port. All further references herein to loading an executable will be using the Win32 function calls used in Windows 95 and NT 3.51 and 4.0 operating systems. The CreateProcess() function which loads files with an .exe extension takes ten parameters:

```
BOOL CreateProcess(// Prototype from Microsoft Visual C++ Help Documentation
  LPCTSTR lpApplicationName,                  // pointer to name of executable module
  LPTSTR lpCommandLine,                       // pointer to command line string
  LPSECURITY_ATTRIBUTES lpProcessAttributes,  // pointer to process security attributes
  LPSECURITY_ATTRIBUTES lpThreadAttributes,   // pointer to thread security attributes
  BOOL bInheritHandles,                       // handle inheritance flag
  DWORD dwCreationFlags,                      // creation flags
  LPVOID lpEnvironment,                       // pointer to new environment block
  LPCTSTR lpCurrentDirectory,                 // pointer to current directory name
  LPSTARTUPINFO lpStartupInfo,                // pointer to STARTUPINFO
  LPPROCESS_INFORMATION lpProcessInformation  // pointer to PROCESS_INFORMATION
);
```

Three of these parameters are pointers to strings that contain an application file name, command line parameters, and the current directory. The other parameters are security, environmental, and process information. The LoadLibrary() function takes one parameter that is a pointer to a string that contains the application file name:

```
HINSTANCE LoadLibrary(// Prototype from Microsoft Visual C++ Help Documentation
  LPCTSTR lpLibFileName      // address of filename of executable module
);
```

The LoadLibraryEx() function takes three parameters the first being the same as LoadLibrary(), the second parameter must be null, and the third tells the operating system whether to load the file as an executable or as a data file in order to retrieve resources such as icons or string table data from it and not load it as an executable:

```
HINSTANCE LoadLibraryEx(// Prototype from Microsoft Visual C++ Help Documentation
  LPCTSTR lpLibFileName,     // points to name of executable module
  HANDLE hFile,              // reserved, must be NULL
  DWORD dwFlags              // entry-point execution flag
);
```

The CreateFile() function is used to create and open files and to load files such as device drivers. This function also requires a pointer to a string that contains the name of a physical file:

```
HANDLE CreateFile(// Prototype from Microsoft Visual C++ Help Documentation
  LPCTSTR lpFileName,                          // pointer to name of the file
  DWORD dwDesiredAccess,                       // access (read-write) mode
  DWORD dwShareMode,                           // share mode
  LPSECURITY_ATTRIBUTES lpSecurityAttributes,  // pointer to security descriptor
  DWORD dwCreationDistribution,                // how to create
  DWORD dwFlagsAndAttributes,                  // file attributes
  HANDLE hTemplateFile                         // handle to file with attributes to copy
);
```

There are other functions such as MapViewOfFile() and MapViewOfFileEx() that map areas of memory to an already opened physical file through a handle to that file. They have the following parameters:

```
LPVOID MapViewOfFile(// Prototype from Microsoft Visual C++ Help Documentation
  HANDLE hFileMappingObject,        // file-mapping object to map into address space
  DWORD dwDesiredAccess,            // access mode
  DWORD dwFileOffsetHigh,           // high-order 32 bits of file offset
  DWORD dwFileOffsetLow,            // low-order 32 bits of file offset
  DWORD dwNumberOfBytesToMap        // number of bytes to map
 );

LPVOID MapViewOfFileEx(// Prototype from Microsoft Visual C++ Help Documentation
  HANDLE hFileMappingObject,        // file-mapping object to map into address space
  DWORD dwDesiredAccess,            // access mode
  DWORD dwFileOffsetHigh,           // high-order 32 bits of file offset
  DWORD dwFileOffsetLow,            // low-order 32 bits of file offset
  DWORD dwNumberOfBytesToMap,       // number of bytes to map
  LPVOID lpBaseAddress              // suggested starting address for mapped view
 );
```

All of the foregoing functions directly use a pointer to a string that is a physical file. The only file functions that do not directly use a physical filename are functions like CreateNamedPipe(), which has the following parameters:

```
HANDLE CreateNamedPipe(// Prototype from Microsoft Visual C++ Help Documentation
  LPCTSTR lpName,                          // pointer to pipe name
  DWORD dwOpenMode,                        // pipe open mode
  DWORD dwPipeMode,                        // pipe-specific modes
  DWORD nMaxInstances,                     // maximum number of instances
  DWORD nOutBufferSize,                    // output buffer size, in bytes
  DWORD nInBufferSize,                     // input buffer size, in bytes
  DWORD nDefaultTimeOut,                   // time-out time, in milliseconds
  LPSECURITY_ATTRIBUTES lpSecurityAttributes  // pointer to security attributes structure
 );
```

The string to which CreateNamedPipe() points using the first parameter is a string that both an existing executable and the operating system know about and does not exist physically. Unfortunately both of the executables that "know" this private name could only be loaded using one of the other procedures that required a physical file. Currently it is not possible to load an executable using a "named pipe" name. Both of or any executables that use the name of the "named pipe" already must have been loaded into memory.

All of the foregoing functions require a physical file because all of them use "file mapping" processes. File mapping allows large executable files to appear to be loaded rapidly since they are rarely completely loaded into memory but rather are mapped into memory. The detriment to this mapping capability is that executable code must remain in physical memory in a file in unencrypted form in order to be loaded, unless there is a middle layer or file system driver that the operating system uses as a physical layer and that decrypts the executable code to the operating system on demand. The potential weakness here is that another file system driver can hook into the operating system to monitor traffic between the operating system and all file system drivers and capture decrypted executable code passing from the file system driver to the operating system. Some operating systems allow such monitoring more than others. Many anti-viral software packages use this technique to prevent computer virus attacks.

One method of loading and executing encrypted executable computer program code is to use a stub executable having two parts. The first part is the normal front end loader code that all executables have. In addition, the first part would perform any authorization which may include receiving a password from the user, then allocate enough memory to hold hidden encrypted code when it is decrypted, either in its entirety or a portion of it, copy the encrypted code into that area of protected (and preferably locked so no disk swapping occurs) memory, decrypt it once it is in memory and only in memory, and then have the operating system load the code only from memory therefore bypassing any file system drivers or TSRs so they have access to only encrypted code.

Some of the file functions listed above and similar functions on other operating systems could be modified easily by a programmer having access to source code for those operating systems, or a new operating system may be made to provide functions which allow direct loading of executable code from memory rather than physical files. For example, in the Win32 commands, a command similar to CreateProcess() command could be provided. The command should have a few extra parameters including the process identifier of the process that contains

the now decrypted executable code, the memory address of the start of the decrypted code, and the size of the decrypted code. The command could also contain a parameter specifying a "call back" function within the first process that would provide decrypted code on demand directly to the operating system through a protected buffer, therefore allowing only a portion of the encrypted code to be decrypted at any one time instead of in its entirety, for better protection and less memory use. The second parameter of the LoadLibraryEx() command that now needs to be NULL could be expanded to hold a structure that contained the same information. Both of these and other similar functions could be changed or created to allow loading executable code either as an .exe, .dll, or other extensions or identifiers, such as by using a "named pipe" name that only the operating system and process that holds decrypted code know about and having the operating system load from the named pipe.

Alternatively, without having such additional capabilities in the operating system, an application program can be divided into two parts. The first part is code that is common to all applications such as code for allocating memory off the heap and code that provides some interaction with the user. This kind of code is generally not code that the content provider is concerned about copying. The second part is the code that the content provider believes is valuable. Typically this valuable code is a business logic code or what would be considered a middle tier of a three-tier environment. A content provider would like to protect this second part of the code, at least much more that the first part of the code. The content provider would place all of the important code to be protected inside a dynamic link library and the code that is not that important would reside in the front end "stub" executable. Both of these would be combined into another executable containing the .dll in encrypted form only, along with any other files, data, information, and/or tables for holding, for example, hardware identifiers. This other executable is the final digital information product.

The first part of the digital information product, i.e., the executable stub, would load and execute normally like any other application. It then would perform any authorization procedures. Once the proper authorization or password was completed successfully, an unwrap procedure would be performed as will now be described in connection with Fig. 8, it would then allocate enough protected memory using a function like VirtualAlloc() as shown in step 150:

```
DWORD nFileSize = 0;
DWORD nPhantomFileSize = 0;
```

```
DWORD exeOffset = 0;
DWORD nPreferredLoadAddress = GetPreCompiledLoadAddress();
CString cCommandFile = UnwrapGetNTCommandFile();
exeOffset = UnwrapGetDllOffset(cCommandFile);
nFileSize = UnwrapGetDllSize(cCommandFile);
nPhantomFileSize = nFileSize + 0x3000; // add any needed extra space
// Increase buffer size to account for page size (currently Intel page size).
DWORD nPageSize = GetPageSize();
nPhantomFileSize += (nPageSize -(nPhantomFileSize % nPageSize));
// Allocate the memory to hold the decrypted executable.
LPVOID lpvBlock = VirtualAlloc((LPVOID) nPreferredLoadAddress,
                    nPhantomFileSize,
                        MEM_RESERVE | MEM_COMMIT, PAGE_READWRITE);
```

This function can request a particular address space. Preferably, this address space is the preferred load address space to which the .dll was linked in order to minimize any needed relocation and fix up code. The stub executable may lock that area of memory in step 152, for example by using VirtualLock() to prevent any memory writes to a swap file, depending on the operating system, as shown below:

```
BOOL bVLock = VirtualLock((LPVOID) nPreferredLoadAddress, nPhantomFileSize);
```

The memory area still should be secure even without this preventive step since the Windows 95 and NT operating systems do not allow any user access to swap files.

The encrypted code is then copied from the digital information product into the allocated protected memory in step 154, for example by using the following command:

```
UnwrapCopyHiddenExeToMem(cCommandFile, exeOffset, nFileSize, (char *) lpvBlock);
```

Once in memory, the stub would then decrypt the code to that same portion of memory in step 156, for example by using the following commands:

```
CwrapDecryptSeed(cPassword.GetBuffer(0), cPassword.GetLength());
CwrapDecrypt((unsigned char *) lpvBlock, 0, nFileSize);
```

Any "fix up and relocation" type services would then be performed in step 158, for example by using the following command:

```
UnwrapFixUpAndRelocateDll(lpvBlock);
```

Possibly, the memory protection may be changed to execute only in step 160, for example by using the VirtualProtect() command as follows:

```
DWORD lpflOldProtect; // variable to get old protection
BOOL bVProtect = VirtualProtect((LPVOID) nPreferredLoadAddress,
                                nPhantomFileSize,
                                PAGE_EXECUTE,
                                &lpflOldProtect);
```

Function calls then can be made into that area of memory that now contains the decrypted code:

```
UnwrapDoDllAlgorithms();
```

Some of the "fix up" operations to be performed above include placing the addresses of external or stub.exe functions into the address place holders of the decrypted .dll or internal code, by using commands similar to the following:

```
WriteAddress((char*) 0x0a406104, (DWORD) &CallBackFunction1);
WriteAddress((char*) 0x0a406100, (DWORD) &CallBackFunction2);
```

For instance a wrapper function could be created in the outer stub.exe that received a size parameter, allocated that amount of memory off of the heap, and passed back the starting address of that block of memory. Another example would be to have encrypted algorithms within the hidden, encrypted .dll which would be called at run time from the front end stub once decrypted

within protected memory. The dynamic link library would be compiled and linked to expect a pointer to a function that took that parameter and/or returned a value by including prototypes in the header file as follows:

```
void (*lpCallBackFunc1)();
void (*lpCallBackFunc2)(unsigned long);
```

Function calls to "external" functions also could be added as follows:

```
(*lpCallBackFunc1)();
unsigned long z = x * x;
(*lpCallBackFunc2)(z);
```

At run time the "fix up" code would take the run time address of that "wrapper function" and place it into the pointer address within the .dll block of code as follows:

```
WriteAddress((char*) 0x0a406104, (DWORD) &CallBackFunction1);
WriteAddress((char*) 0x0a406100, (DWORD) &CallBackFunction2);
```

This information is readily available using the .cod output files from the compiler, an example of which follows:

```
_TestSum PROC NEAR                                          ; COMDAT
; Line 8
 00000    56          push   esi
; Line 23
 00001    ff 15 00 00 00
          00          call   DWORD PTR_lpCallBackFunc1
; Line 24
 00007    8b 44 24 08 mov    eax, DWORD PTR_a$[esp]
 0000b    50          push   eax
 0000c e8 00 00 00 00 call   _TestSquare
```