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	ML	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.

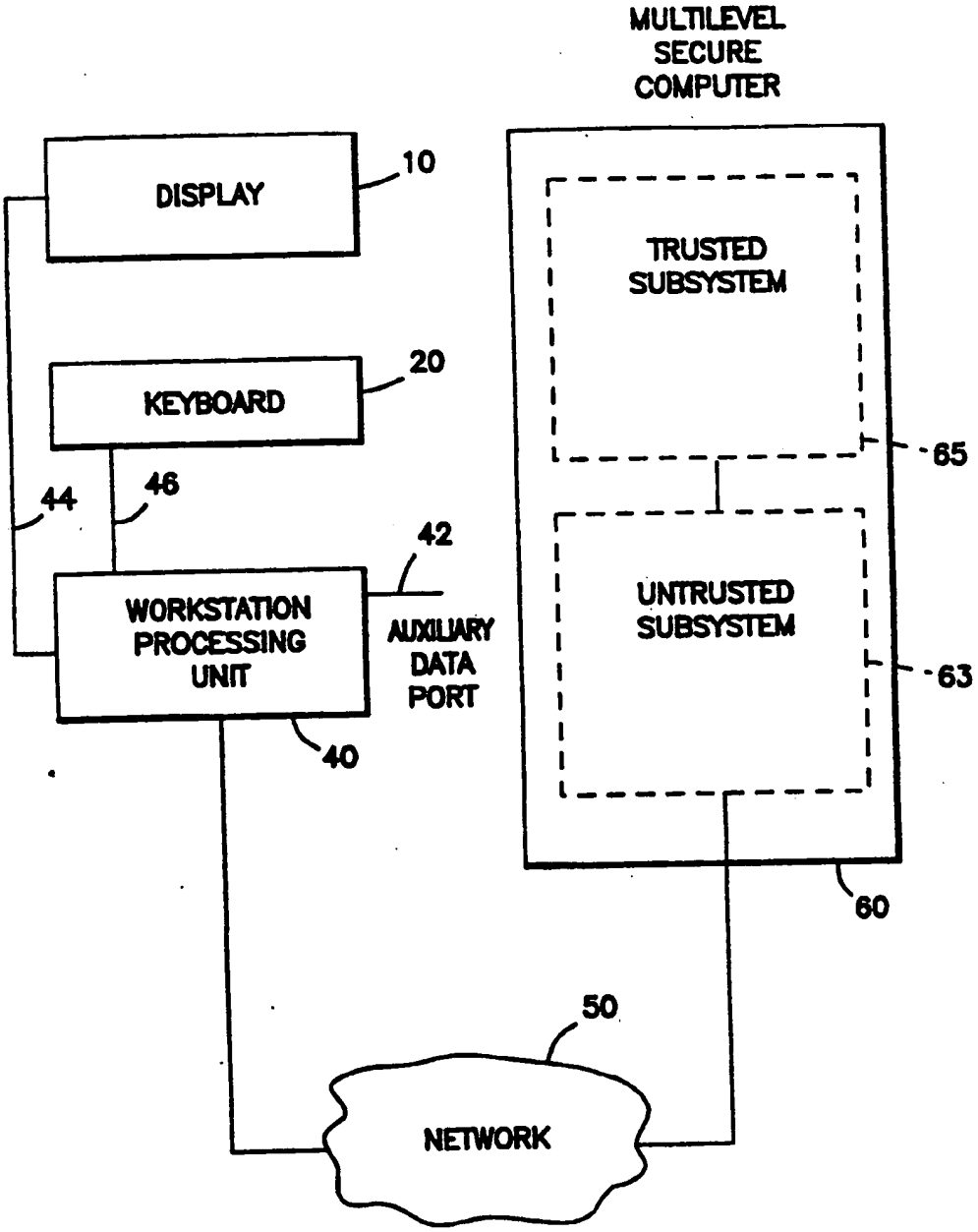


FIG. 1
PRIOR ART

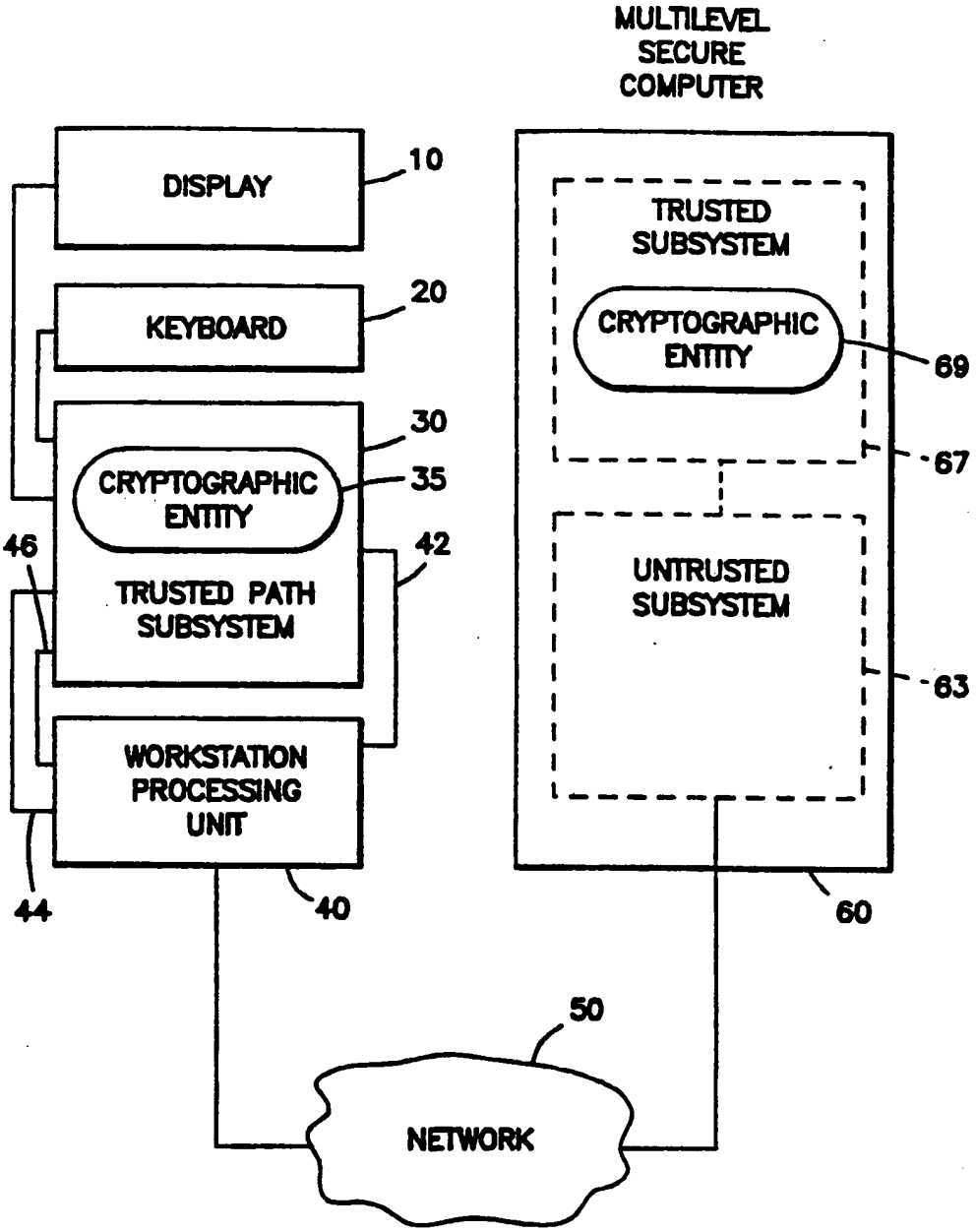

MULTILEVEL SECURE COMPUTER
DISPLAY
10
KEYBOARD
20
CRYPTOGRAPHIC ENTITY
30
35
TRUSTED PATH SUBSYSTEM
46
42
WORKSTATION PROCESSING UNIT
44
40
TRUSTED SUBSYSTEM
CRYPTOGRAPHIC ENTITY
69
67
UNTRUSTED SUBSYSTEM
63
60
50
NETWORK

FIG. 2

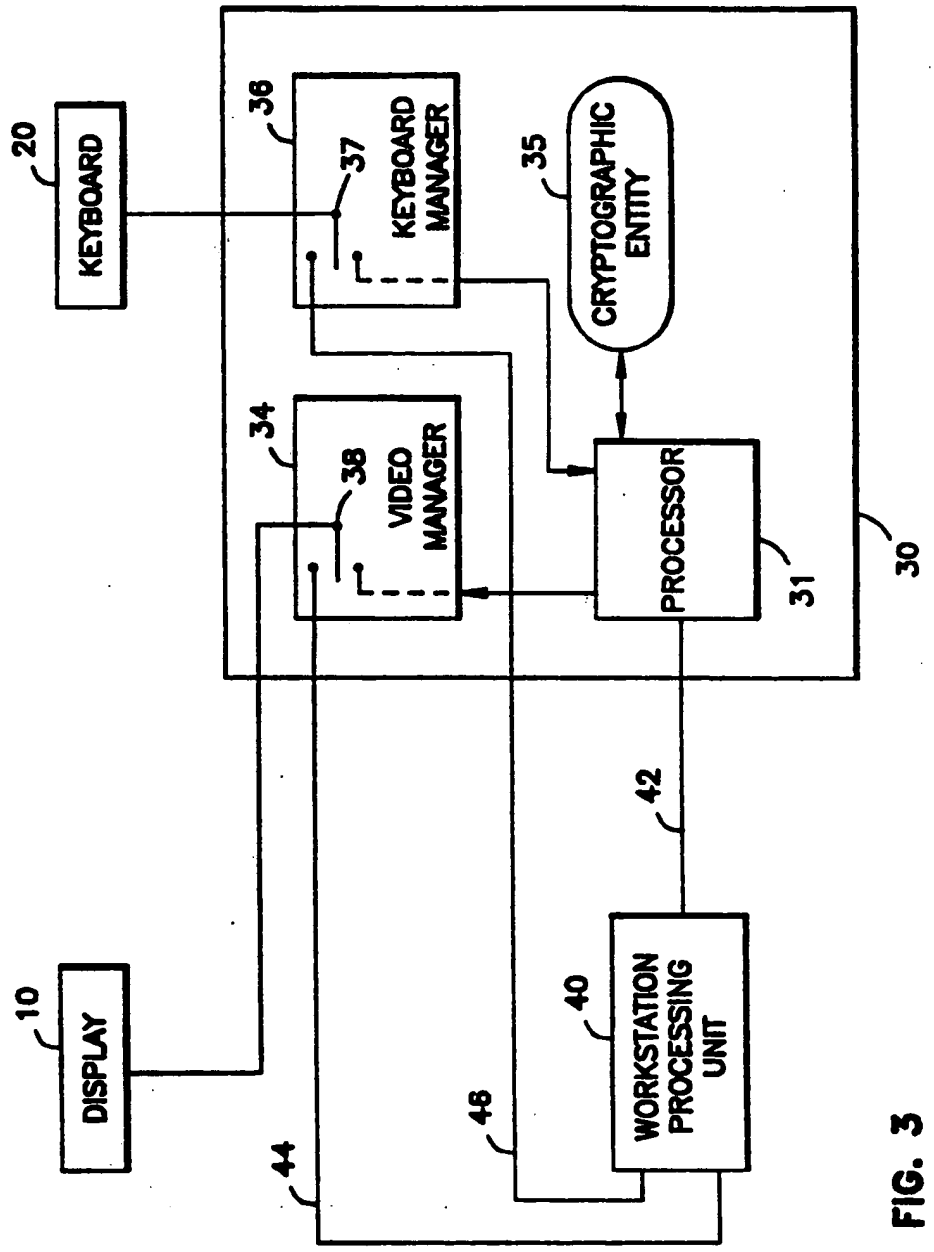

20
KEYBOARD
36
37
KEYBOARD MANAGER
35
CRYPTOGRAPHIC ENTITY
34
38
VIDEO MANAGER
PROCESSOR
31
30
42
10
DISPLAY
40
WORKSTATION PROCESSING UNIT
46
44

FIG. 3

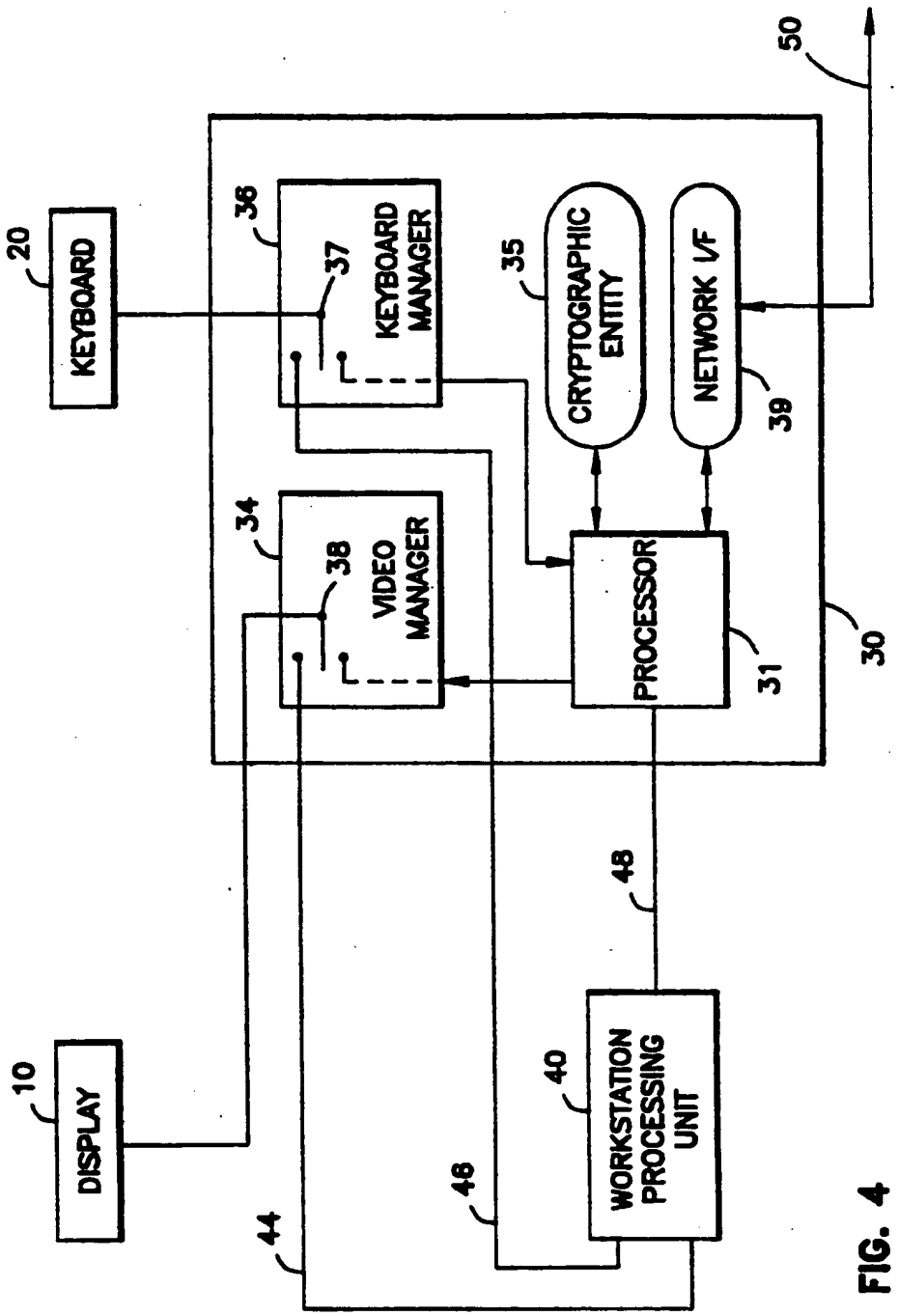

50
20
KEYBOARD
36
37
KEYBOARD MANAGER
35
CRYPTOGRAPHIC ENTITY
NETWORK I/F
39
34
38
VIDEO MANAGER
PROCESSOR
31
30
48
10
DISPLAY
40
WORKSTATION PROCESSING UNIT
46
44

FIG. 4

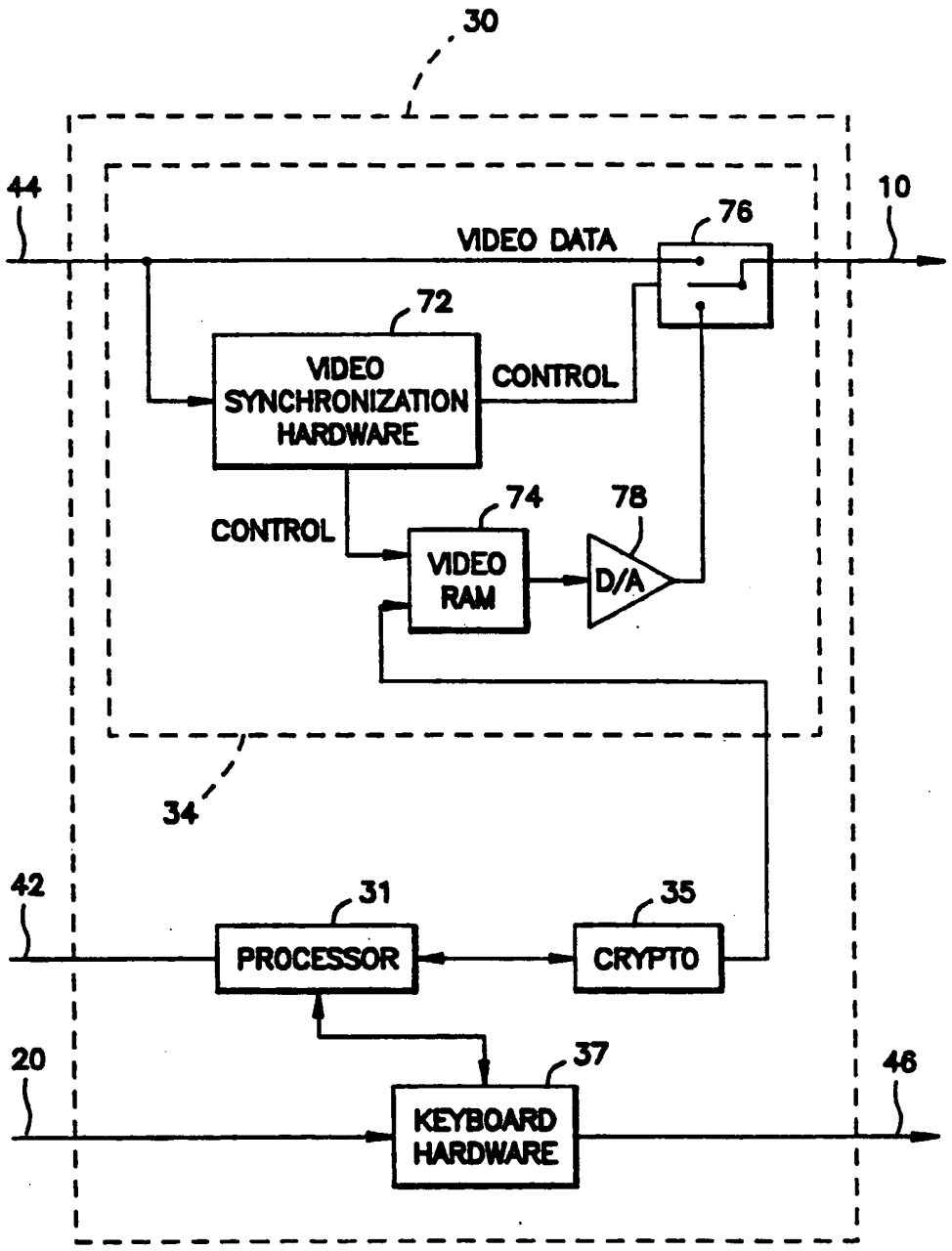

30
44
10
VIDEO DATA
76
72
VIDEO SYNCHRONIZATION HARDWARE
CONTROL
CONTROL
74
78
VIDEO RAM
D/A
34
42
31
35
PROCESSOR
CRYPTO
20
37
46
KEYBOARD HARDWARE

FIG. 5

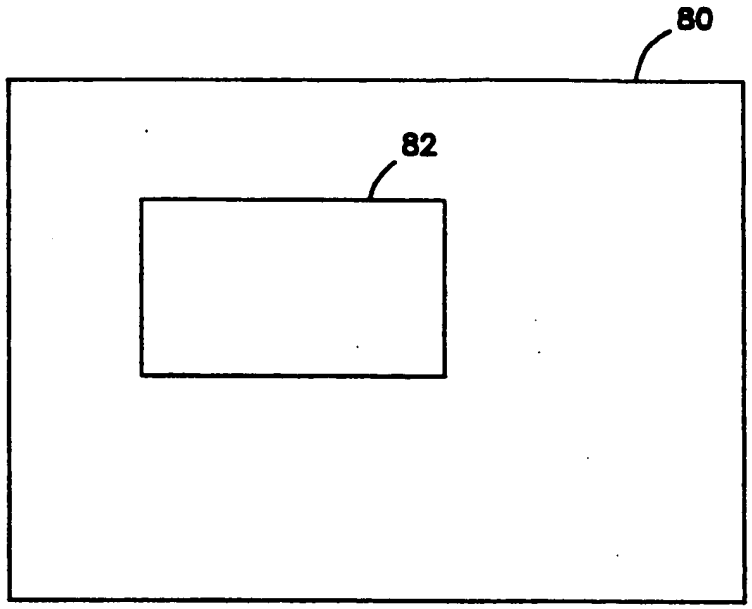


FIG. 6

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)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 93/06511

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

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

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

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

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).

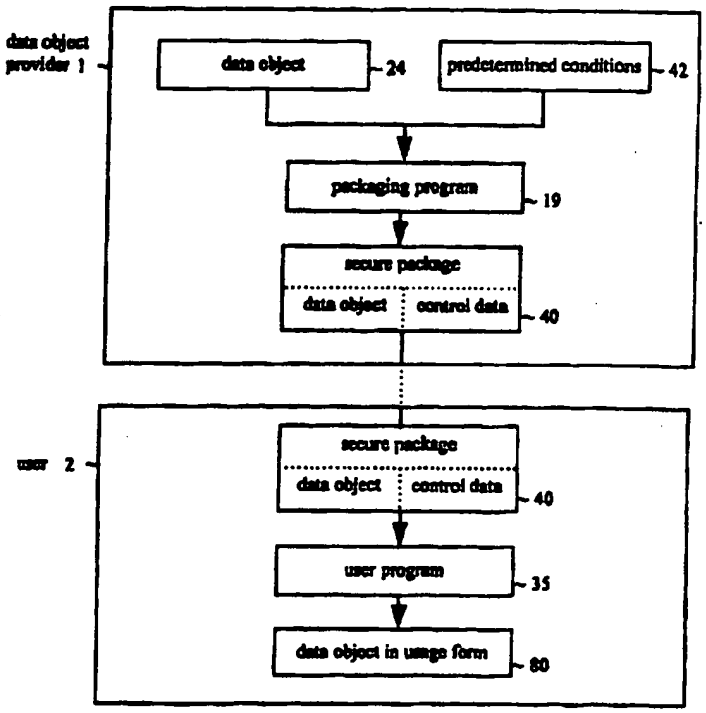
(81) Designated States: AL, AM, AT, AT (Utility model), AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), EE, EE (Utility model), ES, FI, FI (Utility model), GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, 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	Kyrgyzstan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	KZ	Kazakhstan	SG	Singapore
CH	Switzerland	LI	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovakia
CM	Cameroon	LR	Liberia	SN	Senegal
CN	China	LT	Lithuania	SZ	Swaziland
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
CZ	Czech Republic	LV	Latvia	TG	Togo
DE	Germany	MC	Monaco	TJ	Tajikistan
DK	Denmark	MD	Republic of Moldova	TT	Trinidad and Tobago
EE	Estonia	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	UG	Uganda
FI	Finland	MN	Mongolia	US	United States of America
FR	France	MR	Mauritania	UZ	Uzbekistan
GA	Gabon			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

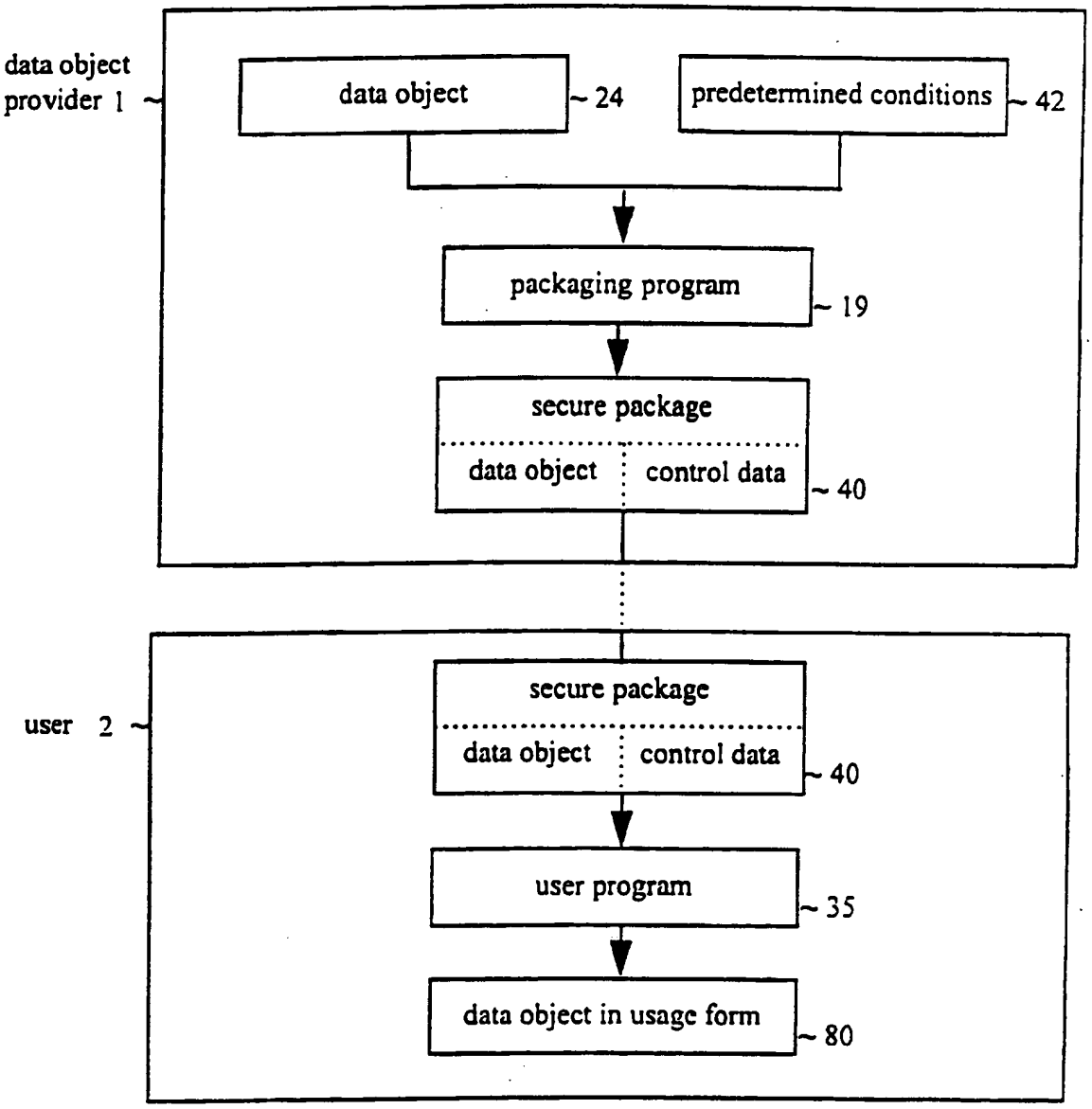


Fig 2

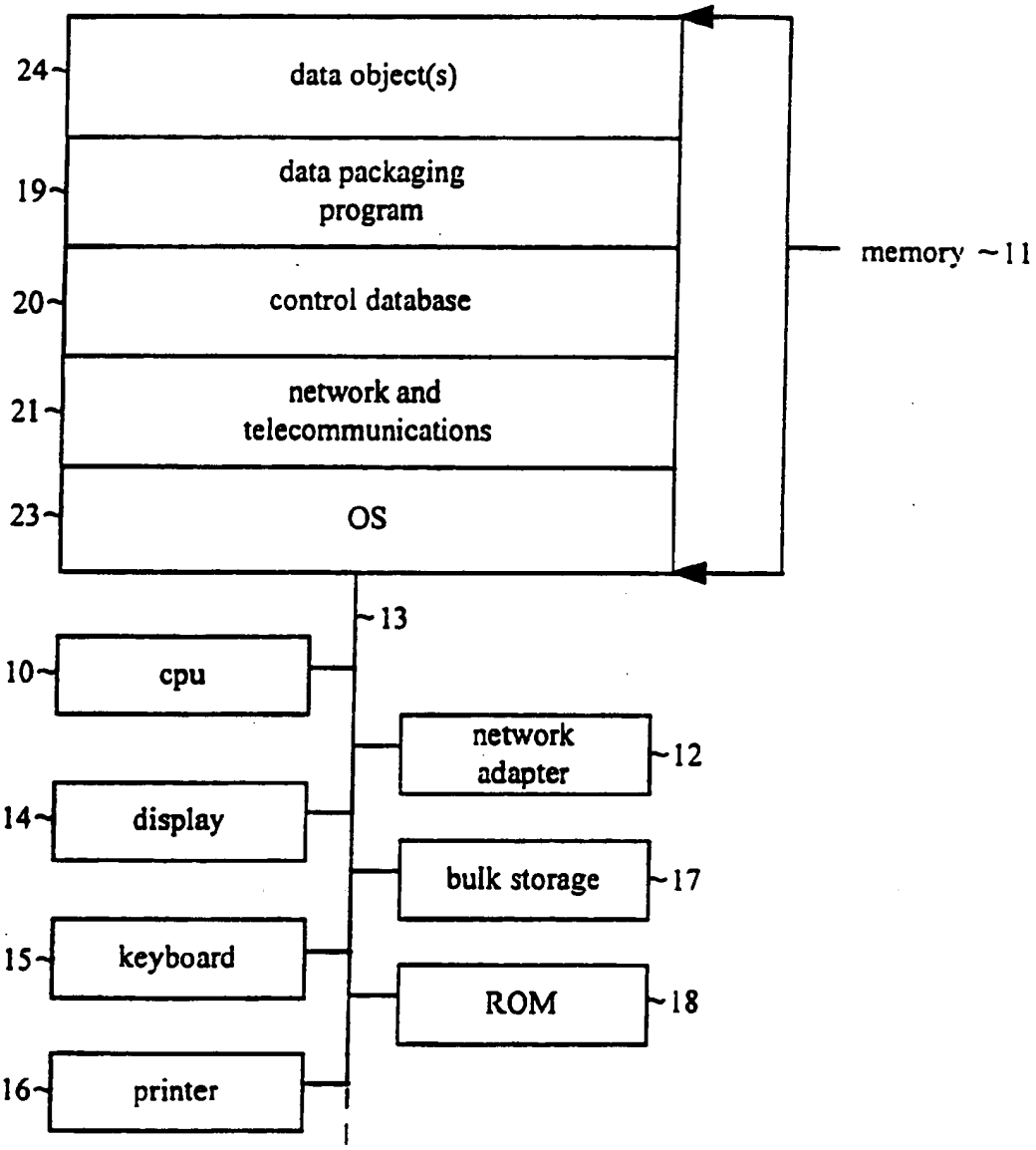


Fig 3

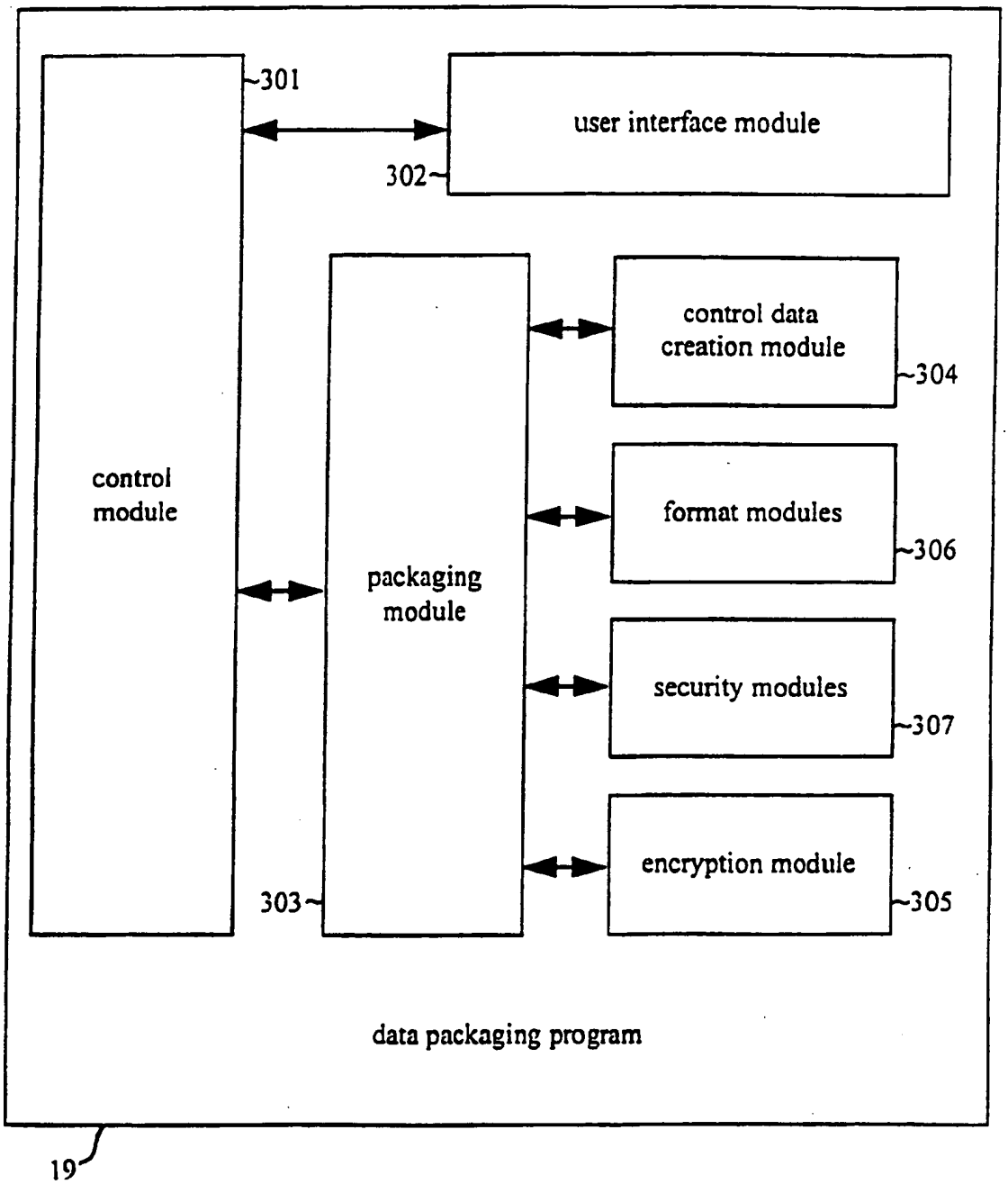


Fig 4

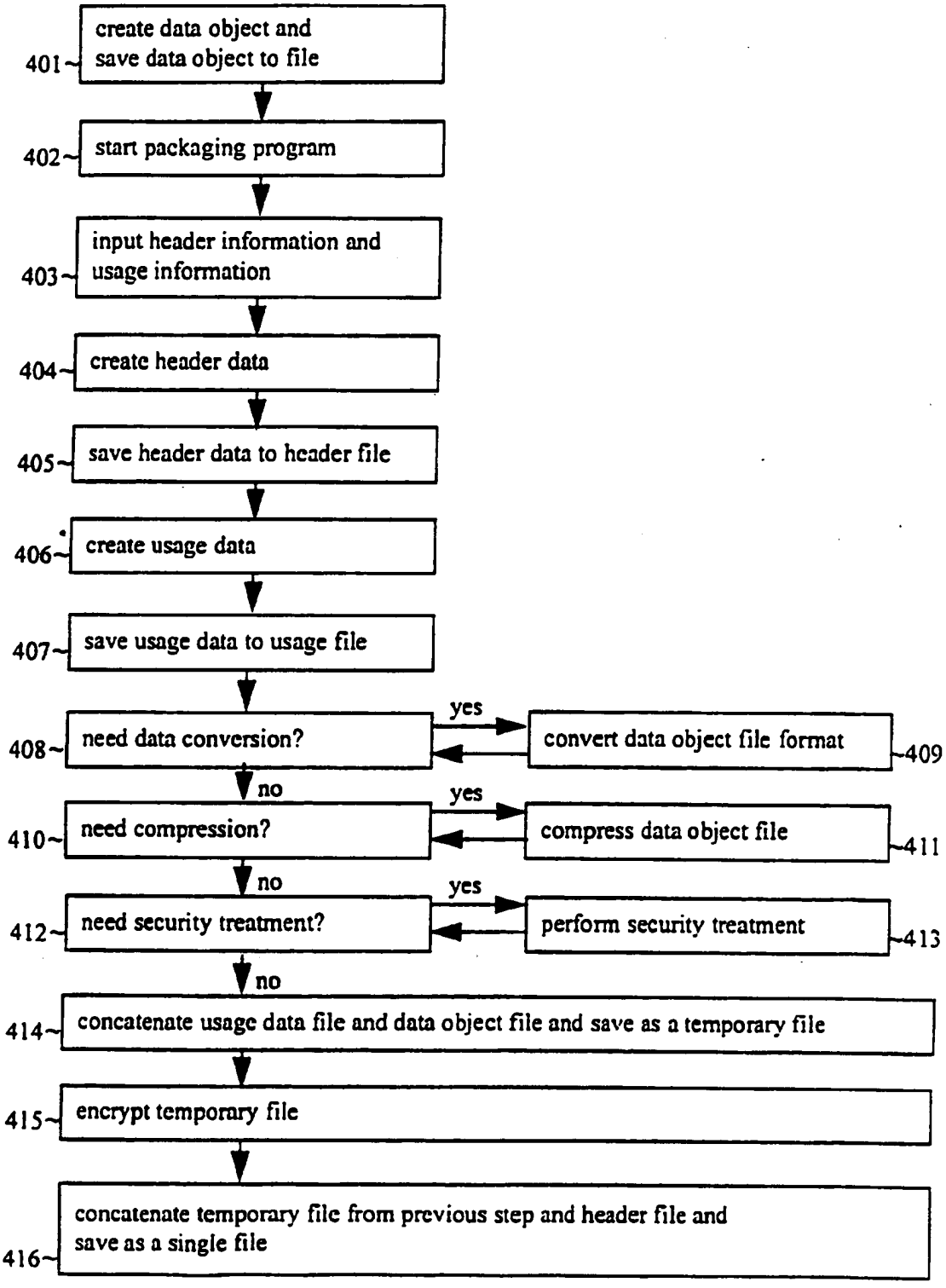


Fig 5

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

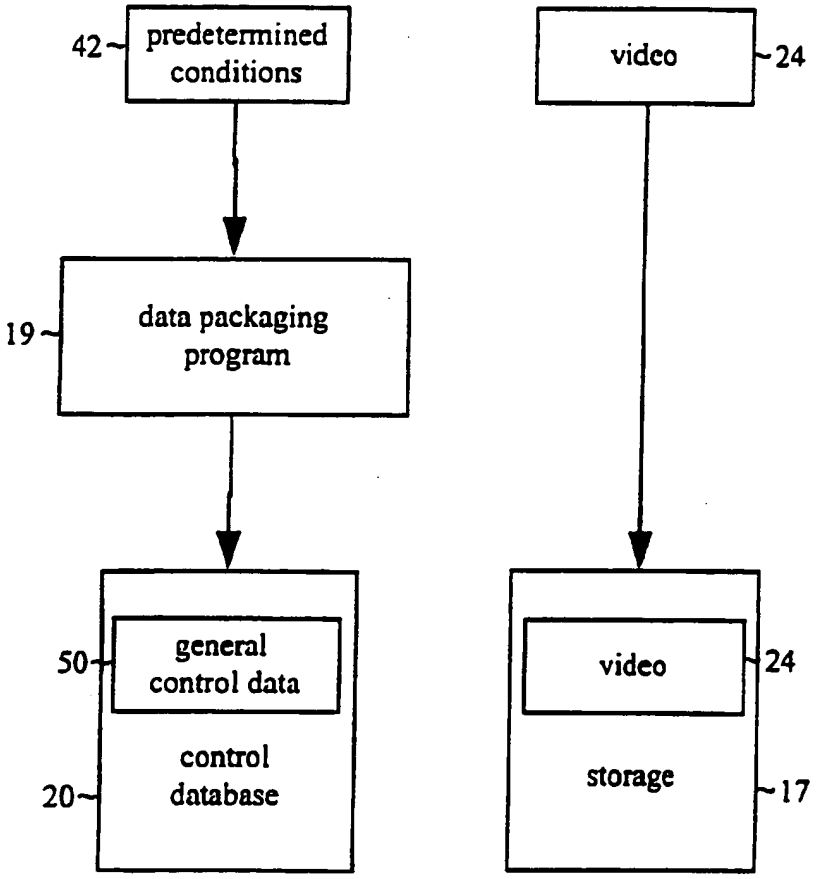


Fig 8a

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

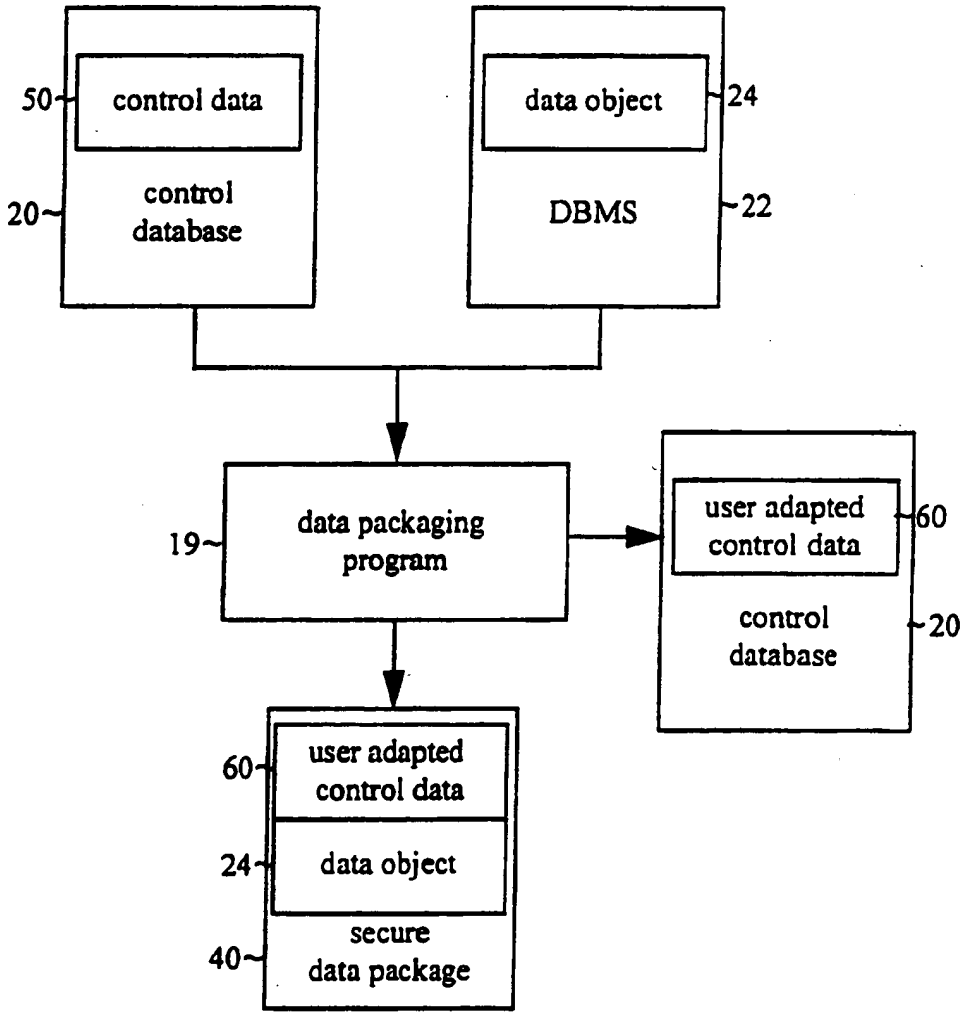


Fig 10

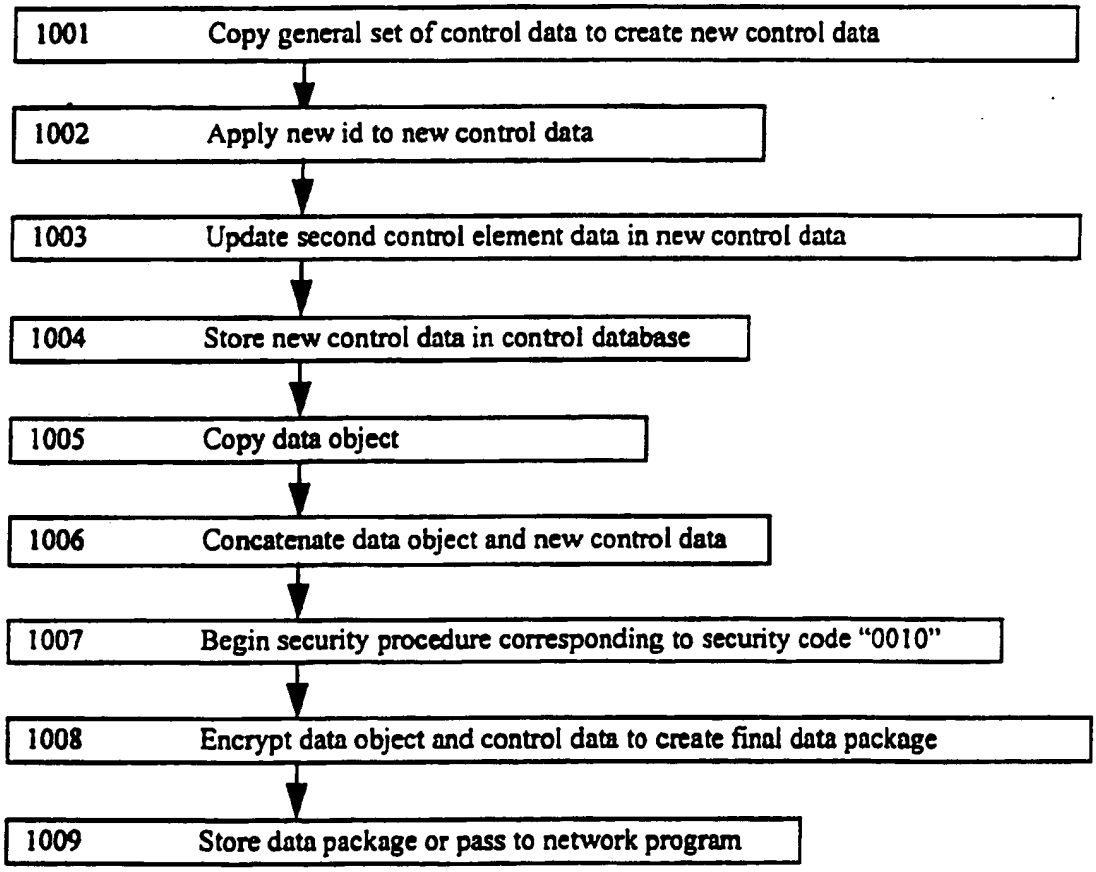
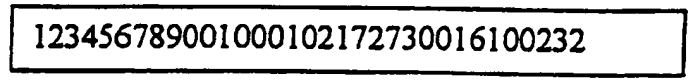


Fig 11

control data in memory



AVI file in memory

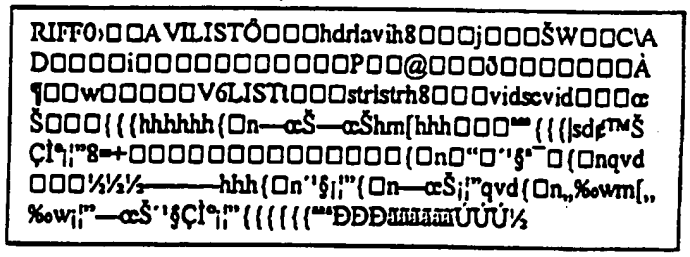


Fig 12a

concatenated control data and AVI file in memory

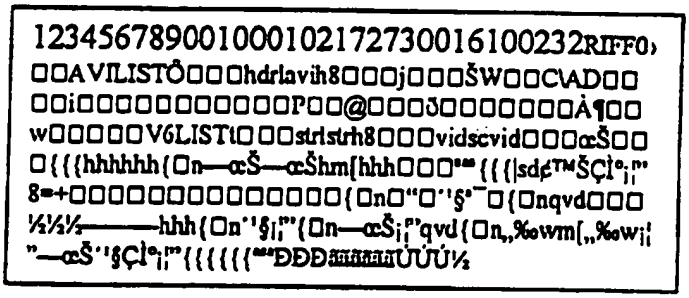
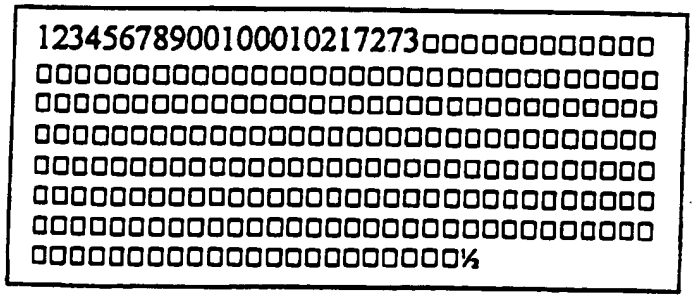


Fig 12b

concatenated and encrypted control data and AVI file in memory



SUBSTITUTE SHEET

Fig 13

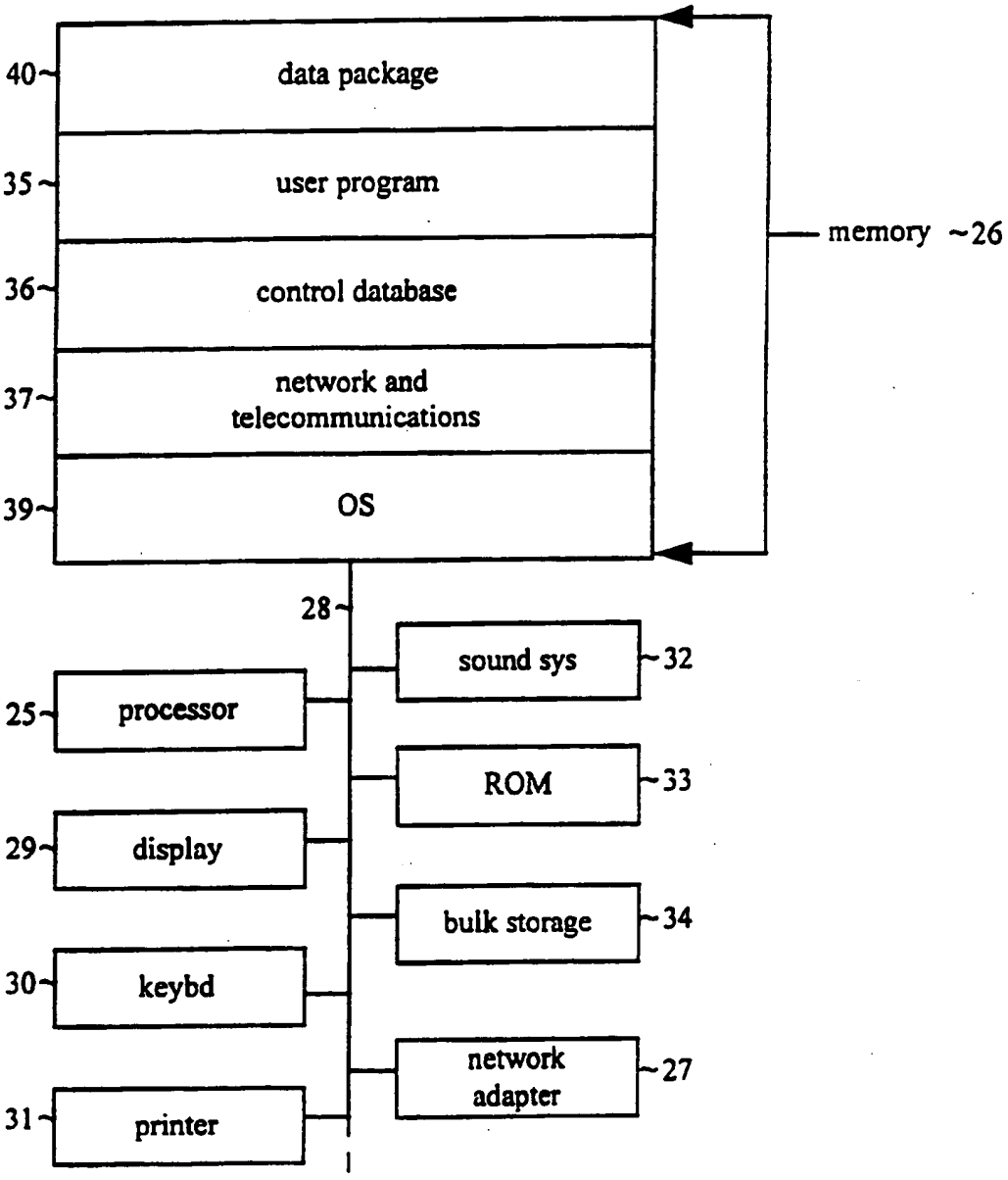


Fig 14

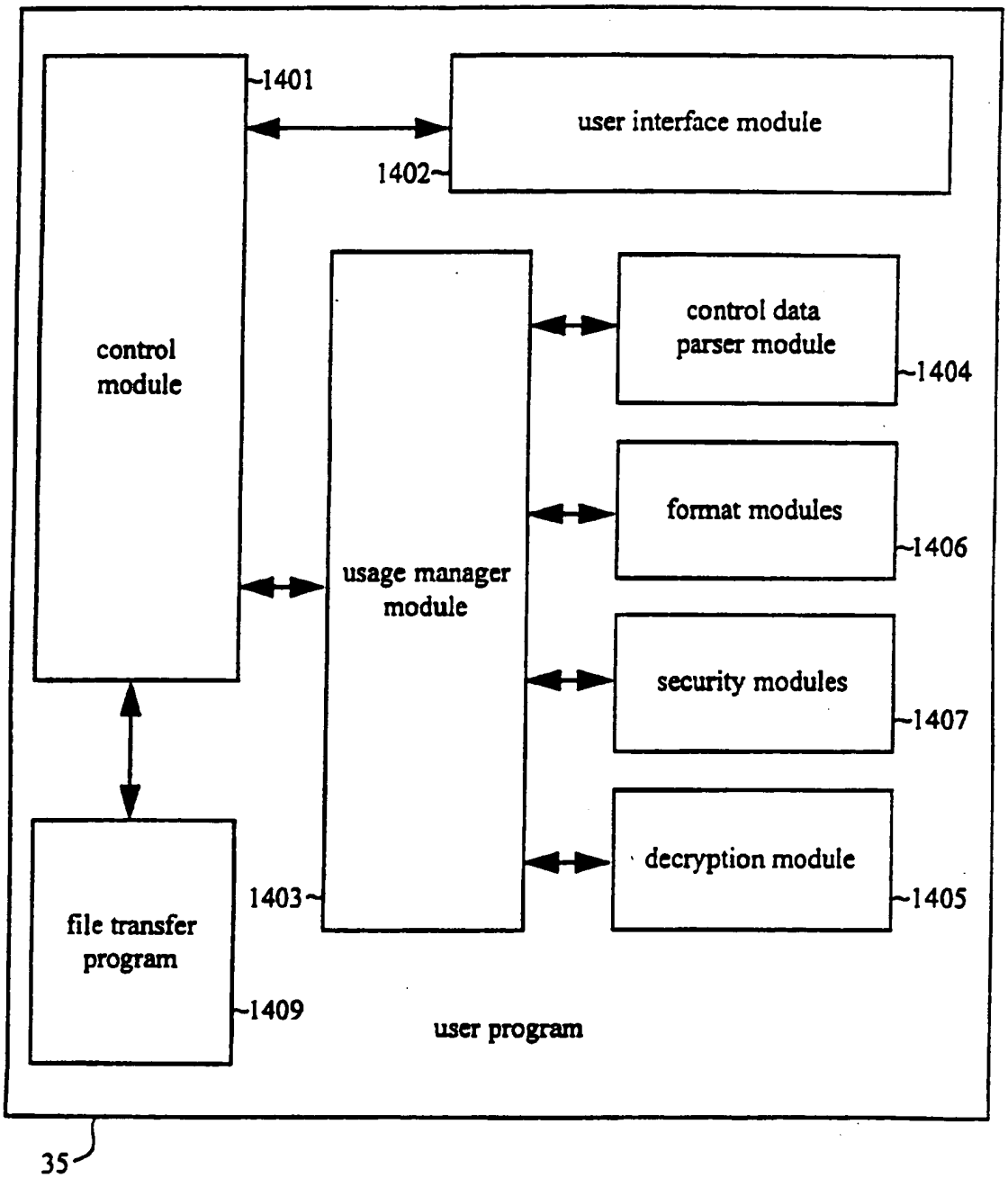

1401
user interface module
1402
control module
usage manager module
control data parser module
1404
format modules
1406
security modules
1407
decryption module
1405
1403
file transfer program
1409
user program
35

Fig 15

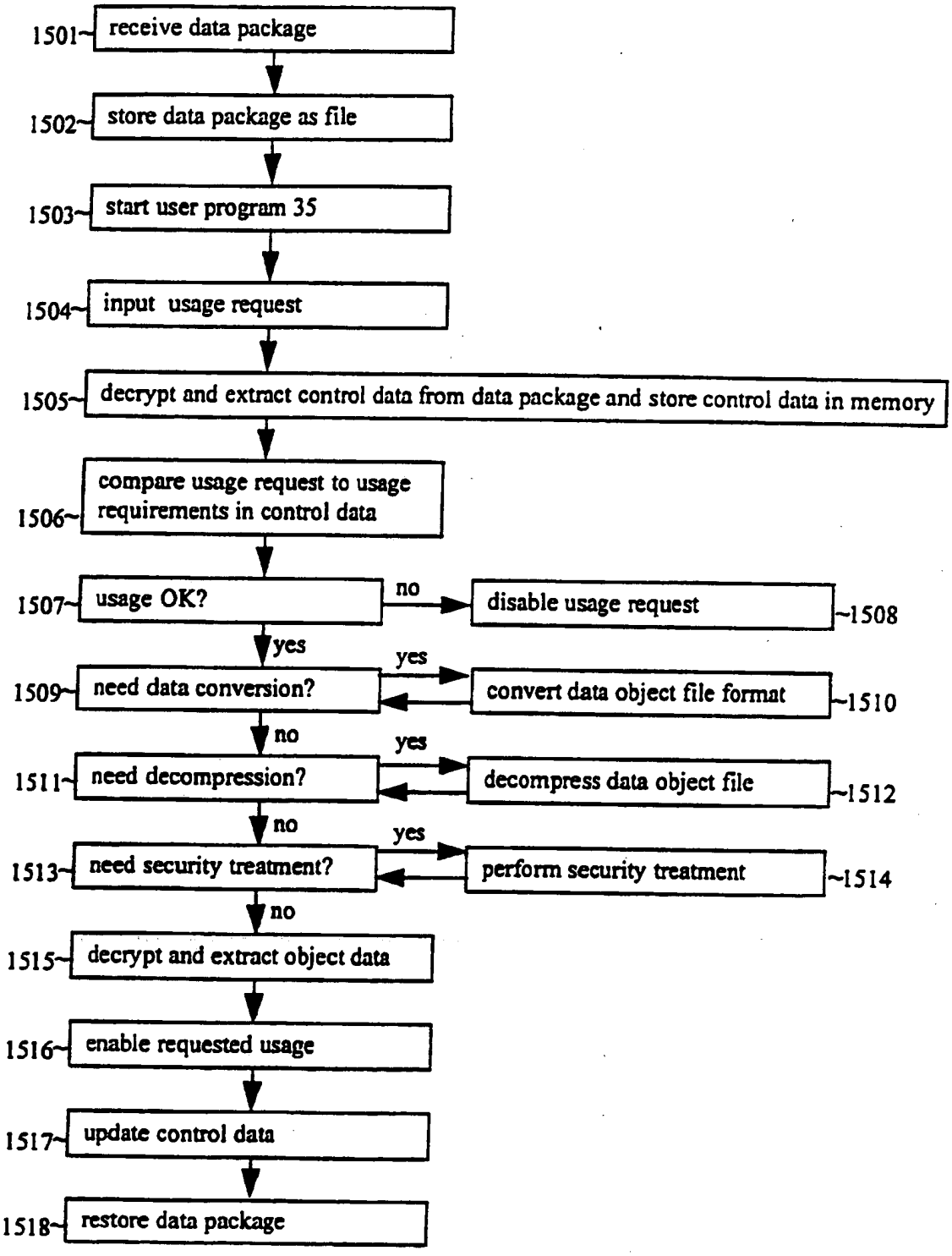


Fig 16

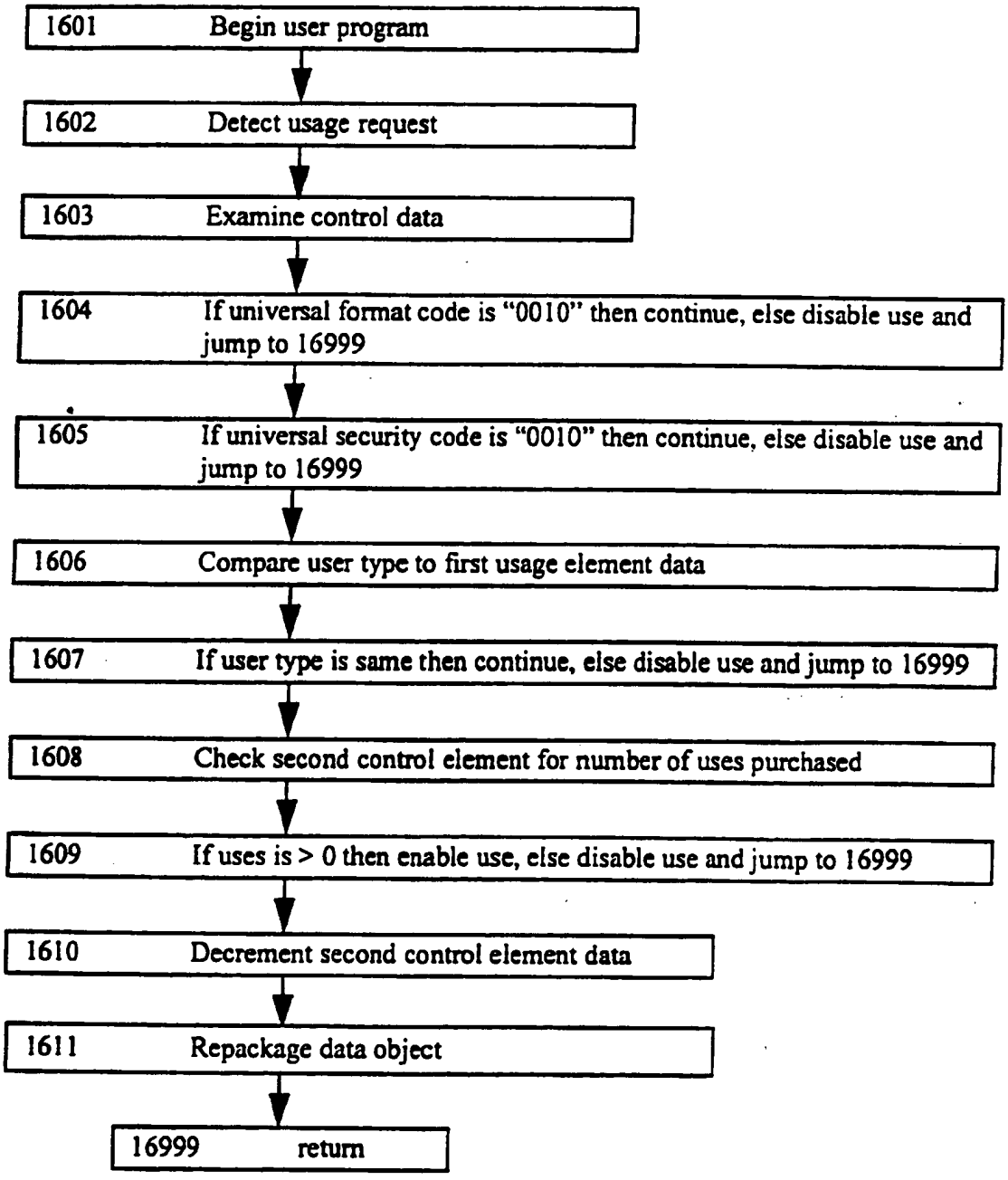
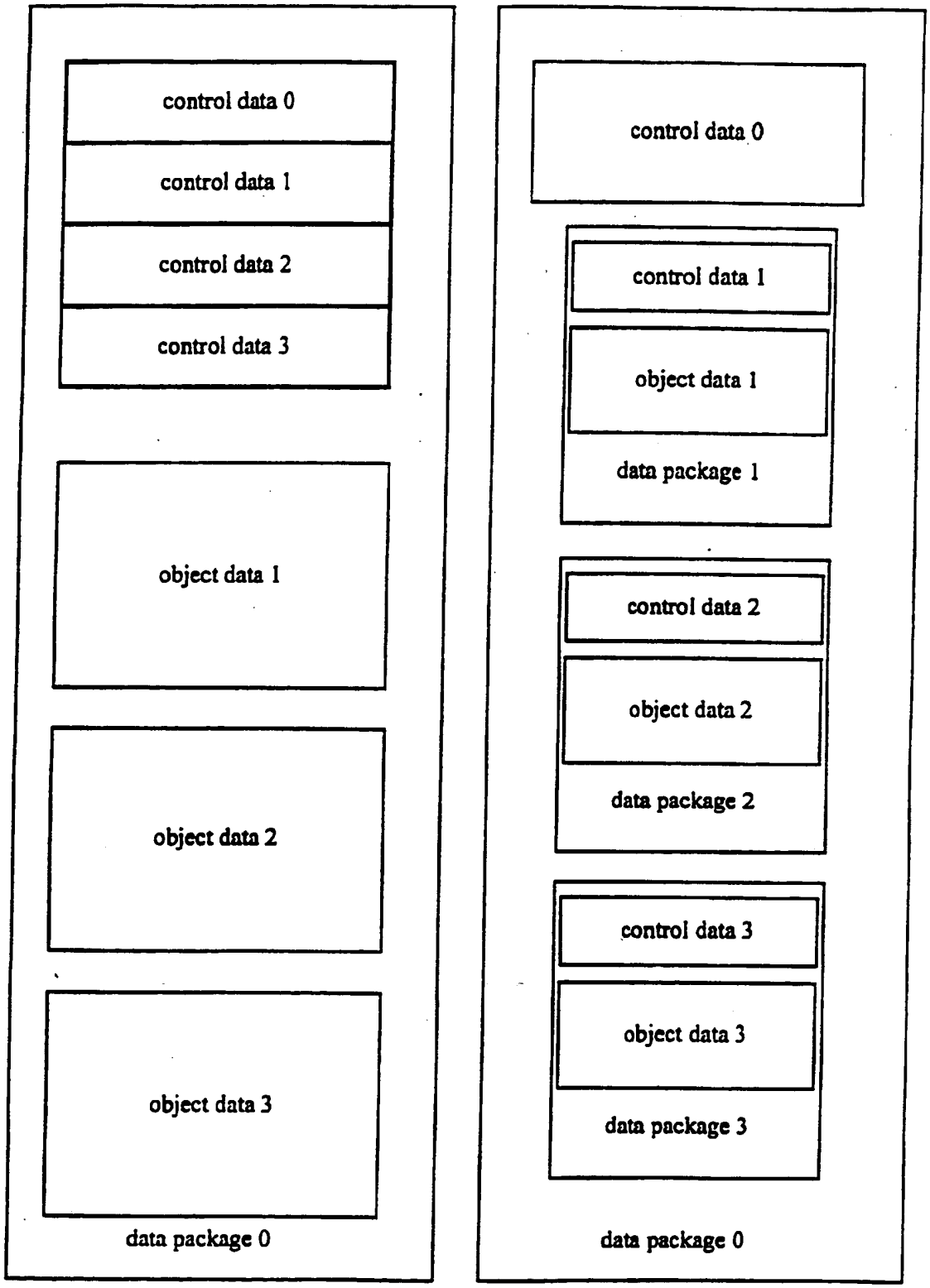


Fig 17



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).

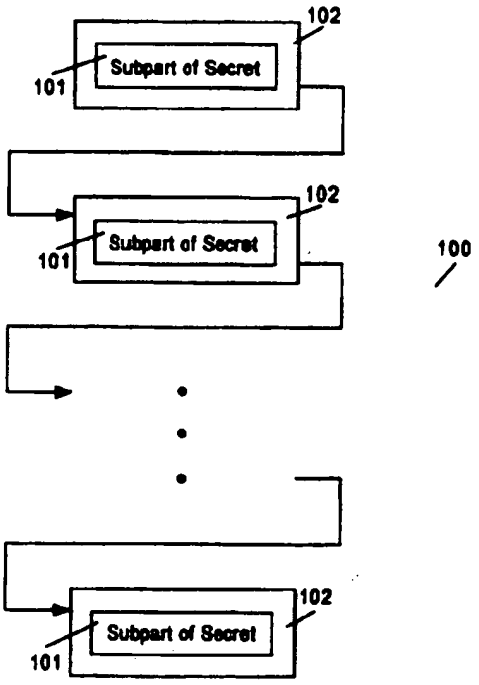
(81) Designated States: AL, AM, AT, AT (Utility model), AU (Petty patent), AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), EE, EE (Utility model), ES, FI, FI (Utility model), GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, 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.

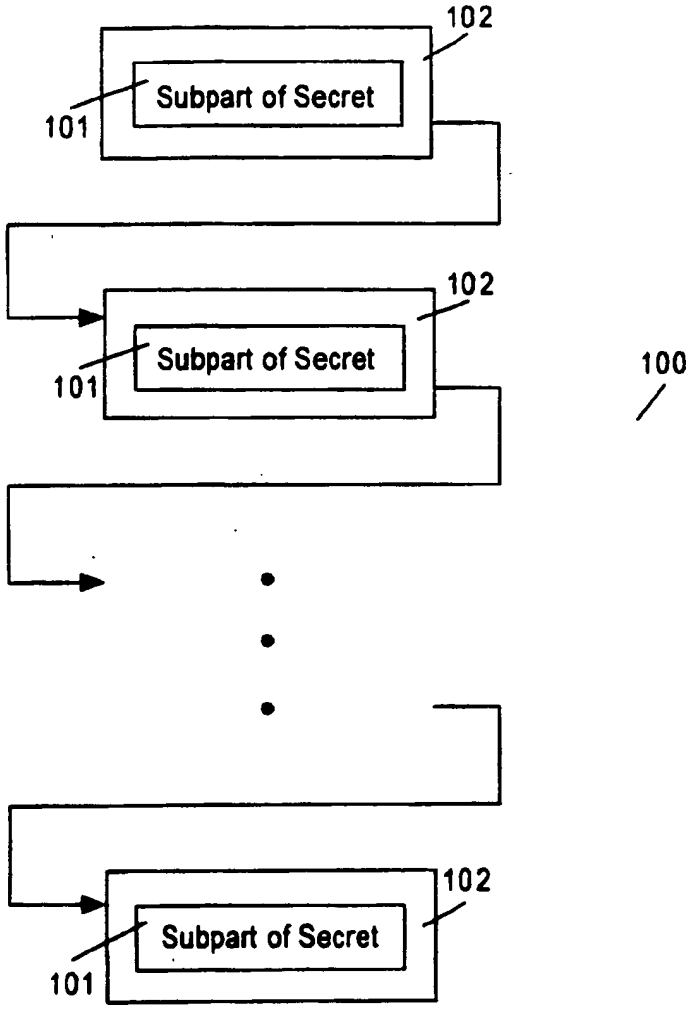

102
Subpart of Secret
101
102
Subpart of Secret
101
100
102
Subpart of Secret
101

Figure 1

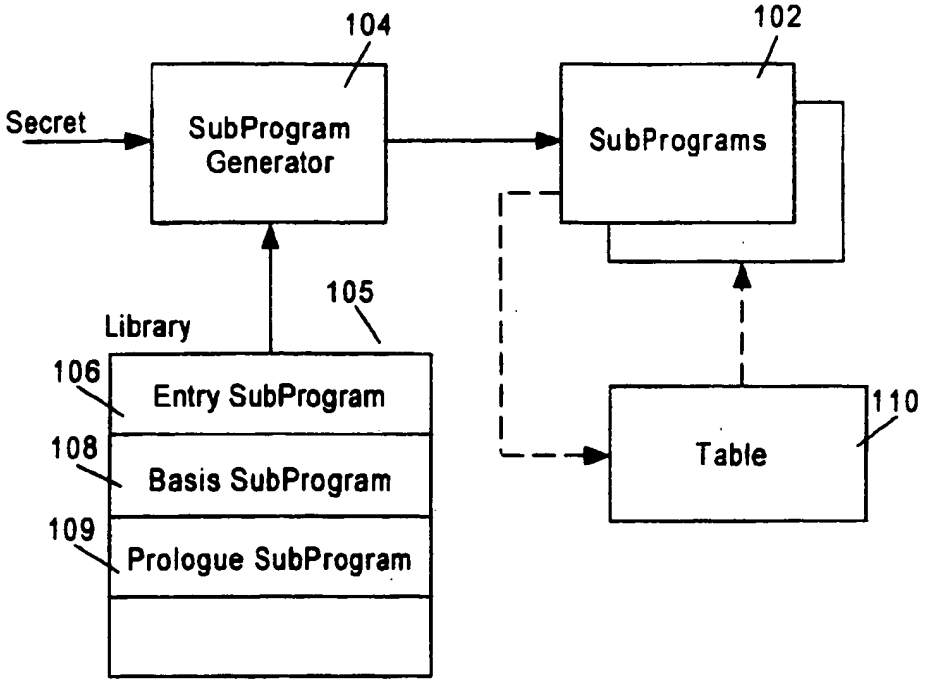

104
102
Secret
SubProgram Generator
SubPrograms
105
Library
106
Entry SubProgram
108
Basis SubProgram
109
Prologue SubProgram
110
Table

Figure 2

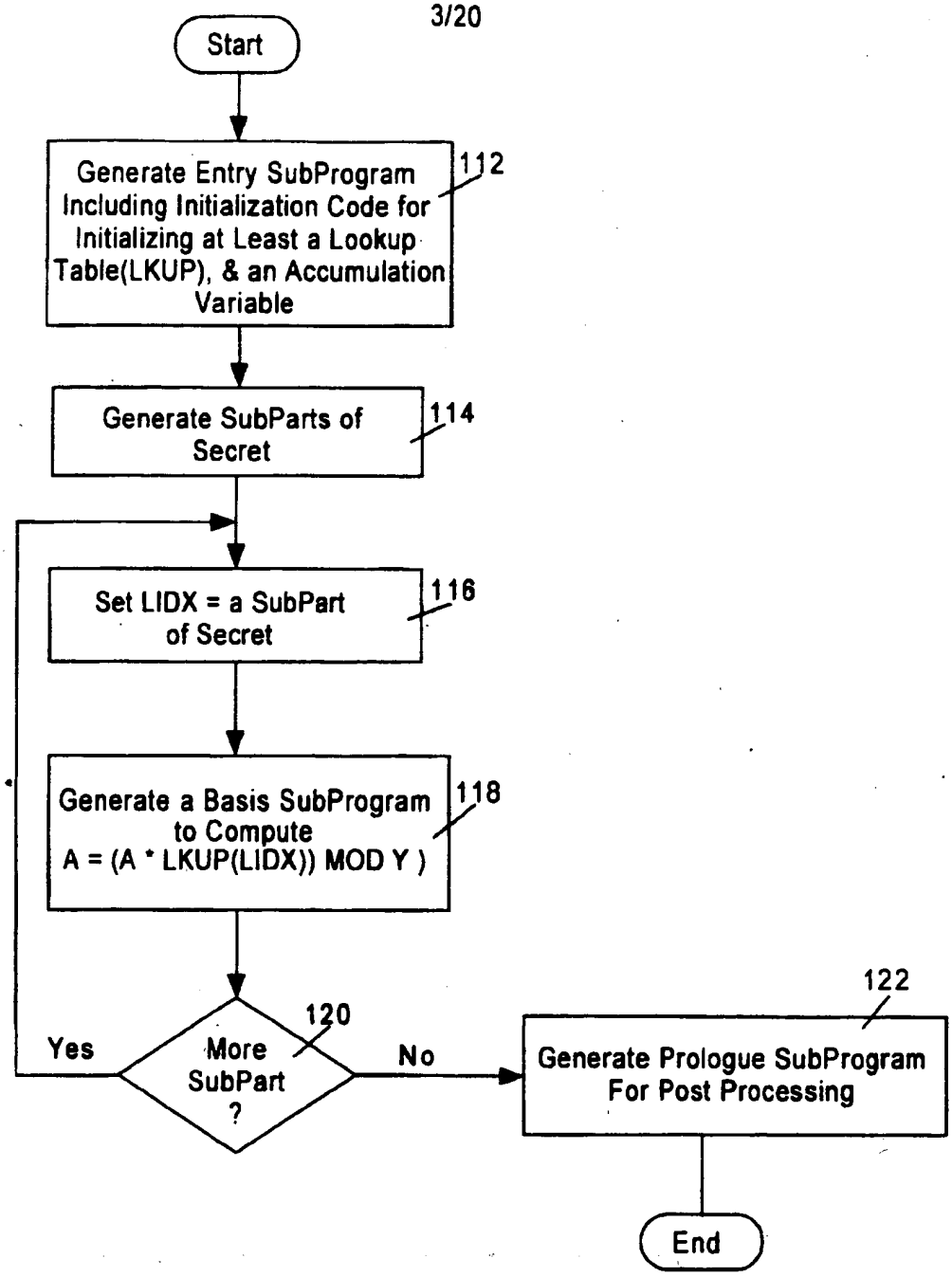

Start
Generate Entry SubProgram Including Initialization Code for Initializing at Least a Lookup Table(LKUP), & an Accumulation Variable
112
Generate SubParts of Secret
114
Set LIDX = a SubPart of Secret
116
Generate a Basis SubProgram to Compute A = (A * LKUP(LIDX)) MOD Y)
118
More SubPart ?
120
Yes
No
Generate Prologue SubProgram For Post Processing
122
End

Figure 3

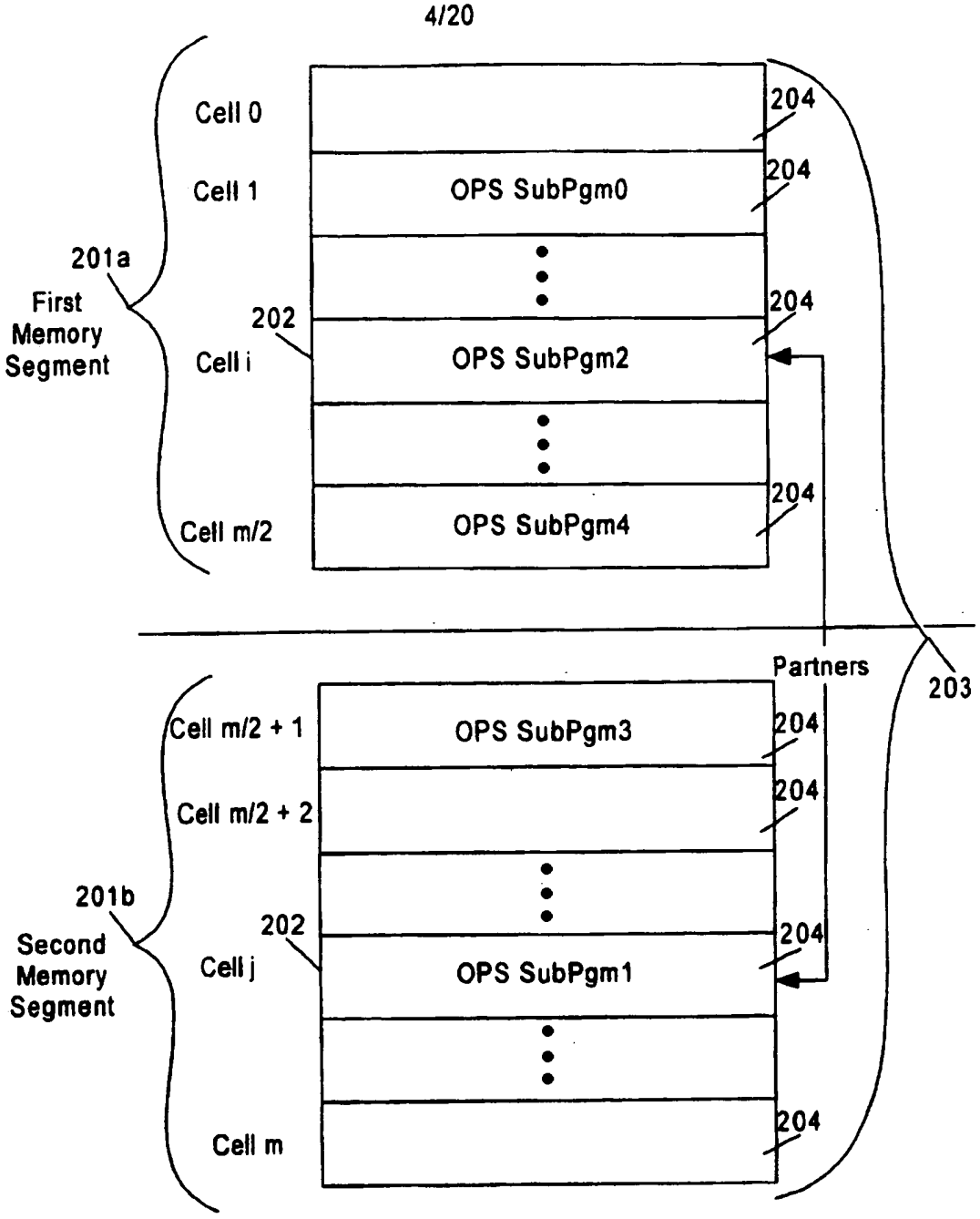

Cell 0
Cell 1
OPS SubPgm0
201a
First Memory Segment
202
Cell i
OPS SubPgm2
Cell m/2
OPS SubPgm4
204
Partners
203
Cell m/2 + 1
OPS SubPgm3
Cell m/2 + 2
201b
Second Memory Segment
Cell j
OPS SubPgm1
Cell m

Figure 4

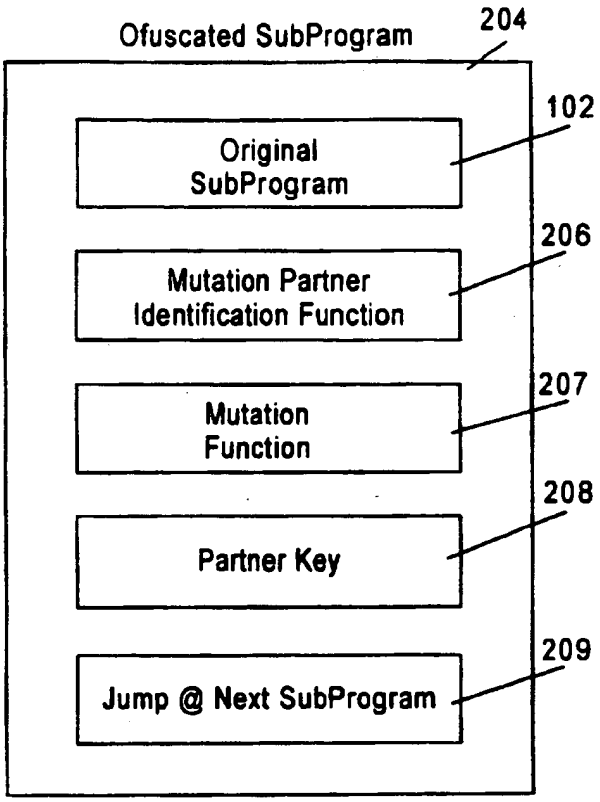

Ofuscated SubProgram
204
Original SubProgram
102
Mutation Partner Identification Function
206
Mutation Function
207
Partner Key
208
Jump @ Next SubProgram
209

Figure 5

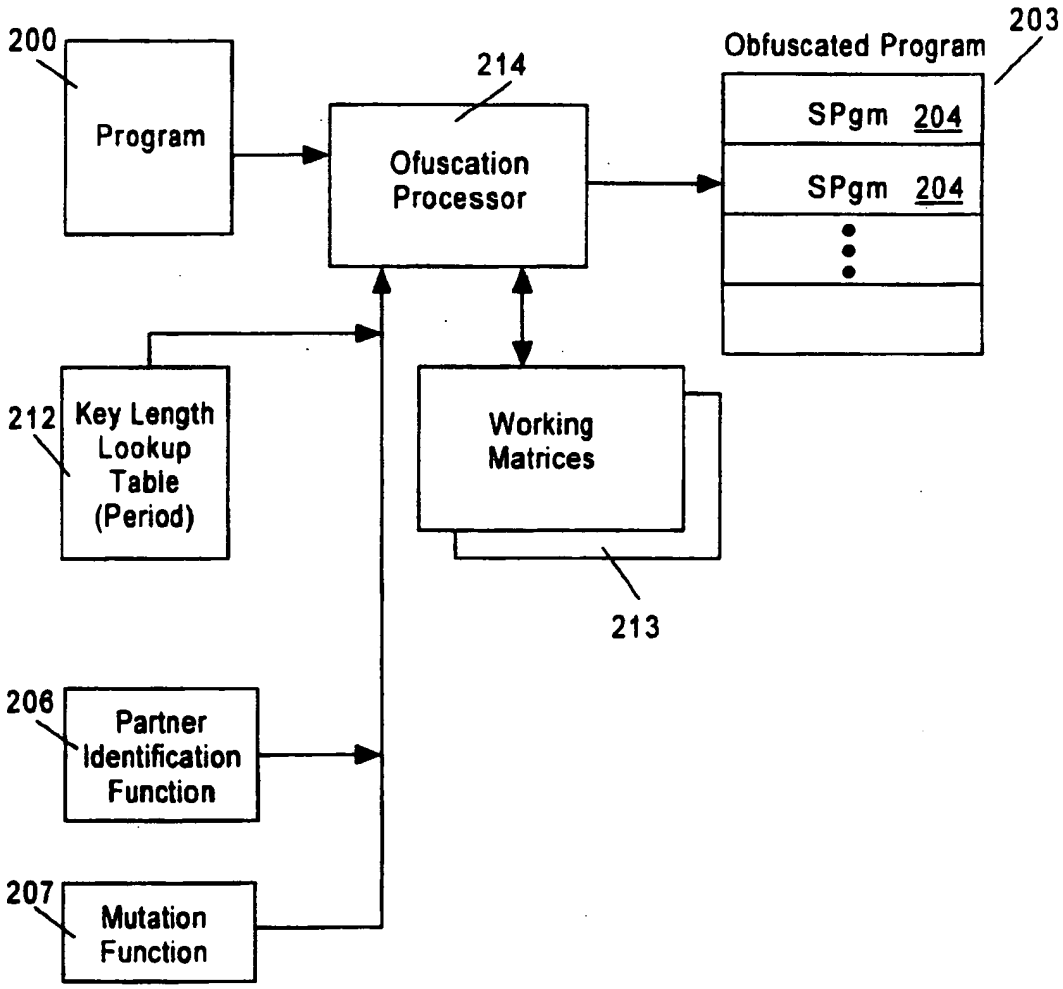

200
Program
214
Ofuscation Processor
Obfuscated Program
203
SPgm 204
SPgm 204
212
Key Length Lookup Table (Period)
Working Matrices
213
206
Partner Identification Function
207
Mutation Function

Figure 6

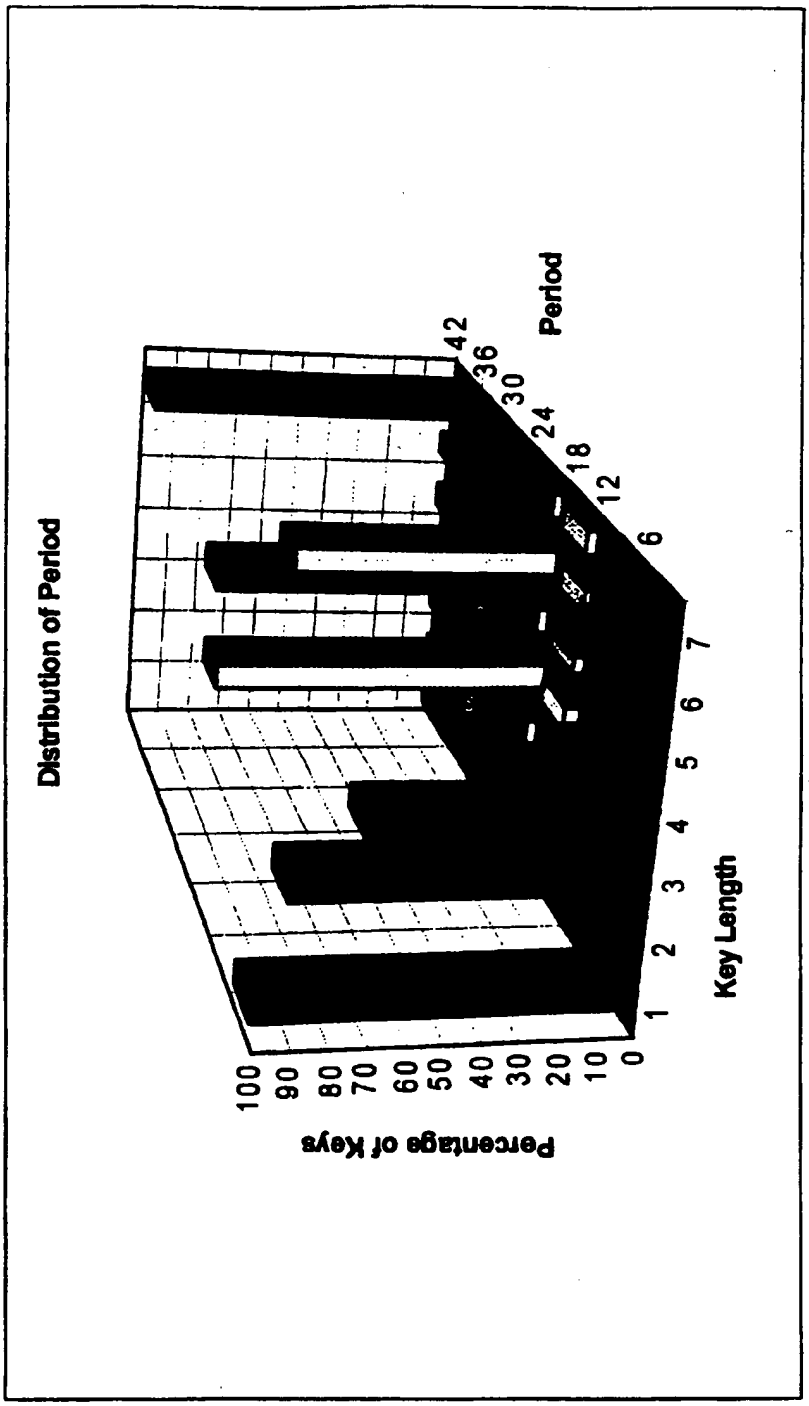

Distribution of Key Period
Distribution of Period
Period
42
36
30
24
18
12
6
Key Length
1
2
3
4
5
6
7
Percentage of Keys
100
90
80
70
60
50
40
30
20
10
0

Figure 7

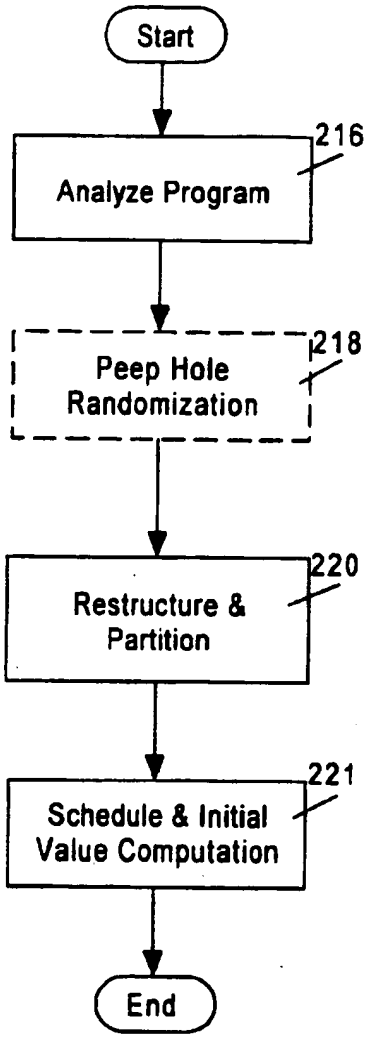

Start
Analyze Program
216
Peep Hole Randomization
218
Restructure & Partition
220
Schedule & Initial Value Computation
221
End

Figure 8a

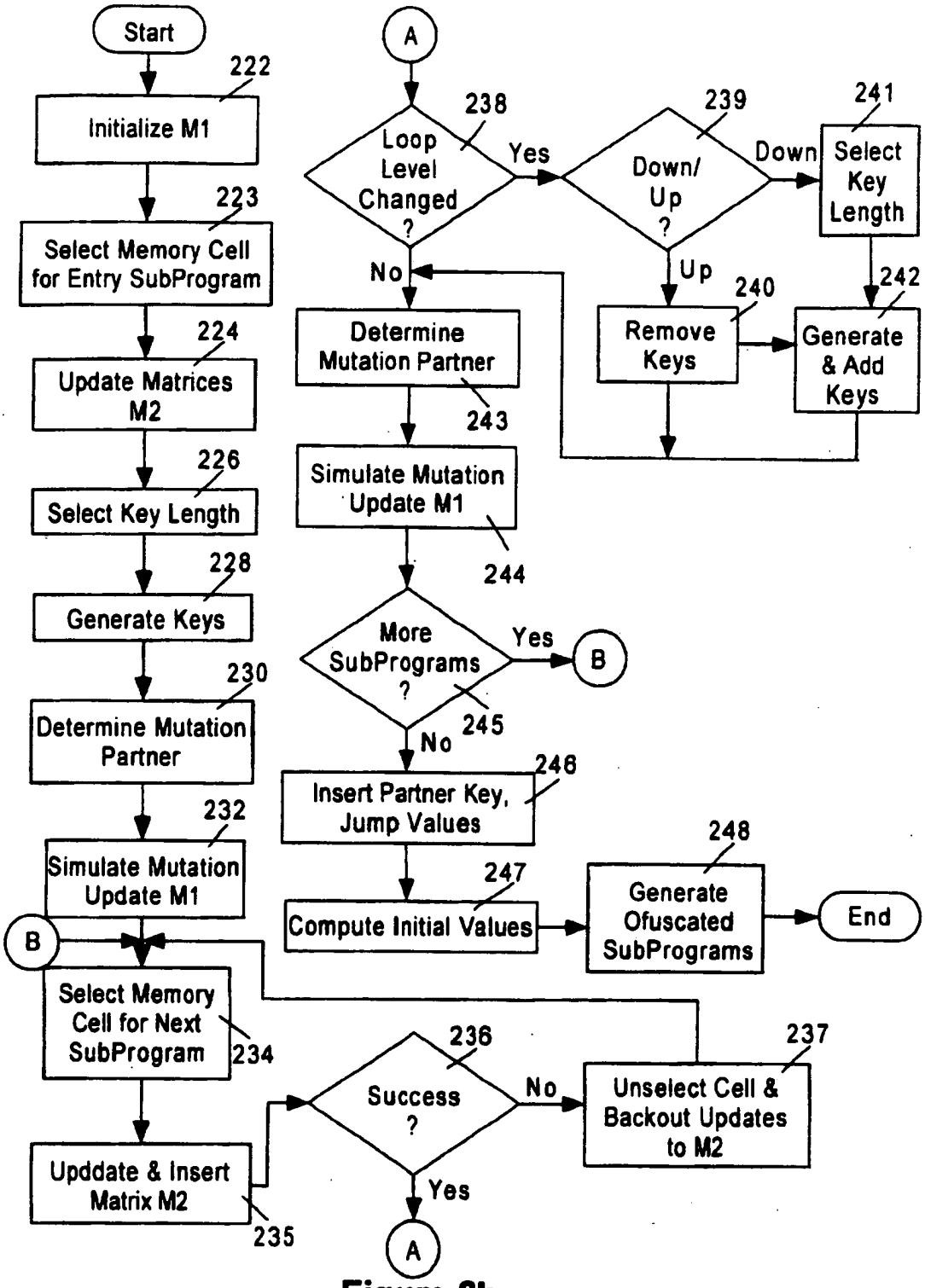

Start
222
Initialize M1
223
Select Memory Cell for Entry SubProgram
224
Update Matrices M2
226
Select Key Length
228
Generate Keys
230
Determine Mutation Partner
232
Simulate Mutation Update M1
B
Select Memory Cell for Next SubProgram
234
Upddate & Insert Matrix M2
235
A
238
Loop Level Changed ?
Yes
239
Down/ Up ?
Down
241
Select Key Length
No
Up
240
Remove Keys
242
Generate & Add Keys
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

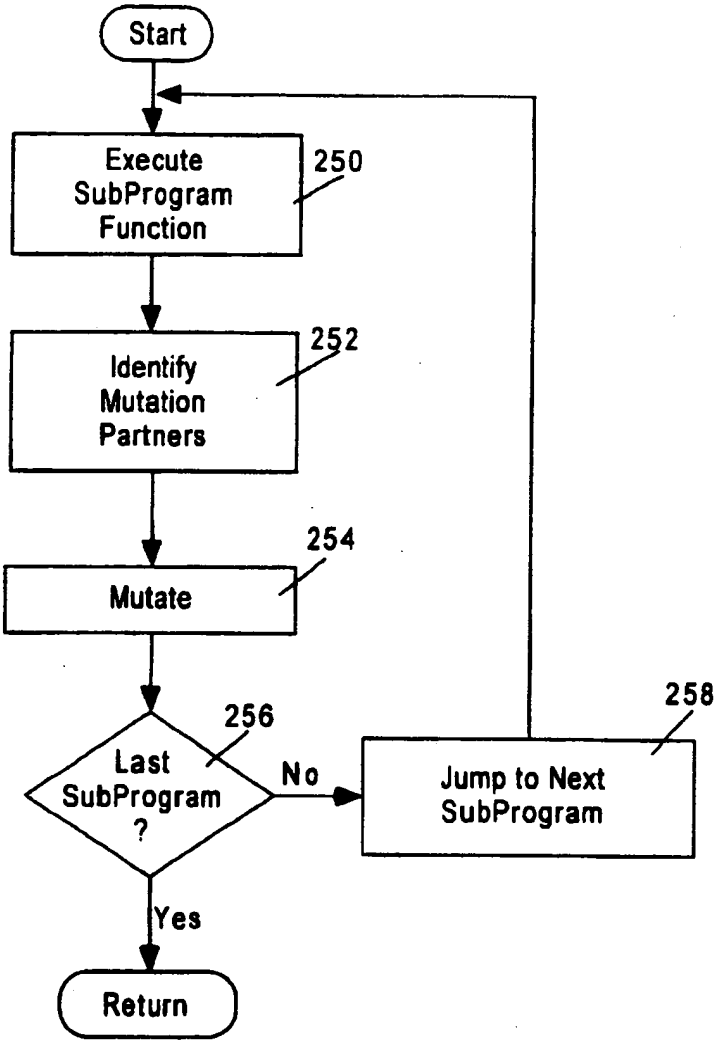

Start
Execute SubProgram Function
250
Identify Mutation Partners
252
Mutate
254
Last SubProgram ?
256
No
Jump to Next SubProgram
258
Yes
Return

Figure 9

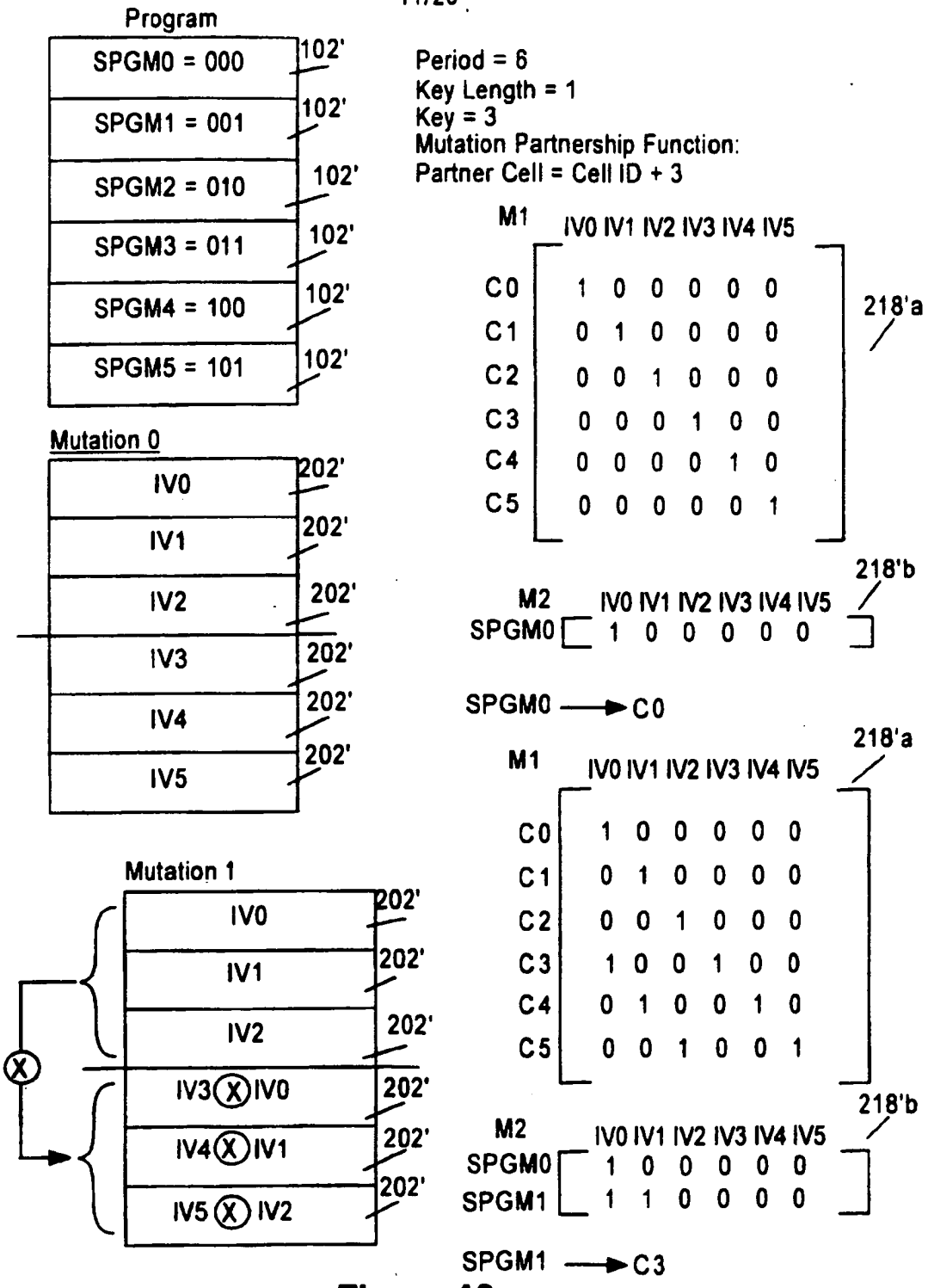

Program
SPGM0 = 000
SPGM1 = 001
SPGM2 = 010
SPGM3 = 011
SPGM4 = 100
SPGM5 = 101
102'
Mutation 0
IV0
IV1
IV2
IV3
IV4
IV5
202'
Mutation 1
IV3 X IV0
IV4 X IV1
IV5 X IV2
Period = 6
Key Length = 1
Key = 3
Mutation Partnership Function:
Partner Cell = Cell ID + 3
M1
IV0 IV1 IV2 IV3 IV4 IV5
C0 1 0 0 0 0 0
C1 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
M1
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
M2
SPGM0 1 0 0 0 0 0
SPGM1 1 1 0 0 0 0
SPGM1 → C3

Figure 10

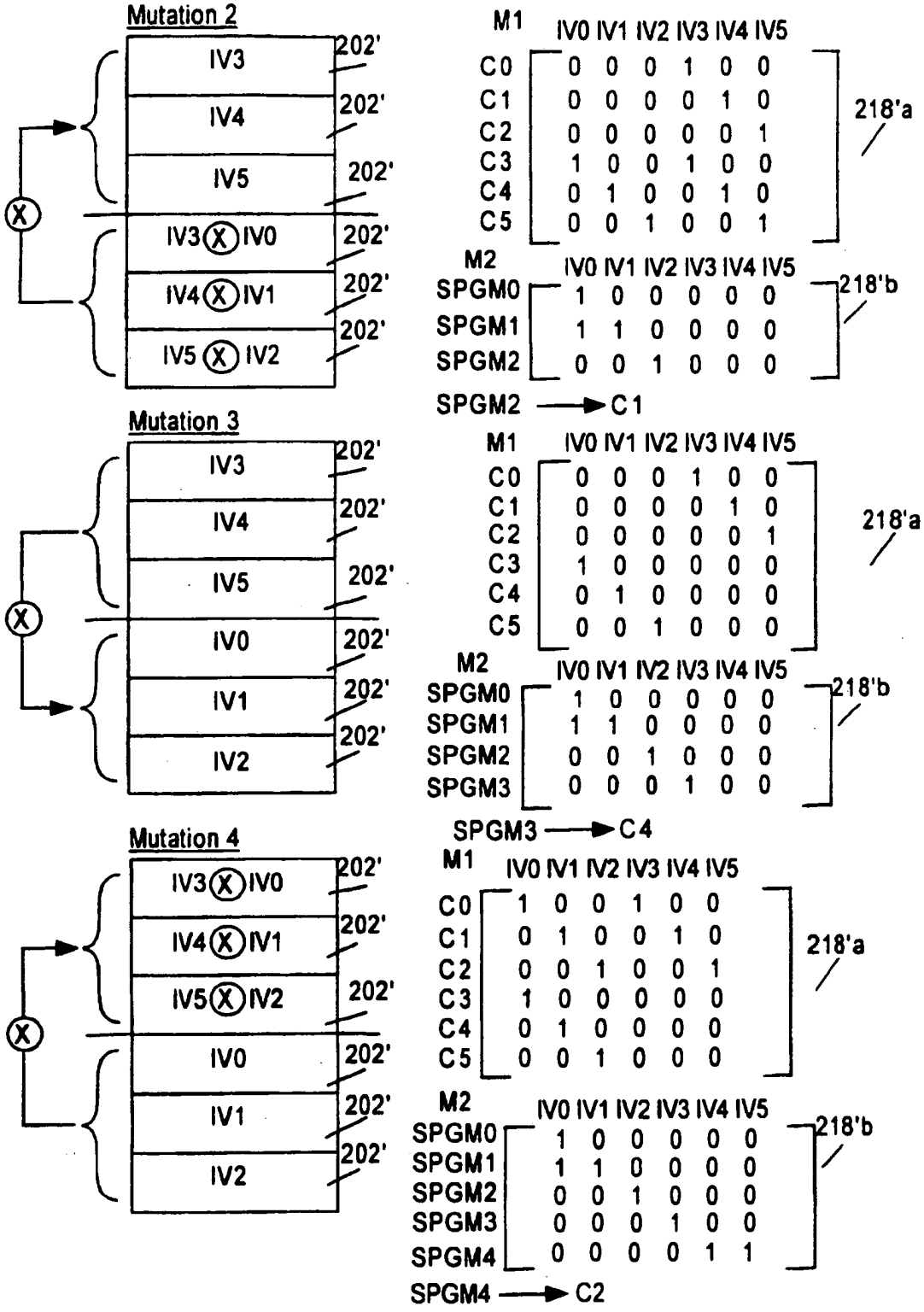

Mutation 2
IV3
IV4
IV5
IV3 X IV0
IV4 X IV1
IV5 X IV2
202'
M1
IV0 IV1 IV2 IV3 IV4 IV5
C0 0 0 0 1 0 0
C1 0 0 0 0 1 0
C2 0 0 0 0 0 1
C3 1 0 0 1 0 0
C4 0 1 0 0 1 0
C5 0 0 1 0 0 1
218'a
M2
IV0 IV1 IV2 IV3 IV4 IV5
SPGM0 1 0 0 0 0 0
SPGM1 1 1 0 0 0 0
SPGM2 0 0 1 0 0 0
218'b
SPGM2 → C1
Mutation 3
IV3
IV4
IV5
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

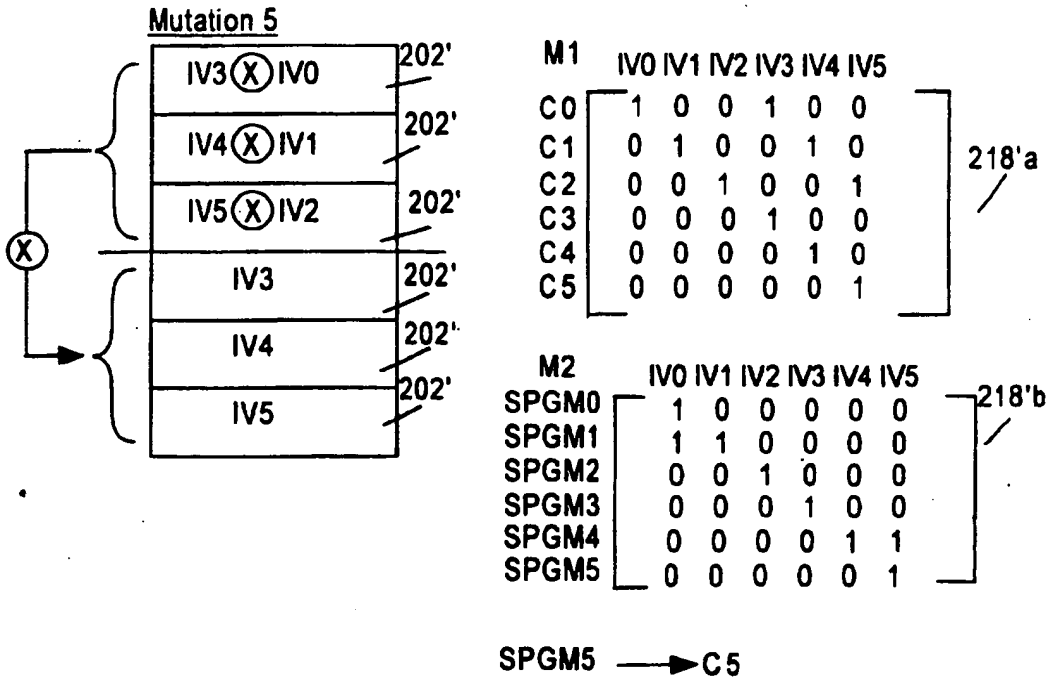

Mutation 5
IV3 X IV0
IV4 X IV1
IV5 X IV2
IV3
IV4
IV5
202'
M1
IV0 IV1 IV2 IV3 IV4 IV5
C0 1 0 0 1 0 0
C1 0 1 0 0 1 0
C2 0 0 1 0 0 1
C3 0 0 0 1 0 0
C4 0 0 0 0 1 0
C5 0 0 0 0 0 1
218'a
M2
IV0 IV1 IV2 IV3 IV4 IV5
SPGM0 1 0 0 0 0 0
SPGM1 1 1 0 0 0 0
SPGM2 0 0 1 0 0 0
SPGM3 0 0 0 1 0 0
SPGM4 0 0 0 0 1 1
SPGM5 0 0 0 0 0 1
218'b
SPGM5 → C5

Figure 12

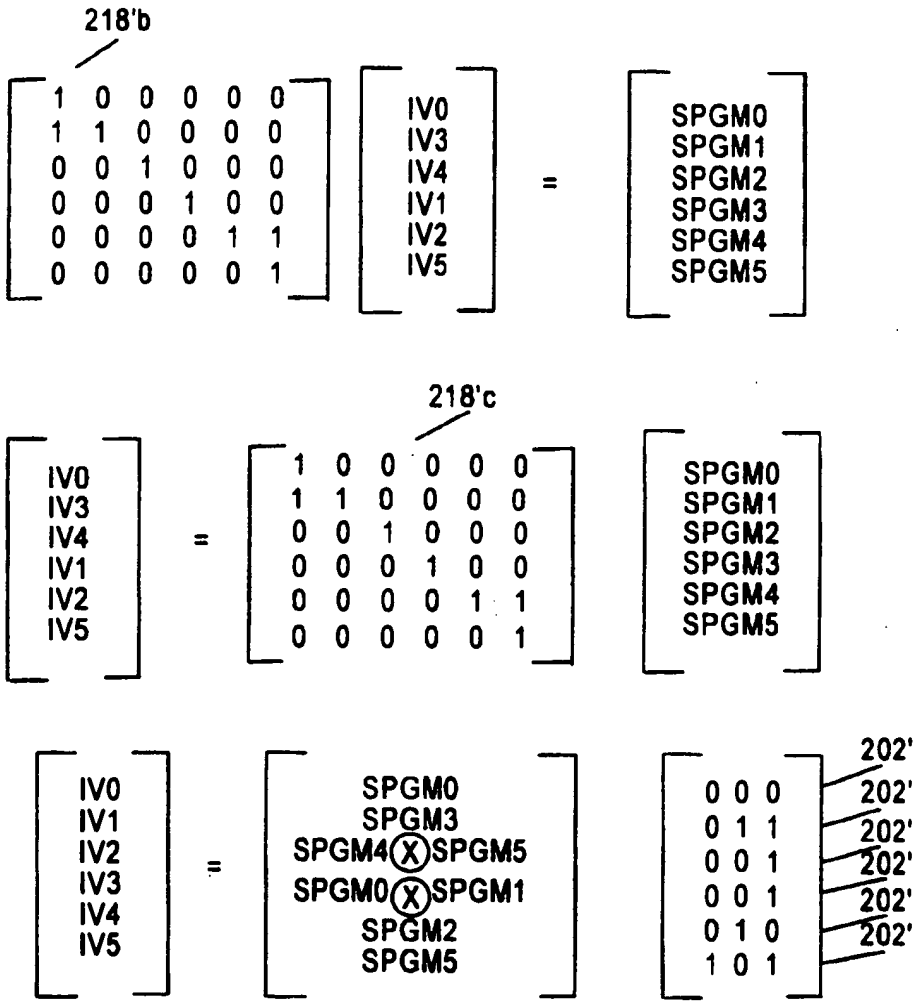

218'b
1 0 0 0 0 0
1 1 0 0 0 0
0 0 1 0 0 0
0 0 0 1 0 0
0 0 0 0 1 1
0 0 0 0 0 1
IV0 IV3 IV4 IV1 IV2 IV5
=
SPGM0 SPGM1 SPGM2 SPGM3 SPGM4 SPGM5
218'c
IV0 IV3 IV4 IV1 IV2 IV5
=
SPGM0 SPGM1 SPGM2 SPGM3 SPGM4 SPGM5
IV0 IV1 IV2 IV3 IV4 IV5
=
SPGM0
SPGM3
SPGM4 X SPGM5
SPGM0 X SPGM1
SPGM2
SPGM5
0 0 0
0 1 1
0 0 1
0 0 1
0 1 0
1 0 1
202'
202'
202'
202'
202'
202'

Figure 13

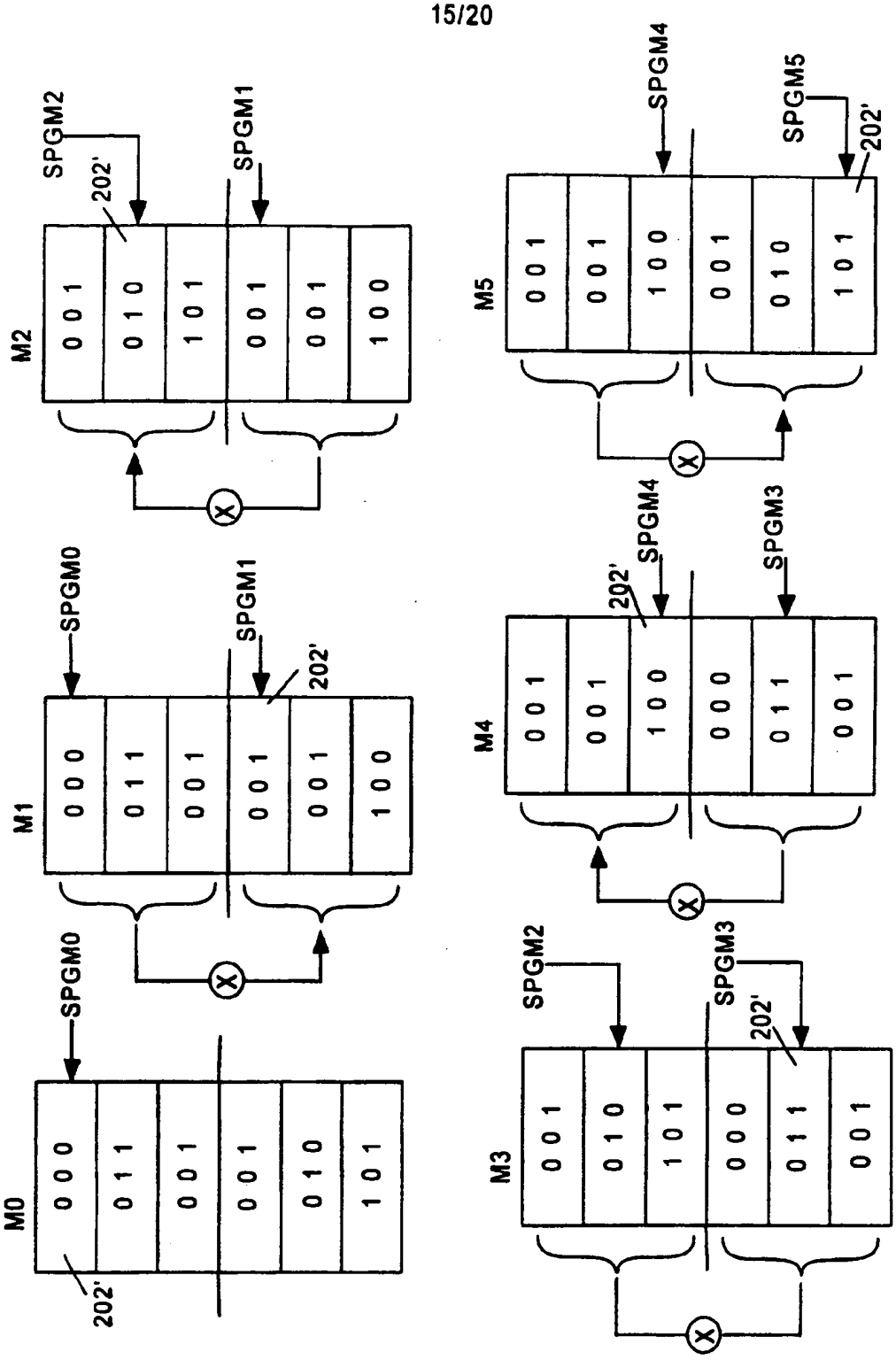

M0
M1
M2
M3
M4
M5
SPGM0
SPGM1
SPGM2
SPGM3
SPGM4
SPGM5
202'
0 0 0
0 1 1
0 0 1
0 1 0
1 0 0
1 0 1

Figure 14

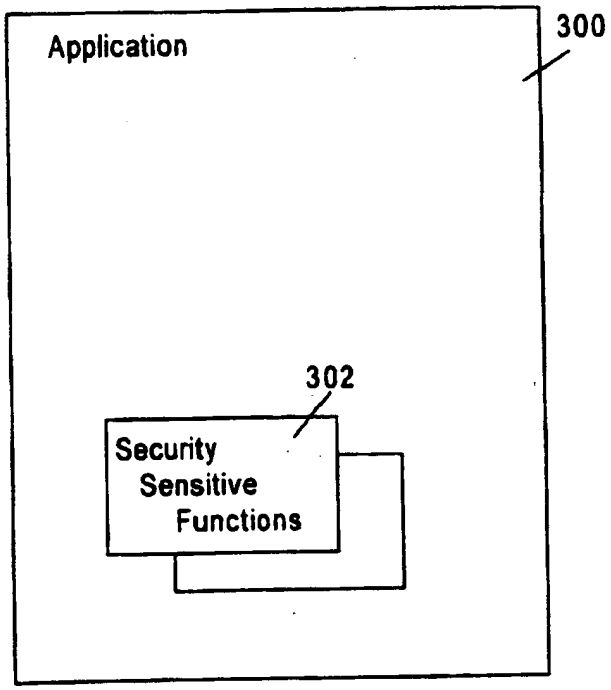


Figure 15

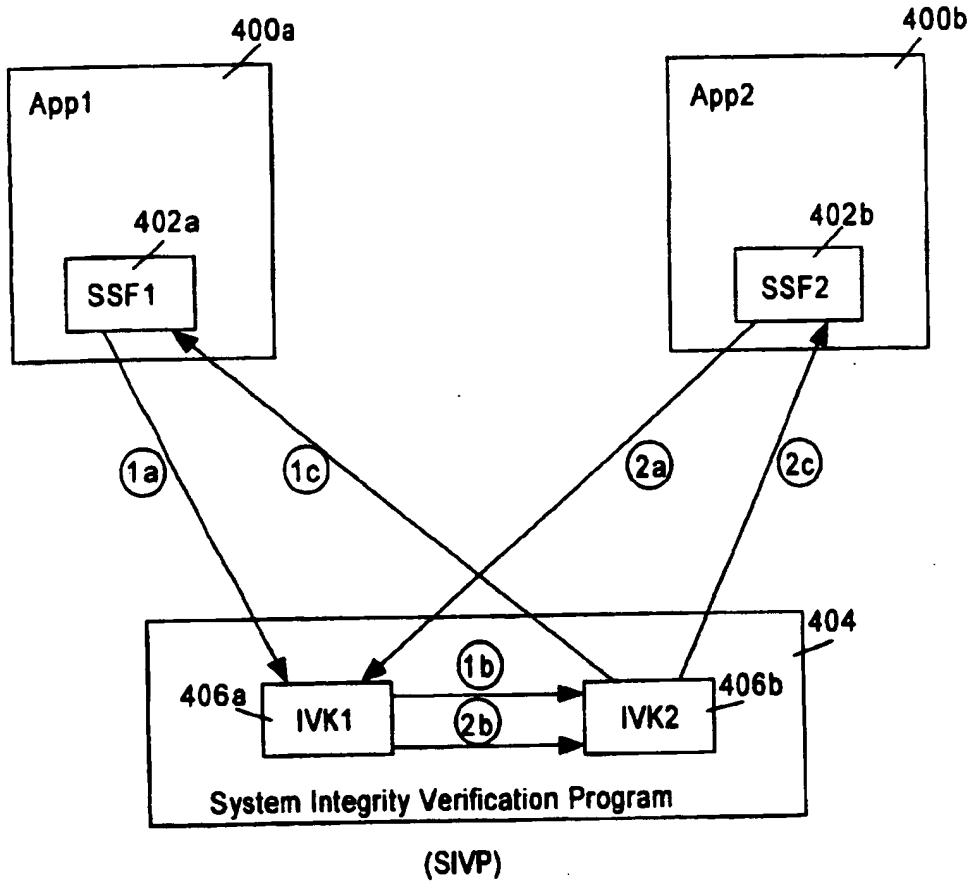

400a
App1
402a
SSF1
400b
App2
402b
SSF2
1a
1c
2a
2c
404
1b
406a
IVK1
2b
IVK2
406b
System Integrity Verification Program
(SIVP)

Figure 16

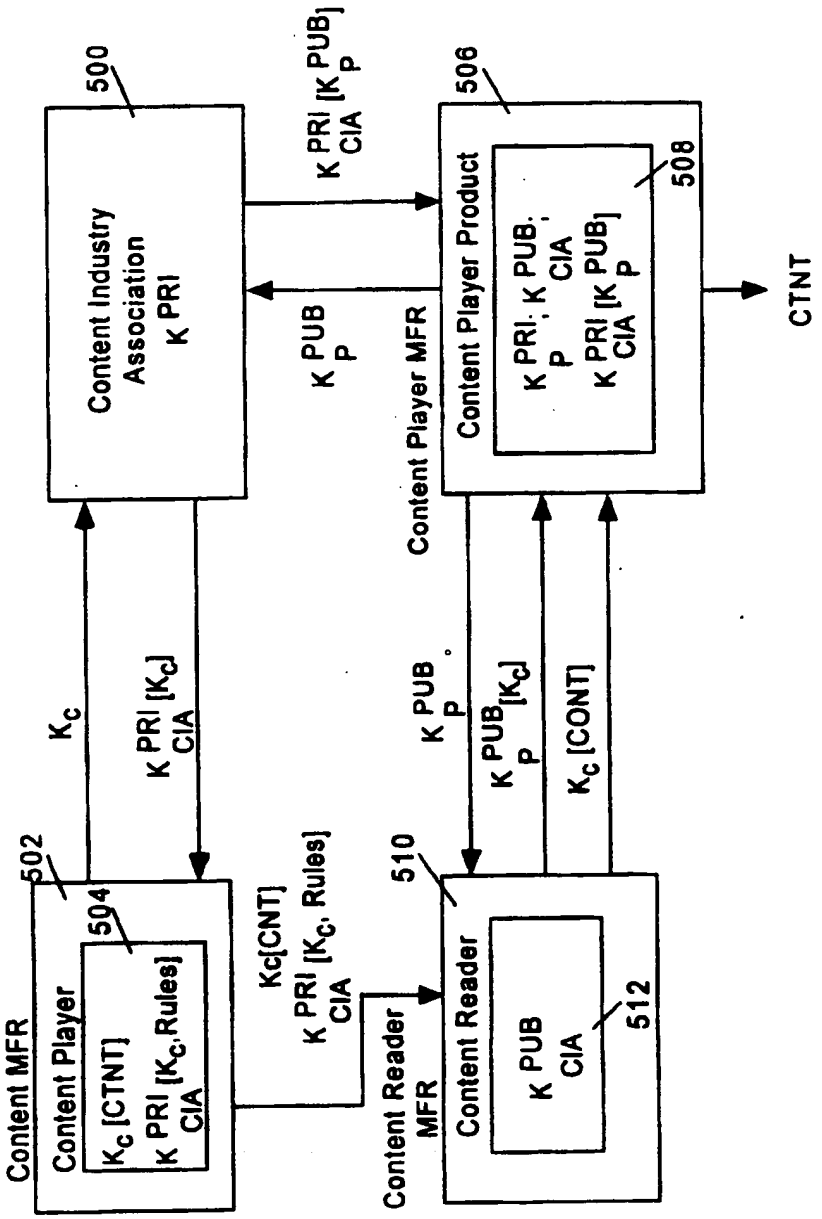

500
Content Industry Association
K PRI
K PRI CIA [K PUB P]
K PUB P
506
Content Player Product
508
K PRI P, K PUB CIA
K PRI CIA [K PUB P]
Content Player MFR
CTNT
K c
K PRI CIA [K c]
K PUB P
K PUB P [K c]
K c [CONT]
502
504
Content Player
K c [CTNT]
K PRI CIA [K c, Rules]
Content MFR
Kc[CNT]
K PRI CIA [K c, Rules]
510
Content Reader
512
K PUB CIA
Content Reader MFR

Figure 17

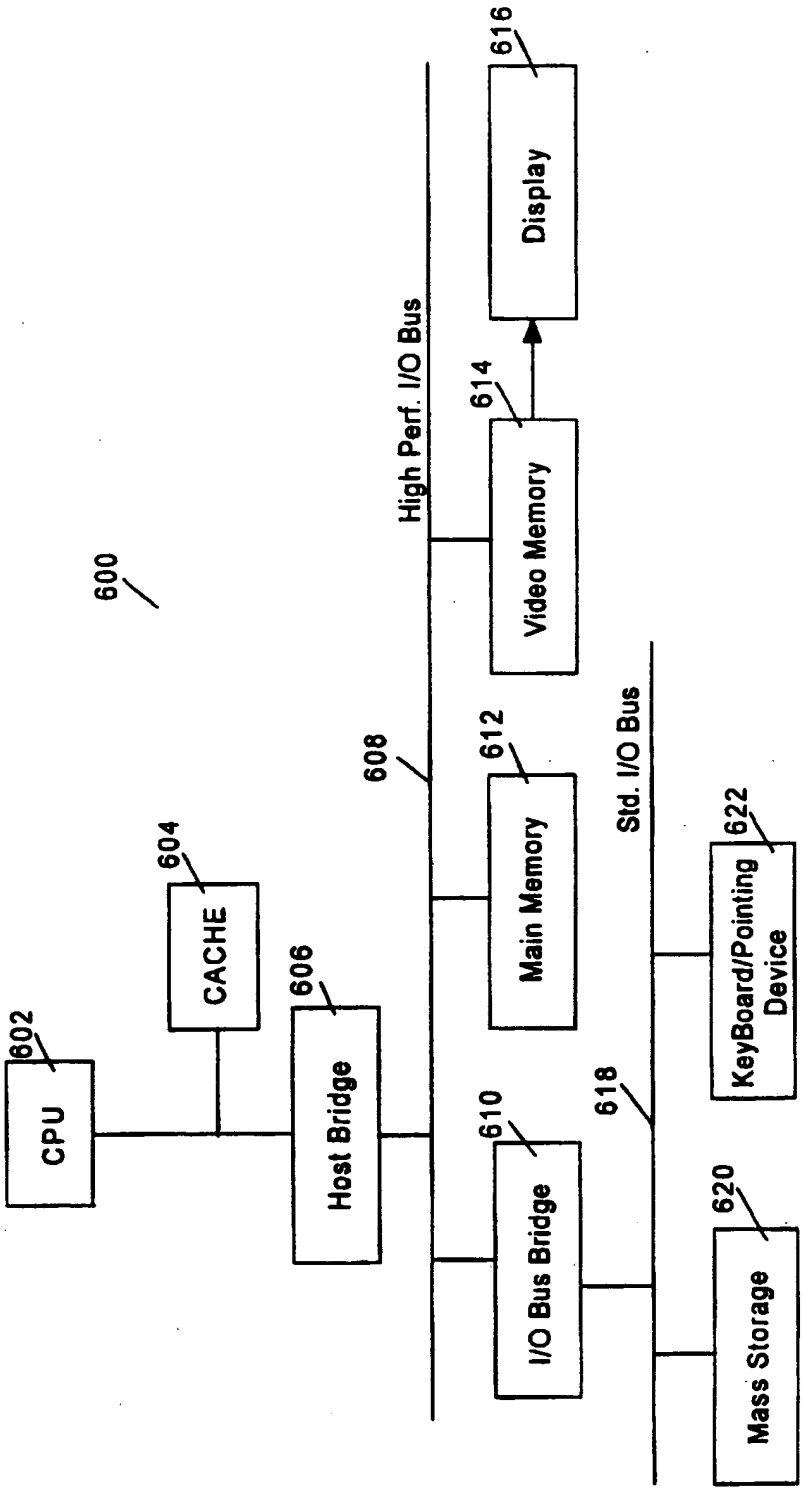

600
CPU
602
CACHE
604
Host Bridge
606
608
High Perf. I/O Bus
Main Memory
612
Video Memory
614
Display
616
I/O Bus Bridge
610
618
Std. I/O Bus
KeyBoard/Pointing Device
622
Mass Storage
620

Figure 18

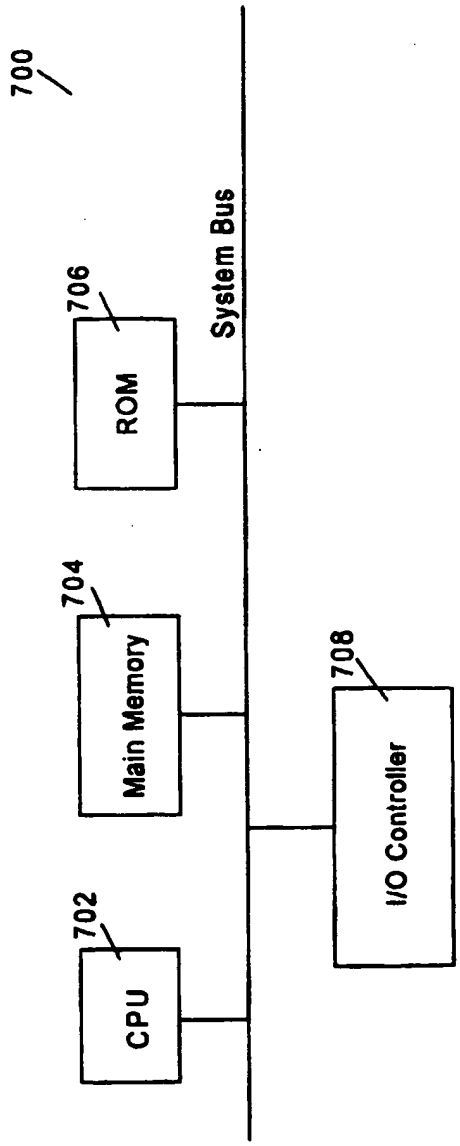

700
System Bus
706
ROM
704
Main Memory
708
I/O Controller
702
CPU

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:
"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
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)*

3621

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	10/162,212
Filing Date	June 5, 2002
First Named Inventor	WANG et al.
Group Art Unit	3621
Examiner Name	unknown
Total Number of Pages in This Submission	
Attorney Docket Number	111325-104

ENCLOSURES *(check all that apply)*

- ☐ Fee Transmittal Form
  - ☐ 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)
- ☐ Response to Missing Parts/ Incomplete Application
  - ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Assignment Papers *(for an Application)*
- ☐ Drawing(s)
- ☐ 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
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to Group (Appeal Notice, Brief, Reply Brief)
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Application Data Sheet
- ☐ Other Enclosure(s) *(please identify below)*:

Remarks

☒ 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	Marc S. Kaufman, Reg. No. 35,212 Nixon Peabody LLP 8180 Greensboro Drive Suite 800 McLean, VA 22102
Signature	
Date	February 12, 2003

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.

NVA255219.1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:	)	Confirmation No. 3700
Xin WANG et al.	)	Examiner: unknown
Serial No. 10/162,212	)	Group Art Unit: 3621
Filed: June 5, 2002	)	Attorney Docket No. 111325-104
For: ***Rights Offering And Granting***	)	

INFORMATION DISCLOSURE STATEMENT

Commissioner for Patents
Washington, DC 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 undersigned certifies that each item of information contained in this information disclosure statement was first cited in a communication from a foreign patent office in connection with a counterpart foreign application not more than three (3) months prior to the filing of this statement.

It is requested that the accompanying PTO-1449 be considered and made of record in the above-identified 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 (111325-104).

Respectfully submitted,

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

MSK:dkt
NIXON PEABODY LLP
8180 Greensboro Drive, Suite 800
McLean, Virginia 22102

Telephone: (703) 770-9300

NVA255212.1

PTO/SB/08A (10-01)
Approved for use through 10/31/2002. OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(use as many sheets as necessary)

Sheet 1 of 1

Complete if Known	
Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin WANG et al.
Art Unit	3621
Examiner Name	Unknown
Attorney Docket Number	111325-104

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	U.S. Patent Document Number - Kind Code[2] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US-5,758,069	05/26/1998	Olsen	
		US-6,169,976 B1	01/02/2001	Colosso	
		US-6,236,971 B1	05/22/2001	Stefik et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3] Number[4]	Kind Code[5] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Application of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]

OTHER PRIOR ART – NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.)., date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		PCT International Search Report, date of mailing January 14, 2003 (PCT/US02/17662)	

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference 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 form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

[1] Applicant's unique citation designation number (optional). [2] Applicant is to place a check mark here if English language Translation is attached.

Burden Hour Statement: This form is estimated to take 2.0 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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:	)	
Xin WANG et al.	)	Examiner: Mary Cheung
Serial No. 10/162,212	)	Group Art Unit: 3621
Filed: June 5, 2002	)	Confirmation No. 3700
For: RIGHTS OFFERING AND GRANTING	)	

INFORMATION DISCLOSURE STATEMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

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 undersigned certifies that either (1) each item of information contained in this information disclosure statement was first cited in a communication from a foreign patent office in connection with a counterpart foreign application not more than three (3) months prior to the filing of this statement, or (2) no item of information contained in this information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application and to my knowledge after making reasonable inquiry, was known to any individual designated in 37 C.F.R. § 1.56(c) more than three months prior to the filing of this statement.

Enclosed is a copy of the International Search Report dated February 11, 2004.

It is requested that the accompanying PTO-1449 be considered and made of record in the above-identified 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. (111325-104)

Respectfully submitted,

By: _____

Marc S. Kaufman
Registration No. 35,212

NIXON PEABODY LLP
Suite 900
401 9th Street, N.W.
Washington, DC 20004-2128

Telephone: (202) 585-8000

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(use as many sheets as necessary)

Sheet 1 of 1

Complete if Known	
Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin WANG
Art Unit	3621
Examiner Name	Mary Cheung
Attorney Docket Number	111325-104

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	U.S. Patent Document Number - Kind Code[2] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US-5,794,207	08-11-1998	Walker et al.	
		US-			
		US-			
		US-			

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3] Number[4]	Kind Code[5] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Application of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]

OTHER PRIOR ART – NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		INTERNATIONAL SEARCH REPORT DATED FEBRUARY 11, 2004	

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference 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 form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

[1] Applicant's unique citation designation number (optional). [2] Applicant is to place a check mark here if English language Translation is attached.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: Xin **WANG**, *et al.*	Examiner: Unassigned
Application No.: 10/162,212	Group Art Unit: 3621
Filed: June 5, 2002	
For: **RIGHTS OFFERING AND GRANTING**	

Commissioner of Patents
U.S. Patent and Trademark Office
220 20th Street S.
Customer Window
Crystal Plaza Two, Lobby, Room 1B03
Arlington, VA 22202

Sir:

INFORMATION DISCLOSURE STATEMENT UNDER 37 C.F.R. § 1.97 (b)

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 listed documents were cited in a communication from the European Patent Office in a counterpart foreign application. The Search Report was mailed on April 26, 2004, which is less than three months ago, therefore no fee or certification is required under 37 C.F.R. § 1.97(b). Enclosed is a copy of the European Search Report dated April 26, 2004.

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

(230400)

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 (111325-104/230300).

Respectfully submitted,

NIXON PEABODY, LLP

By: _____

Marc S. Kaufman
Registration No.: 35,212

Dated: **July 1, 2004**

NIXON PEABODY LLP
Customer No.: 22204
401 9th Street, N.W., Suite 900
Washington, DC 20004-2128
Telephone: (202) 585-8000
FAX: (202) 585-8080

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(use as many sheets as necessary)

Sheet	1	of	1

Complete if Known	
Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin WANG, *et al.*
Art Unit	3621
Examiner Name	Unassigned
Attorney Docket Number	111325-104 (230300)

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	U.S. Patent Document: Number - Kind Code[2] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document: Country Code[3] Number[4] Kind Code[5] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Application of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]
		WO 00/08909 A	February 24, 2000			
		EP 0 715 244 A	June 5, 1996			
		EP 0 715 243 A	June 5, 1996			

OTHER PRIOR ART – NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.)., date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		European Search Report dated April 26, 2004.	

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference 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 form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

[1] Applicant's unique citation designation number (optional). [2] Applicant is to place a check mark here if English language Translation is attached.

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

THIS PAGE BLANK (USPTO)

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification : Not classified

A2

(11) International Publication Number: WO 00/08909

(43) International Publication Date: 24 February 2000 (24.02.00)

(21) International Application Number: PCT/US99/18383

(22) International Filing Date: 12 August 1999 (12.08.99)

(30) Priority Data:

09/133,519	13 August 1998 (13.08.98)	US
09/177,096	22 October 1998 (22.10.98)	US

(71) Applicant *(for all designated States except US)*: INTERNATIONAL BUSINESS MACHINES CORPORATION [US/US]; New Orchard Road, Armonk, NY 10504 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: DORAK, John, Jr. [US/US]; 22238 S.E. 62nd Avenue, Boca Raton, FL 33428 (US). DOWNS, Edgar [US/US]; 2740 N.E. 58th Street, Fort Lauderdale, FL 33308 (US). GRUSE, George, Gregory [US/US]; 4310 N.E. 24th Avenue, Lighthouse Point, FL 33064 (US). HURTADO, Marco [US/US]; 4720 N.W. 28th Avenue, Boca Raton, FL 22434 (US). LEHMAN, Christopher [US/US]; 2663 Hampton Circle S., Delray Beach, FL 33308 (US). LOTSPIECH, Jeffrey [US/US]; 992 Foothill Drive, San Jose, CA 95123 (US). MEDINA, Cesar [US/US]; 4017 N.W. 24th Terrace, Boca Raton, FL 33431 (US). MILSTED, Kenneth [US/US]; 9927 Majestic Way, Boynton Beach, FL 33437–3303 (US).

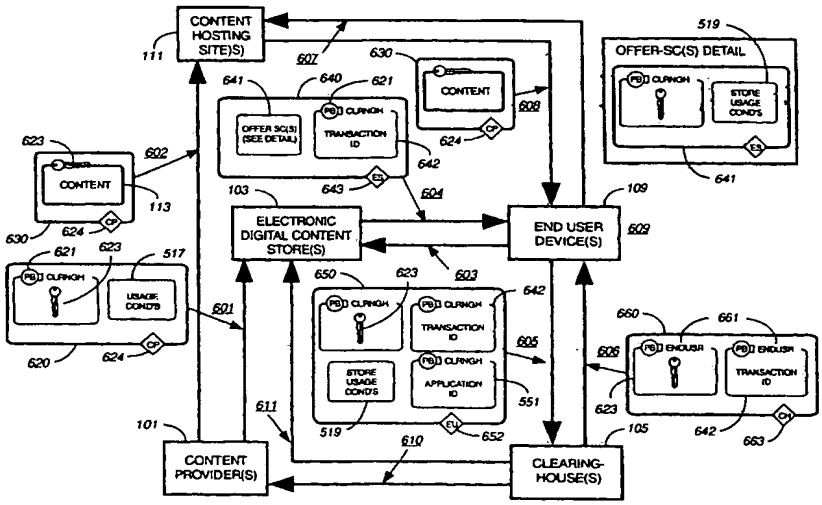
(74) Agent: SOUCAR, Stephen; IBM Corporation, Intellectual Property Law, Building 1, Mail Drop 1140, Route 100, P.O. Box 100, Somers, NY 10589 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, 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, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published

Without international search report and to be republished upon receipt of that report.

(54) Title: SYSTEM FOR TRACKING END–USER ELECTRONIC CONTENT USAGE



(57) Abstract

A system for tracking usage of digital content on user devices. Electronic stores coupled to a network sell licenses to play digital content data to users. Content players, which receive from the network the licensed content data, are used to play the licensed content data. Additionally, a logging site that is coupled to thte network tracks the playing of the content data. In particular, the logging site receives play information from the network, and the play information includes the number of times that the content data has been played by the associated content player. Also provided is a method for tracking usage of digital content on user devices. According to the method, a license to play digital content data is sold to a user, and the licensed content data is transmitted to a content player for the user. Further, information is transmitted to a logging site whenever the content data is played by the content player or copied from the content player to an external medium so that usage of the licensed content data can be tracked.

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	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

SYSTEM FOR TRACKING END-USER ELECTRONIC CONTENT USAGE

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention disclosed broadly relates to the field of electronic commerce and more particularly to a system and related tools for the secure delivery and rights management of digital assets, such as print media, films, games, and music over global communications networks such as the Internet and the World Wide Web.

2. Description of the Related Art

The use of global distribution systems such as the Internet for distribution of digital assets such as music, film, computer programs, pictures, games and other content continues to grow. At the same time owners and publishers of valuable digital content have been slow to embrace the use of the Internet for distribution of digital assets for several reasons. One reason is that owners are afraid of unauthorized copying or pirating of digital content. The electronic delivery of digital content removes several barriers to pirating. One barrier that is removed with electronic distribution is the requirement of the tangible recordable medium itself (e.g., diskettes or CD ROMs). It costs money to copy digital content on to tangible media, albeit, in many cases less than a dollar for a blank tape or recordable CD. However, in the case of electronic distribution, the tangible medium is no longer needed. The cost of the tangible medium is not a factor because content is distributed electronically. A second barrier, is the format of the content itself i.e. is the content stored in an analog format versus a digital format. Content stored in an analog format, for example, a printed picture, when reproduced by photocopying, the copy is of lesser quality than the original. Each subsequent copy of a copy, sometimes called a generation, is of less quality than the original. This degradation in quality is not present when a picture is stored digitally. Each copy, and every generation of copies can be as clear and crisp as the original. The aggregate effect of perfect digital copies combined with the very low cost to distribute content electronically and to distribute content widely over the Internet makes it relatively easy to pirate and distribute unauthorized copies. With a couple of keystrokes, a pirate can send hundreds or even of thousands of perfect copies of digital content over the Internet. Therefore a need exists to ensure the protection and security of digital assets distributed electronically.

Providers of digital content desire to establish a secure, global distribution system for digital content that protects the rights of content owners. The problems with establishing a digital content distribution system includes developing systems for digital content electronic distribution, rights management, and asset protection. Digital content that is distributed electronically includes content such as print media, films, games, programs, television, multimedia, and music.

The deployment of an electronic distribution system provides the digital content providers the ability to achieve fast settlement of payment through immediate sales reporting and electronic reconciliation as well as gain

secondary sources of revenue through redistribution of content. Since the electronic digital content distribution system is not affected by physical inventory outages or returns, the digital content providers and retailers may realize reduced costs and improved margins. Digital content providers could facilitate new, or augment existing, distribution channels for better timed-release of inventory. The transactional data from the electronic distribution system could be used to obtain information regarding consumer buying patterns as well as to provide immediate feedback on electronic marketing programs and promotions. In order to meet these goals, a need exists for digital content providers to use an electronic distribution model to make digital content available to a wide range of users and businesses while ensuring protection and metering of digital assets.

Other commercially available electronic distribution systems for digital content, such as real audio, A2B from AT&T, Liquid Audio Pro from Liquid Audio Pro Corp., City Music Network from Audio Soft and others offer transmission of digital data over secured and unsecured electronic networks. The use of secured electronic networks greatly reduces the requirement of digital content providers of distributing digital to a wide audience. The use of unsecured networks such as the Internet and Web allows the digital content to arrive to an end- user securely such as through the use of encryption. However, once the encrypted digital content is de-encrypted on the end-user's machine, the digital content is readily available to the end-user for unauthorized re-distribution. Therefore a need exists for a secure digital content electronic distribution system that provides protection of digital assets and ensures that the Content Provider(s)' rights are protected even after the digital content is delivered to consumers and businesses. A need thus exists for rights management to allow for secure delivery, licensing authorization, and control of the usage of digital assets.

Another reason owners of digital content have been slow to embrace electronic distribution is their desire to maintain and foster existing channels of distribution. Most content owners sell through retailers. In the music market these U.S. retailers include Tower Records, Peaches, Blockbuster, Circuit City and others. Many of these retailers have Web sites that allow Internet users to makes selections over the Internet and have selections mailed to the end-user. Example music Web sites include @tower, Music Boulevard and Columbia House. The use of electronic distribution can remove the ability of the retail stores from differentiating themselves from each other and differentiate themselves from the content owners, especially on the Web. Therefore a need exists to provide retailers of electronic content such as pictures, games, music, programs and videos a way to differentiate themselves from each other and the content owners when selling music through electronic distribution.

Content owners prepare their digital content for electronic distribution through distribution sites such as electronic stores. Electronic stores on the Internet, or through other online services, want to differentiate themselves from each other by their product offerings and product promotions. A traditional store, i.e. - the non-electronic, non-online analogs to electronic stores - use product promotions, product sales, product samples, liberal return policies and other promotional programs to differentiate themselves from their competitors. However, in the online world where the content providers impose usage conditions on the digital content, the ability of electronic stores to differentiate themselves may be severely limited. Moreover, even if the usage conditions can be changed, electronic stores are faced with the difficult task of processing the metadata associated

with the digital content from the content providers to promote and sell products electronically. Electronic stores need to manage several requirements when processing the metadata. First, the electronic store is required to receive the metadata associated with the digital content from the content providers. Many times, parts of this metadata may be sent encrypted, so the content provider must create a mechanism to decrypt the encrypted content. Second, the electronic store may wish to preview metadata from the content provider either before the content is received from the content provider or after the content is received by the electronic store, in order to assist with product marketing, product positioning and other promotional considerations for the content. Third, the electronic store is required to extract certain metadata used for promotional materials such as graphics and artist information. Often, this promotional material is used directly by the electronic store in its online promotions. Fourth, the electronic stores may wish to differentiate themselves from one another by modifying some of the permitted usage conditions to create different offerings of the digital content. Fifth, the electronic store may have to insert or alter certain address, such as URLs, in the metadata to direct payment reconciliation to an account reconciliation house automatically by the purchaser without the need to go through the electronic store for payment clearance. Sixth, the electronic store may need to create licenses for the permitted use of the copyrighted digital content that match usage conditions. For example, the license may grant the permission to make a limited number of copies of the digital content. A license is needed to reflect the terms and conditions of the permission granted.

In light of all these requirements, to process the metadata related to the digital content, many electronic stores write customized software programs to handle these requirements. The time, cost and testing needed to create these customized software programs can be large. Accordingly, a need exists to provide a solution to these requirements.

Still, another reason owners of digital content have been slow to embrace electronic distribution is the difficulty in preparing content for electronic distribution. Today, many providers of content have thousands or even tens of thousands of titles in their portfolio. In a music example, it is not unusual for a content owner to have a single master sound recording available on several different formats simultaneously (e.g. CD, tape and MiniDisc). In addition, a single format can have a master sound recording re-mastered or re-mixed for a specific distribution channel. As an example, the mixing for broadcast radio may be different than the mixing for a dance club sound track, which may be different than a generally available consumer CD. Inventorying and keeping track of these different mixes can be burdensome. Moreover, many owners of master recordings often times re-issue old recordings in various subsequent collections, such as "The Best Of", or in compilations for musical sound tracks to movies and other collections or compilations. As more content is offered digitally, the need to re-mix and encode the content for electronic distribution grows. Many times providers need to use old recording formats as guides to select the correct master sound recordings and have these sound recordings reprocessed and encoded for release for electronic distribution. This may be especially true for content providers that wish to use their old formats to assist them in re- releasing the old sound recording for electronic distribution. Providers will look through databases to match up titles, artists and sound recordings to set the encoding parameters. This process of manually searching databases for recording portfolios is not without its shortcomings. One shortcoming is the

need to have an operator manually search a database and set the processing parameters appropriately. Another shortcoming is the possibility of operator transcription error in selecting data from a database. Accordingly, a need exists to provide content providers a method to automatically retrieve associated data and master recordings for content such as audio.

Content owners prepare their digital content for electronic distribution through a process known as encoding. Encoding involves taking the content, digitizing it, if the content is presented in an analog format, and compressing it. The process of compressing allows the digital content to be transferred over networks and stored on recordable medium more efficiently because the amount of data transmitted or stored is reduced. However, compression is not without its shortcomings. Most compression involves the loss of some information, and is called lossy compression. Content providers must make decisions on what compression algorithm to use and the compression level required. For example, in music, the digital content or song may have very different characteristics depending on the genre of the music. The compression algorithm and compression level selected for one genre may not be the optimal choice for another genre of music. Content providers may find certain combinations of compression algorithms and compression levels work very well for one genre of music, say classical, but provide unsatisfactory results for another genre of music such as heavy metal. Moreover, audio engineers must often equalize the music, perform dynamic range adjustments and perform other preprocessing and processing settings to ensure the genre of music encoded produces the desired results. The requirement to always have to manually set these encoding parameters such as setting the equalization levels and the dynamic range settings for each digital content can be burdensome. Returning to the music example, a content provider for music with a collection covering a variety of musical genre would have to manually select for each song or set of songs to be encoded, the desired combination of encoding parameters. Accordingly, a need exists to overcome the need for manually selection of process parameters for encoding.

The process to compress content can require a large amount of dedicated computational resources, especially for larger content items such as full-length feature movies. Providers of compression algorithms offer various tradeoffs and advantages associated with their compression techniques. These tradeoffs include: the amount of time and computational resources needed to compress the content; the amount of compression achieved from the original content; the desired bit rate for playback; the performance quality of the compressed content; and other factors. Using an encoding program which take as input a multimedia file and generate an encoded output file with no interim indication of progress or status is a problem. Moreover, in many circumstances, other programs are used to call or to manage an encoding program with no interim indication of progress. This leaves the calling application with no way to gauge the amount of content that has been encoded as a percentage of the entire selection of designated to be encoded. In circumstances where the calling program is trying to schedule several different programs to run at once this can be a problem. Furthermore, this can be especially burdensome in cases where batches of content have been selected for encoding and the content provider wants to determine the progress of the encoding process. Accordingly, a need exists to overcome these problems.

Yet, still another reason digital content providers have been slow to adopt electronic distribution for their content is lack of standards for creating digital players on end-user devices for electronically delivered content. Content providers, electronic stores, or others in the electronic distribution chain may want to offer customized players on a variety of devices such as PCS, set-top boxes, hand-held devices and more. A set of tools that can handle the decryption of the digital content in a tamper resistant environment, that is, an environment to deter the unauthorized access to the content during playing by a third party is needed. Moreover, a set of tools is needed to enable an end user to manage of a local library of digital content without allowing the end user to have access to the content for uses other than what was purchased.

Further information on the background of protecting digital content can be found from the following three sources. "Music on the Internet and the Intellectual Property Protection Problem" by Jack Lacy, James Snyder, David Maher, of AT&T Labs, Florham Park, N.J. available online URL http://www.a2bmusic.com/about/papers/musicipp.htm. Cryptographically protected container, called DigiBox, in the article "Securing the Content, Not the Wire for Information Commerce" by Olin Sibert, David Bernstein and David Van Wie, InterTrust Technologies Corp. Sunnyvale, CA available online URL http://www.intertrust.com/architecture/stc.html. And "Cryptolope Container Technology", an IBM White Paper, available online URL http:///cyptolope.ibm.com/white.htm.

SUMMARY OF THE INVENTION

It is an object of the present invention to remove the above-mentioned drawbacks and to provide a system for tracking usage of content data. One embodiment of the present invention provides a system for tracking usage of digital content on user devices. Electronic stores coupled to a network sell licenses to play digital content data to users. Content players, which receive from the network the licensed content data, are used to play the licensed content data. Additionally, a logging site that is coupled to the network tracks the playing of the content data. In particular, the logging site receives play information from the network, and the play information includes the number of times that the content data has been played by the associated content player.

A further embodiment of the present invention provides a method for tracking usage of digital content on user devices. According to the method, a license to play digital content data is sold to a user, and the licensed content data is transmitted to a content player for the user. Further, information is transmitted to a logging site whenever the content data is played by the content player or copied from the content player to an external medium so that usage of the licensed content data can be tracked.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating an over view of a Secure Digital Content Electronic Distribution System according to the present invention.

FIG. 2 is a block diagram illustrating an example Secure Container (SC) and the associated graphical representations according to the present invention.

FIG. 3 is a block diagram illustrating an overview of the encryption process for a Secure Container (SC) according to the present invention.

FIG. 4 is a block diagram illustrating an overview of the de-encryption process for a Secure Container (SC) according to the present invention.

FIG. 5 is a block diagram illustrating an overview of the layers for the Rights Management Architecture of the Secure Digital Content Distribution System of FIG. 1 according to the present invention.

FIG. 6 is a block diagram illustrating an overview of the Content Distribution and Licensing Control as it applies to the License Control Layer of FIG. 5.

FIG. 7 is an illustration of an example user interface for the Work Flow Manager Tool of FIG. 1 according to the present invention.

FIG. 8 is a block diagram of the major tools, components and processes of the Work Flow Manager corresponding to the user interface in FIG. 7 according to the present invention. FIG. 9 is a block diagram illustrating the major tools, components and processes of an Electronic Digital Content Store of FIG. 1 according to the present invention.

FIG. 10 is a block diagram illustrating the major components and processes of an End- User Device(s) of FIG. 1 according to the present invention.

FIG. 11 is a flow diagram of a method to calculate an encoding rate factor for the Content Preprocessing and Compression tool of FIG. 8 according to the present invention.

FIG. 12 is a flow diagram of a method to automatically retrieve additional information for the Automatic Metadata Acquisition Tool of FIG. 8 according to the present invention.

FIG. 13 is a flow diagram of a method to automatically set the Preprocessing and Compression parameters of the Preprocessing and Compression Tool of FIG. 8 according to the present invention.

FIG. 14 is an example of user interface screens of the Player Application downloading content to a local library as described in FIG. 15 according to the present invention.

FIG. 15 is a block diagram illustrating the major components and processes of a Player Application running on End-User Device of FIG. 9 according to the present invention.

FIG. 16 is an example user interface screens of the Player Application of FIG. 15 according to the present invention.

FIG. 17 is a flow diagram of an alternate embodiment to automatically retrieve additional information for the Automatic Metadata Acquisition Tool of FIG. 8 according to the present invention.

DETAILED DESCRIPTION OF AN EMBODIMENT

A Table of Contents is provided for this present invention to assist the reader in quickly locating different sections in this embodiment.

I. SECURE DIGITAL CONTENT ELECTRONIC DISTRIBUTION SYSTEM

I. SECURE DIGITAL CONTENT ELECTRONIC DISTRIBUTION SYSTEM

A. System Overview

The Secure Digital Content Electronic Distribution System is a technical platform that encompasses the technology, specifications, tools, and software needed for the secure delivery and rights management of Digital Content and digital content-related content to an end-user, client device. The End-User Device(s) include PCS, set top boxes (IRDs), and Internet appliances. These devices may copy the content to external media or portable, consumer devices as permitted by the content proprietors. The term Digital Content or simply Content, refers to information and data stored in a digital format including: pictures, movies, videos, music, programs, multimedia and games.

The technical platform specifies how Digital Content is prepared, securely distributed through point-to-point and broadcast infrastructures (such as cable, Internet, satellite, and wireless) licensed to End-User Device(s), and protected against unauthorized copying or playing. In addition, the architecture of the technical platform allows for the integration and migration of various technologies such as watermarking, compression/encoding, encryption, and other security algorithms as they evolve over time.

The base components of the Secure Digital Content Electronic Distribution System are: (1) rights management for the protection of ownership rights of the content proprietor; (2) transaction metering for immediate and accurate compensation; and (3) an open and well-documented architecture that enables Content Provider(s) to prepare content and permit its secure delivery over multiple network infrastructures for playback on any standard compliant player.

1. Rights Management

Rights management in the Secure Digital Content Electronic Distribution System is implemented through a set of functions distributed among the operating components of the system. Its primary functions include: licensing authorization and control so that content is unlocked only by authorized intermediate or End-User(s) that have secured a license; and control and enforcement of content usage according to the conditions of purchase or license, such as permitted number of copies, number of plays, and the time interval or term the license may be valid. A secondary function of rights management is to enable a means to identify the origin of unauthorized copies of content to combat piracy.

Licensing authorization and control are implemented through the use of a Clearinghouse(s) entity and Secure Container (SC) technology. The Clearinghouse(s) provides licensing authorization by enabling intermediate or End-User(s) to unlock content after verification of a successful completion of a licensing transaction. Secure Containers are used to distribute encrypted content and information among the system components. A SC is a cryptographic carrier of information or content that uses encryption, digital signatures, and digital certificates to provide protection against unauthorized interception or modification of electronic information and content. It also allows for the verification of the authenticity and integrity of the Digital Content. The advantage of these rights management functions is that the electronic Digital Content distribution infrastructure does not have to be secure or trusted. Therefore allowing transmission over network infrastructures such as the Web and Internet. This is due to the fact that the Content is encrypted within Secure Containers and its storage and distribution are separate from the control of its unlocking and use. Only users who have decryption keys can unlock the encrypted Content, and the Clearinghouse(s) releases decryption keys only for authorized and appropriate usage requests. The Clearinghouse(s) will not clear bogus requests from unknown or unauthorized parties or requests that do not comply with the content's usage conditions as set by the content proprietors. In addition, if the SC is tampered with during its transmission, the software in the Clearinghouse(s) determines that the Content in a SC is corrupted or falsified and repudiate the transaction.

The control of Content usage is enabled through the End-User Player Application 195 running on an End-User Device(s). The application embeds a digital code in every copy of the Content that defines the allowable number of secondary copies and play backs. Digital watermarking technology is used to generate the digital code, to keep it hidden from other End- User Player Application 195, and to make it resistant to alteration attempts. In an alternate embodiment, the digital code is just kept as part of the usage conditions associated with the Content 113. When the Digital Content 113 is accessed in a compliant End-User Device(s), the End-User Player Application 195 reads the watermark to check the use restrictions and updates the watermark as required. If the requested use of the content does not comply with the usage conditions, e.g., the number of copies has been exhausted, the End-User Device(s) will not perform the request.

Digital watermarking also provides the means to identify the origin of authorized or unauthorized copies of Content. An initial watermark in the Content is embedded by the content proprietor to identify the content proprietor, specify copyright information, define geographic distribution areas, and add other pertinent information. A second watermark is embedded in the Content at the End-User Device(s) to identify the content

purchaser (or licensee) and End-User Device(s), specify the purchase or license conditions and date, and add any other pertinent information.

Since watermarks become an integral part of the Content, they are carried in the copies independent of whether the copies were authorized or not. Thus the Digital Content always contains information regarding its source and its permitted use regardless of where the content resides or where it comes from. This information may be used to combat illegal use of the Content.

2. Metering

As part of its rights management functions, the Clearinghouse(s) keeps a record of all transactions where a key exchange is cleared through the Clearinghouse(s). This record allows for the metering of licensing authorization and the original conditions of use. The transaction record can be reported to responsible parties, such as, content proprietors or Content Provider(s), retailers, and others, on an immediate or periodic basis to facilitate electronic reconciliation of transaction payments and other uses.

3. Open Architecture

The Secure Digital Content Electronic Distribution System (System) is an open architecture with published specifications and interfaces to facilitate broad implementation and acceptance of the System in the market place while maintaining rights protection for the content proprietors. The flexibility and openness of the System architecture also enable the System to evolve over time as various technologies, transmission infrastructures, and devices are delivered to the marketplace.

The architecture is open regarding the nature of the Content and its format. Distribution of audio, programs, multimedia, video, or other types of Content is supported by the architecture. The Content could be in a native format , such as linear PCM for digital music, or a format achieved by additional preprocessing or encoding, such as filtering, compression, or pre/de- emphasis, and more. The architecture is open to various encryption and watermarking techniques. It allows for the selection of specific techniques to accommodate different Content types and formats and to allow the introduction or adoption of new technologies as they evolve. This flexibility allows Content Provider(s) to pick and evolve the technologies they use for data compression, encryption, and formatting within the Secure Digital Content Electronic Distribution System.

The architecture is also open to different distribution networks and distribution models. The architecture supports content distribution over low-speed Internet connections or high-speed satellite and cable networks and can be used with point-to-point or broadcast models. In addition, the architecture is designed so that the functions in the End-User Device(s) can be implemented on a wide variety of devices, including low cost consumer devices. This flexibility allows Content Provider(s) and retailers to offer Content to intermediate or End-User(s) through a variety of service offerings and enables the users to purchase or license Content, play it back, and record it on various compliant player devices.

B. System Functional Elements

Turning now to FIG. 1, there is shown a block diagram illustrating an overview of a Secure Digital Content Electronic Distribution System 100 according to the present invention. The Secure Digital Content Electronic Distribution System 100 encompasses several business elements that comprise an end-to-end solution, including: Content Provider(s) 101 or the proprietors of the Digital Content, Electronic Digital Content Store(s) 103, Intermediate Market Partners (not shown), Clearinghouse(s) 105, Content Hosting Site 111, Transmission Infrastructures 107, and End-User Device(s) 109. Each of these business elements use various components of the Secure Digital Content Electronic Distribution System 100. A high level description of these business elements and system components, as they pertain specifically to electronic Content 113 distribution, follows.

1. Content Provider(s) 101

Content Provider(s) 101 or content proprietor(s) are owners of original Content 113 and/or distributors authorized to package independent Content 113 for further distribution. Content Provider(s) 101 may exploit their rights directly or license Content 113 to the Electronic Digital Content Store(s) 103, or Intermediate Market Partners (not shown), usually in return for Content usage payments related to electronic commerce revenues. Examples of Content Provider(s) 101 include Sony, Time-Warner, MTV, IBM, Microsoft, Turner, Fox and others.

Content Provider(s) 101 use tools provided as part of the Secure Digital Content Electronic Distribution System 100 in order to prepare their Content 113 and related data for distribution. A Work Flow Manager Tool 154 schedules Content 113 to be processed and tracks the Content 113 as it flows through the various steps of Content 113 preparation and packaging to maintain high quality assurance. The term metadata is used throughout this document to mean data related to the Content 113 and in this embodiment does not include the Content 113 itself. As an example, metadata for a song may be a song title or song credits but not the sound recording of the song. The Content 113 would contain the sound recording. A Metadata Assimilation and Entry Tool 161 is used to extract metadata from the Content Provider(s)' Database 160 or data provided by the Content Provider(s) in a prescribed format (for a music example the Content 113 information such as CD title, artist name, song title, CD artwork, and more) and to package it for electronic distribution. The Metadata Assimilation and Entry Tool 161 is also used to enter the Usage Conditions for the Content 113. The data in Usage Conditions can include copy restriction rules, the wholesale price, and any business rules deemed necessary. A Watermarking Tool is used to hide data in the Content 113 that identifies the content owner, the processing date, and other relevant data. For an embodiment where the Content 113 is audio, an audio preprocessor tool is used to adjust the dynamics and/or equalize the Content 113 or other audio for optimum compression quality, compress the Content 113 to the desired compression levels, and encrypt the Content 113. These can be adapted to follow technical advances in digital content compression/encoding, encryption, and formatting methods, allowing the Content Provider(s) 101 to utilize best tools as they evolve over time in the marketplace.

The encrypted Content 113, digital content-related data or metadata, and encrypted keys are packed in SCs (described below) by the SC Packer Tool and stored in a content hosting site and/or promotional web site for

electronic distribution. The content hosting site can reside at the Content Provider(s) 101 or in multiple locations, including Electronic Digital Content Store(s) 103 and Intermediate Market Partners (not shown) facilities. Since both the Content 113 and the Keys (described below) are encrypted and packed in SCs, Electronic Digital Content Store(s) 103 or any other hosting agent can not directly access decrypted Content 113 without clearance from the Clearinghouse(s) and notification to the Content Provider(s) 101.

2. Electronic Digital Content Store(s) 103

Electronic Digital Content Store(s) 103 are the entities who market the Content 113 through a wide variety of services or applications, such as Content 113 theme programming or electronic merchandising of Content 113. Electronic Digital Content Store(s) 103 manage the design, development, business operations, settlements, merchandising, marketing, and sales of their services. Example online Electronic Digital Content Store(s) 103 are Web sites that provide electronic downloads of software.

Within their services, Electronic Digital Content Store(s) 103 implement certain functions of the Secure Digital Content Electronic Distribution System 100. Electronic Digital Content Store(s) 103 aggregate information from the Content Provider(s) 101, pack content and metadata in additional SCs, and deliver those SCs to consumers or businesses as part of a service or application. Electronic Digital Content Store(s) 103 use tools provided by the Secure Digital Content Electronic Distribution System 100 to assist with: metadata extraction, secondary usage conditions, SC packaging, and tracking of electronic content transactions. The secondary usage conditions data can include retail business offers such as Content 113 purchase price, pay-per-listen price, copy authorization and target device types, or timed-availability restrictions.

Once an Electronic Digital Content Store(s) 103 completes a valid request for electronic Content 113 from an End-User(s), the Electronic Digital Content Store(s) 103 is responsible for authorizing the Clearinghouse(s) 105 to release the decryption key for the Content 113 to the customer. The Electronic Digital Content Store(s) also authorizes the download of the SC containing the Content 113. The Electronic Digital Content Store(s) may elect to host the SCs containing the Digital Content at its local site and/or utilize the hosting and distribution facilities of another Content hosting site.

The Electronic Digital Content Store(s) can provide customer service for any questions or problems that an End-User(s) may have using the Secure Digital Content Electronic Distribution System 100, or the Electronic Digital Content Store(s) 103 may contract their customer service support to the Clearinghouse(s) 105.

3. Intermediate Market Partners (not shown)

In an alternate embodiment, the Secure Digital Content Electronic Distribution System 100 can be used to provide Content 113 securely to other businesses called Intermediate Market Partners. These partners may include digital content-related companies offering a non-electronic service, such as televisions stations or video clubs, radio stations or record clubs, that distribute Content 113. These Partners may also include other trusted parties who handle material as part of making or marketing sound recordings, such as record studios, replicators, and producers. These Intermediate Market Partners requires clearance from the Clearinghouse(s) 105 in order to decrypt the Content 113.

4. Clearinghouse(s) 105

The Clearinghouse(s) 105 provides the licensing authorization and record keeping for all transactions that relate to the sale and/or permitted use of the Content 113 encrypted in a SC. When the Clearinghouse(s) 105 receives a request for a decryption key for the Content 113 from an intermediate or End-User(s), the Clearinghouse(s) 105 validates the integrity and authenticity of the information in the request; verifies that the request was authorized by an Electronic Digital Content Store(s) or Content Provider(s) 101; and verifies that the requested usage complies with the content Usage Conditions as defined by the Content Provider(s) 101. Once these verifications are satisfied, the Clearinghouse(s) 105 sends the decryption key for the Content 113 to the requesting End-User(s) packed in a License SC. The key is encrypted in a manner so that only the authorized user can retrieve it. If the End-User's request is not verifiable, complete, or authorized, the Clearinghouse(s) 105 repudiates the request for the decryption key.

The Clearinghouse(s) 105 keeps a record of all transactions and can report them to responsible parties, such as Electronic Digital Content Store(s) 103 and Content Provider(s) 101, on an immediate, periodic, or restricted basis. This reporting is a means by which Content Provider(s) 101 can be informed of the sale of Content 113 and the Electronic Digital Content Store(s) 103 can obtain an audit trail of electronic delivery to their customers. The Clearinghouse(s) 105 can also notify the Content Provider(s) 101 and/or Electronic Digital Content Store(s) 103 if it detects that information in a SC has been compromised or does not comply with the Content's Usage Conditions. The transaction recording and repository capabilities of the Clearinghouse(s) 105 database is structured for data mining and report generation.

In another embodiment, the Clearinghouse(s) 105 can provide customer support and exception processing for transactions such as refunds, transmission failures, and purchase disputes. The Clearinghouse(s) 105 can be operated as an independent entity, providing a trusted custodian for rights management and metering. It provides billing and settlement as required. Examples of electronic Clearinghouse(s) include Secure-Bank.com and Secure Electronic Transaction (SET) from Visa/Mastercard. In one embodiment, the Clearinghouse(s) 105 are Web sites accessible to the End-User Device(s) 109. In another embodiment, the Clearinghouse(s) 105 is part of the Electronic Digital Content Store(s) 103.

5. End-User Device(s) 109

The End-User Device(s) 109 can be any player device that contains an End-User Player Application 195 (described later) compliant with the Secure Digital Content Electronic Distribution System 100 specifications. These devices may include PCS, set top boxes (IRDs), and Internet appliances. The End-User Player Application 195 could be implemented in software and/or consumer electronics hardware. In addition to performing play, record, and library management functions, the End-User Player Application 195 performs SC processing to enable rights management in the End-User Device(s) 109. The End-User Device(s) 109 manages the download and storage of the SCs containing the Digital Content; requests and manages receipt of the encrypted Digital Content keys from the Clearinghouse(s) 105; processes the watermark(s) every time the Digital Content is copied or played; manages the number of copies made (or deletion of the copy) in accordance with the Digital Content's Usage Conditions; and performs the copy to an external media or portable consumer device if permitted. The portable consumer device can perform a subset of the End-User Player Application 195 functions in order to process the content's Usage Conditions embedded in the watermark. The terms End-User(s) and End-User Player Application 195 are used throughout this to mean through the use or running-on an End-User Device(s) 109.

6. Transmission Infrastructures 107

The Secure Digital Content Electronic Distribution System 100 is independent of the transmission network connecting the Electronic Digital Content Store(s) 103 and End-User Device(s) 109. It supports both point-to-point such as the Internet and broadcast distribution models such as digital broadcast television.

Even though the same tools and applications are used to acquire, package, and track Content 113 transactions over various Transmission Infrastructures 107, the presentation and method in which services are delivered to the customer may vary depending on the infrastructure and distribution model selected. The quality of the Content 113 being transferred may also vary since high bandwidth infrastructures can deliver high-quality digital content at more acceptable response times than lower bandwidth infrastructures. A service application designed for a point-to-point distribution model can be adapted to support a broadcast distribution model as well.

C. System Uses

The Secure Digital Content Electronic Distribution System 100 enables the secure delivery of high-quality, electronic copies of Content 113 to End-User Device(s) 109, whether consumer or business, and to regulate and track usage of the Content 113.

The Secure Digital Content Electronic Distribution System 100 could be deployed in a variety of consumer and business-to-business services using both new and existing distribution channels. Each particular service could use a different financial model that can be enforced through the rights management features of the Secure Digital Content Electronic Distribution System 100. Models such as wholesale or retail purchase, pay-per-listen usage, subscription services, copy/no-copy restrictions, or redistribution could be implemented through the rights management of the Clearinghouse(s) 105 and the End-User Player Application 195 copy protection features.

The Secure Digital Content Electronic Distribution System 100 allows Electronic Digital Content Store(s) 103 and Intermediate Market Partners a great deal of flexibility in creating services that sell Content 113. At the same time it provides Content Provider(s) 101 a level of assurance that their digital assets are protected and metered so that they can receive appropriate compensation for the licensing of Content 113.

II. CRYPTOGRAPHY CONCEPTS AND THEIR APPLICATION TO THE SECURE DIGITAL CONTENT ELECTRONIC DISTRIBUTION SYSTEM

License Control in the Secure Digital Content Electronic Distribution System 100 is based on the use of cryptography. This section introduces basic cryptography technologies of the present invention. The use of public key encryption, symmetric key encryption, digital signatures, digital watermarks and digital certificates is known.

A. Symmetric Algorithms

In the Secure Digital Content Electronic Distribution System 100 the Content Provider(s) 101 encrypts the content using symmetric algorithms. They are called symmetric algorithms because the same key is used to encrypt and decrypt data. The data sender and the message recipient must share the key. The shared key is referred to here as the symmetric key. The Secure Digital Content Electronic Distribution System 100 architecture is independent of the specific symmetric algorithm selected for a particular implementation.

Common symmetric algorithms are DES, RC2 and RC4. Both DES and RC2 are block cipher. A block cipher encrypts the data using a block of data bits at a time. DES is an official US government encryption standard, has a 64-bit block size, and uses a 56-bit key. Triple-DES is commonly used to increase the security achieved with simple DES. RSA Data Security designed RC2. RC2 uses a variable-key-size cipher and has a block size of 64 bits. RC4, also designed by RSA Data Security, is a variable-key-size stream cipher. A stream cipher operates on a single data bit at a time. RSA Data Security claims that eight to sixteen machine operations are required for RC4 per output byte.

IBM designed a fast algorithm called SEAL. SEAL is a stream algorithm that uses a variable-length key and that has been optimized for 32-bit processors. SEAL requires about five elementary machine instructions per data byte. A 50 MHZ, 486-based computer runs the SEAL code at 7.2 megabytes/second if the 160-bit key used has already been preprocessed into internal tables.

Microsoft reports results of encryption performance benchmark in its Overview of CryptoAPI document. These results were obtained by an application using Microsoft's CryptoAPI, running on a 120-MHZ, Pentium-based computer with Windows NT 4.0.

Cipher	Key Size	Key Setup Time	Encryption Speed
DES	56	460	1138519
RC2	40	40	286888

Cipher	Key Size	Key Setup Time	Encryption Speed
DES	56	460	1138519
RC2	40	40	286888
RC4	40	151	2377723
RC4	40	151	2377723

B. Public Key Algorithms

In the Secure Digital Content Electronic Distribution System 100, symmetric keys and other small data pieces are encrypted using public keys. Public key algorithms use two keys. The two keys are mathematically related so that data encrypted with one key can only be decrypted with the other key. The owner of the keys keeps one key private (private key) and publicly distributes the second key (public key).

To secure the transmission of a confidential message using a public key algorithm, one must use the recipient's public key to encrypt the message. Only the recipient, who has the associated private key, can decrypt the message. Public key algorithms are also used to generate digital signatures. The private key is used for that purpose. The following section provides information on digital signatures.

The most common used public-key algorithm is the RSA public-key cipher. It has become the de-facto public key standard in the industry. Other algorithms that also work well for encryption and digital signatures are ElGamal and Rabin. RSA is a variable-key length cipher.

Symmetric key algorithms are much faster than the public key algorithms. In software, DES is generally at least 100 times as fast as RSA. Because of this, RSA is not used to encrypt bulk data. RSA Data Security reports that on a 90 MHZ Pentium machine, RSA Data Security's toolkit BSAFE 3.0 has a throughput for private-key operations (encryption or decryption, using the private key) of 21.6 kilobits/second with a 512-bit modulus and 7.4 kilobits/second with a 1024-bit modulus.

C. Digital Signature

In the Secure Digital Content Electronic Distribution System 100, the issuer of SC(s) protects the integrity of SC(s) by digitally signing it. In general, to create a digital signature of a message, a message owner first computes the message digest (defined below) and then encrypt the message digest using the owner's private key. The message is distributed with its signature. Any recipient of the message can verify the digital signature first by decrypting the signature using the public key of the message owner to recover the message digest. Then, the recipient computes the digest of the received message and compares it with the recovered one. If the message has not being altered during distribution, the calculated digest and recovered digest must be equal.

In the Secure Digital Content Electronic Distribution System 100, since SC(s) contain several data parts, a digest is calculated for each part and a summary digest is calculated for the concatenated part digests. The summary digest is encrypted using the private key of the issuer of the SC(s). The encrypted summary digest is the

issuer's digital signature for the SC(s). The part digests and the digital signature are included in the body of the SC(s). The recipients of SC(s) can verify the integrity of the SC(s) and its parts by means of the received digital signature and part digests.

A one-way hash algorithm is used to calculate a message digest. A hash algorithm takes a variable-length-input message and converts it into a fixed length string, the message digest. A one-way hash algorithm operates only in one direction. That is, it is easy to calculate the digest for an input message, but it is very difficult (computationally infeasible) to generate the input message from its digest. Because of the properties of the one-way hash functions, one can think of a message digest as a fingerprint of the message.

The more common one-way hash functions are MD5 from RSA Data Security and SHA designed by the US National Institute of Technology and Standards (NITS).

D. Digital Certificates

A digital certificate is used to authenticate or verify the identity of a person or entity that has sent a digitally signed message. A certificate is a digital document issued by a certification authority that binds a public key to a person or entity. The certificate includes the public key, the name of the person or entity, an expiration date, the name of the certification authority, and other information. The certificate also contains the digital signature of the certification authority.

When an entity (or person) sends a message signed with its private key and accompanied with its digital certificate, the recipient of the message uses the entity's name from the certificate to decide whether or not to accept the message.

In the Secure Digital Content Electronic Distribution System 100, every SC(s), except those issued by the End-User Device(s) 109, includes the certificate of the creator of the SC(s). The End-User Device(s) 109 do not need to include certificates in their SC(s) because many End- User(s) do not bother to acquire a certificate or have certificates issued by non bona-fide Certification Authorities. In the Secure Digital Content Electronic Distribution System 100, the Clearinghouse(s) 105 has the option of issuing certificates to the Electronic Digital Content Store(s) 103. This allows the End-User Device(s) 109 to independently verify that the Electronic Digital Content Store(s) 103 have been authorized by the Secure Digital Content Electronic Distribution System 100.

E. Guide To The SC(s) Graphical Representation

This document uses a drawing to graphically represent SC(s) that shows encrypted parts, non-encrypted parts, the encryption keys, and certificates. Referring now to FIG. 2 is an example drawing of SC(s) 200. The following symbols are used in the SC(s) figures. Key 201 is a public or private key. The teeth of the key e.g. CLRNGH for Clearinghouse indicate the key owner. PB inside the handle indicates that it is a public key thus key 201 is a Clearinghouse public key. PV inside the handle indicates that it is a private key. Diamond shape is an End- User Digital Signature 202. The initials indicate which private key was used to create the signature thus in EU is the End-User(s) digital signature from table below. Symmetric key 203 is used to encrypt content. An encrypted symmetric key object 204 comprising a symmetric key 203 encrypted with a PB of CLRNGH. The key on the top border of the rectangle is the key used in the encryption of the object. The symbol or text inside the rectangle indicates the encrypted object (a symmetric key in this case). Another encrypted object, in this example a Transaction ID encrypted object 205 is shown. And Usage Conditions 206 for content licensing management as described below. The SC(s) 200 comprises Usage Conditions 206, Transaction ID encrypted object 205, an Application ID encrypted object 207, and encrypted symmetric key object 204, all signed with an End-User Digital Signature 202.

The table below shows the initials that identify the signer of SC(s).

Initial	Component
CP	Content Provider(s) 101
MS	Electronic Digital Content Store(s) 103
HS	Content Hosting Site(s) 111
EU	End-User Device(s) 109
CH	Clearinghouse(s) 105
CA	certification authority(ies) (not shown)

F. Example of a Secure Container Encryption

The tables and diagrams below provide an overview of the encryption and decryption process used to create and recover information from SC(s). The SC(s) that is created and decrypted in this process overview is a general SC(s). It does not represent any of the specific SC(s) types used for rights management in the Secure Digital Content Electronic Distribution System 100. The process consists of the steps described in FIG. 3 for encryption process.

Process Flow for Encryption Process of FIG. 3

Step	Process
301	Sender generates a random symmetric key and uses it to encrypt the content.
302	Sender runs the encrypted content through a hash algorithm to produce the content digest.

303 Sender encrypts the symmetric key using the recipient's public key. PB RECPNT refers to the recipient's public key.

304 Sender runs the encrypted symmetric key through the same hash algorithm used in step 2 to produce the symmetric key digest.

305 Sender runs the concatenation of the content digest and symmetric key digest through the same hash algorithm used in step 2 to produce the SC(s) digest.

306 Sender encrypts the SC(s) digest with the sender's private key to produce the digital signature for the SC(s). PV SENDER refers to the sender's private key.

307B Sender creates a SC(s) file that includes the encrypted content, encrypted symmetric key, content digest, symmetric key digest, sender's certificate, and SC(s) signature.

307A Sender must have obtained the certificate from a certification authority prior to initiating secure communications. The certification authority includes in the certificate the sender's public key, the sender's name and signs it. PV CAUTHR refers to the certifications authority's private key. Sender transmits the SC(s) to the recipient.

Process Flow for Decryption Process of FIG. 4

Step	Process
408	Recipient receives the SC(s) and separates its parts.
409	Recipient verifies the digital signature in the sender's certificate by decrypting it with the public key of the certification authority. If the certificate's digital signature is valid, recipient acquires the sender's public key from the certificate.
410	Recipient decrypts the SC(s) digital signature using the sender's public key. This recovers the SC(s) digest. PB SENDER refers to the sender's public key.
411	Recipient runs the concatenation of the received content digest and encrypted key digest through the same hash algorithm used by the sender to compute the SC(s) digest.
412	Recipient compares the computed SC(s) digest with the one recovered from the sender's digital signature. If they are the same, recipient confirms that the received digests have not been altered and continues with the decryption process. If they are not the same, recipient discards the SC(s) and notifies the sender.
413	Recipient runs the encrypted symmetric key through the same hash algorithm used in step 411 to compute the symmetric key digest.
414	Recipient compares the computed symmetric key digest with the one received in the SC(s). If it is the same, recipient knows that the encrypted symmetric key has not been altered. Recipient continues with the decryption process. If not valid, recipient discards the SC(s) and notifies the sender.
415	Recipient runs the encrypted content through the same hash algorithm used in step 411 to compute the content digest.

416 Recipient compares the computed content digest with the one received in the SC(s). If it is the same, recipient knows that the encrypted content has not been altered. Recipient then continues with the decryption process. If not valid, recipient discards the SC(s) and notifies the sender.

417 Recipient decrypts the encrypted symmetric key using the recipient's private key. This recovers the symmetric key. PV RECPNT refers to the recipient's private key.

418 Recipient uses the symmetric key to decrypt the encrypted content. This recovers the content.

III. SECURE DIGITAL CONTENT ELECTRONIC DISTRIBUTION SYSTEM FLOW

The Secure Electronic Digital Content Distribution System 100, consists of several components that are used by the different participants of the system. These participants include the Content Provider(s) 101, Electronic Digital Content Store(s) 103, End-User(s) via End-User Device(s) 109 and the Clearinghouse(s) 105. A high level system flow is used as an overview of the Secure Digital Content Electronic Distribution System 100. This flow outlined below tracks Content as it flows throughout the System 100. Additionally it outlines the steps used by the participants to conduct the transactions for the purchase, unlocking and use of the Content 113. Some of the assumptions made in the system flow include:

- This is a system flow for a Digital Content service (Point-to-Point Interface to a PC).
- Content Provider(s) 101 submits audio Digital Content in PCM uncompressed format (as a music audio example).
- Content Provider(s) 101 has metadata in an ODBC compliant database or Content Provider(s) 101 will enter the data directly into the Content Information Processing Subsystem, or will have provided data in prescribed ASCII file format(s).
- Financial settlement is done by the Electronic Digital Content Store(s).
- Content 113 is hosted at a single Content Hosting Site(s) 111.

It should be understood by those skilled in the art that these assumptions can be altered to accommodate the exact nature of the Digital Content e.g. music, video and program and electronic distribution systems broadcast.

The following process flow in illustrated in FIG. 1.

Step Process

121 A uncompressed PCM audio file is provided as Content 113 by the Content Provider(s) 101. Its filename is input into the Work Flow Manager 154 Tool along with the Content Provider(s)' 101 unique identifier for the Content 113.

122 Metadata is captured from the Content Provider(s)' Database 160 by the Content Information Processing Subsystem using the Content Provider(s)' 101 unique identifier for the Content 113 and information provided by the Database Mapping Template.

123 The Work Flow Manager Tool 154 is used to direct the content flow through the acquisition and preparation process at the Content Provider(s) 101. It can also be used to track the status of any piece of content in the system at any time.

124 The Usage Conditions for the Content 113 are entered into the Content Information Processing Subsystem, this can be done either manually or automatically. This data includes copy restriction rules and any other business rules deemed necessary. All of the metadata entry can occur in parallel with the Audio Processing for the data.

125 The Watermarking Tool is used to hide data in the Content 113 that the Content Provider(s) 101 deems necessary to identify the content. This could include when it was captured, where it came from (this Content Provider(s) 101), or any other information specified by the Content Provider(s) 101.

- The Content Processing Tool 125 performs equalization, dynamics adjustments and re-sampling to the Content 113 as necessary for the different compression levels supported.
- The Content 113 is compressed using the Content Processing Tool 125 to the desired compression levels. The Content 113 can then be played back to verify that the compression produces the required level of Content 113 quality. If necessary the equalization, dynamics adjustments, compression and playback quality checks can be performed as many times as desired.
- The Content 113 and a subset of its metadata is encrypted with a Symmetric Key by the SC Packer. This tool then encrypts the key using the Public Key of the Clearinghouse(s) 105 to produce an Encrypted Symmetric Key. This key can be transmitted anywhere without comprising the security of the Content 113 since the only entity that can decrypt it is the Clearinghouse(s) 105.

126 The Encrypted Symmetric Key, metadata and other information about the Content 113 is then packed into a Metadata SC by the SC Packer Tool 152.

127 The encrypted Content 113 and metadata are then packed into a Content SC. At this point the processing on the Content 113 and metadata is complete.

128 The Metadata SC(s) is then sent to the Content Promotions Web Site 156 using the Content Disbursement Tool (not shown).

129 The Content Disbursement Tool sends the Content SC(s) to the Content Hosting Site(s) 111. The Content Hosting Site(s) can reside at the Content Provider(s) 101, the Clearinghouse(s) 105 or a special location dedicated for Content Hosting. The URL for this site is part of the metadata that was added to the Metadata SC.

130 The Content Promotions Web Site 156 notifies Electronic Digital Content Store(s) 103 of new Content 113 that is added to the System 100.

131 Using the Content Acquisition Tool, Electronic Digital Content Store(s) 103 then download the Metadata SCs that correspond to the Content 113 they wish to sell.

132 The Electronic Digital Content Store(s) 103 will use the Content Acquisition Tool to pull out any data from the Metadata SC(s) that they want to use to promote the Content 113 on their Web Site. Access to portions of this metadata can be secured and charged for if desired.

133 The Usage Conditions for the Content 113, specific to this Electronic Digital Content Store(s) 103, are entered using the Content Acquisition Tool. These Usage Conditions include the retail prices and copy/play restrictions for the different compression levels of the Content 113.

134 The Electronic Digital Content Store(s) 103 specific Usage Conditions and the original Metadata SC(s) are packed into an Offer SC by the SC Packer Tool.

135 After the Electronic Digital Content Store(s) 103 Web Site is updated, the Content 113 is available to End-User(s) surfing the Web.

136 When an End-User(s) finds Content 113 that they want to buy, they click on a content icon, such as a music icon, and the item is added to his/her shopping cart which is maintained by the Electronic Digital Content Store(s) 103. When the End-User(s) completes shopping they submit the purchase request to the Electronic Digital Content Store(s) 103 for processing.

137 The Electronic Digital Content Store(s) 103 then interacts with credit card clearing organizations to place a hold on the funds in the same way they do business today.

138 Once the Electronic Digital Content Store(s) 103 receives the credit card authorization number back from the credit card clearing organization, it stores this into a database and invokes the SC Packer Tool to build a Transaction SC. This Transaction SC includes all of the Offer SCs for the Content 113 that the End-User(s) has purchased, a Transaction ID that can be tracked back to the Electronic Digital Content Store(s) 103, information that identifies the End-User(s), compression levels, Usage Conditions and the price list for the songs purchased.

139 This Transaction SC is then transmitted to the End-User Device(s) 109.

140 When the Transaction SC arrives on the End-User Device(s) 109, it kicks off the End- User Player Application 195 which opens the Transaction SC and acknowledges the End- User's purchase. The End-User Player Application 195 then opens the individual Offer SCs and in an alternate embodiment, may inform the user with an estimate of the download time. It then asks the user to specify when they want to download the Content 113.

141 Based on the time the End-User(s) requested the download, the End-User Player Application 195 will wake up and initiate the start of the download process by building a Order SC that contains among other things the Encrypted Symmetric Key for the Content 113, the Transaction ID, and End-User(s) information.

142 This Order SC is then sent to the Clearinghouse(s) 105 for processing.

143 The Clearinghouse(s) 105 receives the Order SC, opens it and verifies that none of the data has been tampered with. The Clearinghouse(s) 105 validates the Usage Conditions purchased by the End-User(s). These Usage Conditions must comply with those specified by the Content Provider(s) 101. This information is logged in a database.

144 Once all the checks are complete, the Encrypted Symmetric Key is decrypted using the private key of the Clearinghouse(s) 105. The Symmetric Key is then encrypted using the public key of the End-User(s). This new Encrypted Symmetric Key is then packaged into a License SC by the SC Packer.

145 The License SC is then transmitted to the End-User(s).

146 When the License SC is received at the End-User Device(s) 109 it is stored in memory until the Content SC is downloaded.

147 The End-User Device(s) 109 request from the Content Hosting Facility 111, sending the corresponding License SC for the purchased Content 113.

148 Content 113 is sent to the End-User Device(s) 109. Upon the receipt the Content 113 is de-encrypted by the End-User Device(s) 109 using the Symmetric Key.

IV. RIGHTS MANAGEMENT ARCHITECTURE MODEL

A. Architecture Layer Functions

FIG. 5 is a block diagram of the Rights Management Architecture of the Secure Digital Content Electronic Distribution System 100. Architecturally, four layers represent the Secure Digital Content Electronic Distribution System 100: the License Control Layer 501, the Content Identification Layer 503, Content Usage Control Layer 505, and the Content Formatting Layer 507. The overall functional objective of each layer and the individual key functions for each layer are described in this section. The functions in each of the layers are fairly independent of the functions in the other layers. Within broad limitations, functions in a layer can be substituted with similar functions without affecting the functionality of the other layers. Obviously, it is required that the output from one layer satisfies format and semantics acceptable to the adjacent layer.

The License Control Layer 501 ensures that:

- the Digital Content is protected during distribution against illegal interception and tampering;
- the Content 113 originates from a rightful content owner and is distributed by a licensed distributor, e.g. Electronic Digital Content Store(s) 103;
- the Digital Content purchaser has a properly licensed application;
- the distributor is paid by the purchaser before a copy of the Content 113 is made available to the purchaser or End-User(s); and
- a record of the transaction is kept for reporting purposes.

The Content Identification Layer 503 allows for the verification of the copyright and the identity of the content purchaser. The content's copyright information and identity of the content purchaser enables the source tracking of any, authorized or not, copy of the Content 113. Thus, the Content Identification Layer 503 provides a means to combat piracy.

The Content Usage Control Layer 505 ensures that the copy of the Content 113 is used in the purchaser's device according to the Store Usage Conditions 519. The Store Usage Conditions 519 may specify the number of plays and local copies allowed for the Content 113, and whether or not the Content 113 may be recorded to an external portable device. The functions in the Content Usage Control Layer 505 keep track of the content's copy/play usage and update the copy/play status.

The Content Formatting Layer 507 allows for the format conversion of the Content 113 from its native representation in the content owner's facilities into a form that is consistent with the service features and distribution means of the Secure Digital Content Electronic Distribution System 100. The conversion processing may include compression encoding and its associated preprocessing, such as frequency equalization and amplitude dynamic adjustment. For Content 113 which is audio, at the purchaser's side, the received Content 113 also needs to be processed to achieve a format appropriate for playback or transfer to a portable device.

B. Function Partitioning and Flows

The Rights Management Architectural Model is shown in FIG. 5 and this illustrates the mapping of the architectural layers to the operating components making up the Secure Digital Content Electronic Distribution System 100 and the key functions in each layer.

1. Content Formatting Layer 507

The general functions associated with the Content Formatting Layer 507 are Content Preprocessing 502 and Compression 511 at the Content Provider(s) 101, and Content De- scrambling 513 and Decompression 515 at the End-User Device(s) 109. The need for preprocessing and the examples of specific functions were mentioned above. Content Compression 511 is used to reduce the file size of the Content 113 and its transmission time. Any compression algorithm appropriate for the type of Content 113 and transmission medium can be used in the Secure Digital Content Electronic Distribution System 100. For music, MPEG ½/4, Dolby AC-2 and AC-3, Sony Adaptive Transform Coding (ATRAC), and low-bit rate algorithms are some of the typically used compression algorithms. The Content 113 is stored in the End-User Device(s) 109 in compressed form to reduce the storage size requirement. It is decompressed during active playback. De-scrambling is also performed during active playback. The purpose and type of scrambling will be described later during the discussion of the Content Usage Control Layer 505.

2. Content Usage Control Layer 505

The Content Usage Control Layer 505 permits the specification and enforcement of the conditions or restrictions imposed on the use of Content 113 use at the End-User Device(s) 109. The conditions may specify the number of plays allowed for the Content 113, whether or not a secondary copy of the Content 113 is allowed, the number of secondary copies, and whether or not the Content 113 may be copied to an external portable device. The Content Provider(s) 101 sets the allowable Usage Conditions 517 and transmits them to the Electronic Digital Content Store(s) 103 in a SC (see the License Control Layer 501 section). The Electronic Digital Content Store(s) 103 can add to or narrow the Usage Conditions 517 as long as it doesn't invalidate the original conditions set by the Content Provider(s) 101. The Electronic Digital Content Store(s) 103 then transmits all Store Usage Conditions 519 (in a SC) to the End-User Device(s) 109 and the Clearinghouse(s) 105. The Clearinghouse(s) 105 perform Usage Conditions Validation 521 before authorizing the Content 113 release to an End-User Device(s) 109.

The enforcement of the content Usage Conditions 517 is performed by the Content Usage Control Layer 505 in the End-User Device(s) 109. First, upon reception of the Content 113 copy from the Content Identification Layer 503 in the End-User Device(s) 109 marks the Content 113 with a Copy/Play Code 523 representing the initial copy/play permission. Second, the Player Application 195 cryptographically scrambles the Content 113 before storing it in the End-User Device(s) 109. The Player Application 195 generates a scrambling key for each Content item, and the key is encrypted and hidden in the End-User Device(s) 109. Then, every time the End- User Device(s) 109 accesses the Content 113 for copy or play, the End-User Device(s) 109 verifies the copy/play code before allowing the de-scrambling of the Content 113 and the execution of the play or copy. The End-User Device(s) 109 also appropriately updates the copy/play code in the original copy of the Content 113 and on any new secondary copy. The copy/play coding is performed on Content 113 that has been compressed. That is, there is no need to decompress the Content 113 before the embedding of the copy/play code.

The End-User Device(s) 109 uses a License Watermark 527 to embed the copy/play code within the Content 113. Only the End-User Player Application 195 that is knowledgeable of the embedding algorithm and the associated scrambling key is able to read or modify the embedded data. The data is invisible or inaudible to a human observer; that is, the data introduces no perceivable degradation to the Content 113. Since the watermark survives several steps of content processing, data compression, D-to-A and A-to-D conversion, and signal degradation introduced by normal content handling, the watermark stays with the Content 113 in any representation form, including analog representation. In an alternate embodiment, instead of using a License Watermark 527 to embed the copy/play code within the Content 113, the End- User Player Application 195 uses securely stored Usage Conditions 519.

3. Content Identification Layer 503

As part of the Content Identification Layer 503, the Content Provider(s) 101 also uses a License Watermark 527 to embed data in the Content 113 such as to the content identifier, content owner and other information, such as publication date and geographic distribution region. This watermark is referred to here as the Copyright Watermark 529. Upon reception, the End-User Device(s) 109 watermarks the copy of the Content 113 with the content purchaser's name and the Transaction ID 535 (see the License Control Layer 501 section below), and with other information such as date of license and Usage Conditions 517. This watermark is referred to here as the license watermark. Any copy of Content 113, obtained in an authorized manner or not, and subject to audio processing that preserves the content quality, carries the copyright and license watermarks. The Content Identification Layer 503 deters piracy.

4. License Control Layer 501

The License Control Layer 501 protects the Content 113 against unauthorized interception and ensures that the Content is only released on an individual basis to an End- User(s) that has properly licensed End-User Device(s) 109 and successfully completes a license purchase transaction with an authorized Electronic Digital Content Store(s) 103. The License Control Layer 501 protects the Content 113 by double Encryption 531. The Content 113 is encrypted using an encryption symmetric key generated by the Content Provider(s) 101, and the symmetric key is encrypted using the public key 621 of the Clearinghouse(s). Only the Clearinghouse(s) 105 can initially recover the symmetric key.

License control is designed with the Clearinghouse(s) 105 as the "trusted party". Before releasing permission for the License Request 537, (i.e. the Symmetric Key 623 for the Content 113 to an End-User Device(s) 109), the Clearinghouse(s) 105 verifies that the Transaction 541 and the License Authorization 543 are complete and authentic, that the Electronic Digital Content Store(s) 103 has authorization from the Secure Digital Content Electronic Distribution System 100 for the sale of electronic Content 113, and that the End-User(s) has a properly licensed application. Audit/Reporting 545 allows the generation of reports and the sharing of licensing transaction information with other authorized parties in the Secure Electronic Digital Content Distribution System 100

License control is implemented through SC Processing 533. SC(s) are used to distribute encrypted Content 113 and information among the system operation components (more about the SC(s) detailed structure sections below). A SC is cryptographic carrier of information that uses cryptographic encryption, digital signatures and digital certificates to provide protection against unauthorized interception and modification of the electronic information or Content 113. It also allows for the authenticity verification of the electronic data.

License control requires that the Content Provider(s) 101, the Electronic Digital Content Store(s) 103, and the Clearinghouse(s) 105 have bona-fide cryptographic digital certificates from reputable Certificate Authorities that are used to authenticate those components. The End-User Device(s) 109 are not required to have digital certificates.

C. Content Distribution and Licensing Control

FIG. 6 is a block diagram illustrating an overview of the Content Distribution and Licensing Control as it applies to the License Control Layer of FIG. 5. The figure depicts the case in which the Electronic Digital Content Store(s) 103, End-User Device(s) 109 and the Clearinghouse(s) 105 are interconnected via the Internet, and unicast (point-to-point) transmission is used among those components. The communication between the Content Provider(s) 101 and the Electronic Digital Content Store(s) 103 could also be over the Internet or other network. It is assumed that the Content-purchase commercial transaction between the End-User Device(s) 109 and the Electronic Digital Content Store(s) 103 is based on standard Internet Web protocols. As part of the Web-based interaction, the End-User(s) makes the selection of the Content 113 to purchase, provides personal and financial information, and agrees to the conditions of purchase. The Electronic Digital Content Store(s) 103 could obtain payment authorization from an acquirer institution using a protocol such as SET.

It is also assumed in FIG. 6 that the Electronic Digital Content Store(s) 103 has downloaded the End-User Player Application 195 to an End-User Device(s) 109 based on standard Web protocols. The architecture requires that the Electronic Digital Content Store(s) 103 assigns a unique application ID to the downloaded Player Application 195 and that the End- User Device(s) 109 stores it for later application license verification (see below).

The overall licensing flow starts at the Content Provider(s) 101. The Content Provider(s) 101 encrypts the Content 113 using an encryption symmetric key locally generated, and encrypts the Symmetric Key 623 using the Clearinghouse's 105 public key 621. In an alternate embodiment, the symmetric key instead of being locally generated my be sent to the Content Provider(s) 101 from the Clearinghouse(s) 105. The Content Provider(s) 101 creates a Content SC(s) 630 around the encrypted Content 113, and a Metadata SC(s) 620 around the encrypted Symmetric Key 623, Store Usage Conditions 519, and other Content 113 associated information. There is one Metadata SC(s) 620 and one Content SC(s) 630 for every Content 113 object. The Content 113 object may be a compression level one same song or the Content 113 object may be each song on the album or the Content 113 object may be the entire album. For each Content 113 object, the Metadata SC(s) 620 also carries the Store Usage Conditions 519 associated with the Content Usage Control Layer 505.

The Content Provider(s) 101 distributes the Metadata SC(s) 620 to one or more Electronic Digital Content Store(s) 103 (step 601) and the Content SC(s) 630 to one or more Content Hosting Sites (step 602). Each Electronic Digital Content Store(s) 103, in turn creates an Offer SC(s) 641. The Offer SC(s) 641 typically carries much of the same information as the Metadata SC(s) 620, including the Digital Signature 624 of the Content Provider(s) 101 and the Certificate (not shown of the Content Provider(s) 101. As mentioned above, the Electronic Digital Content Store(s) 103 can add to or narrow the Store Usage Conditions 519 (handled by the Control Usage Control Layer) initially defined by the Content Provider(s) 101. Optionally, the Content SC(s) 630 and/or the Metadata SC(s) 620 is signed with a Digital Signature 624 of the Content Provider(s) 101.

After the completion of the Content-purchase transaction between the End-User Device(s) 109 and the Electronic Digital Content Store(s) 103 (step 603), the Electronic Digital Content Store(s) 103 creates and transfers to the End-User Device(s) 109 a Transaction SC(s) 640 (step 604). The Transaction SC(s) 640 includes a unique Transaction ID 535, the purchaser's name (i.e. End-User(s)') (not shown), the Public Key 661 of the

End-User Device(s) 109, and the Offer SC(s) 641 associated with the purchased Content 113. Transaction Data 642 in FIG. 6 represents both the Transaction ID 535 and the End-User(s) name (not shown). The Transaction Data 642 is encrypted with the Public Key 621 of the Clearinghouse(s) 105. Optionally, the Transaction SC(s) 640 is signed with a Digital Signature 643 of the Electronic Digital Content Store(s) 103.

Upon reception of the Transaction SC(s) 640 (and the Offer SC(s) 641 included in it), the End-User Player Application 195 running on End-User Device(s) 109 solicits license authorization from the Clearinghouse(s) 105 by means of an Order SC(s) 650 (step 605). The Order SC(s) 650 includes the encrypted Symmetric Key 623 and Store Usage Conditions 519 from the Offer SC(s) 641, the encrypted Transaction Data 642 from the Transaction SC(s) 640, and the encrypted Application ID 551 from the End-User Device(s) 109. In another After the completion of the Content-purchase transaction between the End-User Device(s) 109 and the Electronic Digital Content Store(s) 103 (step 603), the Electronic Digital Content Store(s) 103 creates and transfers to the End-User Device(s) 109 a Transaction SC(s) 640 (step 604). The Transaction SC(s) 640 includes a unique Transaction ID 535, the purchaser's name (i.e. End-User(s)') (not shown), the Public Key 661 of the End-User Device(s) 109, and the Offer SC(s) 641 associated with the purchased Content 113. Transaction Data 642 in FIG. 6 represents both the Transaction ID 535 and the End-User(s) name (not shown). The Transaction Data 642 is encrypted with the Public Key 621 of the Clearinghouse(s) 105. Optionally, the Transaction SC(s) 640 is signed with a Digital Signature 643 of the Electronic Digital Content Store(s) 103.

Upon reception of the Transaction SC(s) 640 (and the Offer SC(s) 641 included in it), the End-User Player Application 195 running on End-User Device(s) 109 solicits license authorization from the Clearinghouse(s) 105 by means of an Order SC(s) 650 (step 605). The Order SC(s) 650 includes the encrypted Symmetric Key 623 and Store Usage Conditions 519 from the Offer SC(s) 641, the encrypted Transaction Data 642 from the Transaction SC(s) 640, and the encrypted Application ID 551 from the End-User Device(s) 109. In another embodiment, the Order SC(s) 650 is signed with a Digital Signature 652 of the End-User Device(s) 109.

Upon reception of the Order SC(s) 650 from the End-User Device(s) 109, the Clearinghouse(s) 105 verifies:

1. that the Electronic Digital Content Store(s) 103 has authorization from the Secure Digital Content Electronic Distribution System 100 (exists in the Database 160 of the Clearinghouse(s) 105);
2. that the Order SC(s) 650 has not been altered;
3. that the Transaction Data 642 and Symmetric Key 623 are complete and authentic;
4. that the electronic Store Usage Conditions 519 purchased by the End-User Device(s) 109 are consistent with those Usage Conditions 517 set by the Content Provider(s) 101; and
5. that the Application ID 551 has a valid structure and that it was provided by an authorized Electronic Digital Content Store(s) 103.

If the verifications are successful, the Clearinghouse(s) 105 decrypts the Symmetric Key 623 and the Transaction Data 642 and builds and transfers the License SC(s) 660 to the End-User Device(s) 109 (step 606).

The License SC(s) 660 carries the Symmetric Key 623 and the Transaction Data 642, both encrypted using the Public Key 661 of the End-User Device(s) 109. If any verification is not successful, the Clearinghouse(s) 105 denies the license to the End-User Device(s) 109 and informs the End-User Device(s) 109. The Clearinghouse(s) 105 also immediately informs the Electronic Digital Content Store(s) 103 of this verification failure. In an alternate embodiment, the Clearinghouse(s) 105 signs the License SC(s) 660 with its Digital Signature 663.

After receiving the License SC(s) 660, the End-User Device(s) 109 decrypts the Symmetric Key 623 and the Transaction Data 642 previously received from the Clearinghouse(s) 105 and requests the Content SC(s) 630 (step 607) from a Content Hosting Site(s) 111. Upon arrival of the Content SC(s) 630 (step 608), the End-User Device(s) 109 decrypts the Content 113 using the Symmetric Key 623 (step 609), and passes the Content 113 and the Transaction Data 642 to the other layers for license watermarking, copy/play coding, scrambling, and further Content 113 processing as described previously for FIG. 5.

Finally, the Clearinghouse(s) 105 on a periodic basis transmits summary transaction reports to the Content Provider(s) 101 and the Electronic Digital Content Store(s) 103 for auditing and tracking purposes (step 610).

V. SECURE CONTAINER STRUCTURE

A. General Structure

A Secure Container (SC) is a structure that consists of several parts which together define a unit of Content 113 or a portion of a transaction, and which also define related information such as Usage Conditions, metadata, and encryption methods. SC(s) are designed in such a way that the integrity, completeness, and authenticity of the information can be verified. Some of the information in SC(s) may be encrypted so that it can only be accessed after proper authorization has been obtained.

SC(s) include at least one bill of materials (BOM) part which has records of information about the SC(s) and about each of the parts included in the SC(s). A message digest is calculated, using a hashing algorithm such as MD-5, for each part and then included in the BOM record for the part. The digests of the parts are concatenated together and another digest is computed from them and then encrypted using the private key of the entity creating the SC(s) to create a digital signature. Parties receiving the SC(s) can use the digital signature to verify all of the digests and thus validate the integrity and completeness of the SC(s) and all of its parts.

The following information may be included as records in the BOM along with the records for each part. The SC(s) type determines which records need to be included:

- SC(s) version
- SC(s) ID
- Type of SC(s) (e.g. Offer, Order, Transaction, Content, Metadata or promotional and License.)
- Publisher of the SC(s)
- Date that the SC(s) was created

- Expiration date of the SC(s)
- Clearinghouse(s) URL
- Description of the digest algorithm used for the included parts (default is MD-5)
- Description of the algorithm used for the digital signature encryption (default is RSA)
- Digital signature (encrypted digest of all of the concatenated digests of the included parts)

SC(s) may include more than one BOM. For example, an Offer SC(s) 641 consists of the original Metadata SC(s) 620 parts, including its BOM, as well as additional information added by the Electronic Digital Content Store(s) 103 and a new BOM. A record for the Metadata SC(s) 620 BOM is included in the Offer SC(s) 641 BOM. This record includes a digest for the Metadata SC(s) 620 BOM which can be used to validate its integrity and therefore, the integrity of the parts included from the Metadata SC(s) 620 can also be validated using the part digest values stored in Metadata SC(s) 620 BOM. None of the parts from the Metadata SC(s) 620 have records in the new BOM that was created for the Offer SC(s) 641. Only parts added by the Electronic Digital Content Store(s) 103 and the Metadata SC(s) 620 BOM have records in the new BOM.

SC(s) may also include a Key Description part. Key Description parts include records that contain the following information about encrypted parts in the SC(s):

- The name of the encrypted part.
- The name to use for the part when it is decrypted.
- The encryption algorithm used to encrypt the part.
- Either a Key Identifier to indicate the public encryption key that was used to encrypt the part or an encrypted symmetric key that, when decrypted, is used to decrypt the encrypted part.
- The encryption algorithm used to encrypt the symmetric key. This field is only present when the record in the Key Description part includes an encrypted symmetric key that was used to encrypt the encrypted part.
- A Key Identifier of the public encryption key that was used to encrypt the symmetric key. This field is only present when the record in the Key Description part includes an encrypted symmetric key and the encryption algorithm identifier of the symmetric key that was used to encrypt the encrypted part.

If the SC(s) does not contain any encrypted parts, then there is no Key Description part.

B. Rights Management Language Syntax and Semantics

The Rights Management Language consists of parameters that can be assigned values to define restrictions on the use of the Content 113 by an End-User(s) after the Content 113 purchase. The restrictions on the use of the Content 113 is the Usage Conditions 517. Each Content Provider(s) 101 specifies the Usage Conditions 517 for each of its Content 113 items. Electronic Digital Content Store(s) 103 interpret the Usage Conditions 517 in Metadata SC(s) 620 and use the information to provide select options they wish to offer their customers as well as add retail purchase information for the Content 113. After an End-User(s) has selected a Content 113 item for purchase, the End-User Device(s) 109 requests authorization for the Content 113 based on Store Usage Conditions 519. Before the Clearinghouse(s) 105 sends a License SC(s) 660 to the End-User(s), the Clearinghouse(s) 105 verifies that the Store Usage Conditions 519 being requested are in agreement with the allowable Usage Conditions 517 that were specified by the Content Provider(s) 101 in the Metadata SC(s) 620.

When an End-User Device(s) 109 receives the Content 113 that was purchased, the Store Usage Conditions 519 are encoded into that Content 113 using the Watermarking Tool or encoded in the securely stored Usage Conditions 519. The End-User Player Application 195 running on End-User Device(s) 109 insures that the Store Usage Conditions 519 that were encoded into the Content 113 are enforced.

The following are examples of Store Usage Conditions 519 for an embodiment where the Content 113 is music:

- Song is recordable.
- Song can be played n number of times.

C. Overview of Secure Container Flow and Processing

Metadata SC(s) 620 are built by Content Provider(s) 101 and are used to define Content 113 items such as songs. The Content 113 itself is not included in these SC(s) because the size of the Content 113 is typically too large for Electronic Digital Content Store(s) 103 and End- User(s) to efficiently download the containers just for the purpose of accessing the descriptive metadata. Instead, the SC(s) includes an external URL (Uniform Resource Locators) to point to the Content 113. The SC(s) also includes metadata that provides descriptive information about the Content 113 and any other associated data, such as for music, the CD cover art and/or digital audio clips in the case of song Content 113.

Electronic Digital Content Store(s) 103 download the Metadata SC(s) 620, for which they are authorized, and build Offer SC(s) 641. In short, an Offer SC(s) 641 consists of some of the parts and the BOM from the Metadata SC(s) 620 along with additional information included by the Electronic Digital Content Store(s) 103. A new BOM for the Offer SC(s) 641 is created when the Offer SC(s) 641 is built. Electronic Digital Content Store(s) 103 also use the Metadata SC(s) 620 by extracting metadata information from them to build HTML pages on their web sites that present descriptions of Content 113 to End-User(s), usually so they can purchase the Content 113.

The information in the Offer SC(s) 641 that is added by the Electronic Digital Content Store(s) 103 is typically to narrow the selection of Usage Conditions 517 that are specified in the Metadata SC(s) 620 and promotional data such as a graphic image file of the store's logo and a URL to the store's web site. An Offer SC(s)

641 template in the Metadata SC(s) 620 indicates which information can be overridden by the Electronic Digital Content Store(s) 103 in the Offer SC(s) 641 and what, if any, additional information is required by the Electronic Digital Content Store(s) 103 and what parts are retained in the embedded Metadata SC(s) 620.

Offer SC(s) 641 are included in a Transaction SC(s) 640 when an End-User(s) decides to purchase Content 113 from an Electronic Digital Content Store(s) 103. The Electronic Digital Content Store(s) 103 builds a Transaction SC(s) 640 and includes Offer SC(s) 641 for each Content 113 item being purchased and transmits it to the End-User Device(s) 109. The End-User Device(s) 109 receives the Transaction SC(s) 640 and validates the integrity of the Transaction SC(s) 640 and the included Offer SC(s) 641.

An Order SC(s) 650 is built by the End-User Device(s) 109 for each Content 113 item being purchased. Information is included from the Offer SC(s) 641, from the Transaction SC(s) 640, and from the configuration files of the End-User Device(s) 109. Order SC(s) 650 are sent to the Clearinghouse(s) 105 one at a time. The Clearinghouse(s) 105 URL where the Order SC(s) 650 is included as one of the records in the BOM for the Metadata SC(s) 620 and included again in the Offer SC(s) 641.

The Clearinghouse(s) 105 validates and processes Order SC(s) 650 to provide the End- User Device(s) 109 with everything that is required to a License Watermark 527 and access purchased Content 113. One of the functions of the Clearinghouse(s) 105 is to decrypt the Symmetric Keys 623 that are needed to decrypt the watermarking instructions from the Offer SC(s) 641 and the Content 113 from the Content SC(s) 630. An encrypted Symmetric Key 623 record actually contains more than the actual encrypted Symmetric Key 623. Before executing the encryption, the Content Provider(s) 101 may optionally append its name to the actual Symmetric Key 623. Having the Content Provider(s)' 101 name encrypted together with the Symmetric Key 623 provides security against a pirate Content Provider(s) 101 that has built its own Metadata SC(s) 620 and Content SC(s) 630 from legal SC(s). The Clearinghouse(s) 105 verifies that the name of the Content Provider(s) 101 encrypted together with the Symmetric Keys 623 matches the name of the Content Provider(s) 101 in the SC(s) certificate.

If there are any changes required to be made to the watermarking instructions by the Clearinghouse(s) 105, then the Clearinghouse(s) 105 decrypts the Symmetric Key 623 and then modifies the watermarking instructions and encrypts them again using a new Symmetric Key 623. The Symmetric Key 623 is then re-encrypted using the Public Key 661 of the End-User Device(s) 109. The Clearinghouse(s) 105 also decrypts the other Symmetric Keys 623 in the SC(s) and encrypts them again with the Public Key 661 of the End-User Device(s) 109. The Clearinghouse(s) 105 builds a License SC(s) 660 that includes the newly encrypted Symmetric Keys 623 and updated watermarking instructions and sends it to the End-User Device(s) 109 in response to the Order SC(s) 650. If the processing of the Order SC(s) 650 does not complete successfully, then the Clearinghouse(s) 105 returns to the End-User Device(s) 109 an HTML page or equivalent reporting the failure of the authorization process.

A License SC(s) 660 provides an End-User Device(s) 109 with everything that is needed to access a Content 113 item. The End-User Device(s) 109 requests the appropriate Content SC(s) 630 from the Content Hosting Site(s) 111. Content SC(s) 630 are built by Content Provider(s) 101 and include encrypted Content 113

and metadata parts. The End-User Player Application 195 uses the Symmetric Keys 623 from the License SC(s) 660 to decrypt the Content 113, metadata, and watermarking instructions. The watermarking instructions are then affixed into the Content 113 and the Content 113 is scrambled and stored on the End-User Device(s) 109.

D. Metadata Secure Container 620 Format

The following table shows the parts that are included in a Metadata SC(s) 620. Each box in the Parts column is a separate object included in the SC(s) along with the BOM (with the exception of part names that are surrounded by [] characters). The BOM contains a record for each part included in the SC(s). The Part Exists column indicates whether the part itself is actually included in the SC(s) and the Digest column indicates whether a message digest is computed for the part. Some parts may not be propagated when a SC(s) is included in other SC(s) (as determined by the associated template), although the entire original BOM is propagated. This is done because the entire BOM is required by the Clearinghouse(s) 105 to verify the digital signature in the original SC(s).

The Key Description Part columns of the following table define the records that are included in the Key Description part of the SC(s). Records in the Key Description part define information about the encryption keys and algorithms that were used to encrypt parts within the SC(s) or parts within another SC(s). Each record includes the encrypted part name and, if necessary, a URL that points to another SC(s) that includes the encrypted part. The Result Name column defines the name that is assigned to the part after it is decrypted. The Encrypt Alg column defines the encryption algorithm that was used to encrypt the part. The Key Id/Enc Key column defines either an identification of the encryption key that was used to encrypt the part or a base64 encoding of the encrypted Symmetric Key 623 bit string that was used to encrypt the part. The Sym Key Alg column is an optional parameter that defines the encryption algorithm that was used to encrypt the Symmetric Key 623 when the previous column is an encrypted Symmetric Key 623. The Sym Key ID column is an identification of the encryption key that was used to encrypt the Symmetric Key 623 when the Key Id/Enc Key column is an encrypted Symmetric Key 623.

Parts	*BOM* *Part Exists*	*Digest*	*Key Description Part* *Result Name*	*Encrypt Alg*	*Key Id/ Enc Key*	*Sym Key Alg*	*Sym Key ID*
[Content URL]			Output Part	RC4	Enc Sym Key	RSA	CH Pub Key
[Metadata URL]			Output Part	RC4	Enc Sym Key	RSA	CH Pub Key
	SC Version						
	SC ID						
	SC Type						
	SC Publisher						
	Date						
	Expiration Date						
	Clearinghouse(s) URL						
	Digest Algorithm ID						
	Digital Signature Alg ID						
Content ID	Yes	Yes					
Metadata	Yes	Yes					
Usage Conditions	Yes	Yes					
SC Templates	Yes	Yes					
Watermarking Instructions	Yes	Yes	Output Part	RC4	Enc Sym Key	RSA	CH Pub Key
Key Description Part	Yes	Yes					
Clearinghouse(s) Certificate(s)	Yes	No					
Certificate(s)	Yes	No					
	Digital Signature						

The following describes the terms that are used in the above Metadata SC(s) table:

[Content URL] - A parameter in a record in the Key Description part. This is a URL that points to the encrypted Content 113 in the Content SC(s) 630 that is associated with this Metadata SC(s) 620. The Metadata SC(s) 620 itself does not contain the encrypted Content 113.

[Metadata URL] - A parameter in a record in the Key Description part. This is a URL that points to the encrypted metadata in the Content SC(s) 630 that is associated with this Metadata SC(s) 620. The Metadata SC(s) 620 itself does not contain the encrypted metadata.

Content ID - A part that defines a unique ID assigned to a Content 113 item. There is more than one Content ID included in this part if the Metadata SC(s) 620 references more than one Content 113 item.

Metadata - Parts that contain information related to a Content 113 item such as the artist name and CD cover art in the case of a song. There may be multiple metadata parts, some of which may be encrypted. The internal structure of the metadata parts is dependent on the type of metadata contained therein.

Usage Conditions - A part that contains information that describes usage options, rules, and restrictions to be imposed on an End-User(s) for use of the Content 113.

SC(s) Templates - Parts that define templates that describe the required and optional information for building the Offer, Order, and License SC(s) 660.

Watermarking Instructions - A part that contains the encrypted instructions and parameters for implementing watermarking in the Content 113. The watermarking instructions may be modified by the Clearinghouse(s) 105 and returned back to the End- User Device(s) 109 within the License SC(s) 660. There is a record in the Key Description part that defines the encryption algorithm that was used to encrypt the watermarking instructions, the output part name to use when the watermarking instructions are decrypted, a base64 encoding of the encrypted Symmetric Key 623 bit string that is was used to encrypt the watermarking instructions, the encryption algorithm that was used to encrypt the Symmetric Key 623, and the identification of the public key that is required to decrypt the Symmetric Key 623.

Clearinghouse(s) Certificate(s) - A certificate from a certification authority or from the Clearinghouse(s) 105 that contains the signed Public Key 621 of the Clearinghouse(s) 105. There may be more than one certificate, in which case a hierarchical level structure is used with the highest level certificate containing the public key to open the next lowest level certificate is reached which contains the Public Key 621 of the Clearinghouse(s) 105.

Certificate(s) - A certificate from a certification authority or from the Clearinghouse(s) 105 that contains the signed Public Key 621 of the entity that created the SC(s). There may be more than one certificate, in which case a hierarchical level structure is used with the highest level certificate containing the public key to open the next level certificate, and so on, until the lowest level certificate is reached which contains the public key of the SC(s) creator.

SC Version - A version number assigned to the SC(s) by the SC Packer Tool.

SC ID - A unique ID assigned to the SC(s) by the entity that created the SC(s).

SC Type - Indicates the type of SC(s) (e.g. Metadata, Offer, Order, etc.)

SC Publisher - Indicates the entity that created the SC(s).

Creation Date - Date that the SC(s) was created.

Expiration Date - Date the SC(s) expires and is no longer valid.

Clearinghouse(s) URL - Address of the Clearinghouse(s) 105 that the End-User Player Application 195 should interact with to obtain the proper authorization to access the Content 113.

Digest Algorithm ID - An identifier of the algorithm used to compute the digests of the parts.

Digital Signature Alg ID - An identifier of the algorithm used to encrypt the digest of the concatenated part digests. This encrypted value is the digital signature.

Digital Signature - A digest of the concatenated part digests encrypted with the public key of the entity that created the SC(s).

Output Part - The name to assign to the output part when an encrypted part is decrypted.

RSA and RC4 - Default encryption algorithms used to encrypt the Symmetric Keys 623 and data parts.

Enc Sym Key - A base64 encoding of an encrypted key bitstring that, when decrypted, is used to decrypt a SC(s) part.

CH Pub Key - An identifier that indicates that the Clearinghouse's 105 Public Key 621 was used to encrypt the data.

E. Offer Secure Container 641 Format

The following table shows the parts that are included in the Offer SC(s) 641. The parts, with the exception of some of the metadata parts, and BOM from the Metadata SC(s) 620 are also included in the Offer SC(s) 641.

Parts	*BOM* *Part Exists*	*Digest*	*Key Description Part* *Result Name*	*Encrypt Alg*	*Key ID/ Enc Key*	*Sym Key Alg*	*Sym Key ID*
Metadata SC Parts							
[Content URL]			Output Part	RC4	Enc Sym Key	RSA	CH Pub Key
[Metadata URL]			Output Part	RC4	Enc Sym Key	RSA	CH Pub Key
	SC Version						
	SC ID						
	SC Type						
	SC Publisher						
	Date						
	Expiration Date						
	Clearinghouse(s) URL						
	Digest Algorithm ID						
	Digital Signature Alg ID						
Content ID	Yes	Yes					
Metadata	Some	Yes					
Usage Conditions	Yes	Yes					
SC Templates	Yes	Yes					
Watermarking Instructions	Yes	Yes	Output Part	RC4	Enc Sym Key	RSA	CH Pub Key
Key Description Part	Yes	Yes					
Clearinghouse(s) Certificate(s)	Yes	No					
Certificate(s)	Yes	No					
	Digital Signature						
Offer SC Parts							

	SC Version						
	SC ID						
	SC Type						
	SC Publisher						
	Date						
	Expiration Date						
	Digest Algorithm ID						
	Digital Signature Alg ID						
Metadata SC BOM	Yes	Yes					
Additional and Overridden Fields	Yes	Yes					
Electronic Digital Content Store(s) Certificate	Yes	No					
Certificate(s)	Yes	No					
	Digital Signature						

The following describes the terms that are used in the above Offer SC(s) 641 that were not previously described for another SC(s):

Metadata SC(s) BOM - The BOM from the original Metadata SC(s) 620. The record in the Offer SC(s) 641 BOM includes the digest of the Metadata SC(s) 620 BOM.

Additional and Overridden Fields - Usage conditions information that was overridden by the Electronic Digital Content Store(s) 103. This information is validated by the Clearinghouse(s) 105, by means of the received SC(s) templates, to make sure that anything that the Electronic Digital Content Store(s) 103 overrides is within the scope of its authorization.

Electronic Digital Content Store(s) Certificate - A certificate provided to the Electronic Digital Content Store(s) 103 by the Clearinghouse(s) 105 and signed by the Clearinghouse(s) 105 using its private key. This certificate is used by the End-User Player Application 195 to verify that the Electronic Digital Content Store(s) 103 is a valid distributor of Content 113. The End-User Player Application 195 and Clearinghouse(s) 105 can verify that the Electronic Digital Content Store(s) 103 is an authorized distributor by decrypting the certificate's signature with the Clearinghouse's 105 Public Key 621. The End-User Player Application 195 keeps a local copy of the Clearinghouse's 105 Public Key 621 that it receives as part of its initialization during installation.

F. Transaction Secure Container 640 Format

The following table shows the parts that are included in the Transaction SC(s) 640 as well as its BOM and Key Description parts.

Parts	*BOM* *Part Exists*	*Digest*	*Result Name*	*Key Description Part* *Encrypt Alg*	*Key ID/ Enc Key*	*Sym Key Alg*	*Sym Key ID*
	SC Version						
	SC ID						
	SC Type						
	SC Publisher						
	Date						
	Expiration Date						
	Digest Algorithm ID						
	Digital Signature Alg ID						
Transaction ID	Yes	Yes	Output Part	RSA	CH Pub Key		
End-User(s) ID	Yes	Yes	Output Part	RSA	CH Pub Key		
End-User(s)' Public Key	Yes	Yes					
Offer SC(s)	Yes	Yes					
Selections of Content Use	Yes	Yes					
HTML to Display	Yes	Yes					
Key Description Part	Yes	Yes					
Electronic Digital Content Store(s) Certificate	Yes	No					
	Digital Signature						

The following describes the terms that are used in the above Transaction SC(s) 640 that were not previously described for another SC(s):

- Transaction ID 535 - An ID assigned by the Electronic Digital Content Store(s) 103 to uniquely identify the transaction.
- End-User(s) ID - An identification of the End-User(s) obtained by the Electronic Digital Content Store(s) 103 at the time the End-User(s) makes the buying selection and provides the credit card information.
- End-User(s)' Public Key - The End-User(s)' Public Key 661 that is used by the Clearinghouse(s) 105 to re-encrypt the Symmetric Keys 623. The End-User(s)' Public Key 661 is transmitted to the Electronic Digital Content Store(s) 103 during the purchase transaction.
- Offer SC(s) - Offer SC(s) 641 for the Content 113 items that were purchased.
- Selections of Content Use - An array of Usage Conditions for each Content 113 item being purchased by the End-User(s). There is an entry for each Offer SC(s) 641.
- HTML to Display - One or more HTML pages that the End-User Player Application 195 displays in the Internet browser window upon receipt of the Transaction SC(s) 640 or during the interaction between the End-User Device(s) 109 and the Clearinghouse(s) 105.

When the End-User Device(s) 109 receives a Transaction SC(s) 640, the following steps may be performed to verify the integrity and authenticity of the SC(s):

1. Verify the integrity of the Electronic Digital Content Store(s) 103 certificate using the Public Key 621 of the Clearinghouse(s) 105. The Public Key 621 of the Clearinghouse(s) 105 was stored at the End-User Device(s) 109 after it was received as part of the initialization of the End-User Player Application 195 during its installation process.
2. Verify the Digital Signature 643 of the SC(s) using the public key from the Electronic Digital Content Store(s) 103 certificate.
3. Verify the hashes of the SC(s) parts.
4. Verify the integrity and authenticity of each Offer SC(s) 641 included in the Transaction SC(s) 640.

G. Order Secure Container 650 Format

The following table shows the parts that are included in the Order SC(s) 650 as well as its BOM and Key Description parts. These parts either provide information to the Clearinghouse(s) 105 for decryption and verification purposes or is validated by the Clearinghouse(s) 105. The parts and BOM from the Offer SC(s) 641 are also included in the Order SC(s) 650. The Some string in the Part Exists column of the Metadata SC(s) BOM indicates that the some of those parts are not included in the Order SC(s) 650. The BOM from the Metadata SC(s) 620 is also included without any change so that the Clearinghouse(s) 105 can validate the integrity of the Metadata SC(s) 620 and its parts.

Parts	*BOM* *Part Exists*	*Digest*	*Result Name*	*Key Description Part* *Encrypt Alg*	*Key ID/ Enc Key*	*Sym Key Alg*	*Sym Key ID*
Metadata SC(s) Parts							
[Content URL]			Output Part	RC4	Enc Sym Key	RSA	CH Pub Key
[Metadata URL]			Output Part	RC4	Enc Sym Key	RSA	CH Pub Key
	SC(s) Version						
	SC(s) ID						
	SC(s) Type						
	SC(s) Publisher						
	Date						
	Expiration Date						
	Clearinghouse(s) URL						
	Digest Algorithm ID						
	Digital Signature Alg ID						
Content ID	Yes	Yes					
Metadata	Some	Yes					
Usage Conditions	Yes	Yes					
SC(s) Templates	Yes	Yes					
Watermarking Instructions	Yes	Yes	Output Part	RC4	Enc Sym Key	RSA	CH Pub Key
Key Description Part	Yes	Yes					
Clearinghouse(s) Certificate(s)	Yes	No					
Certificate(s)	Yes	No					
	Digital Signature						
Offer SC(s) Parts							
	SC(s) Version						
	SC(s) ID						
	SC(s) Type						
	SC(s) Publisher						
	Date						
	Expiration Date						
	Digest Algorithm ID						
	Digital Signature Alg ID						
Metadata SC(s) BOM	Yes	Yes					
Additional and Overridden Fields	Yes	Yes					
Electronic Digital Content Store(s) Certificate	Yes	No					
Certificate(s)	Yes	No					

	Digital Signature						
---------- Transaction SC(s) Parts ----------							
	SC(s) Version						
	SC(s) ID						
	SC(s) Type						
	SC(s) Publisher						
	Date						
	Expiration Date						
	Digest Algorithm ID						
	Digital Signature Alg ID						
Transaction ID	Yes	Yes	Output Part	RSA	CH Pub Key		
End-User(s) ID	Yes	Yes	Output Part	RSA	CH Pub Key		
End-User(s)' Public Key	Yes	Yes					
Offer SC(s)	One Offer SC(s)	Yes					
Selections of Content Use	Yes	Yes					
HTML to Display in Browser Wdw	Yes	Yes					
Key Description Part	Yes	Yes					
Electronic Digital Content Store(s) Certificate	Yes	No					
	Digital Signature						
---------- Order SC(s) Parts ----------							
	SC(s) Version						
	SC(s) ID						
	SC(s) Type						
	SC(s) Publisher						
	Date						
	Expiration Date						
	Digest Algorithm ID						
	Digital Signature Alg ID						
Offer SC(s) BOM	Yes	Yes					
Transaction SC(s) BOM	Yes	Yes					
Encrypted Credit Card Info	Yes	Yes	Output Part	RSA	CH Pub Key		
Key Description Part	Yes	Yes					
	Digital Signature						

The following describes the terms that are used in the above Order SC(s) 650 that were not previously described for another SC(s):

Transaction SC(s) BOM - The BOM in the original Transaction SC(s) 640. The record in the Order SC(s) 650 BOM includes the digest of the Transaction SC(s) 640 BOM.

Encrypted Credit Card Info.- Optional encrypted information from the End-User(s) that is used to charge the purchase to a credit card or debit card. This information is required when the Electronic Digital Content Store(s) 103 that created the Offer SC(s) 641 does not handle the customer billing, in which case the Clearinghouse(s) 105 may handle the billing.

H. License Secure Container 660 Format

The following table shows the parts that are included in the License SC(s) 660 as well as its BOM. As shown in the Key Description part, the Symmetric Keys 623 that are required for decrypting the watermarking instructions, Content 113, and Content 113 metadata have been re-encrypted by the Clearinghouse(s) 105 using the End-User(s)' Public Key 661. When the End-User Device(s) 109 receives the License SC(s) 660 it decrypts the Symmetric Keys 623 and use them to access the encrypted parts from the License SC(s) 660 and the Content SC(s) 630.

Parts	BOM Part Exists	BOM Digest	Key Description Part Result Name	Encrypt Alg	Key ID/ Enc Key	Sym Key Alg	Sym Key ID
[Content URL]			Output Part	RC4	Enc Sym Key	RSA	EU Pub Key
[Metadata URL]			Output Part	RC4	Enc Sym Key	RSA	EU Pub Key
	SC(s) Version						
	SC(s) ID						
	SC(s) Type						
	SC(s) Publisher						
	Date						
	Expiration Date						
	Digest Algorithm ID						
	Digital Signature Alg ID						
Content ID	Yes	Yes					
Usage Conditions	Yes	Yes					
Transaction Data	Yes	Yes					
Watermarking Instructions	Yes	Yes	Output Part	RC4	Enc Sym Key	RSA	EU Pub Key
Key Description Part	Yes	Yes					
Certificate(s)	Yes	No					
	Digital Signature						

The following describes the terms that are used in the above License SC(s) 660 that were not previously described for another SC(s):

- EU Pub Key - An identifier that indicates that the End-User(s)' Public Key 661 was used to encrypt the data.
- Order SC(s) 650 ID - The SC(s) ID taken from the Order SC(s) 650 BOM.
- Certificate Revocation List – An optional list of certificate IDs which were previously issued and signed by the Clearinghouse(s) 105, but are no longer considered to be valid. Any SC(s) that have a signature which can be verified by a certificate that is included in the revocation list are invalid SC(s). The End-User Player Application 195 stores a copy of the Clearinghouse's 105 certificate revocation list on the End-User Device(s) 109. Whenever a revocation list is received, the End-User Player Application 195 replaces its local copy if the new one is more up to date. Revocation lists includes a version number or a time stamp (or both) in order to determine which list is the most recent.

I. Content Secure Container Format

The following table shows the parts that are included in the Content SC(s) 630 as well as the BOM:

Parts	*BOM* *Part Exists*	*Digest*
	SC(s) Version	
	SC(s) ID	
	SC(s) Type	
	SC(s) Publisher	
	Date	
	Expiration Date	
	Clearinghouse(s) 105 URL	
	Digest Algorithm ID	
	Digital Signature Alg ID	
Content ID	Yes	Yes
Encrypted Content	Yes	Yes
Encrypted Metadata	Yes	Yes
Metadata	Yes	Yes
Certificate(s)	Yes	No
	Digital Signature	

The following describes the terms used in the above Content SC(s) 630 that were not previously described for another SC(s):

- Encrypted Content - Content 113 that was encrypted by a Content Provider(s) 101 using a Symmetric Key 623.
- Encrypted Metadata - Metadata associated with the Content 113 that was encrypted by a Content Provider(s) 101 using a Symmetric Key 623.

There is no Key Description part included in the Content SC(s) 630 since the keys required to decrypt the encrypted parts are in the License SC(s) 660 that is built at the Clearinghouse(s) 105.

VI. SECURE CONTAINER PACKING AND UNPACKING

A. Overview

The SC(s) Packer is a 32-bit Windows' program with an API (Application Programming Interface) that can be called in either a multiple or single step process to create a SC(s) with all of the specified parts. The SC(s) Packer 151, 152, 153 variety of hardware platforms supporting Windows' program at the Content Provider(s) 101, Clearinghouse(s) 105, Electronic Digital Content Store(s) 103 and other sites requiring SC(s) Packing. A BOM and, if necessary, a Key Description part are created and included in the SC(s). A set of packer APIs allows the caller to specify the information required to generate the records in the BOM and Key Description parts and to include parts in the SC(s). Encryption of parts and Symmetric Keys 623 as well as computing the digests and the digital signature is also be performed by the packer. Encryption and digest algorithms that are supported by the packer are included in the packer code or they are called through an external interface.

The interface to the packer for building a SC(s) is done by an API that accepts the following parameters as input:

- A pointer to a buffer of concatenated structures. Each structure in the buffer is a command to the packer with the information that is required to execute the command. Packer commands include adding a part to the SC(s) with an associated BOM record, adding a record to the BOM, and adding records to the Key Description part.
- A value indicating the number of concatenated structures contained in the above described buffer.
- Name and location of the BOM part.
- A value with each bit being a defined flag or a reserved flag for future use. The following flags are currently defined:
  - Indication as to whether all of the parts of the SC(s) should be bundled together into a single file after all of the structures in the buffer have been processed. Bundling the parts into a single object is the last step that is performed when building a SC(s).
  - Indication as to whether the digital signature is omitted from the BOM part. If this flag is not set, then the digital signature is computed right before the SC(s) is bundled into a single object.

In an alternate embodiment, the interface to the packer for building a SC(s) is done by APIs that accept the following parameters as input:

- First, an API is called to create a Bill of Materials (BOM) part by passing in pointer to a structure that consists of information that is used to initialize SC(s) settings that are denoted as IP records in the SC(s) BOM part, the name to use for the BOM part, a default location to look for parts that will be added, and a flags value. This API returns a SC(s) handle that is used in subsequent Packer APIs.
- The Packer has an API that is used whenever a part is added to a SC(s). This API accepts a SC(s) handle, which was previously returned by a previous Packer API, a pointer to a structure that consists of information about the part that is being added, and a flags value. Information about the part being added includes the name and location of the part, the name to use in the BOM for the part, the type of part that is being added, a hash value for the part, flags, etc.

After all of the parts have been added to the SC(s) a Packer API is called to pack all of the parts, including the BOM part, into a single SC(s) object, which is typically a file. This API accepts a SC(s) handle, which was previously returned by a previous Packer API, the name to use for the packed SC(s), a pointer to a structure with information for signing the SC(s), and a flags value.

Either the packer or the entity calling the packer can use a SC(s) template to build a SC(s). SC(s) templates have information that define parts and records that are required in the SC(s) that is being built. Templates can also define encryption methods and key references to use for encrypting Symmetric Keys 623 and encrypted parts.

The packer has an API that is used to unpack a SC(s). Unpacking a SC(s) is the process of taking a SC(s) and separating it into its individual parts. The packer can then be called to decrypt any of the encrypted parts that were unpacked from the SC(s).

B. Bill of Materials (BOM) Part

The BOM part is created by the packer when a SC(s) is being built. The BOM is a text file that contains records of information about the SC(s) and about the parts that are included in the SC(s). Each record in the BOM is on a single line with a new line indicating the start of a new record. The BOM usually includes digests for each part and a digital signature that can be used to validate the authenticity and integrity of the SC(s).

The record types within a BOM are as follows:

IP An IP record contains a set of Name=Value pairs pertaining to the SC(s). The following Names are reserved for specific properties of SC(s):

V major.minor.fix

The V property specifies the version of the SC(s). This is the version number of the SC(s) specification that the SC(s) was created under. The string that follows should be of the form major.minor.fix, where major, minor, and fix are the major release number, minor release number, and fix level, respectively.

ID value

The ID property is a unique value that is assigned to this specific SC(s) by the entity that is creating this SC(s). The format of the value is defined in a later version of this document.

T value

The T property specifies the type of the SC(s), which should be one of:

ORD - An Order SC(s) 650.

OFF - An Offer SC(s) 641.

LIC - A License SC(s).

TRA - A Transaction SC(s) 640.

MET - A Metadata SC(s) 620.

CON - A Content SC(s) 630.

A value

The A property identifies the author or publisher of the SC(s). Author/publisher identities should be unambiguous and/or registered with the Clearinghouse(s) 105.

D value

The D property identifies the date, and optionally, the time that the SC(s) was created. The value should be of the form yyyy/mm/dd[@hh:mm[:ss[.fsec]][(TZ)]] representing year/month/day@hour:minute:second.decimal-fraction-of-second (time-zone). Optional parts of the value are enclosed in [] characters.

E value

The E property identifies the date, and optionally, the time that the SC(s) expires. The value should be the same form used in the D property that was previously defined. The expiration date/time should be compared, whenever possible, with the date/time at the Clearinghouse(s) 105.

CCURL value

The CCURL property identifies the URL of the Clearinghouse(s) 105. The value should be of the form of a valid external URL.

H value

The H property identifies the algorithm that was used to calculate the message digests for the parts included in the SC(s). An example digest algorithm is MD5.

D A D record is a data or part entry record that contains information that identifies the type of part, the name of the part, the (optional) digest of the part, and an (optional) indication that the part is not included in the SC(s). A - sign immediately after the type identifier is used to indicate that the part is not included in the SC(s). The following are reserved types of data or part records:

K part_name [digest]

Specifies the Key Description part.

W part_name [digest]

Specifies the watermarking instructions part.

C part_name [digest]

Specifies the certificate(s) used to validate the digital signature.

T part_name [digest]

Specifies the Usage Conditions part.

YF part_name [digest]

Specifies the Template part for the Offer SC(s) 641.

YO part_name [digest]

Specifies the Template part for the Order SC(s) 650.

YL part_name [digest]

Specifies the Template part for the License SC(s) 660.

ID part_name [digest]

Specifies the ID(s) of the Content 113 of the item(s) of Content 113 being referenced.

CH part_name [digest]

Specifies the Clearinghouse(s) 105 certificate part.

SP part_name [digest]

Specifies the Electronic Digital Content Store(s) 103 certificate part.

B part_name [digest]

Specifies a BOM part for another SC(s) that has its parts or a subset of its parts included in this SC(s).

BP part_name sc_part_name [digest]

Specifies a BOM part for another SC(s) that is included as a single part in this SC(s). The sc_part_name parameter is the name of the SC(s) part that is included in this SC(s) and that this BOM part defines. A BOM that is identical to this one is also included in the SC(s) that is defined by the sc_part_name parameter.

D part_name [digest]

Specifies a data (or metadata) part.

S An S record is a signature record the is used to define the digital signature of the SC(s). The digital signature is specified as follows:

S key_identifier signature_string signature_algorithm

The S record contains the key_identifier to indicate the encryption key of the signature, the signature_string, which is the base64 encoding of the digital signature bitstring, and the signature algorithm that was used to encrypt the digest to create the digital signature.

C. Key Description Part

The Key Description part is created by the packer to provide information about encryption keys that are needed for decryption of SC(s) encrypted parts. The encrypted parts may be included in the SC(s) being built or may be in other SC(s) which are referred to by the SC(s) being built. The Key Description part is a text file that contains records of information about the encryption keys and the parts for which the encryption keys are used. Each record in the Key Description part is on a single line with a new line indicating the start of a new record.

The following record type is used within a Key Description part and is defined as follows:

K encrypted_part_name; result_part_name; part_encryption_algorithm_identifier; public_key_identifier key_encryption_algorithm and encrypted_symmetric_key.

A K record specifies an encrypted part that may be included in this SC(s) or may be included in another SC(s) that is referred to by this record. The encrypted_part_name is either the name of a part in this SC(s) or a URL pointing to the name of the encrypted part in another SC(s). The result_part_name is the name that is given to the decrypted part. The part_encryption_algorithm_identifier indicates the encryption algorithm that was used to encrypt the part. The public_key_identifier is an identifier of the key that was used to encrypt the Symmetric Key 623.

The key_encryption_algorithm_identifier indicates the encryption algorithm that was used to encrypt the Symmetric Key 623. The encrypted symmetric key is a base64 encoding of the encrypted Symmetric Key 623 bit string that was used to encrypt the part.

VII. CLEARINGHOUSE(S) 105

A. Overview

The Clearinghouse(s) 105 is responsible for the rights management functions of the Secure Digital Content Electronic Distribution System 100. Clearinghouse(s) 105 functions include enablement of Electronic Digital Content Store(s) 103, verification of rights to Content 113, integrity and authenticity validation of the buying transaction and related information, distribution of Content encryption keys or Symmetric Keys 623 to End-User Device(s) 109, tracking the distribution of those keys, and reporting of transaction summaries to Electronic Digital Content Store(s) 103 and Content Provider(s) 101. Content encryption keys are used by End-User Device(s) 109 to unlock Content 113 for which they have obtained rights, typically by a purchase transaction from an authorized Electronic Digital Content Store(s) 103. Before a Content encryption key is sent to an End-User Device(s) 109, the Clearinghouse(s) 105 goes through a verification process to validate the authenticity of the entity that is selling the Content 113 and the rights that the End-User Device(s) 109 has to the Content 113. This is called the SC Analysis Tool 185. In some configurations the Clearinghouse(s) 105 may also handle the financial settlement of Content 113 purchases by co-locating a system at the Clearinghouse(s) 105 that performs the Electronic Digital Content Store(s) 103 functions of credit card authorization and billing. The Clearinghouse(s) 105 uses OEM packages such as ICVerify and Taxware to handle the credit card processing and local sales taxes.

Electronic Digital Content Store(s) Embodiment

An Electronic Digital Content Store(s) 103 that wants to participate as a seller of Content 113 in the Secure Digital Content Electronic Distribution System 100 makes a request to one or more of the Digital Content Provider(s) 101 that provide Content 113 to the Secure Digital Content Electronic Distribution System 100. There is no definitive process for making the request so long as the two parties come to an agreement. After the digital content label such as a Music Label e.g. Sony, Time-Warner, etc. decides to allow the Electronic Digital Content Store(s) 103 to sell its Content 113, the Clearinghouse(s) 105 is contacted, usually via E-mail, with a request that the Electronic Digital Content Store(s) 103 be added to the Secure Digital Content Electronic Distribution System 100. The digital content label provides the name of the Electronic Digital Content Store(s) 103 and any other information that may be required for the Clearinghouse(s) 105 to create a digital certificate for the Electronic Digital Content Store(s) 103. The digital certificate is sent to the digital content label in a secure fashion, and then forwarded by the digital content label to the Electronic Digital Content Store(s) 103. The Clearinghouse(s) 105 maintains a database of digital certificates that it has assigned. Each certificate includes a version number, a unique serial number, the signing algorithm, the name of the issuer (e.g., the name of Clearinghouse(s) 105), a range of dates for which the certificate is considered to be valid, the name Electronic Digital Content Store(s) 103, the public key of the Electronic Digital Content Store(s) 103, and a hash code of all of the other information signed using the private key of the Clearinghouse(s) 105. Entities that have the Public Key 621 of the Clearinghouse(s) 105 can validate the certificate and then be assured that a SC(s) with a signature that can be validated using the public key from the certificate is a valid SC(s).

After the Electronic Digital Content Store(s) 103 has received its digital certificate that was created by the Clearinghouse(s) 105 and the necessary tools for processing the SC(s) from the digital content label, it can begin offering Content 113 that can be purchased by End-User(s). The Electronic Digital Content Store(s) 103 includes its certificate and the Transaction SC(s) 640 and signs the SC(s) using its Digital Signature 643. The End-User Device(s) 109 verifies that the Electronic Digital Content Store(s) 103 is a valid distributor of Content 113 on the Secure Digital Content Electronic Distribution System 100 by first checking the digital certificate revocation list and then using the Public Key 621 of the Clearinghouse(s) 105 to verify the information in the digital certificate for the Electronic Digital Content Store(s) 103. A digital certificate revocation list is maintained by the Clearinghouse(s) 105. The revocation list may be included as one of the parts in a License SC(s) 660 that is created by the Clearinghouse(s) 105. End-User Device(s) 109 keep a copy of the revocation list on the End-User Device(s) 109 so they can use it as part of the Electronic Digital Content Store(s) 103 digital certificate validation. Whenever the End-User Device(s) 109 receives a License SC(s) 660 it determines whether a new revocation list is included and if so, the local revocation list on the End-User Device(s) 109 is updated.

B. Rights Management Processing

Order SC(s) Analysis

The Clearinghouse(s) 105 receives an Order SC(s) 650 from an End-User(s) after the End-User(s) has received the Transaction SC(s) 640, which include the Offer SC(s) 641, from the Electronic Digital Content Store(s) 103. The Order SC(s) 650 consists of parts that contain information relative to the Content 113 and its use, information about the Electronic Digital Content Store(s) 103 that is selling the Content 113, and information about the End-User(s) that is purchasing the Content 113. Before the Clearinghouse(s) 105 begins processing the information in the Order SC(s) 650, it first performs some processing to insure that the SC(s) is in fact valid and the data it contains has not been corrupted in any way.

Validation

The Clearinghouse(s) 105 begins the validation of Order SC(s) 650 by verifying the digital signatures, then the Clearinghouse(s) 105 verifies the integrity of the Order SC(s) 650 parts. To validate the digital signatures, first the Clearinghouse(s) 105 decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity included if signed. (The signing entity could be the Content Provider(s) 101, the Electronic Digital Content Store(s) 103, the End User Device(s) 109 or any combination of them.) Then, the Clearinghouse(s) 105 calculates the digest of the concatenated part digests of the SC(s) and compares it with the digital signature's decrypted Content 113. If the two values match, the digital signature is valid. To verify the integrity of each part, the Clearinghouse(s) 105 computes the digest of the part and compares it to the digest value in the BOM. The Clearinghouse(s) 105 follows the same process to verify the digital signatures and part integrity for the Metadata and Offer SC(s) 641 parts included within the Order SC(s) 650.

The process of verification of the Transaction and Offer SC(s) 641 digital signatures also indirectly verifies that the Electronic Digital Content Store(s) 103 is authorized by the Secure Digital Content Electronic

Distribution System 100. This is based on the fact that the Clearinghouse(s) 105 is the issuer of the certificates. Alternately, the Clearinghouse(s) 105 would be able to successfully verify the digital signatures of the Transaction SC(s) 640 and Offer SC(s) 641 using the public key from the Electronic Digital Content Store(s) 103, but only if the entity signing the SC(s) has ownership of the associated private key. Only the Electronic Digital Content Store(s) 103 has ownership of the private key. Notice that the Clearinghouse(s) 105 does not need to have a local database of the Electronic Digital Content Store(s) 103. Since the store uses the Clearinghouse Public Key to sign the Transaction SC(s) 640 Offer SC(s) 641 public keys.

Then, the Store Usage Conditions 519 of the Content 113 which the End-User(s) is purchasing are validated by the Clearinghouse(s) 105 to insure that they fall within the restrictions that were set in the Metadata SC(s) 620. Recall that the Metadata SC(s) 620 is included within the Order SC(s) 650.

Key Processing

Processing of the encrypted Symmetric Keys 623 and of the watermarking instructions are done by the Clearinghouse(s) 105 after authenticity and the integrity check of the Order SC(s) 650, the validation of the Electronic Digital Content Store(s) 103, and the validation of the Store Usage Conditions 519 have been completed successfully. The Metadata SC(s) 620 portion of the Order SC(s) 650 typically has several Symmetric Keys 623 located in the Key Description part that were encrypted using the Public Key 621 of the Clearinghouse(s) 105. Encryption of the Symmetric Keys 623 are done by the Content Provider(s) 101 when the Metadata SC(s) 620 was created.

One Symmetric Key 623 are used for decrypting the watermarking instructions and the others for decrypting the Content 113 and any encrypted metadata. Since Content 113 can represent a single song or an entire collect of songs on a CD, a different Symmetric Key 623 may be used for each song. The watermarking instructions are included within the Metadata SC(s) 620 portion in the Order SC(s) 650. The Content 113 and encrypted metadata are in the Content SC(s) 630 at a Content Hosting Site(s) 111. The URL and part names of the encrypted Content 113 and metadata parts, within the Content SC(s) 630, are included in the Key Description part of the Metadata SC(s) 620 portion of the Order SC(s) 650. The Clearinghouse(s) 105 uses its private key to decrypt the Symmetric Keys 623 and then encrypts each of them using the Public Key 661 of the End-User Device(s) 109. The Public Key 661 of the End-User Device(s) 109 is retrieved from the Order SC(s) 650. The new encrypted Symmetric Keys 623 are included in the Key Description part of the License SC(s) 660 that the Clearinghouse(s) 105 returns to the End-User Device(s) 109.

During the time of processing the Symmetric Keys 623, the Clearinghouse(s) 105 may want to make modifications to the watermarking instructions. If this is the case, then after the Clearinghouse(s) 105 decrypts the Symmetric Keys 623, the watermarking instructions are modified and re-encrypted. The new watermarking instructions are included as one of the parts within the License SC(s) 660 that gets returned to the End-User Device(s) 109.

If all of the processing of the Order SC(s) 650 is successful, then the Clearinghouse(s) 105 returns a License SC(s) 660 to the End-User Device(s) 109. The End-User Device(s) 109 uses the License SC(s) 660 information to download the Content SC(s) 630 and access the encrypted Content 113 and metadata. The watermarking instructions are also executed by the End-User Device(s) 109.

If the Clearinghouse(s) 105 is not able to successfully process the Order SC(s) 650, then an HTML page is returned to the End-User Device(s) 109 and displayed in an Internet browser window. The HTML page indicates the reason that the Clearinghouse(s) 105 was unable to process the transaction.

In an alternate embodiment, if the user has purchased a copy of the Content 113 prior to the release date set for the sale, the License(s) SC 660 is returned without the Symmetric Keys 623. The License(s) SC 660 is returned to the Clearinghouse(s) 105 on or after the release date to receive the Symmetric Keys 623. As an example, the Content Provider(s) 101 allow users to download a new song prior to the release date for the song to enable customers to download the song and be prepared to play the song before a date set by the Content Provider(s) 101. This allows immediate opening of the Content 113 on the release date without having to content for bandwidth and download time on the release date.

C. Country Specific Parameters

Optionally, the Clearinghouse(s) 105 uses the domain name of the End-User Device(s) 109 and, whenever possible, the credit card billing address to determine the country location of the End-User(s). If there are any restrictions for the sale of Content 113 in the country where the End-User(s) resides, then the Clearinghouse(s) 105 insures that the transaction being processed is not violating any of those restrictions before transmitting License SC(s) 660 to the End-User Device(s) 109. The Electronic Digital Content Store(s) 103 is also expected to participate in managing the distribution of Content 113 to various countries by performing the same checks as the Clearinghouse(s) 105. The Clearinghouse(s) 105 does whatever checking that it can in case the Electronic Digital Content Store(s) 103 is ignoring the country specific rules set by the Content Provider(s) 101.

D. Audit Logs and Tracking

The Clearinghouse(s) 105 maintains a Audit Logs 150 of information for each operation that is performed during Content 113 purchase transactions and report request transactions. The information can be used for a variety of purposes such as audits of the Secure Digital Content Electronic Distribution System 100, generation of reports, and data mining.

The Clearinghouse(s) 105 also maintains account balances in Billing Subsystem 182 for the Electronic Digital Content Store(s) 103. Pricing structures for the Electronic Digital Content Store(s) 103 is provided to the Clearinghouse(s) 105 by the digital content labels. This information can include things like current specials, volume discounts, and account deficit limits that need to be imposed on the Electronic Digital Content Store(s) 103. The Clearinghouse(s) 105 uses the pricing information to track the balances of the Electronic Digital Content Store(s) 103 and insure that they do not exceed their deficit limits set by the Content Provider(s) 101.

The following operations are typically logged by the Clearinghouse(s) 105:

- End-User Device(s) 109 requests for License SC(s) 660
- Credit card authorization number when the Clearinghouse(s) 105 handles the billing
- Dispersement of License SC(s) 660 to End-User Device(s) 109
- Requests for reports
- Notification from the End-User(s) that the Content SC(s) 630 and License SC(s) 660 were received and validated

The following information is typically logged by the Clearinghouse(s) 105 for a License SC(s) 660:

- Date and time of the request
- Date and time of the purchase transaction
- Content ID of the item being purchased
- Identification of the Content Provider(s) 101
- Store Usage Conditions 519
- Watermarking instruction modifications
- Transaction ID 535 that was added by the Electronic Digital Content Store(s) 103
- Identification of the Electronic Digital Content Store(s) 103
- Identification of the End-User Device(s) 109
- End-User(s) credit card information (if the Clearinghouse(s) 105 is handling the billing)

The following information is typically logged by the Clearinghouse(s) 105 for an End- User's credit card validation:

- Date and time of the request
- Amount charged to the credit card
- Content ID of the item being purchased
- Transaction ID 535 that was added by the Electronic Digital Content Store(s) 103
- Identification of the Electronic Digital Content Store(s) 103
- Identification of the End-User(s)
- End-User(s) credit card information
- Authorization number received from the clearer of the credit card

The following information is typically logged by the Clearinghouse(s) 105 when a License SC(s) 660 is sent to an End-User Device(s) 109:

- Date and time of the request
- Content ID of the item being purchased
- Identification of Content Provider(s) 101
- Usage Conditions 517
- Transaction ID 535 that was added by the Electronic Digital Content Store(s) 103
- Identification of the Electronic Digital Content Store(s) 103
- Identification of the End-User(s)

The following information is typically logged when a report request is made:

- Date and time of the request
- Date and time the report was sent out
- Type of report being requested
- Parameters used to generate the report
- Identifier of the entity requesting the report

E. Reporting of Results

Reports are generated by the Clearinghouse(s) 105 using the information that the Clearinghouse(s) 105 logged during End-User(s) purchase transactions. Content Provider(s) 101 and Electronic Digital Content Store(s) 103 can request transaction reports from the Clearinghouse(s) 105 via a Payment Verification Interface 183 so they can reconcile their own transaction databases with the information logged by the Clearinghouse(s) 105. The Clearinghouse(s) 105 can also provide periodic reports to the Content Provider(s) 101 and Electronic Digital Content Store(s) 103.

The Clearinghouse(s) 105 defines a secure electronic interface which allows Content Provider(s) 101 and Electronic Digital Content Store(s) 103 to request and receive reports. The Report Request SC(s) includes a certificate that was assigned by the Clearinghouse(s) 105 to the entity initiating the request. The Clearinghouse(s) 105 uses the certificate and the SC's digital signature to verify that the request originated from an authorized entity. The request also includes parameters, such as time duration, that define the scope of the report. The Clearinghouse(s) 105 validates the request parameters to insure that requesters can only receive information for which they are permitted to have.

If the Clearinghouse(s) 105 determines that the Report Request SC(s) is authentic and valid, then the Clearinghouse(s) 105 generates a report and pack it into a Report SC(s) to be sent to the entity that initiated the request. Some reports may be automatically generated at defined time intervals and stored at the Clearinghouse(s) 105 so they can be immediately sent when a request is received. The format of the data included in the report is defined in a later version of this document.

F. Billing and Payment Verification

Billing of Content 113 can be handled either by the Clearinghouse(s) 105 or by the Electronic Digital Content Store(s) 103. In the case where the Clearinghouse(s) 105 handles the billing of the electronic Content 113, the Electronic Digital Content Store(s) 103 separates the End-User(s)' order into electronic goods and, if applicable, physical goods. The Electronic Digital Content Store(s) 103 then, notifies the Clearinghouse(s) 105 of the transaction, including the End-User(s)' billing information, and the total amount that needs to be authorized. The Clearinghouse(s) 105 authorizes the End-User(s)' credit card and returns a notification back to the Electronic Digital Content Store(s) 103. At the same time the Clearinghouse(s) 105 is authorizing the End-User(s)' credit card, the Electronic Digital Content Store(s) 103 can charge the End-User(s)' credit card for any physical goods that are being purchased. After each electronic item is downloaded by the End-User Device(s) 109, the Clearinghouse(s) 105 is notified so the End-User(s)' credit card can be charged. This occurs as the last step by the End- User Device(s) 109 before the Content 113 is enabled for use at the End-User Device(s) 109.

In the case where the Electronic Digital Content Store(s) 103 handles the billing of the electronic Content 113, the Clearinghouse(s) 105 is not notified about the transaction until the End-User Device(s) 109 sends the Order SC(s) 650 to the Clearinghouse(s) 105. The Clearinghouse(s) 105 is still notified by the End-User Device(s) 109 after each electronic item is downloaded. When the Clearinghouse(s) 105 is notified it sends a notification to the Electronic Digital Content Store(s) 103 so that the Electronic Digital Content Store(s) 103 can charge the End-User(s)' credit card.

G. Retransmissions

The Secure Digital Content Electronic Distribution System 100 provides the ability to handle retransmissions of Content 113. This is typically performed by a Customer Service Interface 184. Electronic Digital Content Store(s) 103 provides a user interface that the End- User(s) can step through in order to initiate a retransmission. The End-User(s) goes to the Electronic Digital Content Store(s) 103 site where the Content 113 item was purchased in order to request a retransmission of the Content 113.

Retransmissions of Content 113 are done when an End-User(s) requests a new copy of a previously purchased Content 113 item because the Content 113 could not be downloaded or the Content 113 that was downloaded is not usable. The Electronic Digital Content Store(s) 103 determines whether the End-User(s) is entitled to do a retransmission of the Content 113. If the End-User(s) is entitled to a retransmission, then the Electronic Digital Content Store(s) 103 builds a Transaction SC(s) 640 that includes the Offer SC(s) 641 of the Content 113 item(s) being retransmitted. The Transaction SC(s) 640 is sent to the End-User Device(s) 109 and the identical steps as for a purchase transaction are performed by the End-User(s). If the End-User Device(s) 109 has a scrambled key(s) in the key library for the Content 113 item(s) undergoing retransmission, then the Transaction SC(s) 640 includes information that instructs the End-User Device(s) 109 to delete the scrambled key(s).

In the case where the Clearinghouse(s) 105 handles the financial settlement of Content 113 purchases, the Electronic Digital Content Store(s) 103 includes a flag in the Transaction SC(s) 640 that is carried forward to the

Clearinghouse(s) 105 in the Order SC(s) 650. The Clearinghouse(s) 105 interprets the flag in the Order SC(s) 650 and proceed with the transaction without charging the End-User(s) for the purchase of the Content 113.

VIII. CONTENT PROVIDER

A. Overview

The Content Provider(s) 101 in the Secure Digital Content Electronic Distribution System 100 is the digital content label or the entity who owns the rights to the Content 113. The role of the Content Provider(s) 101 is to prepare the Content 113 for distribution and make information about the Content 113 available to Electronic Digital Content Store(s) 103 or retailers of the downloadable electronic versions of the Content 113. To provide the utmost security and rights control to the Content Provider(s) 101, a series of tools are provided to enable the Content Provider(s) 101 to prepare and securely package their Content 113 into SC(s) at their premises so that the Content 113 is secure when it leaves the Content Provider(s)' 101 domain and never exposed or accessible by unauthorized parties. This allows Content 113 to be freely distributed throughout a non-secure network, such as the Internet, without fear of exposure to hackers or unauthorized parties.

The end goal of the tools for the Content Provider(s) 101 is to prepare and package a Content 113 such as a song or series of songs into Content SC(s) 630 and to package information describing the song, approved uses of the song (content Usage Conditions 517), and promotional information for the song into a Metadata SC(s) 620. To accomplish this, the following set of tools are provided:

- Work Flow Manager 154 - Schedules processing activities and manages the required synchronization of processes.
- Content Processing Tools 155 - A collection of tools to control Content 113 file preparation including Watermarking, Preprocessing (for an audio example any required equalization, dynamics adjustment, or re-sampling) encoding and compression.
- Metadata Assimilation and Entry Tool 161 - A collection of tools used to gather Content 113 description information from the Database 160 of the Content Provider(s) and/or third party database or data import files and/or via operator interaction and provides means for specifying content Usage Conditions 517. Also provided is an interface for capturing or extracting content such as digital audio content for CDS or DDP files. A Quality Control Tool enables to preview of prepared content and metadata. Any corrections needed to the metadata or resubmission of the content for further processing can be conducted.
- SC(s) Packer Tool 152 - Encrypts and packages all Content 113 and information and calls the SC(s) Packer to pack into SC(s).
- Content Dispersement Tool (not shown) - Disperses SC(s) to designated distribution centers, such as Content Hosting Site(s) 111 and Electronic Digital Content Store(s) 103.
- Content Promotions Web Site 156 - stores Metadata SC(s) 620 and optionally additional promotional material for download by authorized Electronic Digital Content Store(s) 103.

B. Work Flow Manager 154

The purpose of this tool is to schedule, track, and manage Content 113 processing activities. This application enables multi-user access as well as allowing scheduling of Content 113 and status checking from remote locations within the Intranet or extranet of the Content Provider(s) 101. This design also allows for collaborative processing where multiple individuals can be working on multiple pieces of Content 113 in parallel and different individuals can be assigned specific responsibilities and these individuals can be spread throughout the world.

Turning now to FIG. 8 is a block diagram of the major processes of the Work Flow Manager 154 corresponding to FIG. 7. The major processes in FIG. 8 summarizes the Content 113 processing functions provided by the tools described in this section. The Work Flow Manager 154 is responsible for feeding jobs to these processes and directing jobs to the next required process upon completion of its current process. This is accomplished through a series of Application Programming Interfaces (APIs) which each processing tool calls to:

- retrieve the next job to process
- indicate successful completion of a process
- indicate unsuccessful completion of a process and reason for the failure
- provide interim status of a process (to allow initiation of processes that require only partial completion of a dependent process)
- add comments to a product which are made available to the designated processes

The Work Flow Manager 154 also has a user interface, an example Work Flow Manager User Interface 700 is illustrated in FIG. 7 which provides the following functions:

- a configuration panel to allow specification of default values and conditions to be assigned and performed during various stages of processing
- customization of the work flow rules and automated processing flows
- job scheduling
- status queries and reports
- add comments or instructions for a job associated to one or more processes
- job management (i.e. suspend, release, remove, change priority (order of processing))

Each process has a queue associated with it managed by the Work Flow Manager 154. All processes requesting jobs from the Work Flow Manager 154 results in the Work Flow Manager 154 either suspending the process (tool) in a wait state if there are no jobs currently in its associated queue or returning to the process all information about the job needed to perform its respective process. If a process is suspended in a wait state, it resumes processing when a job is placed on its queue by the Work Flow Manager 154.

The Work Flow Manager 154 also manages the flow or order of processing based on a set of defined rules. These rules can be customized by the Content Provider(s) 101 if it has special processing requirements or configures specific defaults rules. When a process reports completion of its assigned task, it notifies the Work

Flow Manager 154 of this status and the Work Flow Manager 154 decides what queue the job gets placed on next based on the defined rules.

Comments indicating special handling instructions or notices may also be attached to the product at any of the processing steps via either the programming API or manually through the Work Flow Manager User Interface 700 or processor interfaces.

The processes in the Work Flow Manager 154 are implemented in Java in the preferred embodiment but other programming languages such as C/C++, Assembler and equivalent can be used. It should be understood that the processes described below for the Work Flow Manager 154 can run on a variety of hardware and software platforms. The Work Flow Manager 154 as a complete system or as any of it's constitute processes may be distributed as an application program in a computer readable medium including but not limited to electronic distribution such as the web or on floppy diskettes, CD ROMS and removable hard disk drives.

Turning now to FIG. 8 is a block diagram of the major processes of the Work Flow Manager 154 corresponding to FIG. 7. The following sections summarize each process and describes the information or action required by each process.

1. Products Awaiting Action/Information Process 801

Jobs are placed on specific processes queues once all information required by that process is available and the job has already successfully completed all dependent processing. A special queue exists in the Work Flow Manager 154 which is used to hold jobs that are not currently available for processing due to missing information or a failure that prevent further processing. These jobs are placed in the Products Awaiting Action/Information Process 801 queue. Each job in this queue has associated status to indicate the action or information it is waiting on, the last process that worked on this job, and the next process(es) this job is queued to once the missing or additional information is provided or the required action is successfully completed.

Completion of any process causes the Work Flow Manager 154 to check this queue and determine if any job in this queue was awaiting the completion of this process (action) or information provided by this process. If so, that job is queued to the appropriate process queue.

2. New Content Request Process 802

The Content Provider(s) 101 determines those products (for example, a product may be a song or a collection of songs) it wishes to sell and deliver electronically. The initial function of the Work Flow Manager 154 is to enable an operator to identify these products and to place them on the queue of the New Content Request Process 802. The Content Provider(s) 101 may specify through configuration options, what information is prompted for on the product selection interface. Enough information is entered to uniquely identify the product. Optionally, additional fields may be included to request manual entry of the information required to initiate the audio processing phase in parallel with the metadata acquisition. If not provided manually, this information can optionally be retrieved from default configuration settings or from the Database 160 of the Content Provider(s), obtained in the first stage of Metadata Processing as in Automatic Metadata Acquisition Process 803. The makeup and capabilities of the Content 113 in the Database 160 of the Content Provider(s) determines the Content selection process.

If the required information needed to perform a query to the Database 160 of the Content Provider(s) 101 is specified, the job is processed by the Automatic Metadata Acquisition Process 803. In a music embodiment, to properly schedule the product for audio processing, the product's genre and the desired compression levels are specified as well as the audio PCM or WAV filename(s). This information may be entered as part of the product selection process or selected via a customized query interface or Web browser function. Specification of this information enables the product to be scheduled for content processing.

The product selection user interface provides an option enabling the operator to specify whether the product can be released for processing or whether it are held pending further information entry. If held, the job is added to the queue of the New Content Request Process 802 awaiting further action to complete data entry and/or release the product for processing. Once the product is released, the Work Flow Manager 154 evaluates the information specified and determines which processes the job is ready to be passed to.

If adequate information is provided to enable an automated query to the Database 160 of the Content Provider(s)' 101, the job is queued for Automatic Metadata Acquisition Process 803. If the database mapping table has not been configured for the Automatic Metadata Acquisition Process 803, the job is queued for Manual Metadata Entry Process 804 (see Automatic Metadata Acquisition Process 803 section for details on the Database Mapping Table).

If the required general information for audio processing and the specific information required for watermarking is specified, the job is queued for Watermarking Process 808 (the first phase of content processing). If any of the required information is missing when the job is released, the job is queued to the queue of the Products Awaiting Action/Information Process 801 along with status indicating the information that is missing.

If the status indicates that the filename of the Content 113, for example where the Content 113 is audio and the PCM or WAV file is missing, this may indicate that a capture (or digital extraction from digital media) is required. The audio processing functions require that the song files be accessible via a standard file system interface. If the songs are located on external media or a file system that is not directly accessible to the audio processing tools, the files are first be copied to an accessible file system. If the songs are in digital format but on

CD or Digital Tape, they are extracted to a file system accessible to the audio processing tools. Once the files are accessible, the Work Flow Manager User Interface 700 is used to specify or select the path and filename for the job so that it can be released to the watermarking process, assuming all other information required for watermarking has also been specified.

3. Automatic Metadata Acquisition Process 803

The Automatic Metadata Acquisition Process 803 performs a series of queries to the Database 160 of the Content Provider(s) 101 or a staging database where data has been imported, in an attempt to obtain as much of the product information as possible in an automated fashion. The Automatic Metadata Acquisition Process 803 requires the following information prior to allowing items to be placed on its queue:

- database mapping table with adequate information to generate queries to the Database 160 of the Content Provider(s) 101
- product information required to perform queries
- adequate product information to uniquely define product

An automated query is performed to the Database 160 of the Content Provider(s) 101 to obtain the information necessary to process this Content 113. For example, if the Content 113 is music, the information needed to perform this query could be the album name or may be a UPC or a specific album or selection ID as defined by the Content Provider(s) 101. Of the information to be obtained, some is designated as required (see the section on Automatic Metadata Acquisition Process 803 for details). If all required information is obtained, the job is next queued for Usage Conditions Process 805. If any required information is missing, the song is queued for Manual Metadata Entry Process 804. If any jobs in the Products Awaiting Action/Information Process 801 queue are waiting for any of the information obtained in this step, the jobs status is updated to indicate that it is no longer waiting for this information. If that job no longer has any outstanding requirements, it is queued to the next defined queue.

4. Manual Metadata Entry Process 804

The Manual Metadata Entry Process 804 provides a means for an operator to enter missing information. It has no dependencies. Once all required information is specified, the job is queued for Usage Conditions Process 805.

5. Usage Conditions Process 805

The Usage Conditions Process 805 allows specification of product uses and restrictions. The Usage Conditions Process 805 may require some metadata. Upon completion of Usage Conditions specifications, the job is eligible to be queued for Metadata SC(s) Creation Process 807 unless the Supervised Release Process 806 option has been requested or is configured as the default in the Work Flow Manager 154 rules. In that case, the job is queued for Supervised Release Process 806. Before queuing to Metadata SC(s) Creation Process 807, the Work Flow Manager 154 will first assure that all dependencies for that process have been met (see below). If not, the job is queued to the Products Awaiting Action/Information Process 801.

6. Supervised Release Process 806

The Supervised Release Process 806 allows a quality check and validation of information specified for the digital content product. It does not have any dependencies. Comments previously attached to the job at any stage of the processing for this product can be reviewed by the Supervisor and appropriate action taken. After reviewing all information and comments, the Supervisor has the following options:

- approve release and queue the product for Metadata SC(s) Creation Process 807
- modify and/or add information and queue the product for Metadata SC(s) Creation Process 807
- add comments to the job and re-queue for Manual Metadata Entry Process 804
- add comments and queue the job to the queue for Products Awaiting Action/Information Process 801

7. Metadata SC(s) Creation Process 807

The Metadata SC(s) Creation Process 807 gathers together all the information collected above as well as other information required for the Metadata SC(s) 620 and calls the SC(s) Packer Process to create the Metadata SC(s) 620. This tool requires the following as input:

- the required metadata
- the usage conditions
- the encryption keys used in the encryption stage of all quality levels for this product

This last dependency requires that the associated audio objects completed the audio processing phase before the Metadata SC(s) 620 can be created. Upon completion of the Metadata SC(s) Creation Process 807, the job is queued to either the queue for Final Quality Assurance Process 813 or Content Dispersement Process 814 based on defined work flow rules.

8. Watermarking Process 808

The Watermarking Process 808 adds copyright and other information to the Content 113. For an embodiment where the Content 113 is a song, this tool requires the following as input:

- song filename(s) (multiple filenames if album)
- watermarking instructions
- watermarking parameters (information to be included in the watermark)

Upon completion of the Watermarking Process 808, the job is queued for Preprocessing and Compression Process 809 if its required input is available or otherwise queued to the Products Awaiting Action/Information Process 801.

9. Preprocessing and Compression Process 809

The Preprocessing and Compression Process 809 encodes the Content 113 to the specified compression level performing any required preprocessing first. Queuing a job to this queue actually create multiple queue entries. A job is created for each compression level of the product desired. The encoding processes can be performed in parallel on multiple systems. This tool requires the following input:

- watermarked content filename(s) (multiple filenames if Content 113 is an album)
- quality levels for product (could be preconfigured)
- compression algorithm (could be preconfigured)
- product genre (if required by preprocessor)

Upon completion of the encoding process, the jobs are queued to the Content Quality Control Process 810 if configured by the work flow rules. If not, the jobs are queued for Encryption Process 811.

If third party providers of encoding tools do not provide a method to display the percentage of the Content 113, such as audio, that has been processed or a method to indicate the amount of Content 113 that has been encoded as a percentage of the entire selection of Content 113 selected, in FIG. 11 there is shown a flow diagram 1100 of a method to determine the encoding rate of Digital Content for the Content Preprocessing and Compression tool of FIG. 8. The method begins with the selection of the desired encoding algorithm and a bit rate, step 1101. Next, a query is made to determine if this algorithm and encoding rate has a previously calculated rate factor, step 1102. The rate factor is the factor used to determine the rate of compression for a specific encoding algorithm and a specific bit rate. If no previously calculated rate factor is stored, a sample of the Content 113 is encoded for a predetermined amount of time. The predetermined period of time in the preferred embodiment is a few seconds. This rate of encoding for a predetermined period of time is used to calculate a new rate factor R_{NEW}. Calculating a new rate factor R_{NEW} knowing the amount of time and the amount of Content 113 encoded is R_{NEW} = (length of Digital Content encoded)/(amount of time), step 1108. The Content 113 is encoded and the encoding status is displayed using the previously calculate rate factor R_{NEW}, step 1109. This encoding rate factor R_{NEW} is then stored, step 1107, for future use for this encoding algorithm and encoding bit rate. If the selected algorithm has a previously calculated rate factor R_{STORED}, step 1103. The Content 113 is encoded and the progression displayed using the previously calculated rate factor R_{STORED}, step 1104. In the meantime, a current rate factor, $R_{current}$ is calculated for this selected algorithm and bit rate, step 1105. This current rate factor $R_{current}$ is used to update the stored rate factor R_{NEW} = AVERAGE OF (R_{STORED} + $R_{CURRENT}$), step 1106. The iterative update of the rate factor enables the determination of the encoding rate to become more and more accurate with each subsequent use for a particular encoding algorithm and bit rate. The new rate R_{NEW} is then stored for future use,

step 1107. The updating of R_{STORED} may not be made if the current rate factor $R_{current}$ is out range for the previously stored rate factor R_{STORED} by a given range or threshold.

The display of the encoding status can then be presented. The encoding status includes along with the current encoding rate, the display of the percentage of the total Content 113 displayed as a progression bar based on the encoding rate and the total length of the file for the Content 113. The encoding status can also include the time remaining for the encoding. The time remaining for the encoding can be calculated by dividing the encoding rate calculated $R_{CURRENT}$ by the total length of the file for Content 113. The encoding status can be transferred to another program that may invoke the calling process. This can help supervisory programs to encoding or co-dependent programs on encoding be operated and be batched for processing more efficiently. It should be understood, in an alternative embodiment, that encoding can include the step of watermarking.

10. Content Quality Control Process 810

The Content Quality Control Process 810 is similar in function to the Supervised Release Process 806. It is an optional step allowing someone to validate the quality of the content processing performed thus far. This has no dependencies other than completion of the Watermarking Process 808 and the encoding portion of the Preprocessing and Compression Process 809. Upon completion of the Content Quality Control Process 810 the following options are available:

- the jobs can be released and queued for Encryption Process 811.
- comments can be attached and one or more of the jobs re-queued for Preprocessing and Compression Process 809.

The last option requires that the unencoded watermarked version of the song file remain available until after Content Quality Control Process 810.

11. Encryption Process 811

The Encryption Process 811 calls the appropriate Secure Digital Content Electronic Distribution Rights Management function to encrypt each of the watermarked/encoded song files. This process has no dependencies other than completion of all other audio processing. Upon completion of the Encryption Process 811 process, the job is queued for Content SC(s) Creation Process 812.

12. Content SC(s) Creation Process 812

The Content SC(s) Creation Process 812 Process may require some metadata files to be included in the Content SC(s) 630. If files other than the Content 113 are required, the files are gathered and the SC(s) Packer Process is called to create a Content SC(s) 630 for each compression level of the Content 113 (e.g. a song) created. Upon completion of the Content SC(s) Creation Process 812, the song is queued to either the Final Quality Assurance Process 813 or Content Dispersement Process 814 queue based on defined work flow rules.

13. Final Quality Assurance Process 813

Final Quality Assurance Process 813 is an optional step that allows a cross reference check between the associated Metadata and Content SC(s) 630 to verify that they match up correctly and that all information and Content 113 contained therein are correct. Upon completion of Final Quality Assurance Process 813, the jobs are queued for Content Dispersement Process 814. If a problem is found, the job in most cases has to be re-queued to the failing stage. Rework at this stage is much more costly since the product has to go through re-encryption and repacking in addition to the reprocessing required to correct the problem. It is highly recommended that the prior assurance stages be used to assure the quality of the Content 113 and accuracy and completeness of the information.

14. Content Dispersement Process 814

The Content Dispersement Process 814 Process is responsible for transferring the SC(s) to the appropriate hosting sites. After the successful transfer of the SC(s), the job completion status is logged and the job is deleted from the queue. If a problem occurs in transferring the SC(s), after a defined number of retries, the job is flagged in the Workflow Manager Tool 154 as having failed along with the error encountered.

15. Work Flow Rules

The Work Flow Rules for FIG. 8 operate in three major systems as follows:

A: Work Flow Manager Tool 154
1. New Content Request Process 802
2. Products Awaiting Action/Information Process 801
3. Final Quality Assurance Process 813
4. Content Dispersement (and Notification) Process 814

B: Metadata Assimilation and Entry Tool 161
1. Automatic Metadata Acquisition Process 803
2. Manual Metadata Entry Process 804
3. Supervised Release Process 806
4. Metadata SC(s) Creation Process 807

C: Content Processing Tools 155
1. Watermarking Process 808 (requires copyright data)
2. Preprocessing and Compression Process 809
3. Content Quality Control Process 810
4. Encryption Process 811
5. Content SC(s) Creation Process 812

Work Flow

The Content 113 selection operator inputs a new product and it starts out queued onto **A1** (New Content Request Process 802).

A1: When the Content 113 selection operator releases it to the Work Flow Manager Tool 154, then it gets queued onto **B1** (the Automatic Metadata Acquisition Process 803).

A2: coming from step **B1** (the Automatic Metadata Acquisition Process 803),
or step **B2** (Manual Metadata Entry Process 804),
or step **B3** (Supervised Release Process 806)
on its way to step **Before** (the Metadata SC(s) Creation Process 807)
[needs the encryption keys].

coming from step **Before** (the Metadata SC(s) Creation Process 807)
on its way to either step **A3** (the Final Quality Assurance Process 813) or step **A4** (the Content Dispersement Process 814)
[needs the Content SC(s) 630].

coming from step **C1** (the Watermarking Process 808)
on its way to step **C2** (the Preprocessing and Compression Process 809)
[needs the metadata for Preprocessing and Compression Process 809].

coming from step **C4** (the Encryption Process 811)
on its way to step **C5** (the Content SC(s) Creation Process 812)
[needs the metadata for Content SC(s) 630 Packing].

coming from step **C5** (the Content SC(s) Creation Process 812)
on its way to either step **A3** (the Final Quality Assurance Process 813) or step **A4** (the Content Dispersement Process 814)
[needs the Metadata SC(s) 620].

A3: After step A3 (the Final Quality Assurance Process 813),
place onto queue **B2** (Manual Metadata Entry Process 804),
or place onto queue **B3** (Supervised Release Process 806),
or place into queue as required by the quality assurance operator.

A4: After step **A4** (Content Dispersement Process 814),
the Work Flow Manager Tool 154 is done for this product.

B1: After step **B1** (the Automatic Metadata Acquisition Process 803),
if the metadata needed for step **C1** (the Watermarking Process 808) is present, then place an entry representing this product onto queue **C1**.
(do the following logic also)

if either 1- any required metadata is missing, or 2- there are comments directed to the manual metadata providers, then also place the product onto queue **B2** (Manual Metadata Entry Process 804),

else if supervised release was requested for this product, then place the product onto queue **B3** (Supervised Release Process 806).

else if the product has all the information from the Content Processing Tools 155 for all of the requested quality levels, then place the product onto queue **Before** (the Metadata SC(s) Creation Process 807),

else flag the product as needs the encryption keys and place the product onto queue **A2** (Products Awaiting Action/Information Process 801).

B2: During step **B2** (Manual Metadata Entry Process 804),

if step **C1** (the Watermarking Process 808) has not been done and the metadata needed for step **C1** is present, then place an entry representing this product onto queue **C1**.

(do the following logic also)

if metadata needed for step **C2** (the Preprocessing and Compression Process 809) just been provided, then

(do the following logic also)

if all the metadata that can be gathered by the Metadata Assimilation and Entry Tool 161 is present, then

if supervised release was requested for this product, then place the product onto queue **B3** (Supervised Release Process 806)

else

if all the information from step **C4** (the Encryption Process 811) of the Content Processing Tools 155 is present, then place this product onto queue **Before** (the Metadata SC(s) Creation Process 807)

else flag the product as needs the encryption keys and place this product onto queue **A2** (Products Awaiting Action/Information Process 801).

else

if the metadata provider requested a forced supervised release, then place the product onto queue **B3** (Supervised Release Process 806)

else do nothing (keep the product on queue **B2** (Manual Metadata Entry Process 804)).

B3: During step **B3** (Supervised Release Process 806),

if this operator is sending the product back to step **B2** (Manual Metadata Entry Process 804), then place the product on queue **B2**.

else if this operator released the product, then

if all the information from step **C4** (the Encryption Process 811) of the Content Processing Tools 155 is present, then place this product onto queue **Before** (the Metadata SC(s) Creation Process)

else flag the product as needs the encryption keys and place this product onto queue **A2** (Products Awaiting Action/Information Process 801).

else the product remains on queue **B3** (Supervised Release Process 806).

Before: After step **Before** (the Metadata SC(s) Creation Process 807),

flag the product Metadata has been packed .

if all the (product/quality level) tuples have been packed, then

if the Content Provider(s)' 101 configuration specifies Quality Assure the SC(s), then place this product onto queue **A3** (the Final Quality Assurance Process 813)

else place this product onto queue **A4** (the Content Dispersement Process 814).

else flag the product as needs the Content 113 SC(s) and place this product onto queue **A2** (Products Awaiting Action/Information Process 801).

C1: After step **C1** (the Watermarking Process 808),

if the metadata needed for step **C2** (the Preprocessing and Compression Process 809) is present, then create an entry for each (product/quality level) tuple and place them onto queue **C2**,

else flag the product as needs the metadata for Preprocessing/Compression and place this product onto queue **A2** (Products Awaiting Action/Information Process 801).

C2: After step **C2** (the Preprocessing and Compression Process 809),

if the Content Provider(s)' 101 configuration specifies Content Quality Control Process 810 , then place this (product/quality level) tuple onto queue **C3** (the Content Quality Control Process 810),

else place this (product/quality level) tuple onto queue **C4** (the Encryption Process 811).

C3: After step **C3** (the Content Quality Control Process 810), then place this (product/quality level) tuple onto queue **C4** (the Encryption Process 811).

C4: After step **C4** (the Encryption Process 811),

provide the needed information (i.e., the Symmetric Key 623 generated by the Process and used to encipher the Content 113) to the Metadata Assimilation and Entry Tool 161.

if all the metadata that's required for the Content SC(s) 630 is present, then place this (product/quality level) tuple onto queue **C5** (the Content SC(s) Creation Process 812),

else flag the product as needs the metadata for Content SC(s) 630 Packing and place this (product/quality level) tuple onto **A2** (Products Awaiting Action/Information Process 801).

C5: After step **C5** (the Content SC(s) Creation Process 812),

flag the quality level the Content 113 at this quality level has been packed .

if all the (product/quality level) tuples have been packed, then

if the product is flagged Metadata has been packed , then

if the Content Provider(s)' 101 configuration specifies Quality Assure the SC(s), then place this product onto queue **A3** (the Final Quality Assurance Process 813)

else place this product onto queue **A4** (the Content Dispersement Process 814)

else flag the product as needs the Metadata SC(s) 620 and place this product onto queue **A2** (Products Awaiting Action/Information Process 801).

else (all the (product/quality level) tuples have not been packed) do nothing (another (product/quality level) tuple triggers an action).

C. Metadata Assimilation and Entry Tool

Metadata consists of the data describing the Content 113 for example in music, title of the recording, artist, author/composer, producer and length of recording. The following description is based upon Content 113 being music but it should be understood by those skilled in the art that other content types e.g., video, programs, multimedia, movies, and equivalent, are within the true scope and meaning of the present invention.

This Subsystem brings together the data the Content Provider(s) 101 provides to the Electronic Digital Content Store(s) 103 to help promote the sale of the product (e.g., for music, sample clips by this artist, history of this artist, list of albums on which this recording appears, genres associated with this artist and/or product), the data the Content Provider(s) 101 provides to the End-User(s) with the purchased product (e.g., artist, producer, album cover, track length), and the different purchase options (the Usage Conditions 517) the Content Provider(s) 101 wants to offer the End-User(s). The data is packaged into a Metadata SC(s) 620 and made available to the Electronic Digital Content Store(s) 103. To accomplish this, the following tools are provided:

- Automatic Metadata Acquisition Tool
- Manual Metadata Entry Tool
- Usage Conditions Tool
- Supervised Release Tool

These tools enable Content Provider(s) 101 to implement the processes described above for Work Flow Manager 154. Tools described here are a toolkit based on Java in the preferred embodiment but other programming languages such as C/C++, Assembler and equivalent can be used.

1. Automatic Metadata Acquisition Tool

The Automatic Metadata Acquisition Tool provides a user the ability to implement the Automatic Metadata Acquisition Process 803 described above. The Automatic Metadata Acquisition Tool is used to access the Database 160 of the Content Provider(s) 101 and to retrieve as much data as possible without operator assistance. Configuration methods are available to automate this process. The Content Provider(s) 101 can tailor the default metadata template to identify the types of data this Content Provider(s) 101 wants to provide to End- User(s) (e.g., composer, producer, sidemen, track length) and the types of promotional data the Content Provider(s) 101 provides to the Electronic Digital Content Store(s) 103 (e.g., for a music example, sample clips by this artist, a history of this artist, the list of albums on which this recording appears, genres associated with this artist). The default metadata template includes data fields which are required by the End-User Device(s) 109, data fields which can be optionally provided to the End-User Device(s) 109 and a sample set of data fields, targeted to the Electronic Digital Content Store(s) 103, that promote the artist, album, and/or single.

To extract the template data fields from the Database 160 of the Content Provider(s) 101 the Automatic Metadata Acquisition Tool uses a table that maps the type of data (e.g., composer, producer, a biography of the artist) to the location within the database where the data can be found. Each of the Content Provider(s) 101 help specify that mapping table for their environment.

The Automatic Metadata Acquisition Tool uses a metadata template of the Content Provider(s) 101 and mapping table to acquire whatever data is available from the Databases 160 of the Content Provider(s) 101. The status of each product is updated with the result of the Automatic Metadata Acquisition Process 803. A product which is missing any required data is queued for Manual Metadata Entry Process 804, otherwise it is available for packing into a Metadata SC(s) 620.

2. Manual Metadata Entry Tool

The Manual Metadata Entry Tool provides a user the ability to implement the Manual Metadata Entry Process 804 described above. The Manual Metadata Entry Tool allows any properly authorized operator to provide the missing data. If the operator determines that the missing data is unavailable, the operator can attach a comment to the product and request supervised release. The Content Provider(s) 101 may require, for quality assurance reasons, that the product undergo supervised release. Once all the required data is present, and if supervised release has not been requested, then the product is available for packing into a Metadata SC(s) 620.

3. Usage Conditions Tool

The Usage Conditions Tool provides a user the ability to implement the Usage Conditions Process 805 described above. The process of offering Content 113 for sale or rent (limited use), using electronic delivery, involves a series of business decisions. The Content Provider(s) 101 decides at which compression level(s) the Content 113 is made available. Then for each compressed encoded version of the Content 113, one or more usage conditions are specified. Each usage condition defines the rights of the End-User(s), and any restrictions on the End- User(s), with regard to the use of the Content 113.

As part of Content Processing Tools 155, a set of usage conditions (End-User(s) rights and restrictions) is attached to the product.

A usage condition defines:

1. the compression encoded version of the Content 113 to which this usage condition applies.
2. the type of user covered by this usage condition (e.g., business, private consumer)
3. whether this usage condition allows for the purchase or the rental of the Content 113.

   For a rental transaction:

   - the measurement unit which is used to limit the term of the rental (e.g., days, plays).
   - the number of the above units after which the Content 113 will no longer play.

   For a purchase transaction:

   - the number of playable copies the End-User(s) is allowed to make.
   - onto what kinds of media can he/she make those copies (e.g., CD-Recordable (CD-R), MiniDisc, Personal Computer).
4. the period of time during which the purchase/rental transaction is allowed to occur (i.e., an End-User(s) can purchase/rent under the terms of this usage condition only after the beginning availability date and before the last date of availability).
5. the countries from which an End-User(s) can transact this purchase (or rental).
6. the price of the purchase/rental transaction under this usage condition
7. the watermarking parameters.
8. the types of events which require notification of the Clearinghouse(s) 105.

An Example of a Set of Usage Conditions

The Content Provider(s) 101 may decide to test the North American market's acceptance to the re-release of the children's song by a popular children's vocalist during the fourth quarter 1997. The test will make the song available in two different compression encoding versions: 384Kbps and 56Kbps. The 384Kbps version can be bought (and one copy made onto MiniDisc) or rented (for two weeks), while the 56Kbps version can only be bought (and no copies made). The watermarking instructions is the same for any purchase/rental, and the Content Provider(s) 101 wants the Clearinghouse(s) 105 to count every copy made. This would create Usage Conditions as follows:

	Usage Condition 1	Usage Condition 2	Usage Condition 3
compressed encoded version	384Kbps	384Kbs	56Kbps
type of user	private consumer	private consumer	private consumer
type of transaction	purchase	rental	purchase
availability dates	1 Oct 1997 - 31 Dec 1997	1 Oct 1997 - 31 Dec 1997	1 Oct 1997 - 31 Dec 1997

	Usage Condition 1	Usage Condition 2	Usage Condition 3
compressed encoded version	384Kbps	384Kbs	56Kbps
type of user	private consumer	private consumer	private consumer
type of transaction	purchase	rental	purchase
availability dates	1 Oct 1997 - 31 Dec 1997	1 Oct 1997 - 31 Dec 1997	1 Oct 1997 - 31 Dec 1997
countries	USA and Canada	USA and Canada	USA and Canada
watermarking	std.	std.	std.
notifying events	copy action	none	none
number of copies	1	0	0
onto what media	MiniDisc	not applicable	not applicable
term of rental	not applicable	14 days	not applicable
price	Price 1	Price 2	Price 3
countries	USA and Canada	USA and Canada	USA and Canada
watermarking	std.	std.	std.
notifying events	copy action	none	none
number of copies	1	0	0
onto what media	MiniDisc	not applicable	not applicable
term of rental	not applicable	14 days	not applicable
price	Price 1	Price 2	Price 3

4. Parts of the Metadata SC(s) 620

Below are some of the kinds of data that the Metadata Assimilation and Entry Tool 161 gathers for inclusion into the Metadata SC(s) 620. An attempt has been made to group the data into SC(s) parts by function and destination.

product ID [src:content provider;] [dest: everybody;]

licensor label company [dest: EMS; end-user;]

licensee label company [dest: EMS; end-user;]

source (publisher) of this object (sublicensee label company) [dest: everybody;]

type of object (i.e., a single object or an array of objects)

object ID [dest: everybody;]

International Standard Recording Code (ISRC)

International Standard Music Number (ISMN)

usage conditions (src: content provider; dest: EMS, end-user, Clearinghouse(s) 105)

purchased usage conditions (src: EMS; dest: end-user, Clearinghouse(s) 105)

the set of usage conditions (consumer restrictions and rights) for the use of the object (sound recording)

an individual entry in the array of usage conditions

the compression encoded version of the Content 113 to which this usage condition applies

whether this usage condition allows for the purchase or the rental of the Content 113
for a rental transaction:
the measurement unit which is used to limit the term of the rental (e.g., days, plays).
the number of the above units after which the Content 113 will no longer play.
for a purchase transaction:
the number of playable copies the End-User(s) is allowed to make.
onto what kinds of media can (s)he make those copies (e.g., CD-Recordable (CD-R), MiniDisc, personal computer).
the period of time during which the purchase/rental transaction is allowed to occur (i.e., an End-User(s) can purchase/rent under the terms of this usage condition only after the beginning availability date and before the last date of availability)
a pointer to the countries from which an End-User(s) can transact this purchase (or rental)
the price of the purchase/rental transaction under this usage condition
a pointer to the encrypted watermarking instructions and parameters
a pointer to the types of events which require notification of the Clearinghouse(s) 105

purchase data (encrypted; optional info; src: EMS; dest: end-user, Clearinghouse(s) 105)
purchase date
purchase price
bill to name and address
consumer name and address
country of the consumer (best guess)

metadata 1 (src: content provider; dest: EMS, end-user)
an array of {
copyright information
for the composition
for the sound recording
title of song
principal artist(s)
}
a pointer to {
the artwork (e.g., album cover);
the format of the artwork (e.g., GIF, JPEG);
}

optional info:

an array of additional information {
composer
publisher
producer
sidemen
date of recording
date of release
lyrics
track name (description) / track length
list of albums on which this recording appears
genre(s)
}

metadata 2(src: content provider; dest: EMS)
an array of structures, each representing different quality levels of the same sound recording {
the sound recording;
the quality level of the sound recording;
the size (in bytes) of the (probably compressed) sound recording;
}

metadata 3(src: content provider; dest: EMS, end-user)
optional info:

promotional material:
a pointer to artist promotion material {
a URL to the artist's web site;
background description(s) of the artist(s);
artist-related interviews (along with format of the interview (e.g., text, audio, video));
reviews (along with format of the reviews (e.g., text, audio, video));
sample clips (and its format and compression level);
recent and upcoming concerts/appearances/events - their dates and locations;

}
a pointer to album promotion material {

sample clip (and its format and compression level);
background description(s) of the producer, and/or the composer, and/or the movie/play/cast, and/or the making of the album, etc.;
non-artist-related interviews (along with format of the interview (e.g., text, audio, video));
reviews (along with format of the reviews (e.g., text, audio, video));
genre(s);

}

single promotions:
sample clip (and its format and compression level)
background description(s) of the producer, and/or the composer, and/or the movie/play/cast, and/or the making of the single, etc.
reviews (along with format of the reviews (e.g., text, audio, video))

5. Supervised Release Tool

Supervised Release Tool provides a user the ability to implement the Supervised Release Process 806 described above. An individual designated by the Content Provider(s) 101 as having supervised release authority, may call up a product awaiting supervised release (i.e., a product on the queue of the Supervised Release Process 806), examine its Contents 113 and its accompanying comments, and either

approve its Contents 113 and release the product for packing into a Metadata SC(s) 620, or
make any necessary corrections and release the product for packing into a Metadata SC(s) 620 or
add a comment specifying the corrective action to take and resubmit the product to the Manual Metadata Entry Process 704

In another embodiment, after the creation of the SC(s), there is another optional quality assurance step where the Content 113 of the SC(s) can be opened and examined for completeness and accuracy, and, at that time, final approval can be given or denied for the product's release to the retail channel.

D. Content Processing Tools

The Content Processing Tools 155 is actually a collection of software tools which are used to process the digital content file to create watermarked, encoded, and encrypted copies of the content. The tools makes use of industry standard digital content processing tools to allow pluggable replacement of watermarking, encoding and encryption technologies as they evolve. If the selected industry tool can be loaded via a command line system call interface and passed parameters or provides a toolkit wherein functions can be called via a DLL interface, the content processing can be automated to some degree. A front end application to each tool queries the appropriate queue in the Content Processing Tools 155 for the next available job, retrieves the required files and parameters and then loads the industry standard content processing tool to perform the required function. Upon completion of the task, manual update to the queue may be required if the tool does not report terminating status.

A generic version of the Content Processing Tools 155 is described, but customization is possible. The Content Processing Tools 155 can be written in Java, C/C++ or any equivalent software. The Content Processing Tools 155 can be delivered by any computer readable means including diskettes, CDS or via a Web site.

1. Watermarking Tool

The Watermarking Tool provides a user the ability to implement the Watermarking Process 808 as described above. This tool applies copyright information of the Content 113 owner to the song file using audio Watermarking technology. The actual information to be written out is determined by the Content Provider(s) 101 and the specific watermarking technology selected. This information is available to the front end Watermarking Tool so that it can properly pass this information to the watermarking function. This imposes a synchronization requirement on the Metadata Assimilation and Entry Tool 161 to assure that it has acquired this information prior to, for example, allowing the song's audio file to be processed. This song will not be available for audio processing until the watermarking information has been obtained.

The watermark is applied as the first step in audio processing since it is common to all encodings of the song created. As long as the watermark can survive the encoding technology, the watermarking process need only occur once per song.

Various watermarking technologies are known and commercially available. The front end Watermarking Tool though is capable of supporting a variety of industry Watermarking Tools.

2. Preprocessing and Compression Tool

The Preprocessing and Compression Tool provides a user the ability to implement the Preprocessing and Compression Process 809 as described above. Audio encoding involves two processes. Encoding is basically the application of a lossy compression algorithm against, for a music content example, a PCM audio stream. The encoder can usually be tuned to generate various playback bit stream rates based on the level of audio quality required. Higher quality results in larger file sizes and since the file sizes can become quite large for high quality Content 113, download times for high quality Content 113 can become lengthy and sometimes prohibitive on standard 28,800 bps modems.

The Content Provider(s) 101 may, therefore, choose to offer a variety of digital content qualities for download to appease both the impatient and low bandwidth customers who don t want to wait hours for a download and the audiophile or high bandwidth customers who either only buys high quality Content 113 or has a higher speed connection.

Compression algorithms vary in their techniques to generate lower bit rate reproductions of Content 113. The techniques vary both by algorithm (i.e. MPEG, AC3, ATRAC) and by levels of compression. To achieve higher levels of compression, typically the data is re-sampled at lower sampling rates prior to being delivered to the compression algorithm. To allow for more efficient compression with less loss of fidelity or to prevent drastic dropout of some frequency ranges, the digital content may sometimes require adjustments to equalization levels of certain frequencies or adjustments to the dynamics of the recording. The content preprocessing requirements are directly related to the compression algorithm and the level of compression required. In some cases, the style of Content 113 (e.g. musical genre) can be successfully used as a base for determining preprocessing requirements since songs from the same genre typically have similar dynamics. With some compression tools, these preprocessing functions are part of the encoding process. With others, the desired preprocessing is performed prior to the compression.

Besides the downloadable audio file for sale, each song also has a Low Bit Rate (LBR) encoded clip to allow the song to be sampled via a LBR streaming protocol. This LBR encoding is also the responsibility of the Content Processing Tools 155. This clip is either provided by the Content Provider(s) 101 as a separate PCM file or as parameters of offset and length.

As with watermarking, it is hoped that the encoding tools can be loaded via a DLL or command line system call interface and passed all the required parameters for preprocessing and compression. The front end Encoding Tool may have a synchronization requirement with the Metadata Assimilation and Entry Tool 161, for example if the content is music, and if it is determined that the song's genre is acquired from the Database 160 of the Content Provider(s) prior to performing any audio preprocessing. This depends on the encoding tools selected and how indeterminate the genre for the song is. If the Content Provider(s) 101 varies the choice of encoded quality levels per song, this information is also be provided prior to the encoding step and agrees with the metadata being generated by the Metadata Assimilation and Entry Tool 161.

A variety of high quality encoding algorithms and tools are known today. The front end Encoding Tool though is capable of supporting a variety of industry encoding tools.

Turning now to FIG. 12 is shown a flow diagram of one embodiment for the Automatic Metadata Acquisition Tool of FIG. 8 according to the present invention. The process starts with reading an identifier from the media the Content Provider(s) 101 is examining. One example of content in an audio CD embodiment. In an audio CD embodiment, the following codes may be available Universal Price Code (UPC), International Standard Recording Code (ISRC), International Standard Music Number (ISMN). This identifier is read in the appropriate player for the content, for example an audio CD Player for audio CD, DVD player for DVD movie, DAT recorder for DAT recording and equivalent, step 1201. Next this Identifier is used to index a Database 160 for the Content

Provider(s) 101, step 1202. Some or all of the information required by the Work Flow Manager Process as described in FIG. 8 is retrieved in Database 160 and any other related sources, step 1203. This information can include the Content 113 and the metadata related to it. In step 1204, the additional information retrieved is used to start the Work Flow Manager 154 for creating electronic Content 113. It should be understood, that several selections of media, such as several audio CDS, can be queued up so as to enable the Automatic Metadata Acquisition Tool to create a series of Content 113 for electronic distribution. For example, all the Content 113 could be created from a series of CDS or even selected tracks from one or more CDS examined by the Content Provider(s) 101.

In an alternate embodiment, the preprocessing parameters can be retrieve from the Database 160 of the Content Provider(s) automatically. Referring now to FIG. 13 is a flow diagram of a method to automatically set the Preprocessing and Compression parameters of the Preprocessing and Compression Tool of FIG. 8 according to the present invention. In this embodiment, the Content 113 is music. In step 1301, music (Content 113) is selected to be encoded in Content Processing Tools 155. The genre of the music selected is determined, step 1302. This can be entered manually or by using other meta data available, such as the additional data retrieved from the process described in FIG. 12. The audio compression level and audio compression algorithms selected are than examined, step 1303. Next, a lookup is made by genre, compression settings and compression algorithms of what compression parameters should be used in the Preprocessing and Compression Process 809, 1304.

3. Content Quality Control Tool

The Content Quality Control Tool provides a user the ability to implement the Content Quality Control Process 810 as described above. This is an optional Content Processing Tool and provides an opportunity for a quality control technician to review the encoded and watermarked content files and approve or reject the content files based on quality judgments. He can re-encode the content making manual preprocessing adjustments until the quality is adequate or can flag the song for reprocessing and attach a note describing the problem.

This process step can be configured by the Content Provider(s) 101 as an optional or required step of the content processing work flow. An additional optional Final Quality Assurance Process 813 step is provided after packaging of all the SC(s) for this content (e.g. each SC(s) for songs on a CD) at which time the quality of the content encoding can be tested but catching a problem early prior to encryption and packaging allows for more efficient content processing. It is, therefore, highly desirable that the content quality be assured at this step as opposed to waiting until final completion of all processing.

4. Encryption Tool

The Encryption Tool provides a user the ability to implement the Encryption Process 811 as described above. Content encryption is the final step of the Content Processing Tools 155. Each of the versions of the content that were created by the Encoding Tool is now encrypted. The encryption tool is a function of the SC(s) Packer. The SC(s) Packer is called to encrypt the song and returns the generated encryption key used. This key is later passed into the SC(s) Packer for use in creation of the Metadata SC(s) 620.

E. Content SC(s) Creation Tool

Once all metadata has been gathered the Content SC(s) Creation Tool groups the metadata into categories based on their intended use. These groups of metadata are written into files to be passed in to the SC(s) Packer Tool as Metadata parts for the Metadata SC(s) 620. Each part (file) has unique processing requirements. Once the associated songs have been processed and encrypted and the target destination (URL of Content Hosting Site(s) 111) has been determined, the Content SC(s) 630 for the Content 113 are ready to be created. The Content 113 which have completed processing and have met all the requirements described above, are queued for packing in the packer queue of the Work Flow Manager 154.

The Content SC(s) Creation Tool now retrieves all the required files created by the previous steps of the Metadata Assimilation and Entry Tool 161 and calls the SC(s) Packer functions to create the Metadata SC(s) 620 and Content SC(s) 630. This process creates a single Metadata SC(s) 620 and multiple Content SC(s) 630 for each song. For example, if the content is music, each of the audio files created during audio processing for the various quality levels of the full song is packed into separate Content SC(s) 630. The audio file created for the sample clip is passed as a metadata file to be included in the Metadata SC(s) 620.

F. Final Quality Assurance Tool

The Final Quality Assurance Tool provides a user the ability to implement the Final Quality Assurance Process 813 as described above. Once all the SC(s) have been built for a content file, the content is available for a final quality assurance check. Quality assurance can be performed at various stages of the Content 113 preparation process. The Content Provider(s) 101 can choose to perform quality assurance as each major step is completed to prevent excessive rework later or may choose to wait until all audio preparation processes are complete and perform quality assurance on everything at once. If the latter is chosen, quality assurance is performed at this point upon completion of the creation of the SC(s). This tool allows each SC(s) for the song to be opened, examined, and the audio played.

Any problem discovered, even minor text changes requires that the SC(s) be rebuilt due to internal security features of SC(s). To avoid unnecessary re-processing time, it is highly recommended that the interim quality assurance steps be utilized to assure accuracy of the metadata and that this specific quality assurance step be reserved for validating appropriate cross references between the SC(s) associated with this song. If problems are found, the assurer can enter a problem description to be attached to the song and have it re-queued to the appropriate processing queue for reprocessing. Status is updated appropriately in the Work Flow Manager 154 to

indicate the status of all related components of the song. If no problems are discovered, the Content 113 is marked or flagged as ready for release.

G. Content Dispersement Tool

The Content Dispersement Tool provides a user the ability to implement the Content Dispersement Process 814 as described above. Once the Content 113 has been approved for release, the SC(s) for the Content 113 are placed in the queue of the Content Dispersement Process. The Content Dispersement Tool monitors the queue and performs immediate transfer of the SC(s) files or batch transfer of a group of SC(s) files based on the configuration settings provided by the Content Provider(s) 101. The Content Provider(s) 101 can also optionally configure the Content Dispersement Tool to automatically hold all SC(s) in this queue until they are manually flagged for release. This allows the Content Provider(s) 101 to prepare content in advance of their scheduled release date and hold them until they wish to release them e.g., a new song, movie or game. The SC(s) can also control access to Content 113 based on a defined release date so there is no requirement for the Content Provider(s) 101 to actually hold up delivery of the SC(s) but this manual release option can still be used for this purpose or used to manage network bandwidth required to transfer these large files.

When flagged for release, the Content SC(s) 630 for the Content 113 are transferred via FTP to the designated Content Hosting Site(s) 111. The Metadata SC(s) 620 is transferred via FTP to the Content Promotions Web Site 156. Here the SC(s) are staged to a new Content 113 directory until they can be processed and integrated into the Content Promotions Web Site 156.

FIG. 17 is a flow diagram of an alternate embodiment to automatically retrieve additional information for the Automatic Metadata Acquisition Tool of FIG. 8 according to the present invention. The process is similar for that described in FIG. 8 above. However, the quality checks of Supervised Release 806 and Content Quality Control 809 are combined into one quality check called Quality Control 1704. Performing quality checks prior to Metadata SC Creation 807 and Content SC Creation 812. Performing quality check prior to SC creation, eliminates the steps of unpacking the Content 113 and the associated Metadata SC(s) 620. In addition, in this embodiment, the queue of Products Awaiting Action/Information 801 have been eliminated. The jobs are placed on the specific process queues depending on what action is being requested. For example, if the job requires Manual Metadata, i.e. additional Metadata to be entered, the job is place on the Manual Metadata entry queue. Also the Automatic Metadata Acquisition 803 has been merged with New Content Request to occur up front prior to the Metadata Assimilation and Entry Tool 161 and the Content Processing Tool 155. Finally, it is important to point out that the Usage Conditions 804 are entered both at the Automatic Metadata Acquisition 803 and during the Manual Metadata Entry 803. Since, many of the usage conditions can be automatically filled-in during the Automatic Metadata Acquisition 803 step.

H. Content Promotions Web Site

To most effectively disperse information on what the Content Provider(s) 101 is making available for sale via digital download, and to get the necessary files to the Electronic Digital Content Store(s) 103 to enable it to make this Content 113 available for download to its customers, each Content Provider(s) 101 should have a secure web site housing this information. This is similar to the method used today by some Content Provider(s) 101 to make promotional content available to their retailers and others with a need for this information. In the case where this type of service already exists, an additional section can be added to the web site where Electronic Digital Content Store(s) 103 can go to see a list of the content available for sale via download.

The Content Provider(s) 101 has complete control over the design and layout of this site or can choose to use a turnkey web server solution provided as part of the toolkit for Secure Digital Content Electronic Distribution System 100. To implement their own design for this service, the Content Provider(s) 101 need only provide links to the Metadata SC(s) 620 for Electronic Digital Content Store(s) 103 who access their site. This is accomplished using the toolkit for the Secure Digital Content Electronic Distribution System 100. The selection process and what information is shown is the discretion of the Content Provider(s) 101.

Metadata SC(s) 620 received into a new content directory via FTP from the Content Dispersement Tool is processed by the Content Promotions Web Site 156. These containers can be opened with the SC(s) Preview Tool to display or extract information from the container. This information can then be used to update HTML Web pages and/or add information to a searchable database maintained by this service. The SC(s) Preview Tool is actually a subset of the Content Acquisition Tool used by the Electronic Digital Content Store(s) 103 to open and process Metadata SC(s) 620. See the Content Acquisition Tool section for more details. The Metadata SC(s) 620 file should then be moved to a permanent directory maintained by the Content Promotions Web Site 156.

Once the Metadata SC(s) 620 has been integrated into the Content Promotions Web Site 156, its availability is publicized. The Content Provider(s) 101 can send a notification to all subscribing Electronic Digital Content Store(s) 103 as each new Metadata SC(s) 620 is added to the site or can perform a single notification daily (or any defined periodicity) of all Metadata SC(s) 620 added that day (or period). This notification is performed via a standard HTTP exchange with the Electronic Digital Content Store(s) 103 Web Server by sending a defined CGI string containing parameters referencing the Metadata SC(s) 620 added. This message is handled by the Notification Interface Module of the Electronic Digital Content Store(s) 103 which is described later.

I. Content Hosting

The Entertainment Industry produces thousands of content titles, such as CDS, movies and games every year, adding to the tens of thousands of content titles that are currently available. The Secure Digital Content Electronic Distribution System 100 is designed to support all of the content titles available in stores today.

The numbers of content titles that the Secure Digital Content Electronic Distribution System 100 may eventually download to customers on a daily basis is in the thousands or tens of thousands. For a large number of titles, this requires a large amount of bandwidth. The computer disk space and bandwidth needs call for a

distributed, scalable implementation with multiple Content Hosting Site(s) 111. The system also supports customers all over the world. This requires overseas sites to speed delivery to the global customers.

Content hosting on the Secure Digital Content Electronic Distribution System 100 is designed to allow the Content Provider(s) 101 to either host their own Content 113 or share a common facility or a set of facilities.

Content hosting on the Secure Digital Content Electronic Distribution System 100 consists of multiple Content Hosting Site(s) 111 that collectively contain all of the Content 113 offered by the Secure Digital Content Electronic Distribution System 100 and several Secondary Content Sites (not shown) that contain the current hot hits offered by the Content Provider(s) 101. The number of Content Hosting Site(s) 111 changes depending on the number of End-User(s) using the system. The Secondary Content sites host a limited number of songs, but they will represent a large percentage of the bandwidth used on the system. The secondary sites are brought on line as the volume on the primary sites increases to the point of maximum capacity. The secondary sites can be located close to Network Access Points (NAPs) which helps speed up download times. They may also be placed in different geographic areas around the world to speed up download times.

Should the Content Provider(s) 101 choose to host all of their Content 113 in their own system, they can act as a single Content Hosting Site 111 with or without additional Secondary Content Sites. This allows them to build their own scalable distributed system. In another embodiment, Electronic Digital Content Store(s) 103 can also act as Content Hosting Site(s) 111 for certain Content 113. This embodiment requires a special financial agreement between the Electronic Digital Content Store(s) 103 and the Content Provider(s) 101.

1. Content Hosting Sites

Content 113 is added to the Content Hosting Site(s) 111 via FTP or HTTP by the Content Disbursement Tool described in the Content Provider(s) Section of this specification or via offline means such as content delivery on tape, CD Rom, flash, or other computer readable media. The Metadata SC(s) 620 created by the Content Provider(s) 101 contain a field that indicates the URL locating the Content SC(s) 630 for this Content 113. This URL corresponds to a Content Hosting Site(s) 111. Electronic Digital Content Store(s) 103 can override this URL if allowed by the Content Provider(s) 101 in the Offer SC(s) 641. The End-User Device(s) 109 communicates to this Content Hosting Site(s) 111 when it wants to download the Content SC(s) 630.

The End-User Device(s) 109 initiates the request for a Content SC(s) 630 by sending the License SC(s) 660 to the Content Hosting Site(s) 111. This is the same License SC(s) 660 returned by the Clearinghouse(s) 105. The Digital Signature of the License SC(s) 660 can be verified to determine if it is a valid License SC(s) 660. If it is a valid License SC(s) 660 either the download is initiated, or the download request may be redirected to another Content Hosting Site(s) 111.

2. Content Hosting Site(s) 111 provided by the Secure Digital Content Electronic Distribution System 100

For the Secure Digital Content Electronic Distribution System 100 the decision of which site should be used to download the Content 113 is made by the primary content site that received the initial request for a Content SC(s) 630. This site uses the following information to make this decision:

- Are there secondary content sites that host the Content 113 requested? (The majority of Content 113 offered by the Secure Digital Content Electronic Distribution System 100 is only located at primary sites);
- Where is the End-User Device(s) 109 geographically located? (This information can be obtained from the End-User Device(s) 109 when the request is initiated at the End-User Device(s) 109, this is passed up to the Clearinghouse(s) 105 in the Order SC(s) 650;
- Is the appropriate secondary site up and operational? (Sometimes the secondary sites may be off-line);
- What is the load of the secondary sites? (In some cases where a secondary site is swamped with activity another site that is less busy may be selected.

Before transmitting the Content SC(s) 630 to the End-User Device(s) 109, analysis and verifications are performed on the End-User's request. A database is kept of all of the License SC IDs that have been used to download Content 113. This database can be checked to ensure that the End-User Device(s) 109 only makes one request for each piece of Content 113 purchased. This prevents malicious users from repeatedly accessing the Content Hosting Site(s) 111 in hopes of slowing down the Content Hosting Site(s) 111 and prevents unauthorized download of the Content SC(s) 630.

The promotion and demotion of Content 113 to the Secondary Content sites is done periodically based on customer demand for the individual pieces of Content 113.

Content Hosting Router

The Content Hosting Router (not shown) resides in the Content Hosting Site(s) 111 and receives all requests from End-User(s) wanting to download Content 113. It performs validation checks on the End-User(s) request to ensure they indeed bought the Content 113. A database is maintained on the status of the Secondary Content Sites that includes what Content 113 is on them and their current status. This current status includes the amount of activity on the sites and whether a site is down for maintenance.

The only interface to the Content Hosting Router is the License SC(s) 660 that is sent by the End-User Device(s) 109 when Content 113 is required to be downloaded. The License SC(s) 660 includes information that indicates the user is allowed to download the Content 113.

Secondary Content Sites

The Secondary Content Sites (not shown) host the popular Content 113 of the Secure Digital Content Distribution System 100. These sites are geographically dispersed across the world and are located near Network Access Points (NAPs) to improve download times. These sites are added to the system as demand on the primary Content Hosting Site(s) 111 nears maximum capacity

IX. ELECTRONIC DIGITAL CONTENT STORE(S)

A. Overview - Support for Multiple Electronic Digital Content Store(s) 103

Electronic Digital Content Store(s) 103 are essentially the retailers. They are the entities who market the Content 113 to be distributed to the customer. For distribution of Content 113, this would include Digital Content Retailing Web Sites, Digital Content Retail Stores, or any business who wishes to get involved in marketing electronic Content 113 to consumers. These businesses can market the sale of electronic Content 113 only or can choose to just add the sale of electronic goods to whatever other merchandise they currently offer for sale. Introduction of downloadable electronic goods into the service offering of the Electronic Digital Content Store(s) 103 is accomplished via a set of tools developed for the Electronic Digital Content Store(s) 103 as part of the Secure Digital Content Electronic Distribution System 100.

These tools are used by the Electronic Digital Content Store(s) 103 to:

- acquire the Metadata SC(s) 620 packaged by the Content Provider(s) 101
- extract Content 113 from these SC(s) to be used as input to building their service offering
- create Offer SC(s) 641 describing the downloadable Content 113 they are offering for sale
- handle the acknowledgment of the sale and initiation of the download by creating and sending Transaction SC(s) 640 to the End-User Device(s) 109
- manage a transaction log of sales of downloadable Content 113 and the status of each download
- handle status notifications and transaction authentication requests
- perform account reconciliation

The tools are designed to allow flexibility in how the Electronic Digital Content Store(s) 103 wishes to integrate sale of downloadable electronic Content 113 into its service. The tools can be used in such a way as to request that all financial settlements for downloadable Content 113 purchased be handled by the Clearinghouse(s) 105 although this is not required. These tools also enable Electronic Digital Content Store(s) 103 to completely service their customers and handle the financial transactions themselves, including providing promotions and special offers. The tools enable the Electronic Digital Content Store(s) 103 to quickly integrate the sale of downloadable Content 113 into its existing services. In addition, the Electronic Digital Content Store(s) 103 is not required to host the downloadable Content 113 and does not have to manage its dispersement. This function is performed by the Content Hosting Site(s) 111 selected by the Content Provider(s) 101.

The tools for the Electronic Digital Content Stores(s) 103 are implemented in Java in the preferred embodiment but other programming languages such as C/C++, Assembler and equivalent can be used. It should be understood that the tools described below for the Electronic Digital Content Stores(s) 103 can run on a variety of hardware and software platforms. The Electronic Digital Content Stores(s) 103 as a complete system or as any of it's constitute components may be distributed as an application program in a computer readable medium including but not limited to electronic distribution such as the web or on floppy diskettes, CD ROMS and removable hard disk drives.

In another embodiment, the components of the Electronic Digital Content Stores(s) 103 is part of a programmer's software toolkit. This toolkit enables predefined interfaces to the components of the generic Electronic Digital Content Stores(s) 103 components and tools discussed below. These predefined interfaces are in the form of APIs or Application Programming Interfaces. A developer using these APIs can implement any of the functionality of the components from a high level application program. By providing APIs to these components, a programmer can quickly develop a customized Electronic Digital Content Stores(s) 103 without the need to re-created these functions and resources of any of these components.

Electronic Digital Content Store(s) 103 are not limited to Web based service offerings. The tools provided are used by all Electronic Digital Content Store(s) 103 wishing to sell downloadable electronic Content 113 regardless of the transmission infrastructure or delivery mode used to deliver this Content 113 to End-User(s). Broadcast services offered over satellite and cable infrastructures also use these same tools to acquire, package, and track electronic Content 113 sales. The presentation of electronic merchandise for sale and the method in which these offers are delivered to the End-User(s) is the main variant between the broadcast based service offering and the point-to-point interactive web service type offering.

B. Point-to-Point Electronic Digital Content Distribution Service

Point-to-Point primarily means a one-to-one interactive service between the Electronic Digital Content Store(s) 103 and the End-User Device(s) 109. This typically represents an Internet web based service provided via telephone or cable modem connection. Networks other than the Internet are supported in this model as well, as long as they conform to the Web Server/Client Browser model. FIG. 9 is a block diagram illustrating the major tools, components and processes of an Electronic Digital Content Store(s) 103.

1. Integration Requirements

The Secure Digital Content Electronic Distribution System 100 not only creates new online businesses but provides a method for existing businesses to integrate the sale of downloadable electronic Content 113 to their current inventory. The suite of tools provided to the Electronic Digital Content Store(s) 103 simplify this integration effort. The Content Acquisition Tool 171 and SC(s) Packer Tool 153 provides a method for the Electronic Digital Content Store(s) 103 to acquire information from the participating Content Provider(s) 101 on what they have available for sale and to create the files required to reference these downloadable objects as items in their own inventory. This process is batch driven and can be largely automated and is executed only to integrate new Content 113 into the site.

The tools for the Secure Digital Content Electronic Distribution have been designed to allow integration of sale of electronic downloadable Content 113 into typical implementations of web based Electronic Digital Content Store(s) 103 (i.e. Columbia House online, Music Boulevard, @Tower) and equivalent with minimal change to their current Content 113 retailing paradigm. Several methods of integration are possible and in the preferred embodiment, the Electronic Digital Content Store(s) 103 provides support for all product searches, previews,

selections (shopping cart), and purchases. Each Electronic Digital Content Store(s) 103 establishes customer loyalty with its customers and continues to offer its own incentives and market its products as it does today. In the Secure Digital Content Electronic Distribution System 100, it would simply need to indicate which products in its inventory are also available for electronic download and allow its customers to select the electronic download option when making a purchase selection. In another embodiment, the customer's shopping cart could contain a mixture of electronic (Content 113) and physical media selections. After the customer checks out, and the Electronic Digital Content Store(s) 103 has completed the financial settlement and logged or notified its shipping and handling functions to process the physical merchandise purchased, the commerce handling function of the Electronic Digital Content Store(s) 103 then calls the Transaction Processor Module 175 to handle all electronic downloads. It simply passes the required information and all processing from that point on is handled by the toolset for the Secure Digital Content Electronic Distribution System 100. In another embodiment, other methods of transaction handling are also possible using tools for the Secure Digital Content Electronic Distribution System 100 to handle the financial settlement should the Electronic Digital Content Store(s) 103 wish to sell downloadable merchandise only or to segregate the financial settlement of physical and downloadable merchandise.

To handle the downloading of merchandise, the Electronic Digital Content Store(s) 103 is given a Product ID (not shown) for each downloadable product that it acquires from the Content Promotions Web Site 156 for the Content Provider(s) 101. This Product ID is associated to a customer's purchase selection to the downloadable product. The Product ID is what the Electronic Digital Content Store(s) 103 passes to the Transaction Processor Module 175 to identify the product that the user has purchased. The SC(s) (Offer SC(s) 641) that were created to describe the products, are isolated from the Electronic Digital Content Store(s) 103 and kept in an Offer Database 181 in an effort to simplify management of these objects and make their existence transparent to the Electronic Digital Content Store(s) 103.

The Transaction Processor Module 175 and other additional functions are provided as web server side executables (i.e. CGI and NSAPI, ISAPI callable functions) or simply APIs into a DLL or C object library. These functions handle run time processing for End-User(s) interactions and optional interactions with the Clearinghouse(s) 105. These functions interact with the web server's commerce services to create and download to the End-User Device(s) 109 the files necessary to initiate the Content 113 download process. They also handle optional interactions to provide authorizations and accept notifications of completion of activities.

An Accounting Reconciliation Tool 179 is also provided to assist the Electronic Digital Content Store(s) 103 in contacting the Clearinghouse(s) 105 to reconcile accounts based on its own and the transaction logs of the Clearinghouse(s) 105.

2. Content Acquisition Tool 171

The Content Acquisition Tool 171 is responsible for interfacing with the Content Promotions Web Site 156 to preview and download Metadata SC(s) 620. Since the Content Promotions site is a standard web site, a web browser is used by the Electronic Digital Content Store(s) 103 to navigate this site. The navigation features varies based on the site design of the Content Provider(s) 101. Some sites may provide extensive search capabilities with many screens of promotional information. Others may have a simple browser interface with lists of titles, performers or new releases to select from. All sites include the selection of Metadata SC(s) 620 containing all the promotional and descriptive information of a song or album.

Alternatively, the Electronic Store(s) 103 may subscribe to content updates and receive updates automatically via FTP.

Viewing Metadata

The Content Acquisition Tool 171 is a web browser helper application which launches whenever a Metadata SC(s) 620 link is selected at the Content Promotions Web Site 156. Selection of the SC(s) causes it to be downloaded to the Electronic Digital Content Store(s) 103, and launch the helper application. The Content Acquisition Tool 171 opens the Metadata SC(s) 620 and display the non-encrypted information contained therein. Displayed information includes Extracted Metadata 173, for a music example, the graphic image(s) associated with the song and the information describing the song, a preview clip of the song can also be listened to if included in the Metadata SC(s) 620. In an example where the Content 113 is music, promotional information about the song or album, the album title, and the artist is also shown if provided by the Content Provider(s) 101. This information is displayed as a series of linked HTML pages in the browser window. Purchasable Content 113 such as the song and the lyrics and whatever other metadata the Content Provider(s) 101 wishes to protect, is not accessible to the Retail Content Web Site 180.

In another embodiment, the Content Provider(s) 101 provides optional promotional content for a fee. In this embodiment such promotional content is encrypted in the Metadata SC(s) 620. Financial settlement to open this data can be handled via the Clearinghouse(s) 105 with the account for the Electronic Digital Content Store(s) 103 being charged the designated fee. .

Extracting Metadata

Besides the preview capabilities, this tool provides two additional features: metadata extraction and preparation of an Offer SC(s) 641. Selection of the metadata extraction option prompts the Electronic Digital Content Store(s) 103 to enter the path and filenames to where the metadata is to be stored. Binary metadata such as graphics and the audio preview clip is stored as separate files. Text metadata is stored in an ASCII delimited text file which the Retail Content Web Site 180 can then import into its database. A table describing the layout of the ASCII delimited file is also be created in a separate TOC file. Additional options is available to allow extraction into other National Language Support (NLS) supported formats.

One important piece of information provided in the extracted data is the Product ID. This Product ID is what the commerce handling function for the Electronic Digital Content Store(s) 103 needs to identify to the Transaction Processor Module 175 (for more information refer to Transaction Processing section), the Content 113 that the user has purchased. The Transaction Processor Module 175 uses this Product ID to properly retrieve the appropriate Offer SC(s) 641 from the Offer Database 181 for subsequent download to the End-User Device(s) 109. The Electronic Digital Content Store(s) 103 has full control over how it presents the offer of downloadable Content 113 on its site. It only needs to retain a cross reference of the Content 113 being offered to this Product ID to properly interface with the tools for the Secure Digital Content Electronic Distribution System 100. Providing this information here, allows the Electronic Digital Content Store(s) 103 to integrate this product or Content 113 into its inventory and sales pages (database) in parallel with the Offer SC(s) 641 creation process since both processes uses the same Product ID to reference the product. This is described further below.

Offer SC(s) Creation Packer 153

The Electronic Digital Content Store(s) 103 is required to create an Offer SC(s) 641 describing the downloadable Content 113 that is for sale. Most of the information that goes into the Offer SC(s) 641 is derived from the Metadata SC(s) 620. The Content Acquisition Tool 171 creates the Offer SC(s) 641 by:

- removing parts from the Metadata SC(s) 620 that are not required to be included in the Offer SC(s) 641 as defined by the Offer SC(s) Template in the Metadata SC(s) 620
- adding additional required parts as defined by defaults specified by the configuration options in this tool for the Electronic Digital Content Store(s) 103
- prompting for additional required inputs or selections as defined by the Offer SC(s) Template in the Metadata SC(s) 620
- calling the SC(s) Packer 153 to pack this information into the SC(s) format

Metadata to be displayed by the Player Application 195 (further described later) on the End-User Device(s) 109 is kept in the Metadata SC(s) 620. Other promotional metadata that was only used by the Electronic Digital Content Store(s) 103 as input to his web service database is removed from the Metadata SC(s) 620. Rights management information provided by the Content Provider(s) 101, such as watermarking instructions, encrypted Symmetric Keys 623, and Usage Conditions 517 defining the permitted uses of the object, are also retained.

This stripped down Metadata SC(s) 620 is then included in the Offer SC(s) 641. The Electronic Digital Content Store(s) 103 also attaches its own Usage Conditions called Store Usage Conditions 519 or purchase options to the Offer SC(s) 641. This can be accomplished interactively or automatically through a set of defaults. If configured to be processed interactively, the Electronic Digital Content Store(s) 103 is prompted with the set of permitted object Usage Conditions 517 as defined by the Content Provider(s) 101. He then selects the option(s) he wishes to offer to his customers. These now become the new Usage Conditions or Store Usage Conditions 519. To process automatically, the Electronic Digital Content Store(s) 103 configures a set of default purchase options to be offered for all Content 113. These default options are automatically checked against the permitted Usage

Conditions 517 defined by the Content Provider(s) 101 and is set in the Offer SC(s) 641 if there are no discrepancies.

Once the Offer SC(s) 641 is created, it is stored in an Offer Database 181 and is indexed with the Product ID pre-assigned in the Metadata SC(s) 620. This Product ID is used later by the Electronic Digital Content Store(s) 103 to identify the downloadable Content 113 being purchased by a customer when interfacing with the Offer Database 181 to retrieve the Offer SC(s) 641 for packaging and transmittal to the End-User(s). See the Transaction Processor Module 175 section for more details.

In another embodiment, the Electronic Digital Content Store(s) 103 hosts the Content SC(s) 641 at his site. This embodiment requires changes to the Offer SC(s) 641 such as the replacement of the URL of the Content Hosting Site(s) 111 with the URL of the Electronic Digital Content Store(s) 103.

3. Transaction Processing Module 175

Electronic Digital Content Store(s) 103 directs billing to Clearinghouse(s) 105. Alternatively, the Electronic Digital Content Store(s) 103 may request financial clearance direct from the Clearinghouse(s) 105. There are two basic modes for processing End-User(s) purchase requests for downloadable Content 113. If the Electronic Digital Content Store(s) 103 does not wish to handle the financial settlement of the purchase and has no special promotions or incentives governing the sale of the merchandise and does not use a shopping cart metaphor for batching the purchase requests, it may opt to provide links on its Content 113 download pages directly to the Offer SC(s) 641 files. These Offer SC(s) 641 would have to have been built with retail pricing information included in the metadata. Also included in the Offer SC(s) 641 is a special HTML offer page presenting the purchase options with terms and conditions of the sale. This page is built from a template created when the Offer SC(s) 641 was built. When the End- User(s) clicks on the direct link to the Offer SC(s) 641, the Offer SC(s) 641 is downloaded to the browser End-User Device(s) 109 launching a helper application which opens the container and present the offer page included in the Offer SC(s) 641. This page contains a form to collect customer information including credit card information and purchase option selection. The form then gets submitted directly to the Clearinghouse(s) 105 for financial settlement and processing. Optionally, this form may contain the fields needed to use the End-User(s)' credit information or industry standard local transaction handler.

An embodiment where the Electronic Digital Content Store(s) 103 handles billing is now described. The more typical mode of handling purchase requests is to allow the Electronic Digital Content Store(s) 103 to process the financial settlement and then submit the download authorization to the End-User(s). This method allows the Electronic Digital Content Store(s) 103 to integrate sale of downloadable Content 113 with other merchandise offered for sale at his site, allows batch processing of purchase requests with only one consolidated charge to the customer (via a shopping cart metaphor) instead of individual charges for each download request, and allows the Electronic Digital Content Store(s) 103 to directly track his customers buying patterns and offer special promotions and club options. In this environment, the offer of downloadable Content 113 is included in his shopping pages which get added to a shopping cart when selected by the End-User(s) and get processed and financially settled as is

done in the Electronic Digital Content Store(s)' 103 current shopping model. Once the financial settlement is completed, the commerce handling process of the Electronic Digital Content Store(s) 100 then calls the Transaction Processor Module 175 to complete the transaction.

Transaction Processor Module 175

The role of the Transaction Processor Module 175 is to put together the information needed by the End-User Device(s) 109 to initiate and process the download of the Content 113 purchased. This information is packaged into a Transaction SC(s) 640 which is sent back to the End-User Device(s) 109 by the Web Server as the response to the purchase submission. The Transaction Processor Module 175 requires three pieces of information from the commerce handling process of the Electronic Digital Content Store(s) 103: the Product IDs for the Content 113 purchased, Transaction Data 642, and an HTML page or CGI URL acknowledging the purchase settlement.

The Product ID is the value provided to the Electronic Digital Content Store(s) 103 in the Metadata SC(s) 620 associated to the Content 113 just sold. This Product ID is used to retrieve the associated Offer SC(s) 641 from the Offer Database 181.

The Transaction Data 642 is a structure of information provided by the transaction processing function of the Electronic Digital Content Store(s) 103 which is later used to correlate the Clearinghouse(s) 105 processing with the financial settlement transaction performed by the Electronic Digital Content Store(s) 103 and to provide user identity information to be included in the watermark of the Content 113 downloaded to the End-User Device(s) 109. When the Clearinghouse(s) 105 receives a valid Order SC(s) 650, it logs a transaction indicating the Content 113 that was sold, which Electronic Digital Content Store(s) 103 sold it and the associated Transaction Data 642 including the End-User's Name and a Transaction ID 535. The Transaction ID 535 provides a reference to the financial settlement transaction. This information is later returned by the Clearinghouse(s) 105 to the Electronic Digital Content Store(s) 103 for use in reconciling its accounts with the billing statements received from the Content Provider(s) 101 (or his agent). The Clearinghouse Transaction Log 178 can be used by the Content Provider(s) 101 to determine what Content 113 of his has been sold and enables him to create a bill to each Electronic Digital Content Store(s) 103 for royalties owed him. Other electronic means besides billing can alternatively be used to settle accounts between the Content Provider(s) 101 and Electronic Digital Content Store(s) 103.

The information provided in the Transaction SC(s) 640 and the security and integrity of the Transaction SC(s) 640 provide sufficient authenticity to the Clearinghouse(s) 105 that the purchase transaction is valid and thus no further validation is required prior to the logging of this sale by the Clearinghouse(s) 105. The Electronic Digital Content Store(s) 103, however, has the option to request authentication before its accounts are charged (transaction logged at the Clearinghouse(s) 105 indicating to the Content Provider(s) 101 that this Electronic Digital Content Store(s) 103 has collected money for the sale of this Content 113). This request for authentication/notification is indicated by a flag in the Transaction Data 642. In this scenario, the Clearinghouse(s) 105 contacts the Electronic Digital Content Store(s) 103 and receive authorization from the

Electronic Digital Content Store(s) 103 before the charge to his account and the release of the encryption Key 623. The Transaction ID 535 is passed to the Electronic Digital Content Store(s) 103 from the Clearinghouse(s) 105 as part of this authentication request to enable the Electronic Digital Content Store(s) 103 to associate this request to a prior transaction performed with the End-User(s). This Transaction ID 535 can be any unique value the Electronic Digital Content Store(s) 103 wishes to use and is solely for its benefit.

The Transaction Data 642 also contains a customer name. This name can be from the user name field of the purchase form filled out by the user when making his purchase, or from information logged previously during some user registration process with the Electronic Digital Content Store(s) 103, or the official name obtained from credit card information associated with the card used in this transaction. This name is later included in the License Watermark 527.

The Transaction Data 642 also contains the Store Usage Conditions 519 purchased by the End-User(s). This information is included in the License Watermark 527 and used by the End- User Device(s) 109 in Copy and Play Control.

The final parameter required by the Transaction Processor Module 175 is the HTML page or CGI URL acknowledging the purchase settlement. The purpose of this is to allow the Electronic Digital Content Store(s) 103 to respond to the End-User(s) with an acknowledgment of the financial settlement and whatever other information he wishes to include in the response. This HTML page or CGI URL is included in the Transaction SC(s) 640 and is displayed in the browser window of the End-User Device(s) 109 when the Transaction SC(s) 640 is received and processed.

The Transaction SC(s) 640 is the HTTP response to the End-User(s) from the Electronic Digital Content Store(s) 103 after processing the purchase submission. Sending a SC(s) as the direct HTTP response forces the automatic loading on the End-User Device(s) 109 of a SC(s) Processor Helper Application thus allowing automatic completion of the transaction without depending on further End-User(s) initiated actions. This process is described in more detail in the End-User Device(s) 109 and Player Application 195 section later.

When the Transaction Processor Module 175 is called with the required parameters, it builds a Transaction SC(s) 640 containing the Transaction Data 642, the transaction acknowledgment HTML page or reference URL other required security features of the SC(s), and retrieves and imbeds the Offer SC(s) 641 associated with the purchase. It also logs information about this transaction for later use by the Notification Interface Module 176 and the Account Reconciliation Tool 179.

4. Notification Interface Module 176

The Notification Interface Module 176 is a Web Server side executable routine (CGI or function callable by NSAPI, ISAPI or equivalent). It handles optional requests and notifications from the Clearinghouse(s) 105, the End-User Device(s) 109, the Content Hosting Site(s) 111, and the Content Provider(s) 101. The events that the Electronic Digital Content Store(s) 103 can optionally request notification for are:

Notification from the Clearinghouse(s) 105 that the End-User Device(s) 109 requested an encryption Key 623 and the Clearinghouse(s) 105 is releasing the encryption Key 623 for the specified Content 113. This notification can optionally be configured to require authentication from the Electronic Digital Content Store(s) 103 prior to the encryption Key 623 being sent to the End-User Device(s) 109.

Notification from the Content Hosting Site(s) 111 that the Content SC(s) 630 has been sent to the End-User Device(s) 109.

Notification from the End-User Device(s) 109 that the Content SC(s) 630 and the License SC(s) 660 have been received and successfully used to process the Content 113 or was found to be corrupt.

Notification from the Content Provider(s) 101 that new Content 113 has been placed in the Content Promotions Web Site 156.

None of these notifications are a required step in the Secure Digital Content Electronic Distribution System flows 100 but are provided as options to allow the Electronic Digital Content Store(s) 103 the opportunity to close its records on the satisfaction of completion of the sale. It also provides information that may be needed to handle customer service requests by letting the Electronic Digital Content Store(s) 103 know what functions have transpired since financial settlement of the transaction or what errors occurred during an attempt to complete the sale. Alternatively, much of this status can be obtained from the Clearinghouse(s) 105 through the Customer Service Interface 184 as needed.

Frequency of notification of new Content 113 available at the Content Promotions Web Site 156 is determined by the Content Provider(s) 101. Notification may be provided as each new Metadata SC(s) 620 is added or just daily with all new Metadata SC(s) 620 added that day.

All of these notifications result in entries being made to the Transaction Log 178. If the Electronic Digital Content Store(s) 103 wishes to perform his own processing on these notifications, he can intercept the CGI call, perform his unique function and then optionally pass the request on to the Notification Interface Module 176.

5. Account Reconciliation Tool 179

This Account Reconciliation Tool 179 contacts the Clearinghouse(s) 105 to compare the Transaction Log 178 with the log of the Clearinghouse(s) 105. This is an optional process which is available to help the Electronic Digital Content Store(s) 103 feel comfortable with the accounting for the Secure Digital Content Electronic Distribution System 100.

In another embodiment, this tool can be updated to provide electronic funds transfers for automated periodic payments to the Content Provider(s) 101 and the Clearinghouse(s) 105. It can also be designed to automatically process payments upon reception of an electronic bill from the Clearinghouse(s) 105 after reconciling the bill against the Transaction Log 178.

C. Broadcast Electronic Digital Content Distribution Service

Broadcast primarily refers to a one to many transmission method where there is no personal interaction between the End-User Device(s) 109 and the Electronic Digital Content Store(s) 103 to customize on-demand viewing and listening. This is typically provided over a digital satellite or cable infrastructure where the Content 113 is preprogrammed so that all End-User Device(s) 109 receive the same stream.

A hybrid model can also be defined such that an Electronic Digital Content Store(s) 103 provides a digital content service organized in such a way that it can offer both a web distribution interface via an Internet connection as well as a higher bandwidth satellite or cable distribution interface via a broadcast service, with a great deal of commonality to the site design. If the IRD backchannel serial interface were connected to the web, and the IRD supported web navigation, the End-User(s) could navigate the digital content service in the usual way via the backchannel Internet interface, previewing and selecting Content 113 to purchase. The user can select high quality downloadable Content 113, purchase these selections, and receive the required License SC(s) 660 all via an Internet connection and then request delivery of the Content 113 (Content SC(s) 630) over the higher bandwidth broadcast interface. The Web service can indicate which Content 113 would be available for download in this manner based on the broadcast schedule or could build the broadcast streams based totally on purchased Content 113. This method would allow a Web based digital content service to contract with a broadcast facility to deliver high quality Content 113 to users equipped with the proper equipment making a limited number of specific Content 113 (e.g. songs or CDS) available daily in this manner and the entire catalog available for download in lower quality via the web interface.

Other broadcast models can be designed where there is no web interface to the End-User Device(s) 109. In this model, promotional content is packaged in specially formatted digital streams for broadcast delivery to the End-User Device(s) 109 (i.e. IRD) where special processing is performed to decode the streams and present the End-User(s) with the promotional content from which purchase selections can be made.

The actual purchase selections would still be initiated via backchannel communications from the End-User Device(s) 109 to the Clearinghouse(s) 105 and would utilize SC(s) to perform all data exchange. The toolset provided to the Electronic Digital Content Store(s) 103 has been architected and developed in such a way that most of the tools apply to both a point-to-point Internet service offering as well as a broadcast satellite or cable offering. The tools used by a Digital Content Web Site Electronic Digital Content Store(s) 103 to acquire and manage Content 113 as well as prepare SC(s) is also used by a satellite based Electronic Digital Content Store(s) 103 to manage and prepare Content 113 for distribution on a broadcast infrastructure. The SC(s) distributed over a Web service are the same as those distributed over a broadcast service.

X. END-USER DEVICE(S) 109

The applications in the End-User Device(s) 109 for the Secure Digital Content Electronic Distribution System 100 perform two main functions: first the SC(s) processing and copy control; and second playback of encrypted Content 113. Whether the End-User Device(s) 109 is a Personal Computer or a specialized electronic consumer device, it has to be capable of performing these base functions. The End-User Device(s) 109 also provides a variety of additional features and functions like creating play lists, managing the digital content library, displaying information and images during content playback, and recording to external media devices. These functions vary based on the services these applications are supporting and the type of devices the applications are designed for.

A. Overview

Referring now to FIG. 10, shown is the major components and processes and End-User Device(s) 109 Functional Flow. The applications designed to support a PC based web interface Content 113 service consists of two executable software applications: the SC(s) Processor 192 and the Player Application 195. The SC(s) Processor 192 is an executable application which is configured as a Helper Application into the End-User(s) Web Browser 191 to handle SC(s) File/MIME Types. This application is launched by the Browser whenever SC(s) are received from the Electronic Digital Content Store(s) 103, the Clearinghouse(s) 105, and the Content Hosting Site(s) 111. It is responsible for performing all required processing of the SC(s) and eventually adding Content 113 to the Digital Content Library 196 of the End-User(s).

The Player Application 195 is a stand alone executable application which the End-User(s) loads to perform Content 113 in his Digital Content Library 196, manage his Digital Content Library 196 and create copies of the Content 113 if permitted. Both the Player Application 195 and SC(s) Processor 192 applications can be written in Java, C/C++ or any equivalent software. In the preferred embodiment, the applications can be downloaded from computer readable means such as website. However, other delivery mechanisms are also possible such as being delivered on computer readable media such as diskettes or CDS.

The searching and browsing of Content 113 information, previewing of, for example, song clips, and selecting songs for purchase is all handled via the End-User(s) Web Browser 191. Electronic Digital Content Store(s) 103 provides the shopping experience in the same way that is offered today by many Content 113 retailing web sites. The difference to the End-User(s) over today's web based Content 113 shopping is that they may now select downloadable Content 113 objects to be added to their shopping cart. If the Electronic Digital Content Store(s) 103 has other merchandise available for sale in addition to the downloadable objects, the End-User(s) may have a combination of physical and electronic downloadable merchandise in his shopping cart. The Secure Digital Content Electronic Distribution End-User Device(s) 109 are not involved until after the End-User(s) checks out and submits his final purchase authorization to the Electronic Digital Content Store(s) 103. Prior to this point, all interaction is between the Web Server for the Electronic Digital Content Store(s) 103 and the Browser 191 on the End-User Device(s) 109. This includes preview of sample Digital Content clips. Digital Content clips are not packaged into SC(s) but instead are integrated into the web service of the Electronic Digital Content Store(s) 103

as downloadable files or fed from a streaming server. The format of the Content 113 clip is not dictated by the system architecture. In another embodiment, the Player Application 195 could interact directly with the Electronic Digital Content Store(s) 103 or Clearinghouse(s) 105 or offline using a promotional CD.

B. Application Installation

The Player Application 195 and the Helper Application 1981 are packaged into a self installing executable program which is available for download from many web sites. The Clearinghouse(s) 105 acts as a central location which hosts the master download page at a public web site. It contains links to the locations from which the installation package can be downloaded. The installation package is available at all Content Hosting Site(s) 111 to provide geographic dispersal of the download requests. Each participating Electronic Digital Content Store(s) 103 can also make the package available for download from their site or may just provide a link to the master download page at the public web site of the Clearinghouse(s) 105.

Any End-User(s) wishing to purchase downloadable Content 113, downloads and install this package. The installation is self contained in this downloadable package. It unpacks and installs both the Helper Application 198 and the Player Application 195 and also configure the Helper Application 198 to the installed Web Browser(s).

As part of the installation, a Public/Private Key 661 pair is created for the End-User Device(s) 109 for use in processing Order and License SC(s) 660. A random Symmetric Key (Secret User Key) is also generated for use in protecting song encryption keys in the License Database 197. The Secret User Key (not shown) is protected by breaking the key into multiple parts and storing pieces of the key in multiple locations throughout the End-User(s)' computer. This area of the code is protected with Tamper Resistant Software technology so as not to divulge how the key is segmented and where it is stored. Preventing access to this key by even the End-User(s) helps to prevent piracy or sharing of the Content 113 with other computers. See the SC(s) Processor 192 section for more details on how these keys are used.

Tamper-resistant software technology is a method to deter unauthorized entry into a computer software application by a hacker. Typically a hacker wants to understand and/or modify the software to remove the restrictions on the usage. In practicality, no computer program exists that cannot be hacked; that is why tamper-resistant software is not called "tamper-proof". But the amount of effort required to hack a tamper-resistance protect application usually deters most hackers because the effort is not worth the possible gain. Here the effort would be to gain access to a key to one piece of Content 113, perhaps a single song on a CD.

One type of tamper-resistant software technology is from IBM. One product this code was introduced is in the IBM ThinkPad 770 laptop computer. Here, the tamper-resistant software was used to protect the DVD movie player in the computer. Digital Content Provider(s) such as Hollywood studios, concerned about the advent of digital movies and the ease at which perfect copies can be made, have insisted that movies on DVD disc(s) contain copy protection mechanisms. IBM's tamper-resistant software made it difficult to circumvent these copy protection mechanisms. This is a very typical application for tamper-resistant software; the software is used to enforce rules on the usage of some protected type of Content 113.

IBM's tamper-resistant software puts several types of obstacles in the path of the attacker. First, it contains techniques to defeat, or at least reduce the effectiveness of, the standard software tools that the hacker uses: debuggers and disassemblers. Second it contains self- integrity checking, so that single modifications, or even small handfuls of modifications, will be detected and cause incorrect operation. Finally, it contains obfuscations to mislead hackers regarding its true operation. The latter technique is largely ad hoc, but the first two build upon well-known tools in cryptography: encryption and digital signatures.

C. Secure Container Processor 192

When the End-User(s) submits the final purchase authorization to the Electronic Digital Content Store(s) 103 for the merchandise he has collected in his shopping cart, his Web Browser remains active waiting for a response from the Web Server. The Web Server at the Electronic Digital Content Store(s) 103 processes the purchase and performs the financial settlement and then returns a Transaction SC(s) 640 to the End-User Device(s) 109. The SC(s) Processor 192 (Helper Application 198) is launched by the Web Browser to process the SC(s) mime type associated with the Transaction SC(s) 640. FIG. 14 is an example of user interface screens of the Player Application 195 downloading content to a local library as described in FIG. 10 according to the present invention.

The SC(s) Processor 192 opens the Transaction SC(s) 640 and extract the Response HTML page and Offer SC(s) 641 contained within. The Response HTML page is displayed in the Browser window acknowledging the End-User(s)' purchase. The Offer SC(s) 641 are then opened and the Content 113 (e.g. song or album) names along with the projected download times are extracted from them, step 1401. A new window is then displayed with this information and the End-User(s) is presented with options to schedule the download(s) of the Content 113 (e.g. for music, songs or entire albums), step 1402. The End-User(s) can select immediate download or can schedule the download to occur at a later time. If a later time is selected, the download schedule information is saved in a log and the download is initiated at the scheduled time if the End-User Device(s) 109 is powered on at that time. If the computer is not active at the scheduled download time or the communication link is not active, the End-User(s) is prompted to reschedule the download when the computer is next powered up.

When the scheduled download time occurs or if immediate download was requested, the SC(s) Processor 192 creates Order SC(s) 650 from information in the Transaction SC(s) 640, Offer SC(s) 641, and the Public Key 661 of the End-User(s) generated at install time. This Order SC(s) 650 is sent via HTTP request to the Clearinghouse(s) 105. When the Clearinghouse(s) 105 returns the License SC(s) 660, the Helper Application 198 is re-invoked to process the License SC(s) 660. The License SC(s) 660 is then opened and the URL of the Content Hosting Site(s) 111 is extracted from the referenced Order SC(s) 650. The License SC(s) 660 is then sent to the specified Content Hosting Site 111, via http request through the Browser, requesting download of the Content SC(s) 630. When the Content SC(s) 630 comes back to the Browser, the Helper Application 198 is re-invoked again. The SC(s) Processor 192 displays the name of the Content 113 being downloaded along with a download progress indicator and an estimated time to completion.

As the Content 113 is being received by the SC(s) Processor 192, it loads the Content 113 data into memory buffers for decryption. The size of the buffers depends on the requirements of the encryption algorithm and watermarking technology 193 and is the minimum size possible to reduce the amount of unencrypted Content 113 exposed to hacker code. As a buffer is filled, it is decrypted using the Key 623 (corresponding to the Public Key 661) of the End-User(s) extracted from the License SC(s) 660, which itself is first decrypted using the Private Key. The decrypted buffer is then passed to the watermarking function.

The watermarking 193 extracts the watermarking instructions from the License SC(s) 660 and decrypt the instructions using the Private Key of the End-User(s). The watermarking data is then extracted from the License SC(s) 660 which includes transaction information such as the purchaser's name as registered with the Electronic Digital Content Store(s) 103 from which this Content 113 was purchased or derived from the credit card registration information if the Electronic Digital Content Store(s) 103 does not provide a registration function. Also included in the watermark is the purchase date and the Transaction ID 535 assigned by the Electronic Digital Content Store(s) 103 to reference the specific records logged for this transaction. The Store Usage Conditions 519 are also included to be used by the Copy Control of the Player Application 195.

The Watermarking 193 is protected with Tamper Resistant Code technology so as not to divulge the watermarking instructions thus preventing a hacker from discovering the location and technique of the watermark. This prevents removal or modification of the watermark by a hacker.

After inscribing any required watermark to this content buffer, the buffer is passed to the scrambling function for Re-Encryption 194. A processor efficient secure encryption algorithm such as IBM's SEAL encryption technology is used to re-encrypt the Content 113 using a random Symmetric Key. Once the download and Decryption and Re-Encryption 194 process is complete, the encryption Key 623 used by the Content Provider(s) 101 to originally encrypt the Content 113 is now destroyed and the new SEAL key is itself encrypted using the Secret User Key created and hidden at installation time. This new encrypted Seal Key is now stored in the License Database 107.

Unlike source performed at the Content Provider(s) 101 and user watermarking performed at the End User Device(s) 109 may need to become an industry standard to be effective. These standards are still evolving. The technology is available to allow control information to be embedded in the music and updated a number of times. Until such time as the copy control standards are more stable, alternative methods of copy control have been provided in the Secure Digital Content Electronic Distribution System 100 so that it does not rely on the copy control watermark in order to provide rights management in the consumer device. Storage and p lay/record usage conditions security is implemented utilizing encrypted DC Library Collections 196 that are tied to the End User Device(s) 109 and protected via the Tamper Resistant Environment. Software hooks are in place to support copy control watermarking when standards have been adopted. Support exists today for watermarking AAC and other encoded audio streams at a variety of compression levels but this technology is still somewhat immature at this time to be put to use as a sole method of copy control.

The Decryption and Re-Encryption 194 process is another area of the code that is protected with Tamper Resistant Code technology so as not to divulge the original Content 113 encryption key, the new SEAL key, the Secret User Key, and where the Secret User Key segments are stored and how the key is segmented.

The process of Decryption and Re-Encryption 194 serves two purposes. Storing the Content 113 encrypted with an algorithm like SEAL enables faster than real-time decryption and requires much less processor utilization to perform the decryption than does a more industry standard type algorithm like DES. This enables the Player Application 195 to perform a real-time concurrent decryption-decode-playback of the Content 113 without the need to first decrypt the entire file for the Content 113 prior to decode and playback. The efficiency of the SEAL algorithm and a highly efficient decode algorithm, allows not only concurrent operation (streaming playback from the encrypted file) but also allows this process to occur on a much lower powered system processor. Thus this application can be supported on a End-User Device(s) 109 as low end as a 60MHz Pentium system and perhaps lower. Separating the encryption format in which the Content 113 is finally stored from the original encryption format, allows for greater flexibility in the selection of the original content encryption algorithm. Thus use of widely accepted and proven industry standard algorithms can be used thus further enhancing Digital Content Industry acceptance of the Secure Digital Content Electronic Distribution System 100.

The second purpose of this Decryption and Re-Encryption 194 process is to remove the requirement that the original master encryption Key 623, used by the Content Provider(s) 101 to encrypt this Content 113, be stored on every End-User Device(s) 109 which has licensed this Content 113. The encrypted master Key 623, as part of the License SC(s) 660, is only cached on the hard disk of the End-User Device(s) 109 for a very short time and is in the clear only in memory and for a very short time. During this execution phase, the Key 623 is protected via Tamper Resistant Code technology. Not having to retain this Key 623 in any form on the End- User Device(s) 109 once this Decryption and Re-Encryption 194 phase has completed, greatly lessens the possibility of piracy from hackers.

Once the song has been re-encrypted, it is stored in the Digital Content Library 196. All metadata required for use by the Player Application 195, is extracted from the associated Offer SC(s) 641 and also stored in the Digital Content Library 196, step 1403. Any parts of the metadata which are encrypted, such as the song lyrics, are decrypted and re-encrypted in the same manner as described above for the other content. The same SEAL key used to encrypt the Content 113 is used for any associated metadata needing to be encrypted.

D. The Player Application 195

1. Overview

The Secure Digital Content Electronic Distribution Player Application 195 (referred to here as the Player Application 195) is analogous to both a CD, DVD or other Digital Content player and to a CD, DVD, or other digital content storage management system. At its simplest, it performs Content 113, such as playing songs or videos. At another level, it provides the End- User(s) a tool for managing his/her Digital Content Library 196. And just as importantly, it provides for editing and playing of collections of content, such as songs, (referred to here as Play-lists).

The Player Application 195 is assembled from a collection of components that may be individually selected and customized to the requirements of the Content Provider(s) 101 and Electronic Digital Content Store(s) 103. A generic version of the player is described, but customization is possible.

Referring now to FIG. 15 there is shown a block diagram of the major components and processes of the Player Application 195 running on End-User Device(s) 109 of FIG. 10.

There are several component-sets that make up the subsystems of the Player Object Manager 1501:

1. End-User Interface Components 1509
2. Copy/Play Management Components 1504
3. Decryption 1505, Decompression 1506, Playback Components 1507 and may include recording.
4. Data Management 1502 and Library Access Components 1503
5. Inter-application Communication Components 1508
6. Other miscellaneous (Installation, etc) Components

Components from within each of these sets may be selected, based on the requirements of:

- the platform (Windows, Unix, or equivalent)
- communications protocols (network, cable, etc)
- Content Provider(s) 101 or Electronic Digital Content Store(s) 103
- Hardware (CD, DVD, etc)
- Clearinghouse(s) 105 technology and more.

The sections below detail the various component sets. The final section details how these components are put together in the generic player, and discusses how the components can be customized.

In another embodiment, the components of the Player Application 195 and the SC(s) Processor 192 are available as part of a programmer's software toolkit. This toolkit enables predefined interfaces to the components of the generic player application listed above. These predefined interfaces are in the form of APIs or Application Programming Interfaces. A developer using these APIs can implement any of the functionality of the components from a high level application program. By providing APIs to these components, a programmer can quickly develop a customized Player Application 195 without the need to re-created these functions and resources of any of these components.

2. End-User Interface Components 1509

Components from this set combine to provide the on-screen manifestation of the Player Application 195. Note that the design establishes no definitive layout of these components. One such layout is provided in the generic player. Based on requirements from Content Provider(s) 101 and/or Electronic Digital Content Store(s) and other requirements, alternate layouts are possible.

This set is grouped into subgroups, starting with the components used to present End-User Display 1510 and handle controls called End-User Controls 1511 used for such low-level functions as audio playback , and presentation of metadata. Next, the End-User Display Component 1510 is further divided by special function groupings (Play-list, Digital Content Library), and then object-container components used for grouping and placing of those lower-level components.

Within the component listings below, any reference to creating CDS or copying of Content 113 to a CD or other recordable medium only applies to the case where the Player Application 195 has such functionality enabled. Also note that the term CD in that context is a generic one, that can also represent various other external recording devices, such as MiniDisc or DVD.

FIG. 16 is an example user interface screens of the Player Application 195 of FIG. 15 according to the present invention. Function for the End-User Controls 1511 include (corresponding screens of an End-User Interface are shown 1601-1605):

Controls for performing the Content 113:

- Play/Stop button
- Play button
- Stop button
- Pause button
- Skip forward button
- Skip backward button
- Volume control
- Track position control/display
- Audio channel volume level display and more.

Controls for the displaying metadata associated with the Content 113

- Cover Picture button
- Cover Picture object
- Artist Picture button
- Artist Picture object
- Track List button
- Track List Information object

- Track List Selector object (click to play)
- Track Name object
- Track Information object
- Track Lyrics button
- Track Lyrics object
- Track Artist Name object
- Track Credits button
- Track Credits object
- CD Name object
- CD Credits button
- CD Credits object
- Generic (Configurable) Metadata button
- Generic Metadata object and more.

Function for the End-User Display 1510 include (corresponding screens of an End-User Interface are shown 1601 - 1605):

Play-list of display container

- Play-list Management button
- Play-list Management window
- Digital Content search button
- Digital Content search Definition object
- Digital Content search Submit button
- Digital Content search Results object
- Copy Selected Search Result Item To Play-list button
- Play-list object (editable)
- Play-list Save button
- Play-list Play button
- Play-list Pause button
- Play-list Restart button
- Create CD from Play-list button and more.

Display of Digital Content Library 196

- Digital content library button
- Digital content librarian window
- Digital content categories button
- Digital content categories object

- By-artist button
- By-genre button
- By-label button
- By-category button
- Delete button
- Add-to-Play-list button
- Copy to CD button
- Song List object
- Song List display container and more

Containers and Misc.

- Player window container
- Audio controls container
- Metadata controls container
- Metadata display container
- Toolbar container object
- Sample button
- Download button
- Purchase button
- Record button
- Player Name object
- Label/Provider/Store Advertisement object
- Label/Provider/Store URL button
- Artist URL Button and more

3. Copy/Play Management Components 1504

These components handle set up of encryption keys, Watermark processing, Copy management, and more. Interfaces also exist for communication with the Clearinghouse(s) 105, transmission of purchase requests, and more, for special services such as pay per listen or cases where each access to the Content 113 is accounted for. Currently, the communications to the Clearinghouse(s) 105 functions are handled by the SC(s) Processor 192.

The use of the Content 113 by the Player Applications 195 on End User Device(s) 109 is logged into a database such as the License Database 197. The tracking of each use of Content 113 by the Player Application 195 can be transmitted to one or more logging sites such as the Clearing House(s) 105 or Content Provider(s) 101 or Electronic Digital Content Store(s) 103 or any site designated and coupled to Transmission Infrastructures 107. This transmission can be scheduled at predetermined times to upload the usage information to a logging site. One predetermined time contemplated is early in the morning when Transmission Infrastructures 107 may not be as

congested with network traffic. The Player Application 195 using known techniques, wakes-up at a scheduled time, and transmit the information from the local logging database to the logging site. By reviewing the logging site information, the Content Provider(s) 101 can measure the popularity of their Content 113.

In another embodiment, the instead of logging the usage of Content 113 for later uploading to a logging site, the use of the Content 113 is uploaded to the logging site during every use of the Content 113. For example, when duplicating or copying the Content 113 stored at the End User Device(s) 109, on to an external device such as DVD Disc, digital tape, flash memory, mini Disc or equivalent read/writeable removable media, the use is updates to the logging site. This may be a precondition to copying the Content 113 in the usage conditions 206 that is transmitted when the Content 113 is purchased. This ensures the Content Provider(s) 101 can accurately track the usage of their Content 113 during their playing, duplicating or other actions upon the Content 113.

In addition, other information about the Content 113 can be uploaded to the logging site. For example the last time (e.g., hour and day) the Content 113 was performed; how many times the Content 113 was performed; if the Content 113 has been duplicated or copied to an authorized external device such as DVD Disc, digital tape or mini-Disc. In cases where there are multiple distinct users of a single Player Application 195 on the End User Device(s) 109, such as different members of a family, the identifications of the user of the Content 113 is transmitted along with the usage information to the logging site. By reviewing the usage information uploaded to the logging site, the Content Provider(s) 101 can measure the popularity of the Content 113 base on the actual usage, the identification of the user and the number of times the Content 113 has been performed. The actual usage measurement makes this system more factual driven over systems using sampling methods, such as a Nielsen Rating scheme for televisions, or telephone surveys, where only a limited number of users are sampled at any one time and the results extrapolated. In this present embodiment, the actual usage can be measures for the users logging back onto a designated web site such as the Electronic Digital Content Store(s) 103 or Content Provider(s) 101.

4. Decryption 1505, Decompression 1506 and Playback Components 1506

These components use the keys acquired by the Copy/Play Management components to unlock the audio data acquired from the Data Management and Library Access components, apply the appropriate decompression to prepare it for playback, and use system audio services to play it. In an alternate embodiment, the audio data acquired from the Data Management and Library Access components may be copied to removable media such as CDS, diskettes, tapes or MiniDisks.

5. Data Management 1502 and Library Access Components 1503

These components are used to store and retrieve song data on various storage devices on the End-User(s)' system, as well as handle requests for information about the stored songs.

6. Inter-application Communication Components 1508

These components are used for coordination between the Secure Digital Content Electronic Distribution Player and other applications (e.g., Browser, helper-app and/or plug-in, etc) that may invoke the Player Application 195, or that the Player Application 195 needs to use when carrying out its functions. For example, when a URL control is activated, it invokes the appropriate browser and instruct it to load the appropriate page.

7. Other Miscellaneous Components

Individual components that don t fall into the categories above (e.g., Installation) are grouped here.

8. The Generic Player

In this section the combining of the components above into a version of the Player Application 195 is discussed. This is just one of many different examples possible, since the Player Application 195 is designed for customization by being based on software objects. The Player Object Manager 1501 is a software framework holding all the other components together. As discussed in the sections above, the blocks below the Player Object Manager 1501 in this diagram are required for any player, but may be replaced by specialized versions depending on such things as form of encryption or scrambling being used, types of audio compression, access methods for the Content 113 library, and more.

Above the Player Object Manager 1501 are Variable Objects 1512, which are mostly derived from the metadata associated with the Content 113 being played or searched. These Variable Objects are made available to the End-User Device(s) 109 by way of the End-User Display 1510 and received input from the End-User Controls 1511. All objects are configurable, and the layouts of all containers are customizable. These objects may be implemented in C/C++, Java or any equivalent programming language.

Using the Player Application 195

The following embodiment is for an example where the Player Application 195 running on End-User Device(s) 109 is an audio player where Content 113 is music. It should be understood to those skilled in the art that other types of Content 113 can be supported by the Player Application 195. A typical audio enthusiast has a library of CDS holding songs. All of these are available within the Secure Digital Content Electronic Distribution System 100. The set of songs that have been purchased from Electronic Digital Content Store(s) 103 are stored within a Digital Content Library 196 on his or her system. The groupings of songs that are analogous to physical CDS are stored as Play-lists. In some cases a Play-list exactly emulates a CD (e.g., all tracks of a commercially available CD has been purchased from an Electronic Digital Content Store(s) 103 as an on-line version of the CD and is defined by a Play-list equivalent to that of the CD). But most Play-lists is put together by End-User(s) to group songs they have stored in the Digital Content Libraries on their systems. However for the purposes of the ensuing discussions, an example of a custom made music CD is used when the term a Play- list is mentioned.

When the End-User(s) starts the Player Application 195 explicitly, rather than having it start up via invocation from the SC(s) Processor 192 Application, it pre-loads to the last Play-list that was accessed. If no

Play-lists exist in the Digital Content Library 196, the Play-list editor is started automatically (unless the user has turned off this feature via a preference setting). See The Play-list, below for further details.

The Player Application 195 may also be invoked with a specific song as an argument, in which case it immediately enters Song-play mode. Optionally, the song may be prepared for play but await action by the End-User(s) before proceeding. See Song Play, below for more on this situation.

The Play-list (corresponding screen of an End-User Interface 1603):

When the End-User(s) has invoked the Play-list function, these are the available functions:

* Open Play-list
* Digital Content Librarian is invoked to display a list of stored Play-lists for selection. Also see Digital Content Librarian below for more info.
* Edit Play-list
* Invokes the Play-list Editor (see below), primed with the current Play-list if one has been loaded already. Otherwise the editor creates an empty Play-list to start with.
* Run Play-list
* Songs are played one at a time starting with the selected song (or the beginning of the play-list, if no song is selected). Options set in the Play-list Editor affect the sequencing of the playback. However there is controls available here to override those options for this play of the Play-list.
* Play song
* Only the selected song from the Play-list is played. See Song Play below for more info.
* Play-list Info
* Display information about the Play-list.
* Song Info
* Display information about the selected song within the Play-list.
* Visit web site
* Load web site associated with this Play-list into browser.
* Librarian
* Open the Digital Content Librarian window. Also see Digital Content Librarian below for more info.

The Play-list Editor (corresponding screen of an End-User Interface 1603):

When invoking the Play-list editor, these are the End-User(s)' options:

* View/Load/Delete Play-lists
* Digital Content Librarian is invoked to display a list of stored Play-lists for selection of one to load or delete. Also see Digital Content Librarian below for more info.
* Save Play-list
* Current version of Play-list is saved in the Digital Content Library 196.
* Delete Song

* Currently selected song is deleted from Play-list.
* Add Song
* Digital Content Librarian is invoked in song-search mode, for selection of song to add to the Play-list. Also see Digital Content Librarian below for more info.
* Set Song Information
* Display and allow changes to information about the selected song within the play-list. This information is stored within the Play-list, and does not alter information about the song stored within the Digital Content Library 196. These things can be changed:
* Displayed Song Title
* End-User(s) notes about the song
* Lead-in delay on playing the song
* Follow-on delay after playing the song
* Start-point within song when playing
* End-point within song when playing
* Weighting for random mode
* Volume adjustment for this song and more.

Set Play-list attributes: Display and allow changes to the attributes of this Play-list. These attributes may be set:

* Play-list title
* Play-list mode (random, sequential, etc)
* Repeat mode (play once, restart when done, etc)
* End-User(s) notes about this Play-list

Librarian (corresponding screen of an End-User Interface 1601):

* Open the Digital Content Librarian window. Also see Digital Content Librarian below for more info.

Song Play

When a song has been prepared for play, either by invoking the Player Application 195 with the song as an argument or by selecting a song for play from a Play-list or within the Digital Content Librarian, these are the End-User(s)' options: (corresponding screen of an End-User Interface 1601):

* Play
* Pause
* Stop
* Skip Backward
* Skip Forward
* Adjust Volume
* Adjust Track Position

* View Lyrics
* View Credits
* View CD Cover
* View Artist Picture
* View Track Information
* View other metadata
* Visit web site
* Play-list
* Librarian and more.

Digital Content Librarian

The Digital Content Librarian can be invoked implicitly when selecting songs or Play-lists (see above) or may be opened in its own window for management of the Song Library on the End-User(s)' system. In that case, these are the End-User(s)' options:

Working with songs:

Sort All by Artist, Category, Label, other

Select Songs by Artist, Category, Label, other

Add selected songs to Current Play-list

Copy Song to CD (if enabled)

Delete Song

Add Song to Category and more.

Work with Play-lists:

Sort by Name

Sort by Category

Search by Keyword

Search by Included Song Title

Load Selected Play-list

Rename Play-list

Delete Play-list

Create CD from Selected Play-list (if enabled) and more.

Although a specific embodiment of the invention has been disclosed, it will be understood by those having skill in the art that changes can be made to this specific embodiment without departing from the spirit and scope of the invention. The scope of the invention is not to be restricted, therefore, to the specific embodiment, and it is intended that the appended claims cover any and all such applications, modifications, and embodiments within the scope of the present invention.

WHAT IS CLAIMED IS:

1. A clearinghouse that enables securely providing data, the clearinghouse being capable of communicating with a system, the system being capable of receiving both data encrypted with a first encryption key and an encrypted first encryption key, the encrypted first encryption key being the first encryption key encrypted by a second encryption key, the clearinghouse comprising:

decryption of a first decryption key from the encrypted first decrypting key; and

transfer of the decrypted first decrypting key to a system.

2. A system for securely providing data comprising:

encryption of data by a first encrypting key to generate encrypted data;

encryption of a first decrypting key by a second encrypting key to generate an encrypted first decrypting key;

transfer of the encrypted data to a second system;

transfer of the encrypted first decrypting key to a second system;

transfer of the encrypted first decrypting key to a clearinghouse that possesses a second decrypting key;

decryption of the first decrypting key by the second decrypting key; and

transfer of the first decrypting key to a second system.

3. The system of claim 2, further comprising, prior to the transfer of the first decrypting key re-encryption of the first decrypting key by a third encrypting key and where the transfer of the first decrypting key to a second system is the transfer of the decrypted and re-encrypted first decrypting key to a second system.

4. The system of claim 3, wherein the third encrypting key is a public key of a second system.

5. The system of claim 2, wherein the second encrypting key is a public key of the clearinghouse and the second decrypting key is a corresponding private key of the clearinghouse.

6. The system of claim 2, further comprising confirming that the data was paid.

7. A system of securely providing data to a second system, the data being encrypted so as to be decryptable by a data decrypting key, the data decrypting key being encrypted using a first public key, said method comprising the steps of:

receiving the encrypted data decrypting key by a clearinghouse;

decrypting the data decrypting key using a first private key;

re-encrypting the data decrypting key using a second public key, the second public key having a corresponding second private key; and;

transferring the re-encrypted data decrypting key to a second system.

8. The system of claim 7, wherein the encrypted data decrypting key is received from another system.

9. The system of claim 8, further comprising authorization for the data prior to transferring the encrypted data decrypting key to a second system.

10. A system for managing content data, associated metadata, and associated usage condition data comprising:

metadata and usage condition data for associated content data;

altering at least one of the metadata and the usage condition data to create promotional data; and

transferring the promotional data.

11. The system of claim 10 wherein the content data includes music data, and the usage condition data includes at least one of a time restriction on the playing of the music data, a maximum number of copies of the music data that can be made and a maximum number of times the music data can be played.

12. The system of claim 10,

wherein the content data includes music data, and

the metadata includes at least one of a link to a content data host, a description of the content of the music data, artwork associated with the music data, and a selected portion of the music data.

13. An electronic content management system for managing content data, associated metadata, and associated usage condition data, said system comprising:

a content provider capable of transmitting associated metadata of content data, and associated usage condition data of content data; and

an electronic store capable of receiving the metadata and the usage condition data from the content provider and generating altered promotional data from at least one of a portion of the metadata and a portion of the usage condition data.

14. The system of claim 13 wherein the content provider is further capable of encrypting the content data with a first encrypting key, encrypting the first encrypting key with a second encrypting key; and transmitting the encrypted first encrypting key.

15. The system of claim 13,

wherein the content data includes music data, and

the usage condition data includes at least one of time restrictions on the playing of the music data a maximum number of copies of the music data that can be made and a maximum number of times the music data can be played.

16. The system of claim 13,

wherein the content data includes music data, and

the metadata transmitted at least one of the information identifying the content provider, a link to the content host; and at least one of a description of the content of the music data, artwork associated with the music data, and a selected portion of the music data.

17. The system of claim 13 wherein the content provider is capable of transmitting content data, the system further comprising a content host capable of receiving the content data from the content provider.

18. A digital content data player for playing digital content data, said data player comprising a transmitter for transmitting usage information, the usage information being at least one of the occurance of the playing or copying of the digital content data, the number of times the digital content data was played or copied, the time the digital content data was played or copied, and the identification of a user that plays or copies the digital content data.

19. The data player as defined in claim 18, wherein the digital content data includes digital music data.

20. A system for tracking usage of digital content comprising:

a license to play or copy digital content data;

licensed digital content data; and

information on at least one of the occurrence of playing or copying of the licensed digital content data, the number of times the licensed digital content data was played or copied, the time the licensed digital content data was played or copied and the identification of a user that plays or copies the licensed digital content data.

21. The system of claim 20 further comprising prohibiting further playing or copying based on the information.

22. The system in claim 20, wherein the digital content data includes digital music data.

23. The system of claim 20, wherein the information is transmitted at a predetermined time or at a predetermined interval.

26. A computer readable medium containing program instructions for tracking usage of digital content data on user devices, comprising instructions for:

obtaining a license to play or copy digital content data;

accepting the licensed digital content; and

transmitting information on at least one of playing of the digital content data, copying of the digital content data, the time the digital content data was played or copied and the identification of a user that plays or copies the digital content data.

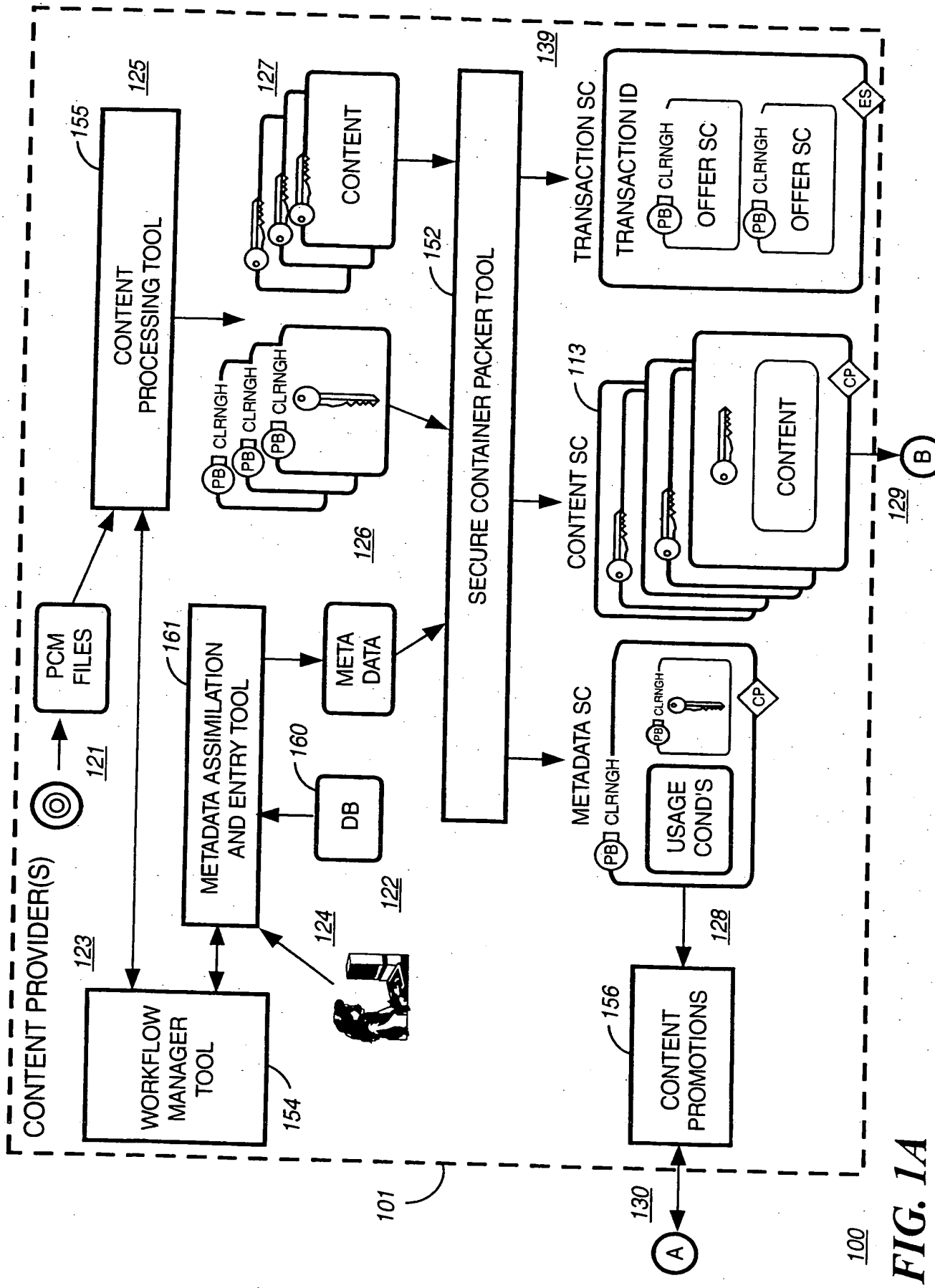

CONTENT PROVIDER(S)
PCM FILES
121
CONTENT PROCESSING TOOL
125
155
METADATA ASSIMILATION AND ENTRY TOOL
161
DB
160
META DATA
126
CONTENT
127
CLRNGH
PB
SECURE CONTAINER PACKER TOOL
152
139
TRANSACTION SC
TRANSACTION ID
OFFER SC
ES
CONTENT SC
113
CONTENT
CP
129
B
METADATA SC
USAGE COND'S
128
WORKFLOW MANAGER TOOL
154
123
124
122
CONTENT PROMOTIONS
156
130
A
101
100

FIG. 1A

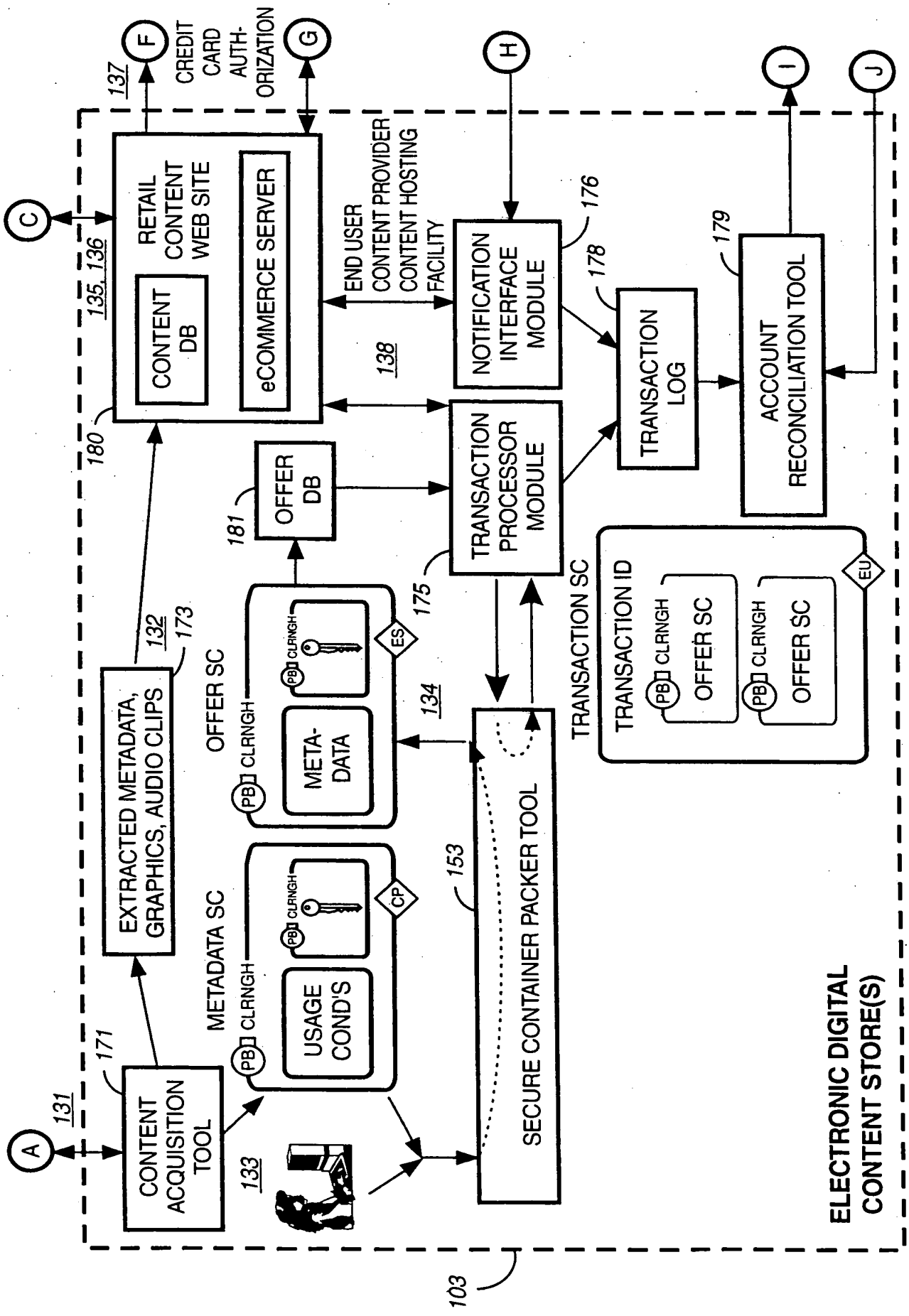

A
131
CONTENT ACQUISITION TOOL
171
EXTRACTED METADATA, GRAPHICS, AUDIO CLIPS
132
173
133
METADATA SC
PB CLRNGH
USAGE COND'S
CP
OFFER SC
PB CLRNGH
META-DATA
ES
134
175
SECURE CONTAINER PACKER TOOL
153
OFFER DB
181
TRANSACTION PROCESSOR MODULE
TRANSACTION SC
TRANSACTION ID
PB CLRNGH
OFFER SC
EU
TRANSACTION LOG
178
ACCOUNT RECONCILIATION TOOL
179
NOTIFICATION INTERFACE MODULE
176
H
I
J
C
RETAIL CONTENT WEB SITE
135, 136
CONTENT DB
eCOMMERCE SERVER
180
END USER
CONTENT PROVIDER
CONTENT HOSTING
FACILITY
138
137
F
CREDIT CARD AUTH-ORIZATION
G
ELECTRONIC DIGITAL CONTENT STORE(S)
100
103

FIG. 1B

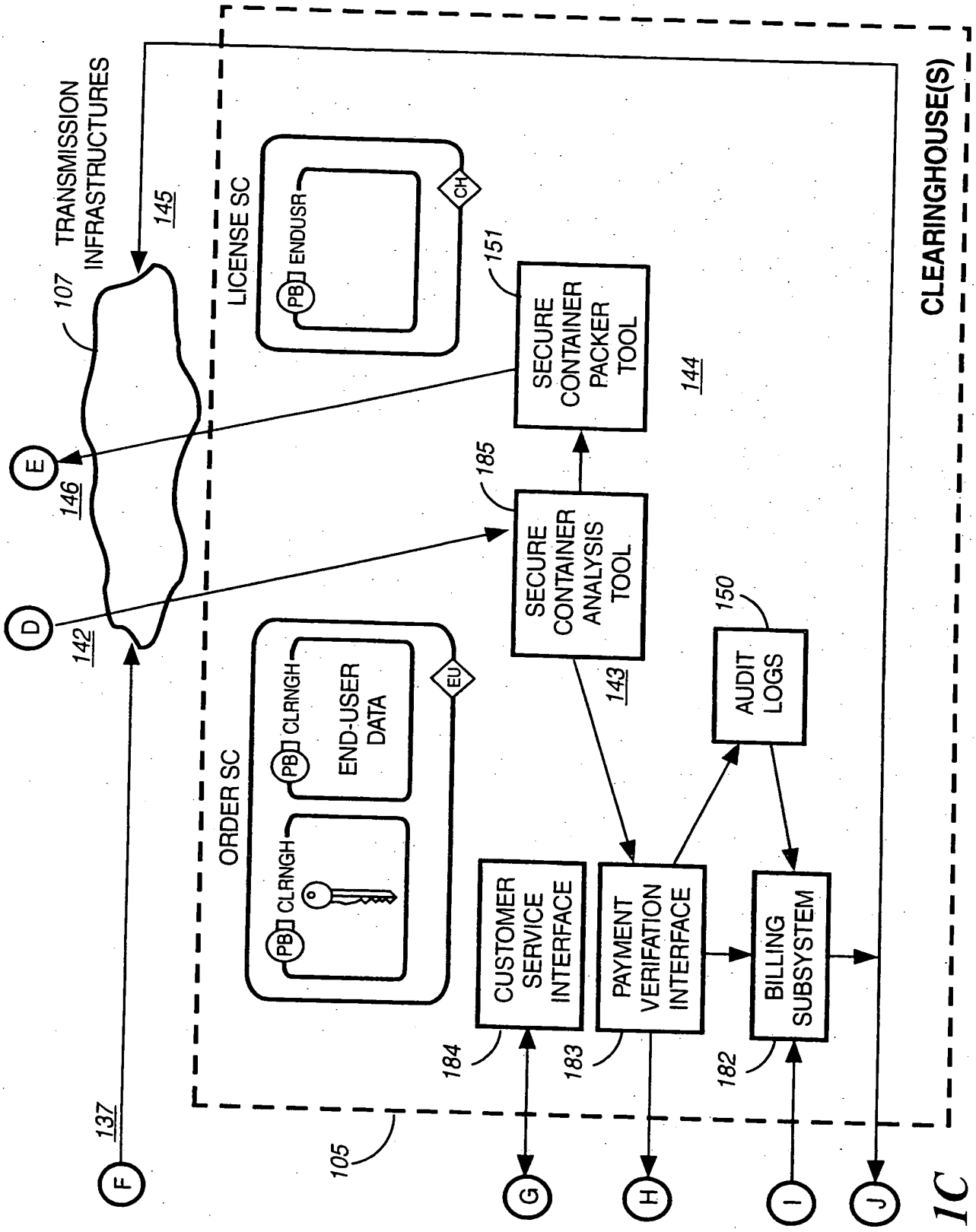

CLEARINGHOUSE(S)
100
TRANSMISSION INFRASTRUCTURES
107
145
LICENSE SC
ENDUSR
PB
CH
151
SECURE CONTAINER PACKER TOOL
144
185
SECURE CONTAINER ANALYSIS TOOL
143
150
AUDIT LOGS
ORDER SC
CLRNGH
END-USER DATA
EU
CLRNGH
CUSTOMER SERVICE INTERFACE
PAYMENT VERIFATION INTERFACE
BILLING SUBSYSTEM
184
183
182
105
137
142
146
D
E
F
G
H
I
J

FIG. 1C

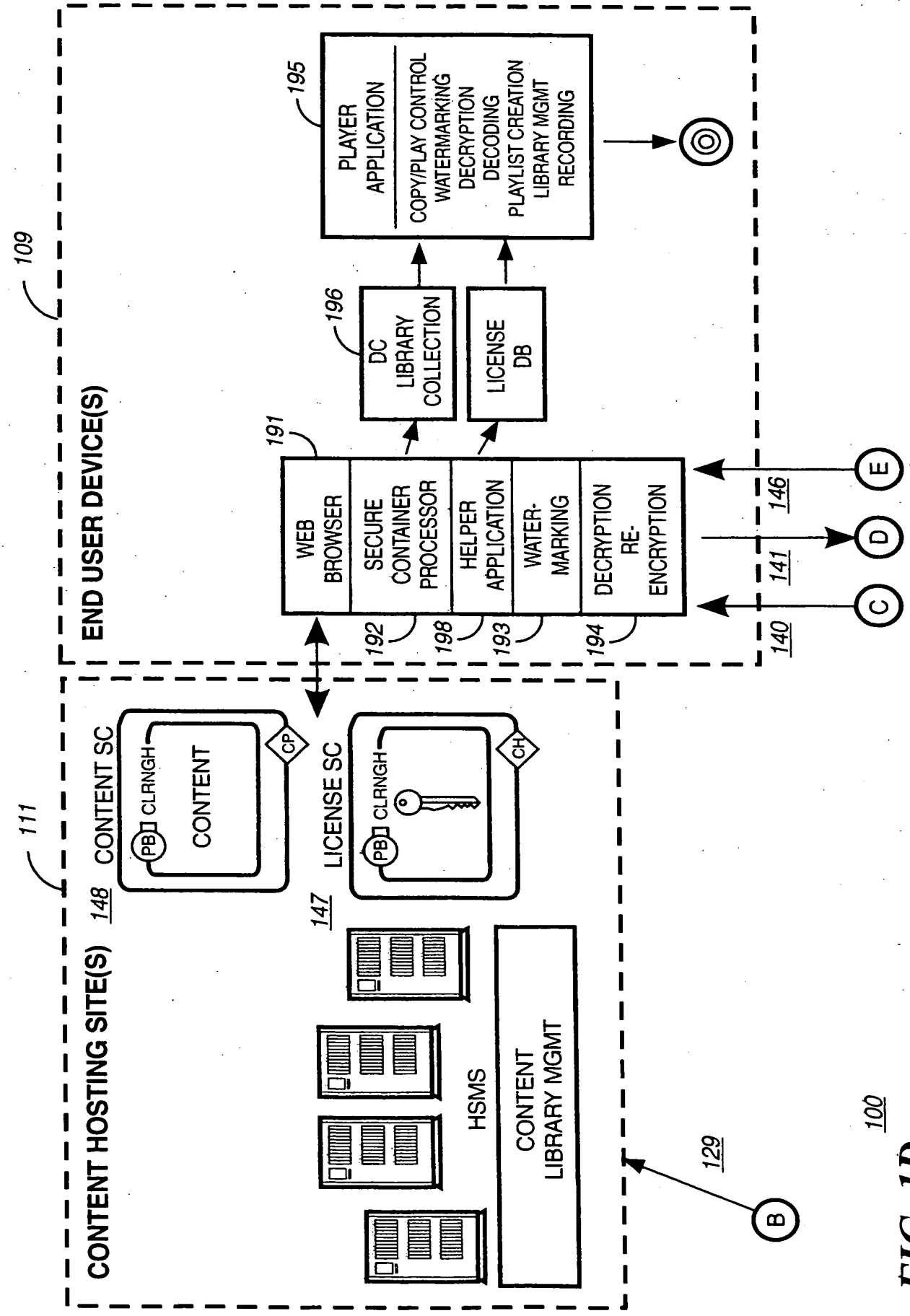

109
END USER DEVICE(S)
195
PLAYER APPLICATION
COPY/PLAY CONTROL
WATERMARKING
DECRYPTION
DECODING
PLAYLIST CREATION
LIBRARY MGMT
RECORDING
196
DC LIBRARY COLLECTION
LICENSE DB
191
WEB BROWSER
SECURE CONTAINER PROCESSOR
HELPER APPLICATION
WATER-MARKING
DECRYPTION RE-ENCRYPTION
192
198
193
194
140
146
141
E
D
C
111
CONTENT HOSTING SITE(S)
148
CONTENT SC
PB
CLRNGH
CONTENT
CP
147
LICENSE SC
PB
CLRNGH
CH
HSMS
CONTENT LIBRARY MGMT
129
B
100

FIG. 1D

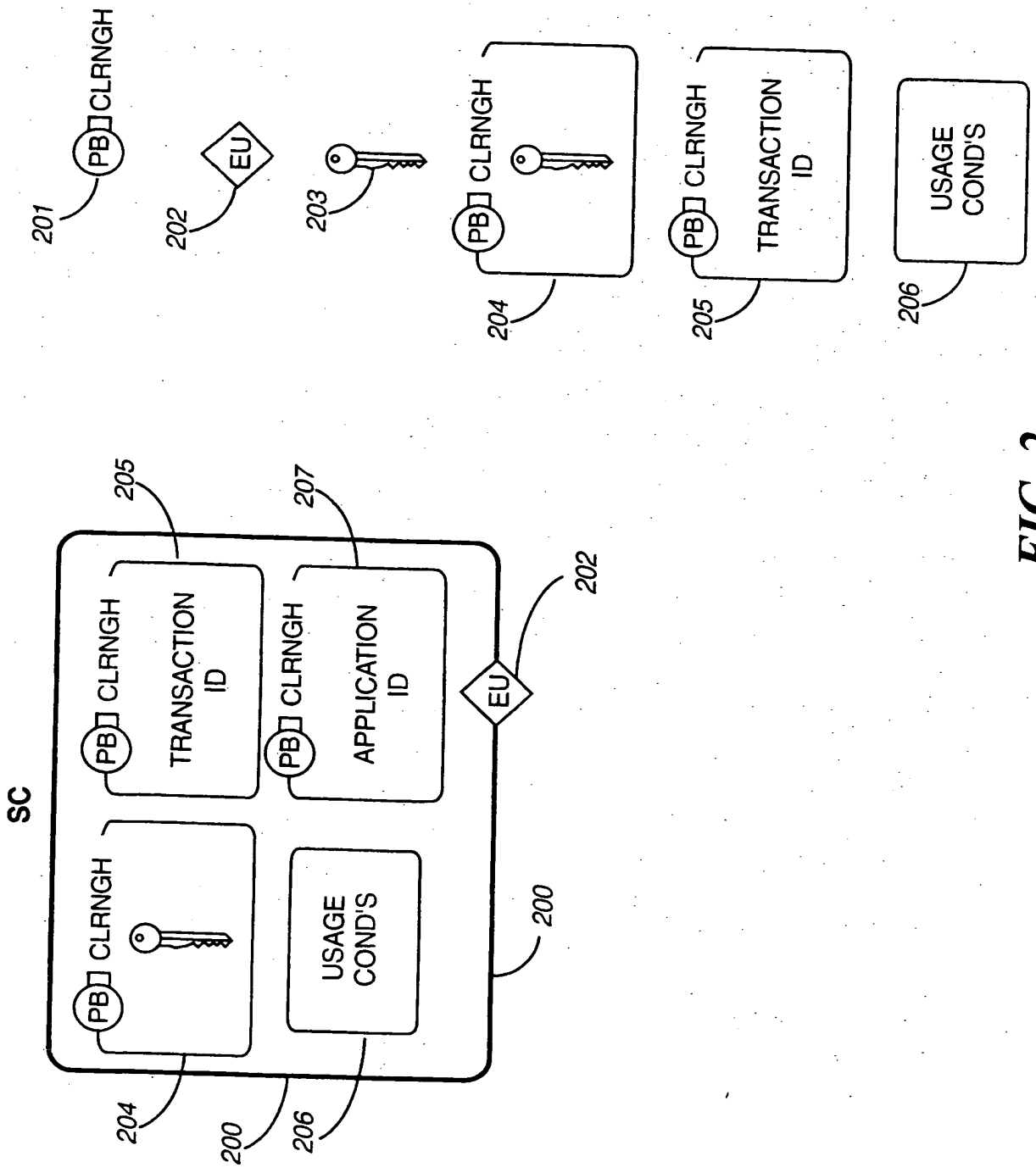

201
PB
CLRNGH
202
EU
203
204
PB
CLRNGH
205
PB
CLRNGH
TRANSACTION ID
206
USAGE COND'S
SC
205
TRANSACTION ID
207
APPLICATION ID
202
200
204
206

FIG. 2

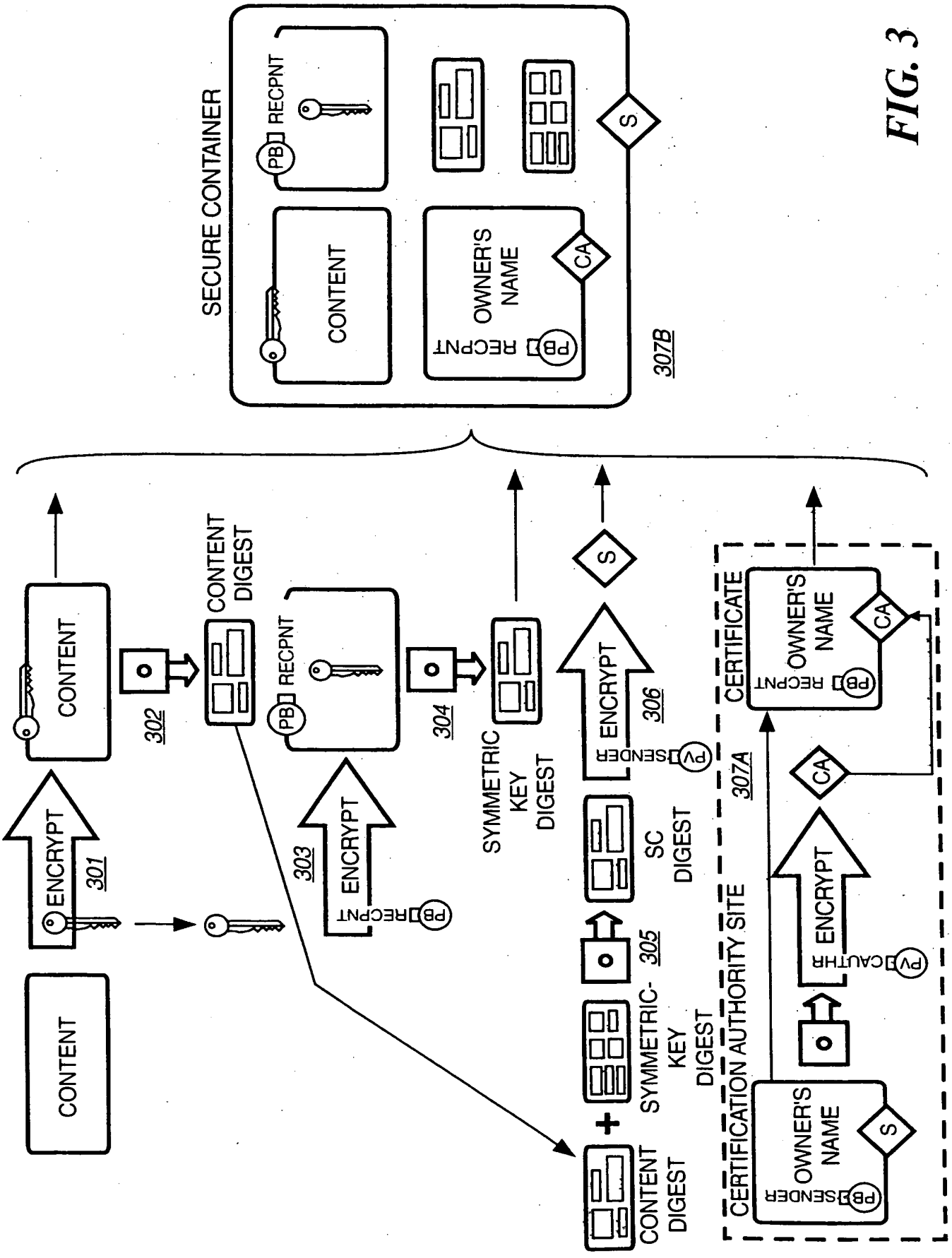

SECURE CONTAINER
RECPNT
PB
CONTENT
OWNER'S NAME
CA
RECPNT
PB
S
307B
CONTENT
ENCRYPT
301
CONTENT
302
CONTENT DIGEST
RECPNT
PB
303
ENCRYPT
RECPNT
PB
304
SYMMETRIC KEY DIGEST
S
ENCRYPT
306
SENDER
PV
SC DIGEST
SYMMETRIC-KEY DIGEST
305
CONTENT DIGEST
CERTIFICATE
OWNER'S NAME
CA
RECPNT
PB
CA
ENCRYPT
CAUTHR
PV
CERTIFICATION AUTHORITY SITE
307A
OWNER'S NAME
S
SENDER
PB

FIG. 3

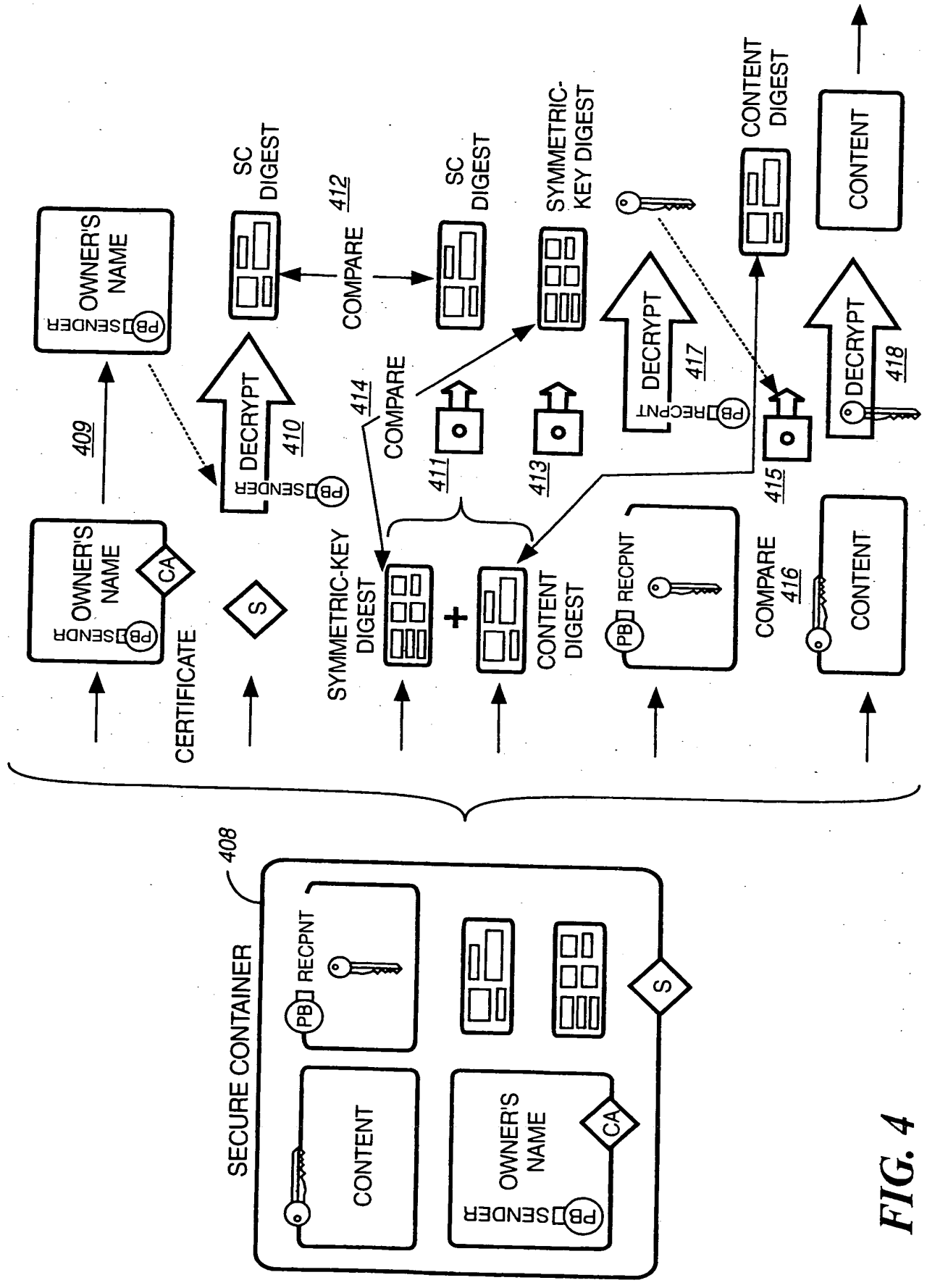

SECURE CONTAINER
408
PB RECPNT
CONTENT
OWNER'S NAME
CA
PB SENDER
S
409
OWNER'S NAME
CA
PB SENDER
CERTIFICATE
DECRYPT
410
PB SENDER
SC DIGEST
412
COMPARE
SC DIGEST
414
COMPARE
411
413
SYMMETRIC-KEY DIGEST
SYMMETRIC-KEY DIGEST
CONTENT DIGEST
+
PB RECPNT
DECRYPT
417
PB RECPNT
CONTENT DIGEST
415
COMPARE
416
CONTENT
DECRYPT
418
CONTENT

FIG. 4

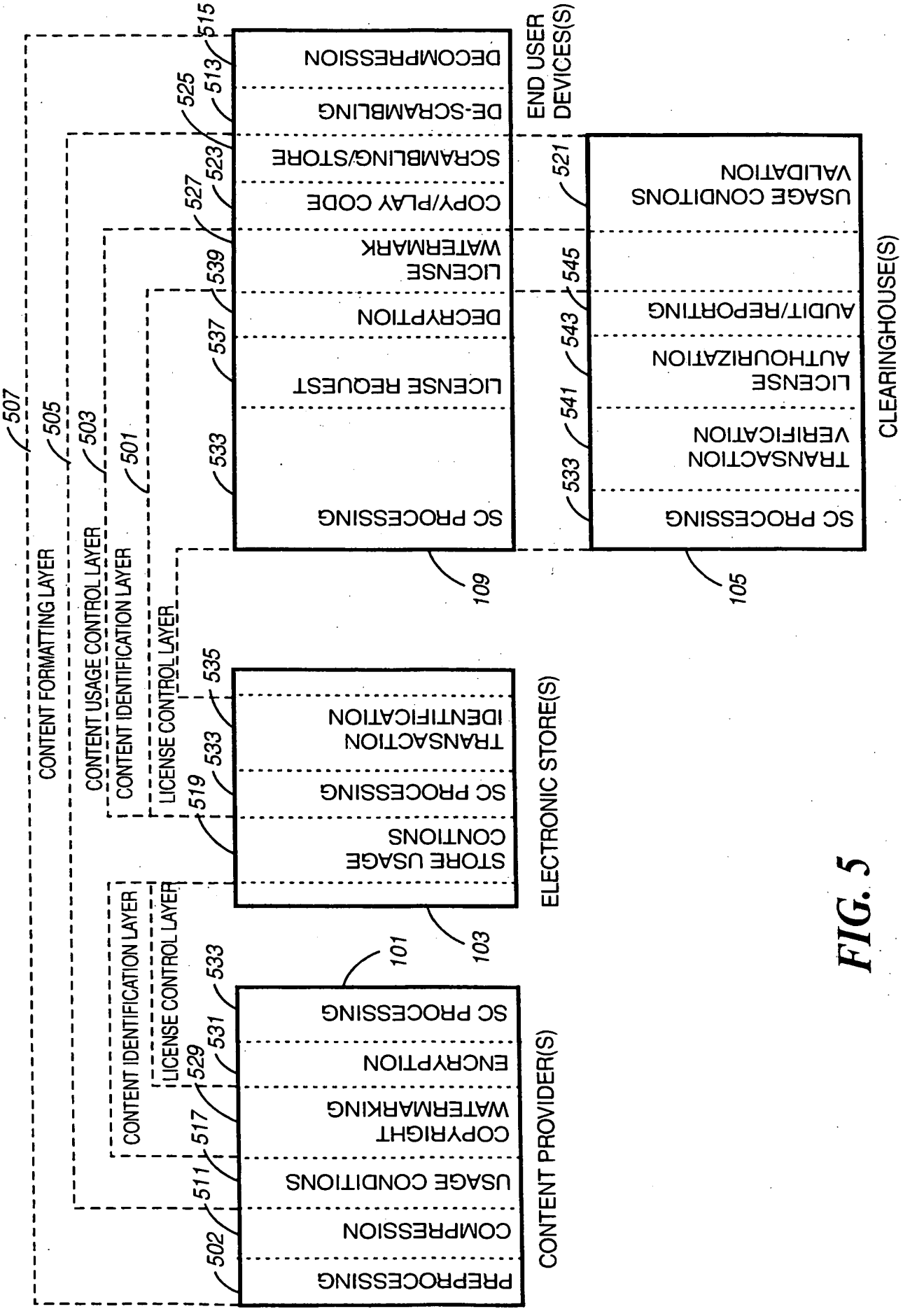


FIG. 5

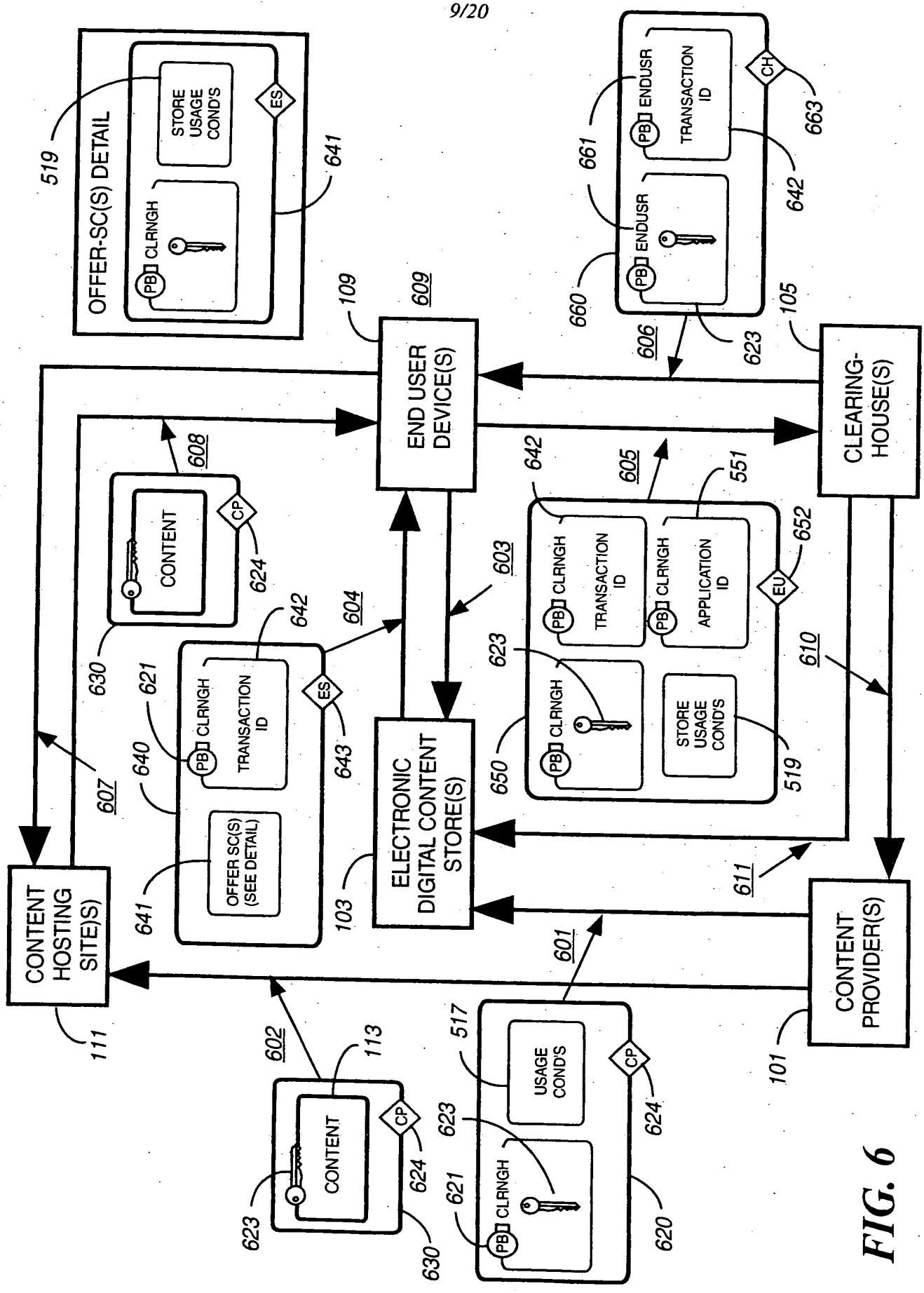


FIG. 6

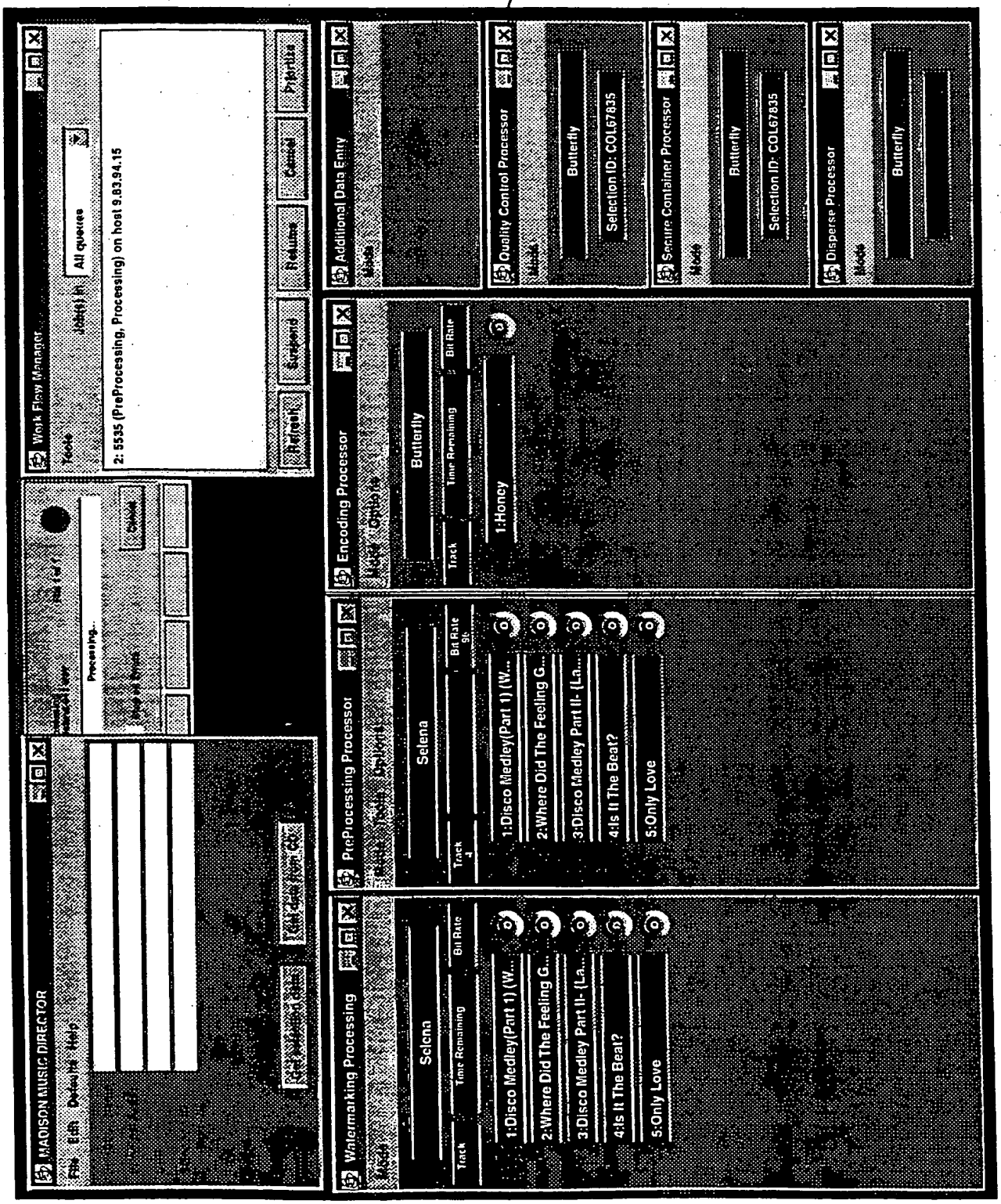

700
MADISON MUSIC DIRECTOR
Work Flow Manager
Tools
All queues
2: 5535 (PreProcessing, Processing) on host 9.83.94.15
Refresh
Suspend
Refilter
Cancel
Priorities
Additional Data Entry
Quality Control Processor
Secure Container Processor
Disperse Processor
Butterfly
Selection ID: COL67835
Encoding Processor
PreProcessing Processor
Watermarking Processing
Selena
Track
Time Remaining
Bit Rate
1:Disco Medley(Part 1) (W...
2:Where Did The Feeling G...
3:Disco Medley Part II- (La...
4:Is It The Beat?
5:Only Love
1:Honey

FIG. 7

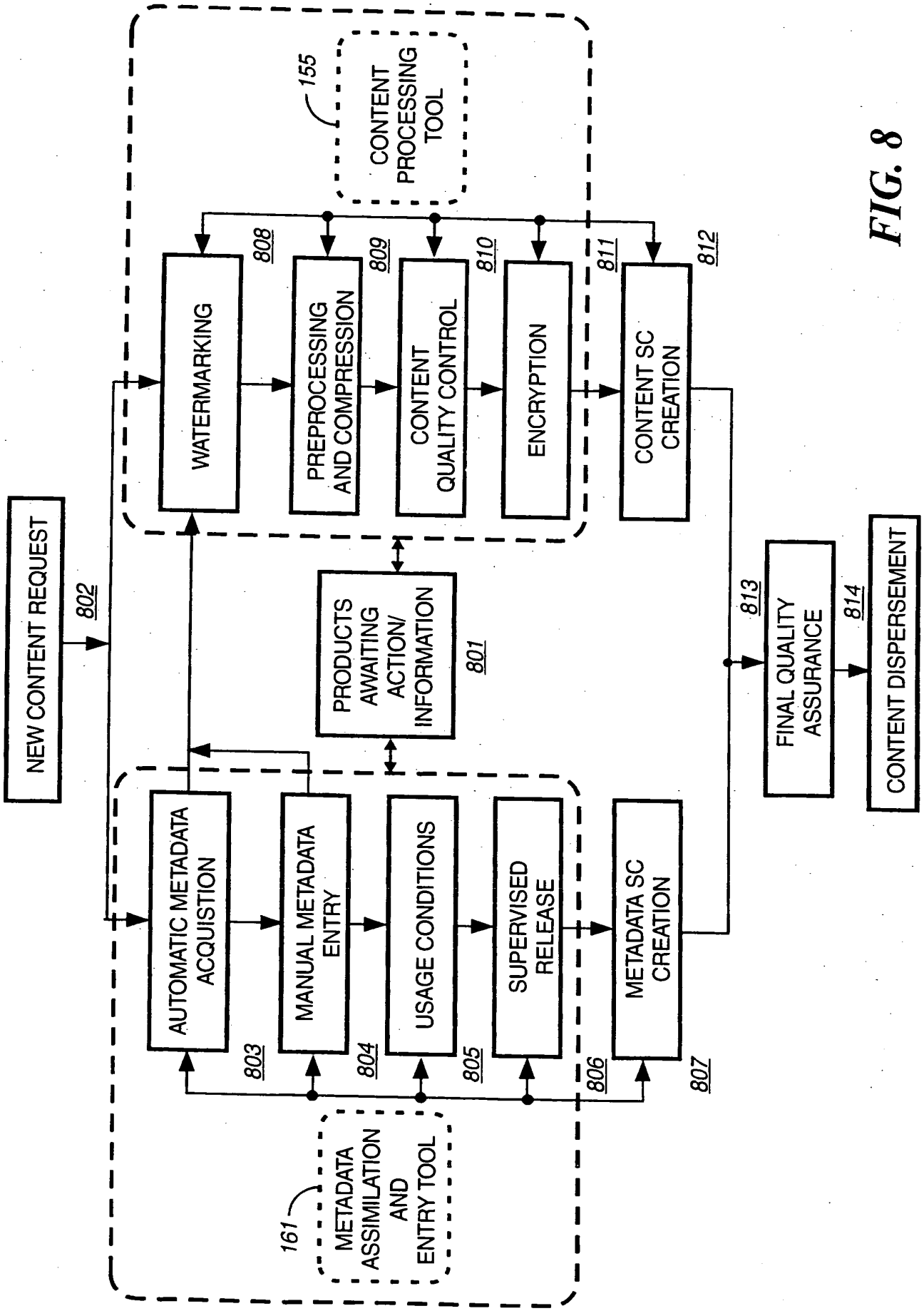


FIG. 8

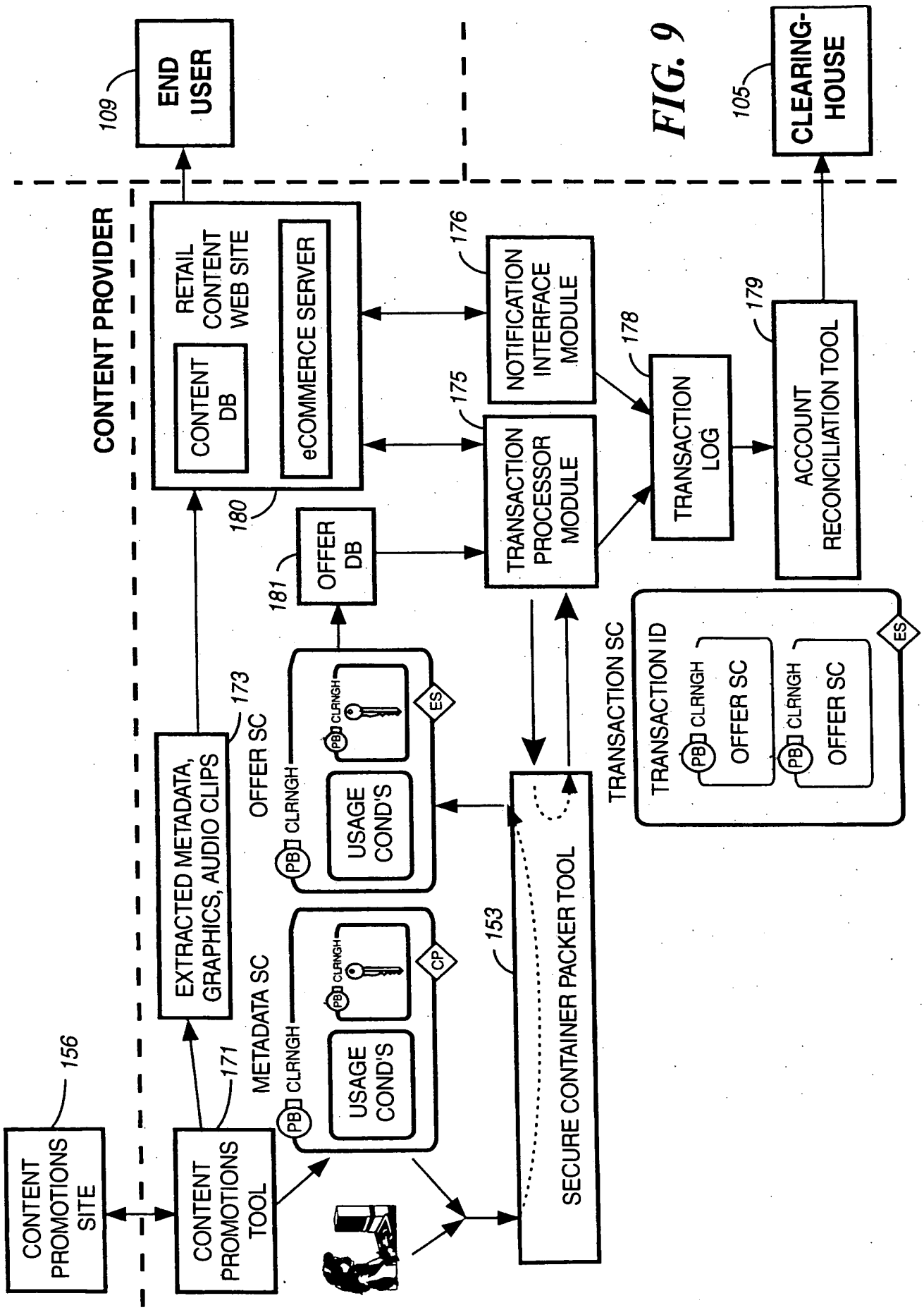

CONTENT PROVIDER
109
END USER
105
CLEARING-HOUSE
RETAIL CONTENT WEB SITE
CONTENT DB
eCOMMERCE SERVER
180
176
NOTIFICATION INTERFACE MODULE
175
TRANSACTION PROCESSOR MODULE
178
TRANSACTION LOG
179
ACCOUNT RECONCILIATION TOOL
181
OFFER DB
173
EXTRACTED METADATA, GRAPHICS, AUDIO CLIPS
OFFER SC
PB
CLRNGH
USAGE COND'S
ES
METADATA SC
CP
153
SECURE CONTAINER PACKER TOOL
TRANSACTION SC
TRANSACTION ID
OFFER SC
156
CONTENT PROMOTIONS SITE
171
CONTENT PROMOTIONS TOOL

FIG. 9

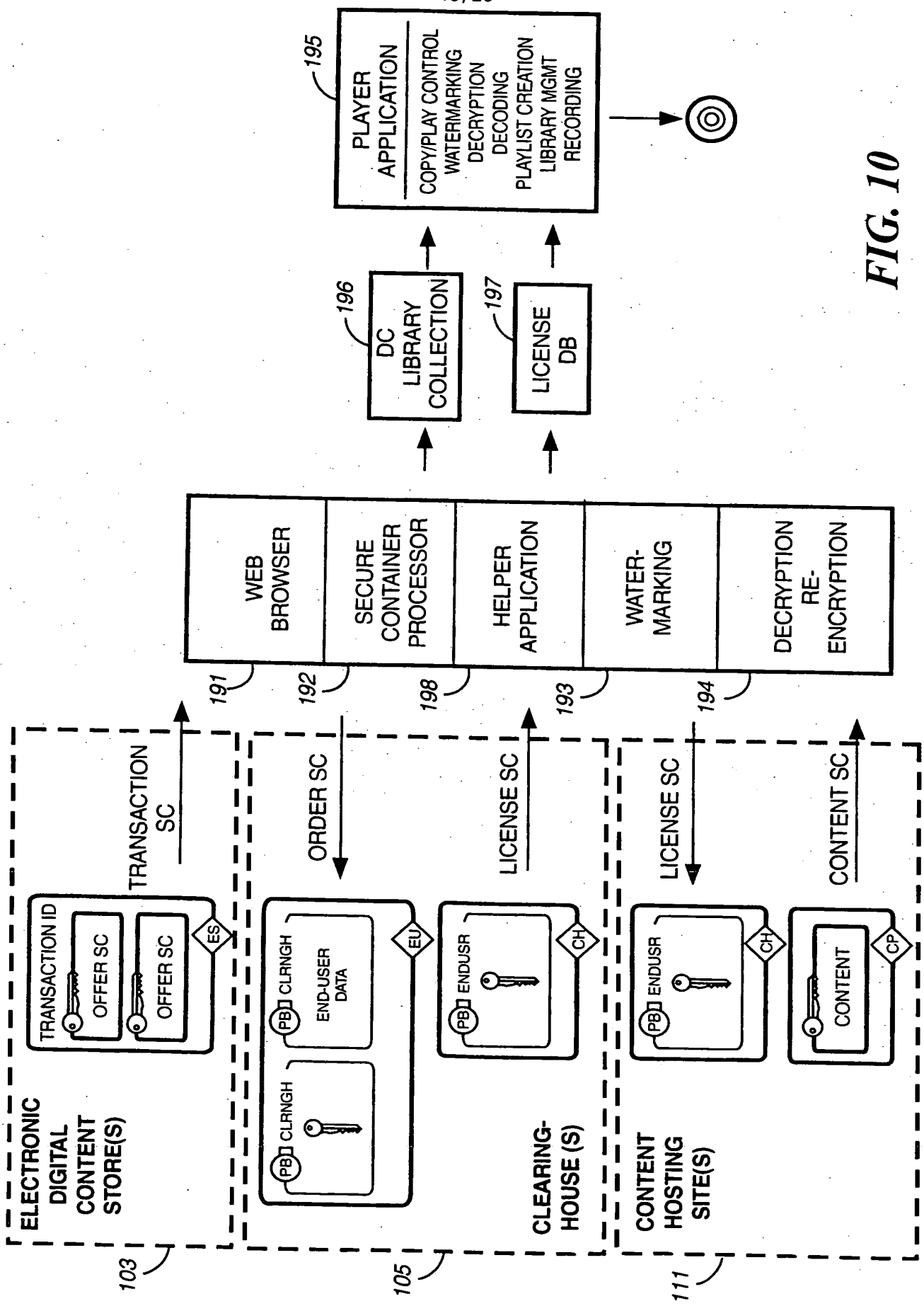


FIG. 10

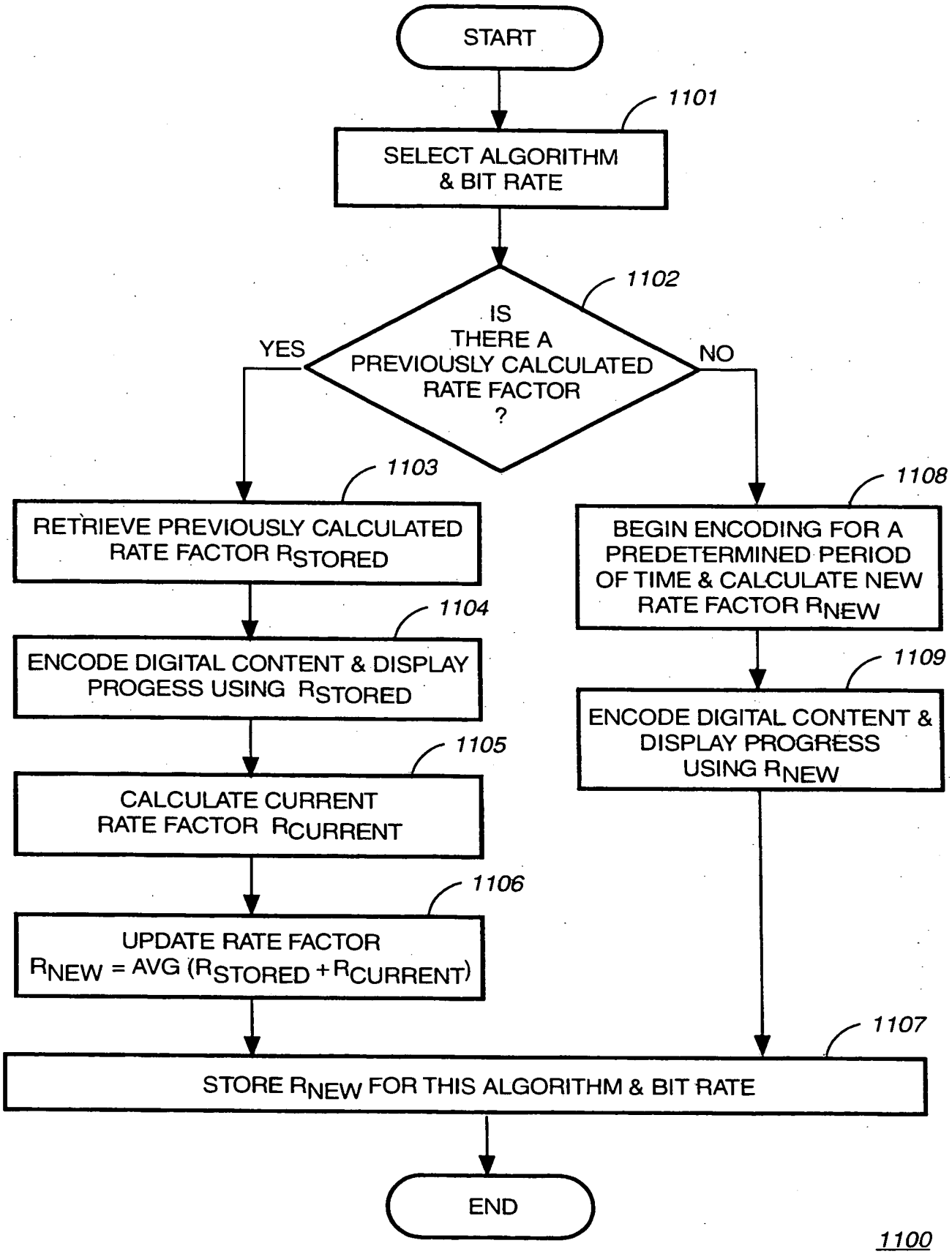

START
1101
SELECT ALGORITHM & BIT RATE
1102
IS THERE A PREVIOUSLY CALCULATED RATE FACTOR ?
YES
NO
1103
RETRIEVE PREVIOUSLY CALCULATED RATE FACTOR RSTORED
1104
ENCODE DIGITAL CONTENT & DISPLAY PROGESS USING RSTORED
1105
CALCULATE CURRENT RATE FACTOR RCURRENT
1106
UPDATE RATE FACTOR RNEW = AVG (RSTORED + RCURRENT)
1108
BEGIN ENCODING FOR A PREDETERMINED PERIOD OF TIME & CALCULATE NEW RATE FACTOR RNEW
1109
ENCODE DIGITAL CONTENT & DISPLAY PROGRESS USING RNEW
1107
STORE RNEW FOR THIS ALGORITHM & BIT RATE
END
1100

FIG. 11

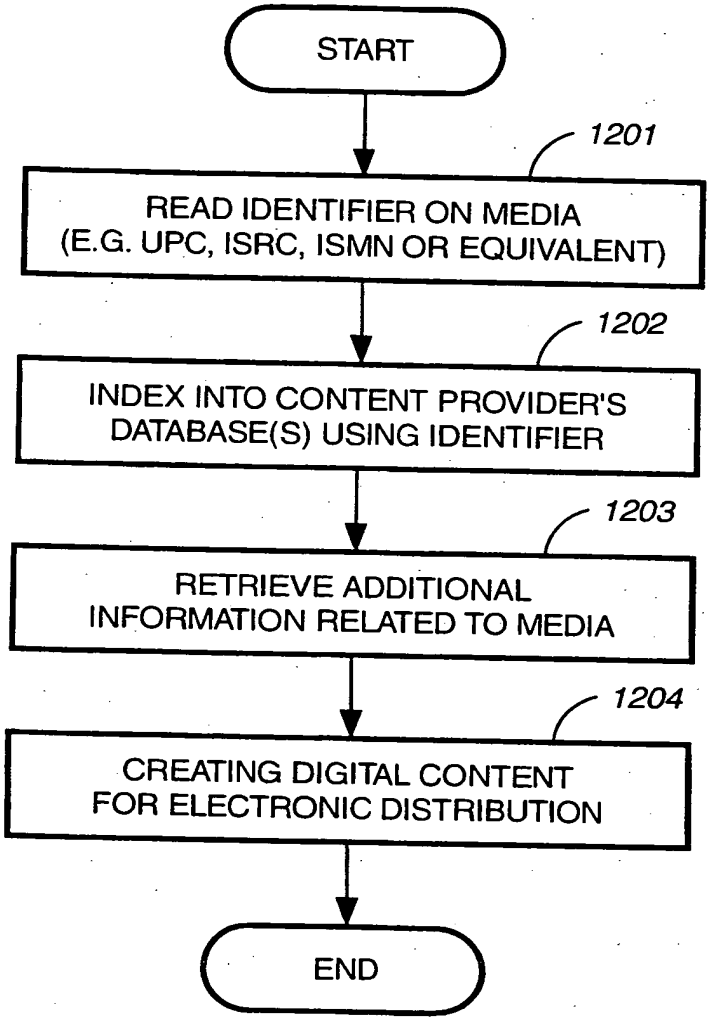

START
1201
READ IDENTIFIER ON MEDIA
(E.G. UPC, ISRC, ISMN OR EQUIVALENT)
1202
INDEX INTO CONTENT PROVIDER'S
DATABASE(S) USING IDENTIFIER
1203
RETRIEVE ADDITIONAL
INFORMATION RELATED TO MEDIA
1204
CREATING DIGITAL CONTENT
FOR ELECTRONIC DISTRIBUTION
END

FIG. 12

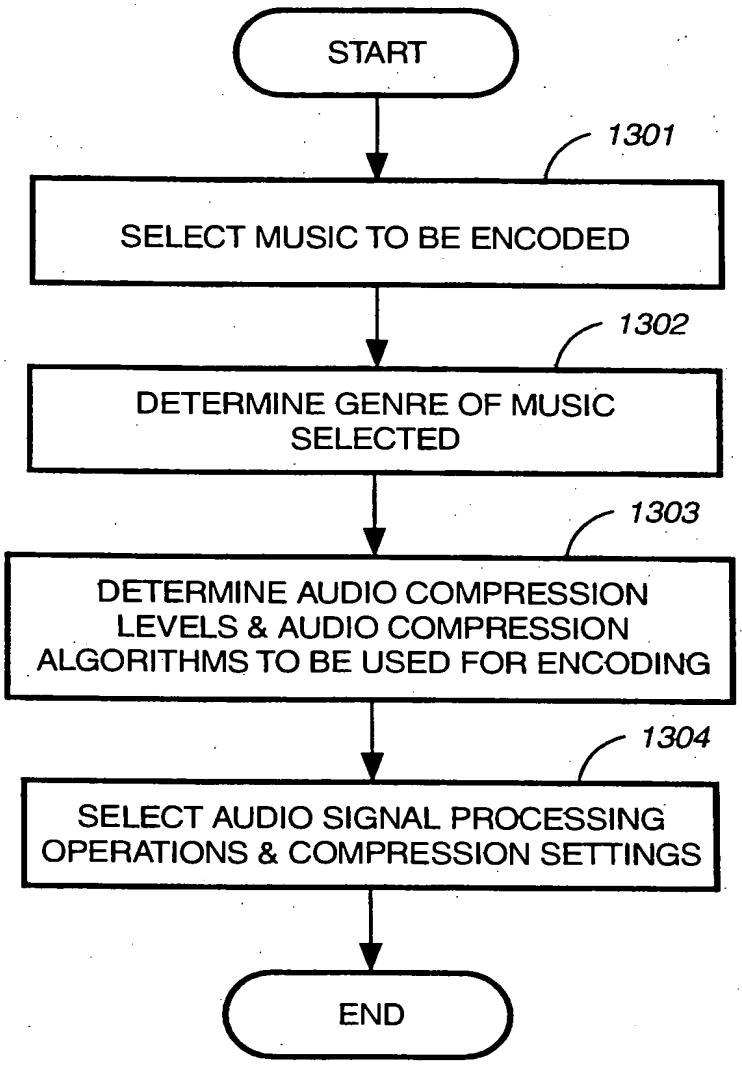

START
1301
SELECT MUSIC TO BE ENCODED
1302
DETERMINE GENRE OF MUSIC SELECTED
1303
DETERMINE AUDIO COMPRESSION LEVELS & AUDIO COMPRESSION ALGORITHMS TO BE USED FOR ENCODING
1304
SELECT AUDIO SIGNAL PROCESSING OPERATIONS & COMPRESSION SETTINGS
END

FIG. 13

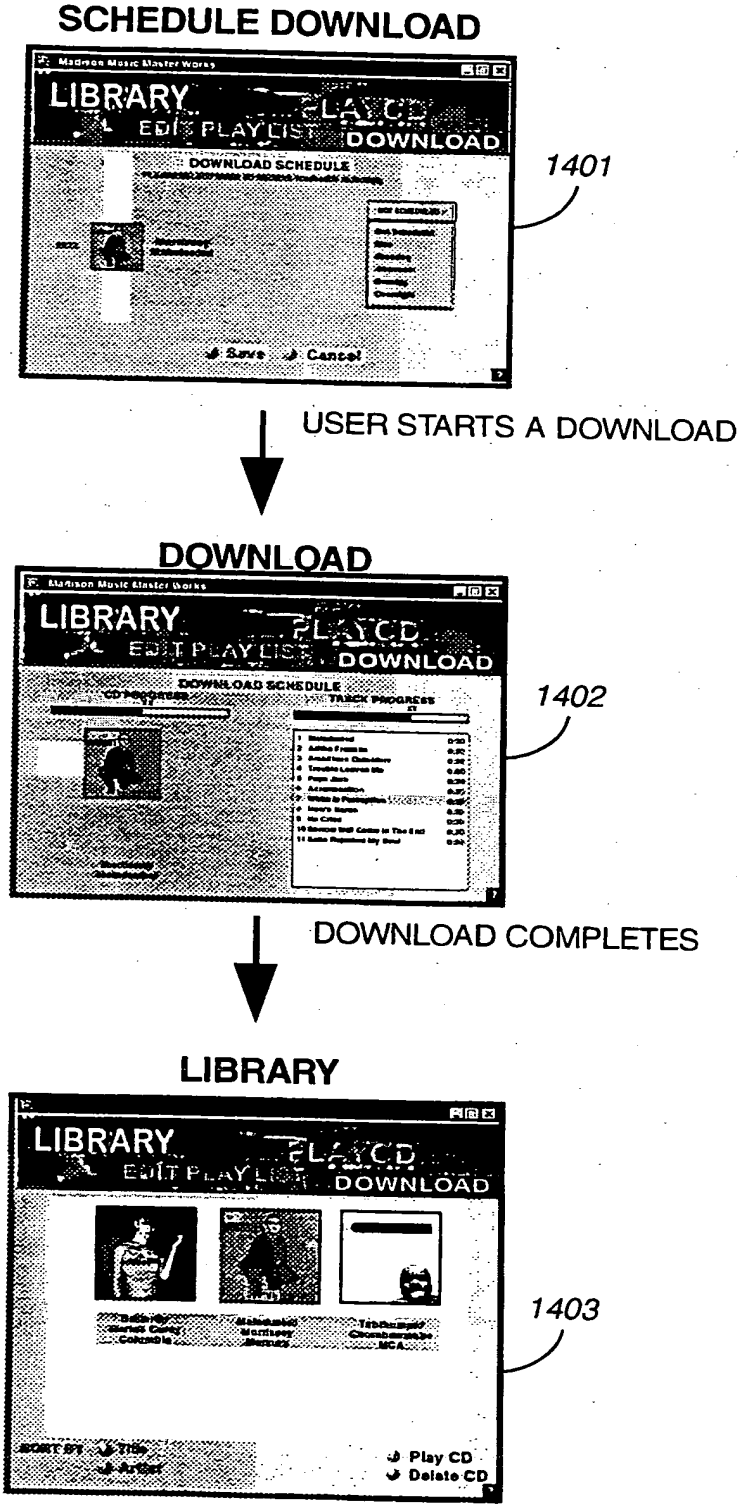

SCHEDULE DOWNLOAD
LIBRARY
PLAY CD
EDIT PLAYLIST
DOWNLOAD
DOWNLOAD SCHEDULE
Save
Cancel
1401
USER STARTS A DOWNLOAD
DOWNLOAD
LIBRARY
PLAY CD
EDIT PLAYLIST
DOWNLOAD
DOWNLOAD SCHEDULE
TRACK PROGRESS
1402
DOWNLOAD COMPLETES
LIBRARY
LIBRARY
PLAY CD
EDIT PLAYLIST
DOWNLOAD
SORT BY
Title
Artist
Play CD
Delete CD
1403

FIG. 14

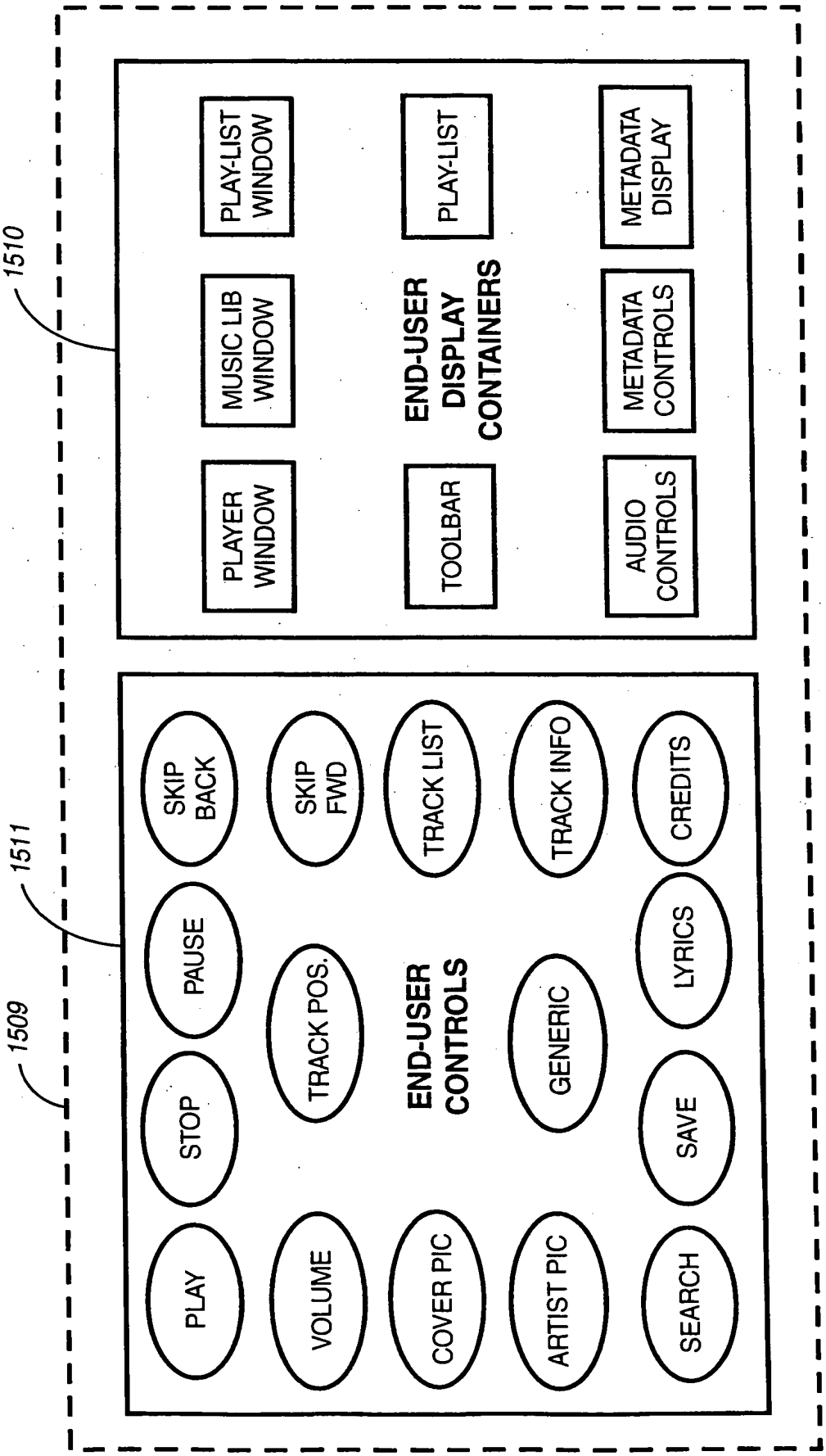

1509
1510
1511
END-USER CONTROLS
PLAY
STOP
PAUSE
VOLUME
TRACK POS.
SKIP BACK
SKIP FWD
COVER PIC
ARTIST PIC
SEARCH
SAVE
GENERIC
TRACK LIST
TRACK INFO
LYRICS
CREDITS
END-USER DISPLAY CONTAINERS
PLAYER WINDOW
MUSIC LIB WINDOW
PLAY-LIST WINDOW
TOOLBAR
PLAY-LIST
AUDIO CONTROLS
METADATA CONTROLS
METADATA DISPLAY

FIG. 15A

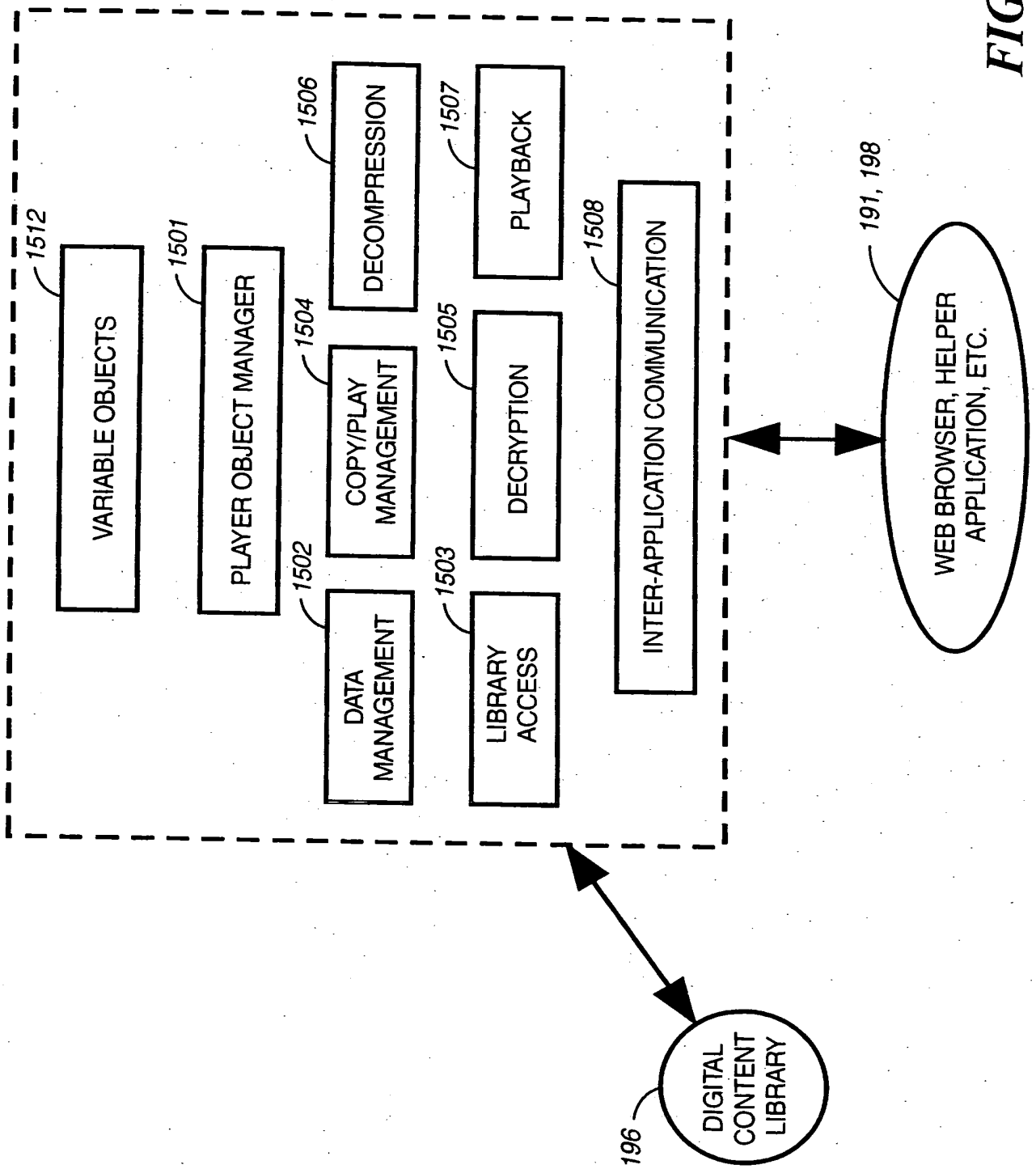

VARIABLE OBJECTS
1512
PLAYER OBJECT MANAGER
1501
DATA MANAGEMENT
1502
COPY/PLAY MANAGEMENT
1504
LIBRARY ACCESS
1503
DECRYPTION
1505
DECOMPRESSION
1506
PLAYBACK
1507
INTER-APPLICATION COMMUNICATION
1508
WEB BROWSER, HELPER APPLICATION, ETC.
191, 198
DIGITAL CONTENT LIBRARY
196

FIG. 15B

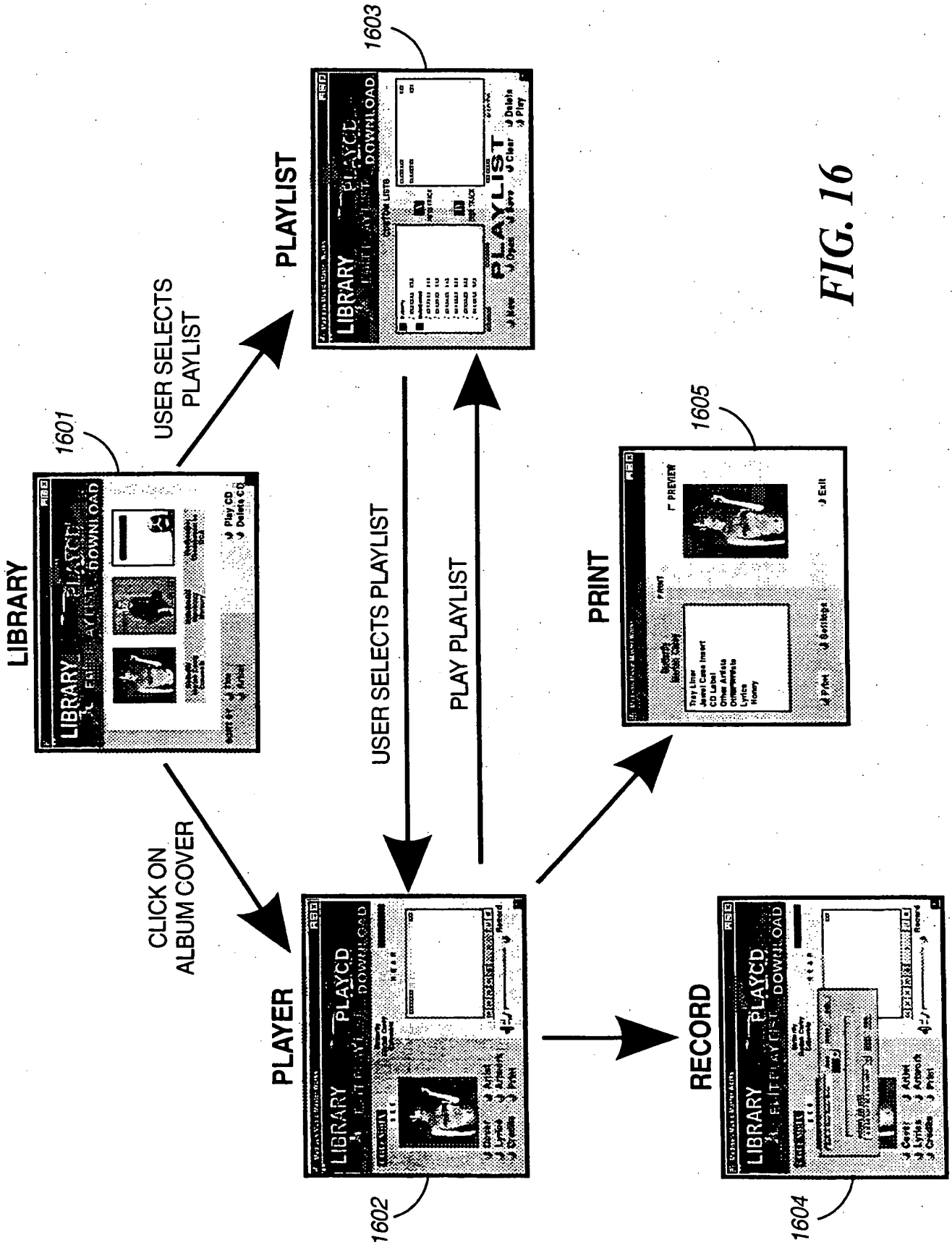

LIBRARY
1601
USER SELECTS PLAYLIST
PLAYLIST
1603
USER SELECTS PLAYLIST
PLAY PLAYLIST
PRINT
1605
CLICK ON ALBUM COVER
PLAYER
1602
RECORD
1604

FIG. 16

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7:
G06F 1/00, H04L 29/06, G06F 17/60

A3

(11) International Publication Number: **WO 00/08909**

(43) International Publication Date: 24 February 2000 (24.02.00)

(21) International Application Number: PCT/US99/18383

(22) International Filing Date: 12 August 1999 (12.08.99)

(30) Priority Data:

09/133,519	13 August 1998 (13.08.98)	US
09/177,096	22 October 1998 (22.10.98)	US

(71) Applicant *(for all designated States except US)*: INTERNATIONAL BUSINESS MACHINES CORPORATION [US/US]; New Orchard Road, Armonk, NY 10504 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: DORAK, John, Jr. [US/US]; 22238 S.E. 62nd Avenue, Boca Raton, FL 33428 (US). DOWNS, Edgar [US/US]; 2740 N.E. 58th Street, Fort Lauderdale, FL 33308 (US). GRUSE, George, Gregory [US/US]; 4310 N.E. 24th Avenue, Lighthouse Point, FL 33064 (US). HURTADO, Marco [US/US]; 4720 N.W. 28th Avenue, Boca Raton, FL 22434 (US). LEHMAN, Christopher [US/US]; 2663 Hampton Circle S., Delray Beach, FL 33308 (US). LOTSPIECH, Jeffrey [US/US]; 992 Foothill Drive, San Jose, CA 95123 (US). MEDINA, Cesar [US/US]; 4017 N.W. 24th Terrace, Boca Raton, FL 33431 (US). MILSTED, Kenneth [US/US]; 9927 Majestic Way, Boynton Beach, FL 33437-3303 (US).

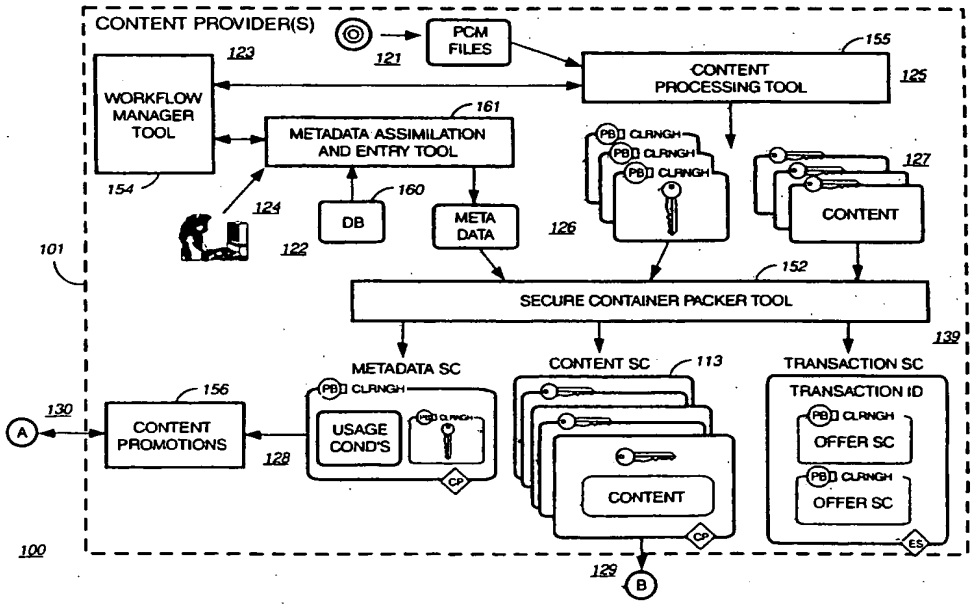
(74) Agent: SOUCAR, Stephen; IBM Corporation, Intellectual Property Law, Building 1, Mail Drop 1140, Route 100, P.O. Box 100, Somers, NY 10589 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, 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, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(88) Date of publication of the international search report:
16 November 2000 (16.11.00)

(54) Title: SYSTEM FOR TRACKING END-USER ELECTRONIC CONTENT USAGE



(57) Abstract

A system for tracking usage of digital content on user devices. Electronic stores coupled to a network sell licenses to play digital content data to users. A system for securely providing data, the system being capable of receiving both data encrypted with a first encryption key and a encrypted first encryption key.

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		

INTERNATIONAL SEARCH REPORT

International Application No
P /US 99/18383

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/00 H04L29/06 G06F17/60

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 7 G06F H04L

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.
A	WO 97 43717 A (CORP FOR NATIONAL RESEARCH INI) 20 November 1997 (1997-11-20) page 32, line 15 - line 24 ---	1-9
A	WO 98 13970 A (WALLENSTEIN & WAGNER LTD) 2 April 1998 (1998-04-02) page 6, line 12 -page 8, line 9 ---	1-9
A	US 5 237 157 A (KAPLAN JOSHUA D) 17 August 1993 (1993-08-17) column 3, line 10 - line 28 column 3, line 47 - line 52 column 5, line 6 - line 21 --- -/--	10-17

[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

Date of the actual completion of the international search	Date of mailing of the international search report
1 September 2000	06.09.00

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	Siebel, C

2

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No
P /US 99/18383

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	HOUSLEY R ET AL: "METERING: A PRE-PAY TECHNIQUE" PROCEEDINGS OF SPIE,US,BELLINGHAM, SPIE, vol. 3022, 13 February 1997 (1997-02-13), pages 527-531, XP000742405 ISBN: 0-8194-2433-1 page 527, line 1 - line 6 ---	18-26
A	US 5 649 013 A (ROBERTS JON L ET AL) 15 July 1997 (1997-07-15) column 2, line 41 - line 58 -----	18-26

2

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 99/18383

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☒ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☐ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

1. Claims: 1-9

A clearinghouse and a system for securely providing data

2. Claims: 10-17

A system for managing content data and an electronic content management system

3. Claims: 18-26

A digital content data player, a system for tracking of digital content and a computer readable medium containing programm instructions for tracking usage of digital content data

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
P· 'US 99/18383

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9743717 A	20-11-1997	AU 3000897 A	05-12-1997
WO 9813970 A	02-04-1998	AU 4599997 A	17-04-1998
US 5237157 A	17-08-1993	US 5963916 A	05-10-1999
US 5649013 A	15-07-1997	NONE	

EP 25664 (5)

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

(11) EP 0 715 244 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
05.06.1996 Bulletin 1996/23

(51) Int Cl.[6]: **G06F 1/00**

(21) Application number: **95308417.5**

(22) Date of filing: **23.11.1995**

(84) Designated Contracting States:
DE FR GB

(30) Priority: **23.11.1994 US 334041**

(71) Applicant: **XEROX CORPORATION**
Rochester New York 14644 (US)

(72) Inventor: **Stefik, Mark J.**
Woodside, California 94062 (US)

(74) Representative: **Goode, Ian Roy**
Rank Xerox Ltd
Patent Department
Parkway
Marlow Buckinghamshire SL7 1YL (GB)

(54) System for controlling the distribution and use of digital works utilizing a usage rights grammar

(57) A system for controlling use and distribution of digital works. The present invention allows the owner of a digital work to attach usage rights (1450) to their work. The usage rights define how the individual digital work may be used and distributed (1451). Instances of usage rights are defined using a flexible and extensible usage rights grammar. Conceptually, a right in the usage rights grammar is a label associated with a predetermined behavior and conditions to exercising the right. The behavior of a usage right is embodied in a predetermined set (1452) of usage transactions steps. The usage transaction steps further check all conditions (1453-1457) which must be satisfied before the right may be exercised. These usage transaction steps define a protocol for requesting the exercise of a right and the carrying out of a right.

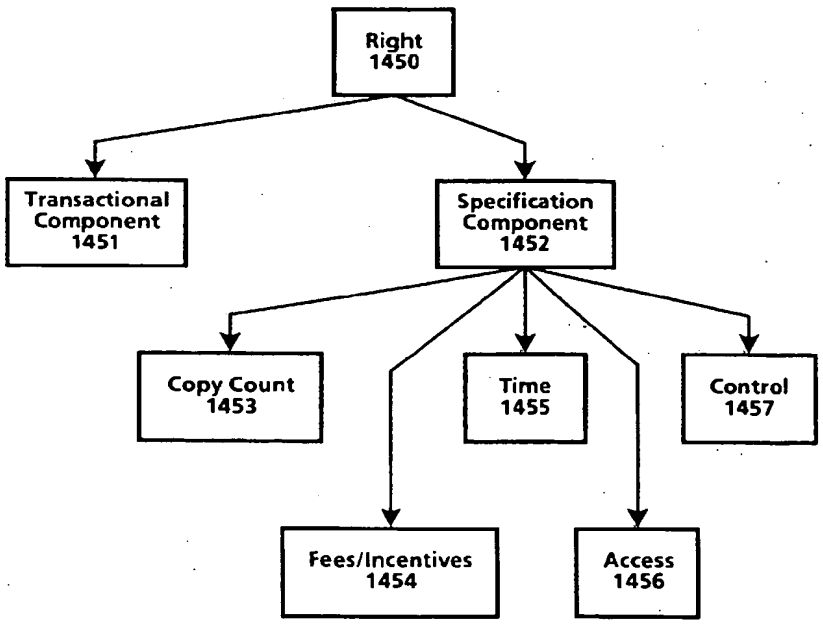


Fig.14

EP 0 715 244 A1

Printed by Jouve, 75001 PARIS (FR)

Description

The present invention relates to the field of distribution and usage rights enforcement for digitally encoded works.

A fundamental issue facing the publishing and information industries as they consider electronic publishing is how to prevent the unauthorized and unaccounted distribution or usage of electronically published materials. Electronically published materials are typically distributed in a digital form and recreated on a computer based system having the capability to recreate the materials. Audio and video recordings, software, books and multimedia works are all being electronically published. Companies in these industries receive royalties for each accounted for delivery of the materials, e.g. the sale of an audio CD at a retail outlet. Any unaccounted distribution of a work results in an unpaid royalty (e.g. copying the audio recording CD to another digital medium.)

The ease in which electronically published works can be "perfectly" reproduced and distributed is a major concern. The transmission of digital works over networks is commonplace. One such widely used network is the Internet. The Internet is a widespread network facility by which computer users in many universities, corporations and government entities communicate and trade ideas and information. Computer bulletin boards found on the Internet and commercial networks such as CompuServ and Prodigy allow for the posting and retrieving of digital information. Information services such as Dialog and LEXIS/NEXIS provide databases of current information on a wide variety of topics. Another factor which will exacerbate the situation is the development and expansion of the National Information Infrastructure (the NII). It is anticipated that, as the NII grows, the transmission of digital works over networks will increase many times over. It would be desirable to utilize the NII for distribution of digital works without the fear of widespread unauthorized copying.

The most straightforward way to curb unaccounted distribution is to prevent unauthorized copying and transmission. For existing materials that are distributed in digital form, various safeguards are used. In the case of software, copy protection schemes which limit the number of copies that can be made or which corrupt the output when copying is detected have been employed. Another scheme causes software to become disabled after a predetermined period of time has lapsed. A technique used for workstation based software is to require that a special hardware device must be present on the workstation in order for the software to run, e.g., see US-A-4,932,054 entitled "Method and Apparatus for Protecting Computer Software Utilizing Coded Filter Network in Conjunction with an Active Coded Hardware Device." Such devices are provided with the software and are commonly referred to as dongles.

Yet another scheme is to distribute software, but which requires a "key" to enable its use. This is employed in distribution schemes where "demos" of the software are provided on a medium along with the entire product. The demos can be freely used, but in order to use the actual product, the key must be purchased. These schemes do not hinder copying of the software once the key is initially purchased.

It is an object of the present invention to provide an improved system and method for controlling the use and distribution of digital works.

The invention accordingly provides a system and method as claimed in the accompanying claims.

A system for controlling use and distribution of digital works is disclosed. A digital work is any written, aural, graphical or video based work that has been translated to or created in a digital form, and which can be recreated using suitable rendering means such as software programs. The present invention allows the owner of a digital work to attach usage rights to their work. The usage rights define how the digital work may be used and distributed. These usage rights become part of the digital work and are always honored.

Instances of usage rights are defined using a flexible and extensible usage rights grammar. Conceptually, a right in the usage rights grammar is a label associated with a predetermined behavior and conditions to exercising the right. For example, a COPY right denotes that a copy of the digital work may be made. A condition to exercising the right is that the requester must pass certain security criteria. Conditions may also be attached to limit the right itself. For example, a LOAN right may be defined so as to limit the duration of which a work may be LOANed.

In the present invention a usage right is comprised of a right code along with the various conditions for exercising the right. Such conditions include a copy-count condition for limiting the number of times a right can be concurrently exercised (e.g. limit the number of copies on loan to some predetermined number), a security class condition for insuring that a repository has an appropriate level of security, access conditions for specifying access tests that must be passed, a time specification for indicating time based constraints for exercising a right and a fee specification for indicating usage fees for the exercise of a right. A digital work may have different versions of a right attached thereto. A version of a right will have the same right code as other versions, but the conditions (and typically the fees) would be different.

Digital works and their attached usage rights are stored in repositories. Digital works are transmitted between repositories. Repositories interact to exchange digital works according to a predetermined set of usage transactions steps. The behavior of a usage right is embodied in a predetermined set of usage transactions steps. The usage transaction steps further check all conditions which must be satisfied before the right may be exercised. These usage transaction steps define a protocol used by the repositories for requesting the exercise of a right and the carrying out of a right.

A system and method in accordance with the invention will now be described, by way of example, with reference to the accompanying drawings, in which:-

Figure 1 is a flowchart illustrating a simple instantiation of the operation of the currently preferred embodiment of the present invention.

Figure 2 is a block diagram illustrating the various repository types and the repository transaction flow between them in the currently preferred embodiment of the present invention.

Figure 3 is a block diagram of a repository coupled with a credit server in the currently preferred embodiment of the present invention.

Figures 4a and 4b are examples of rendering systems as may be utilized in the currently preferred embodiment of the present invention.

Figure 5 illustrates a contents file layout for a digital work as may be utilized in the currently preferred embodiment of the present invention.

Figure 6 illustrates a contents file layout for an individual digital work of the digital work of Figure 5 as may be utilized in the currently preferred embodiment of the present invention.

Figure 7 illustrates the components of a description block of the currently preferred embodiment of the present invention.

Figure 8 illustrates a description tree for the contents file layout of the digital work illustrated in Figure 5.

Figure 9 illustrates a portion of a description tree corresponding to the individual digital work illustrated in Figure 6.

Figure 10 illustrates a layout for the rights portion of a description block as may be utilized in the currently preferred embodiment of the present invention.

Figure 11 is a description tree wherein certain d-blocks have PRINT usage rights and is used to illustrate "strict". and "lenient" rules for resolving usage rights conflicts.

Figure 12 is a block diagram of the hardware components of a repository as are utilized in the currently preferred embodiment of the present invention.

Figure 13 is a block diagram of the functional (logical) components of a repository as are utilized in the currently preferred embodiment of the present invention.

Figure 14 is diagram illustrating the basic components of a usage right in the currently preferred embodiment of the present invention.

Figure 15 lists the usage rights grammar of the currently preferred embodiment of the present invention.

Figure 16 is a flowchart illustrating the steps of certificate delivery, hotlist checking and performance testing as performed in a registration transaction as may be performed in the currently preferred embodiment of the present invention.

Figure 17 is a flowchart illustrating the steps of session information exchange and clock synchronization as may be performed in the currently preferred embodiment of the present invention, after each repository in the registration transaction has successfully completed the steps described in Figure 16.

Figure 18 is a flowchart illustrating the basic flow for a usage transaction, including the common opening and closing step, as may be performed in the currently preferred embodiment of the present invention.

Figure 19 is a state diagram of server and client repositories in accordance with a transport protocol followed when moving a digital work from the server to the client repositories, as may be performed in the currently preferred embodiment of the present invention.

OVERVIEW

A system for controlling use and distribution of digital works is disclosed. The present invention is directed to supporting commercial transactions involving digital works.

Herein the terms "digital work", "work" and "content" refer to any work that has been reduced to a digital representation. This would include any audio, video, text, or multimedia work and any accompanying interpreter (e.g. software) that may be required for recreating the work. The term composite work refers to a digital work comprised of a collection of other digital works. The term "usage rights" or "rights" is a term which refers to rights granted to a recipient of a digital work. Generally, these rights define how a digital work can be used and if it can be further distributed. Each usage right may have one or more specified conditions which must be satisfied before the right may be exercised.

Figure 1 is a high level flowchart omitting various details but which demonstrates the basic operation of the present invention. Referring to Figure 1, a creator creates a digital work, step 101. The creator will then determine appropriate usage rights and fees, attach them to the digital work, and store them in Repository 1, step 102. The determination of appropriate usage rights and fees will depend on various economic factors. The digital work remains securely in Repository 1 until a request for access is received. The request for access begins with a session initiation by another repository. Here a Repository 2 initiates a session with Repository 1, step 103. As will be described in greater detail below, this session initiation includes steps which helps to insure that the respective repositories are trustworthy. As-

suming that a session can be established, Repository 2 may then request access to the Digital Work for a stated purpose, step 104. The purpose may be, for example, to print the digital work or to obtain a copy of the digital work. The purpose will correspond to a specific usage right. In any event, Repository 1 checks the usage rights associated with the digital work to determine if the access to the digital work may be granted, step 105. The check of the usage rights essentially involves a determination of whether a right associated with the access request has been attached to the digital work and if all conditions associated with the right are satisfied. If the access is denied, repository 1 terminates the session with an error message, step 106. If access is granted, repository 1 transmits the digital work to repository 2, step 107. Once the digital work has been transmitted to repository 2, repository 1 and 2 each generate billing information for the access which is transmitted to a credit server, step 108. Such double billing reporting is done to insure against attempts to circumvent the billing process.

Figure 2 illustrates the basic interactions between repository types in the present invention. As will become apparent from Figure 2, the various repository types will serve different functions. It is fundamental that repositories will share a core set of functionality which will enable secure and trusted communications. Referring to Figure 2, a repository 201 represents the general instance of a repository. The repository 201 has two modes of operation; a server mode and a requester mode. When in the server mode, the repository will be receiving and processing access requests to digital works. When in the requester mode, the repository will be initiating requests to access digital works. Repository 201 is general in the sense that its primary purpose is as an exchange medium for digital works. During the course of operation, the repository 201 may communicate with a plurality of other repositories, namely authorization repository 202, rendering repository 203 and master repository 204. Communication between repositories occurs utilizing a repository transaction protocol 205.

Communication with an authorization repository 202 may occur when a digital work being accessed has a condition requiring an authorization. Conceptually, an authorization is a digital certificate such that possession of the certificate is required to gain access to the digital work. An authorization is itself a digital work that can be moved between repositories and subjected to fees and usage rights conditions. An authorization may be required by both repositories involved in an access to a digital work.

Communication with a rendering repository 203 occurs in connection with the rendering of a digital work. As will be described in greater detail below, a rendering repository is coupled with a rendering device (e.g. a printer device) to comprise a rendering system.

Communication with a master repository 205 occurs in connection with obtaining an identification certificate. Identification certificates are the means by which a repository is identified as "trustworthy". The use of identification certificates is described below with respect to the registration transaction.

Figure 3 illustrates the repository 201 coupled to a credit server 301. The credit server 301 is a device which accumulates billing information for the repository 201. The credit server 301 communicates with repository 201 via billing transactions 302 to record billing transactions. Billing transactions are reported to a billing clearinghouse 303 by the credit server 301 on a periodic basis. The credit server 301 communicates to the billing clearinghouse 303 via clearinghouse transactions 304. The clearinghouse transactions 304 enable a secure and encrypted transmission of information to the billing clearinghouse 303.

RENDERING SYSTEMS

A rendering system is generally defined as a system comprising a repository and a rendering device which can render a digital work into its desired form. Examples of a rendering system may be a computer system, a digital audio system, or a printer. A rendering system has the same security features as a repository. The coupling of a rendering repository with the rendering device may occur in a manner suitable for the type of rendering device.

Figure 4a illustrates a printer as an example of a rendering system. Referring to Figure 4, printer system 401 has contained therein a printer repository 402 and a print device 403. It should be noted that the the dashed line defining printer system 401 defines a secure system boundary. Communications within the boundary are assumed to be secure. Depending on the security level, the boundary also represents a barrier intended to provide physical integrity. The printer repository 402 is an instantiation of the rendering repository 205 of Figure 2. The printer repository 402 will in some instances contain an ephemeral copy of a digital work which remains until it is printed out by the print engine 403. In other instances, the printer repository 402 may contain digital works such as fonts, which will remain and can be billed based on use. This design assures that all communication lines between printers and printing devices are encrypted, unless they are within a physically secure boundary. This design feature eliminates a potential "fault" point through which the digital work could be improperly obtained. The printer device 403 represents the printer components used to create the printed output.

Also illustrated in Figure 4a is the repository 404. The repository 404 is coupled to the printer repository 402. The repository 404 represents an external repository which contains digital works.

Figure 4b is an example of a computer system as a rendering system. A computer system may constitute a "multi-

function" device since it may execute digital works (e.g. software programs) and display digital works (e.g. a digitized photograph). Logically, each rendering device can be viewed as having its own repository, although only one physical repository is needed. Referring to Figure 4b, a computer system 410 has contained therein a display/execution repository 411. The display/execution repository 411 is coupled to display device, 412 and execution device 413. The dashed box surrounding the computer system 410 represents a security boundary within which communications are assumed to be secure. The display/execution repository 411 is further coupled to a credit server 414 to report any fees to be billed for access to a digital work and a repository 415 for accessing digital works stored therein.

STRUCTURE OF DIGITAL WORKS

Usage rights are attached directly to digital works. Thus, it is important to understand the structure of a digital work. The structure of a digital work, in particular composite digital works, may be naturally organized into an acyclic structure such as a hierarchy. For example, a magazine has various articles and photographs which may have been created and are owned by different persons. Each of the articles and photographs may represent a node in a hierarchical structure. Consequently, controls, i.e. usage rights, may be placed on each node by the creator. By enabling control and fee billing to be associated with each node, a creator of a work can be assured that the rights and fees are not circumvented.

In the currently preferred embodiment, the file information for a digital work is divided into two files: a "contents" file and a "description tree" file. From the perspective of a repository, the "contents" file is a stream of addressable bytes whose format depends completely on the interpreter used to play, display or print the digital work. The description tree file makes it possible to examine the rights and fees for a work without reference to the content of the digital work. It should be noted that the term description tree as used herein refers to any type of acyclic structure used to represent the relationship between the various components of a digital work.

Figure 5 illustrates the layout of a contents file. Referring to Figure 5, a digital work is comprised of story A 510, advertisement 511, story B 512 and story C 513. It is assumed that the digital work is stored starting at a relative address of 0. Each of the parts of the digital work are stored linearly so that story A 510 is stored at approximately addresses 0-30,000, advertisement 511 at addresses 30,001-40,000, story B 512 at addresses 40,001-60,000 and story C 513 at addresses 60,001-85K. The detail of story A 510 is illustrated in Figure 6. Referring to Figure 6, the story A 510 is further broken down to show text 614 stored at address 0-1500, soldier photo 615 at addresses 1501-10,000, graphics 616 stored at addresses 10,001-25,000 and sidebar 617 stored address 25,001-30,000. Note that the data in the contents file may be compressed (for saving storage) or encrypted (for security).

From Figures 5 and 6 it is readily observed that a digital work can be represented by its component parts as a hierarchy. The description tree for a digital work is comprised of a set of related descriptor blocks (d-blocks). The contents of each d-block is described with respect to Figure 7. Referring to Figure 7, a d-block 700 includes an identifier 701 which is a unique identifier for the work in the repository, a starting address 702 providing the start address of the first byte of the work, a length 703 giving the number of bytes in the work, a rights portion 704 wherein the granted usage rights and their status data are maintained, a parent pointer 705 for pointing to a parent d-block and child pointers 706 for pointing to the child d-blocks. In the currently preferred embodiment, the identifier 701 has two parts. The first part is a unique number assigned to the repository upon manufacture. The second part is a unique number assigned to the work upon creation. The rights portion 704 will contain a data structure, such as a look-up table, wherein the various information associated with a right is maintained. The information required by the respective usage rights is described in more detail below. D-blocks form a strict hierarchy. The top d-block of a work has no parent; all other d-blocks have one parent. The relationship of usage rights between parent and child d-blocks and how conflicts are resolved is described below.

A special type of d-block is a "shell" d-block. A shell d-block adds no new content beyond the content of its parts. A shell d-block is used to add rights and fee information, typically by distributors of digital works.

Figure 8 illustrates a description tree for the digital work of Figure 5. Referring to Figure 8, a top d-block 820 for the digital work points to the various stories and advertisements contained therein. Here, the top d-block 820 points to d-block 821 (representing story A 510), d-block 822 (representing the advertisement 511), d-block 823 (representing story B 512) and and d-block 824 (representing story C 513).

The portion of the description tree for Story A 510 is illustrated in Figure 9. D-block 925 represents text 614, d-block 926 represents photo 615, d-block 927 represents graphics 616 by and d-block 928 represents sidebar 617.

The rights portion 704 of a descriptor block is further illustrated in Figure 10. Figure 10 illustrates a structure which is repeated in the rights portion 704 for each right. Referring to Figure 10, each right will have a right code field 1050 and status information field 1052. The right code field 1050 will contain a unique code assigned to a right. The status information field 1052 will contain information relating to the state of a right and the digital work. Such information is indicated below in Table 1. The rights as stored in the rights portion 704 may typically be in numerical order based on the right code.

TABLE 1

DIGITAL WORK STATE INFORMATION		
Property	**Value**	**Use**
Copies-in-Use	Number	A counter of the number of copies of a work that are in use. Incremented when another copy is used; decremented when use is completed.
Loan-Period	Time-Units	Indicator of the maximum number of time-units that a document can be loaned out
Loaner-Copy	Boolean	Indicator that the current work is a loaned out copy of an authorized digital work.
Remaining-Time	Time-Units	Indicator of the remaining time of use on a metered document right.
Document-Descr	String	A string containing various identifying information about a document. The exact format of this is not specified, but it can include information such as a publisher name, author name, ISBN number, and so on.
Revenue-Owner	RO-Descr	A handle identifying a revenue owner for a digital work. This is used for reporting usage fees.
Publication-Date	Date-Descr	The date that the digital work was published.
History-list	History-Rec	A list of events recording the repostories and dates for operations that copy, transfer, backup, or restore a digital work.

The approach for representing digtal works by separating description data from content assumes that parts of a file are contiguous but takes no position on the actual representation of content. In particular, it is neutral to the question of whether content representation may take an object oriented approach. It would be natural to represent content as objects. In principle, it may be convenient to have content objects that include the billing structure and rights information that is represented in the d-blocks. Such variations in the design of the representation are possible and are viable alternatives but may introduce processing overhead, e.g. the interpretation of the objects.

Digital works are stored in a repository as part of a hierarchical file system. Folders (also termed directories and sub-directories) contain the digital works as well as other folders. Digital works and folders in a folder are ordered in alphabetical order. The digital works are typed to reflect how the files are used. Usage rights can be attached to folders so that the folder itself is treated as a digital work. Access to the folder would then be handled in the same fashion as any other digital work As will be described in more detail below, the contents of the folder are subject to their own rights. Moreover, file management rights may be attached to the folder which define how folder contents can be managed.

ATTACHING USAGE RIGHTS TO A DIGITAL WORK

It is fundamental to the present invention that the usage rights are treated as part of the digital work. As the digital work is distributed, the scope of the granted usage rights will remain the same or may be narrowed. For example, when a digital work is transferred from a document server to a repository, the usage rights may include the right to loan a copy for a predetermined period of time (called the original rights). When the repository loans out a copy of the digital work, the usage rights in the loaner copy (called the next set of rights) could be set to prohibit any further rights to loan out the copy. The basic idea is that one cannot grant more rights than they have.

The attachment of usage rights into a digital work may occur in a variety of ways. If the usage rights will be the same for an entire digital work, they could be attached when the digital work is processed for deposit in the digital work server. In the case of a digital work having different usage rights for the various components, this can be done as the digital work is being created. An authoring tool or digital work assembling tool could be utilized which provides for an automated process of attaching the usage rights.

As will be described below, when a digital work is copied, transferred or loaned, a "next set of rights" can be specified. The"next set of rights" will be attached to the digital work as it is transported.

Resolving Conflicting Rights

Because each part of a digital work may have its own usage rights, there will be instances where the rights of a "contained part" are different from its parent or container part. As a result, conflict rules must be established to dictate when and how a right may be exercised. The hierarchical structure of a digital work facilitates the enforcement of such

rules. A "strict" rule would be as follows: a right for a part in a digital work is sanctioned if and only if it is sanctioned for the part, for ancestor d-blocks containing the part and for all descendent d-blocks. By sanctioned, it is meant that (1) each of the respective parts must have the right, and (2) any conditions for exercising the right are satisfied.

It also possible to implement the present invention using a more lenient rule. In the more lenient rule, access to the part may be enabled to the descendent parts which have the right, but access is denied to the descendents which do not.

An example of applying both the strict rule and lenient is illustrated with reference to Figure 11. Referring to Figure 11, a root d-block 1101 has child d-blocks 1102-1105. In this case, root d-block represents a magazine, and each of the child d-blocks 1102-1105 represent articles in the magazine. Suppose that a request is made to PRINT the digital work represented by root d-block 1101 wherein the strict rule is followed. The rights for the root d-block 1101 and child d-blocks 1102-1105 are then examined. Root d-block 1101 and child d-blocks 1102 and 1105 have been granted PRINT rights. Child d-block 1103 has not been granted PRINT rights and child d-block 1104 has PRINT rights conditioned on payment of a usage fee.

Under the strict rule the PRINT right cannot be exercised because the child d-block does not have the PRINT right. Under the lenient rule, the result would be different. The digital works represented by child d-blocks 1102 and 1105 could be printed and the digital work represented by d-block 1104 could be printed so long as the usage fee is paid. Only the digital work represented by d-block 1103 could not be printed. This same result would be accomplished under the strict rule if the requests were directed to each of the individual digital works.

The present invention supports various combinations of allowing and disallowing access. Moreover, as will be described below, the usage rights grammar permits the owner of a digital work to specify if constraints may be imposed on the work by a container part. The manner in which digital works may be sanctioned because of usage rights conflicts would be implementation specific and would depend on the nature of the digital works.

REPOSITORIES

In the description of Figure 2, it was indicated that repositories come in various forms. All repositories provide a core set of services for the transmission of digital works. The manner in which digital works are exchanged is the basis for all transaction between repositories. The various repository types differ in the ultimate functions that they perform. Repositories may be devices themselves, or they may be incorporated into other systems. An example is the rendering repository 203 of Figure 2.

A repository will have associated with it a repository identifier. Typically, the repository identifier would be a unique number assigned to the repository at the time of manufacture. Each repository will also be classified as being in a particular security class. Certain communications and transactions may be conditioned on a repository being in a particular security class. The various security classes are described in greater detail below.

As a prerequisite to operation, a repository will require possession of an identification certificate. Identification certificates are encrypted to prevent forgery and are issued by a Master repository. A master repository plays the role of an authorization agent to enable repositories to receive digital works. Identification certificates must be updated on a periodic basis. Identification certificates are described in greater detail below with respect to the registration transaction.

A repository has both a hardware and functional embodiment. The functional embodiment is typically software executing on the hardware embodiment. Alternatively, the functional embodiment may be embedded in the hardware embodiment such as an Application Specific Integrated Circuit (ASIC) chip.

The hardware embodiment of a repository will be enclosed in a secure housing which if compromised, may cause the repository to be disabled. The basic components of the hardware embodiment of a repository are described with reference to Figure 12. Referring to Figure 12, a repository is comprised of a processing means 1200, storage system 1207, clock 1205 and external interface 1206. The processing means 1200 is comprised of a processor element 1201 and processor memory 1202. The processing means 1201 provides controller, repository transaction and usage rights transaction functions for the repository. Various functions in the operation of the repository such as decryption and/or decompression of digital works and transaction messages are also performed by the processing means 1200. The processor element 1201 may be a microprocessor or other suitable computing component. The processor memory 1202 would typically be further comprised of Read Only Memories (ROM) and Random Access Memories (RAM). Such memories would contain the software instructions utilized by the processor element 1201 in performing the functions of the repository.

The storage system 1207 is further comprised of descriptor storage 1203 and content storage 1204. The description tree storage 1203 will store the description tree for the digital work and the content storage will store the associated content. The description tree storage 1203 and content storage 1204 need not be of the same type of storage medium, nor are they necessarily on the same physical device. So for example, the descriptor storage 1203 may be stored on a solid state storage (for rapid retrieval of the description tree information), while the content storage 1204 may be on

a high capacity storage such as an optical disk.

The clock 1205 is used to time-stamp various time based conditions for usage rights or for metering usage fees which may be associated with the digital works. The clock 1205 will have an uninterruptable power supply, e.g. a battery, in order to maintain the integrity of the time-stamps. The external interface means 1206 provides for the signal connection to other repositories and to a credit server. The external interface means 1206 provides for the exchange of signals via such standard interfaces such as RS-232 or Personal Computer Manufacturers Card Industry Association (PCMCIA) standards, or FDDI. The external interface means 1206 may also provide network connectivity.

The functional embodiment of a repository is described with reference to Figure 13. Referring to Figure 13, the functional embodiment is comprised of an operating system 1301, core repository services 1302, usage transaction handlers 1303, repository specific functions, 1304 and a user interface 1305. The operating system 1301 is specific to the repository and would typically depend on the type of processor being used. The operating system 1301 would also provide the basic services for controlling and interfacing between the basic components of the repository.

The core repository services 1302 comprise a set of functions required by each and every repository. The core repository services 1302 include the session initiation transactions which are defined in greater detail below. This set of services also includes a generic ticket agent which is used to "punch" a digital ticket and a generic authorization server for processing authorization specifications. Digital tickets and authorizations are specific mechanisms for controlling the distribution and use of digital works and are described in more detail below. Note that coupled to the core repository services are a plurality of identification certificates 1306. The identification certificates 1306 are required to enable the use of the repository.

The usage transactions handlers 1303 comprise functionality for processing access requests to digital works and for billing fees based on access. The usage transactions supported will be different for each repository type. For example, it may not be necessary for some repositories to handle access requests for digital works.

The repository specific functionality 1304 comprises functionality that is unique to a repository. For example, the master repository has special functionality for issuing digital certificates and maintaining encryption keys. The repository specific functionality 1304 would include the user interface implementation for the repository.

Repository Security Classes

For some digital works the losses caused by any individual instance of unauthorized copying is insignificant and the chief economic concern lies in assuring the convenience of access and low-overhead billing. In such cases, simple and inexpensive handheld repositories and network-based workstations may be suitable repositories, even though the measures and guarantees of security are modest.

At the other extreme, some digital works such as a digital copy of a first run movie or a bearer bond or stock certificate would be of very high value so that it is prudent to employ caution and fairly elaborate security measures to ensure that they are not copied or forged. A repository suitable for holding such a digital work could have elaborate measures for ensuring physical integrity and for verifying authorization before use.

By arranging a universal protocol, all kinds of repositories can communicate with each other in principle. However, creators of some works will want to specify that their works will only be transferred to repositories whose level of security is high enough. For this reason, document repositories have a ranking system for classes and levels of security. The security classes in the currently preferred embodiment are described in Table 2.

TABLE 2

REPOSITORY SECURITY LEVELS	
Level	Description of Security
0	Open system. Document transmission is unencrypted. No digital certificate is required for identification. The security of the system depends mostly on user honesty, since only modest knowledge may be needed to circumvent the security measures. The repository has no provisions for preventing unauthorized programs from running and accessing or copying files. The system does not prevent the use of removable storage and does not encrypt stored files.
1	Minimal security. Like the previous class except that stored files are minimally encrypted, including ones on removable storage.
2	Basic security. Like the previous class except that special tools and knowledge are required to compromise the programming, the contents of the repository, or the state of the clock. All digital communications are encrypted. A digital certificate is provided as identification. Medium level encryption is used. Repository identification number is unforgeable.

TABLE 2 (continued)

REPOSITORY SECURITY LEVELS	
Level	Description of Security
3	General security. Like the previous class plus the requirement of special tools are needed to compromise the physical integrity of the repository and that modest encryption is used on all transmissions. Password protection is required to use the local user interface. The digital clock system cannot be reset without authorization. No works would be stored on removable storage. When executing works as programs, it runs them in their own address space and does not give them direct access to any file storage or other memory containing system code or works. They can access works only through the transmission transaction protocol.
4	Like the previous class except that high level encryption is used on all communications. Sensors are used to record attempts at physical and electronic tampering. After such tampering, the repository will not perform other transactions until it has reported such tampering to a designated server.
5	Like the previous class except that if the physical or digital attempts at tampering exceed some preset threshold that threaten the physical integrity of the repository or the integrity of digital and cryptographic barriers, then the repository will save only document description records of history but will erase or destroy any digital identifiers that could be misused if released to an unscrupulous party. It also modifies any certificates of authenticity to indicate that the physical system has been compromised. It also erases the contents of designated documents.
6	Like the previous class except that the repository will attempt wireless communication to report tampering and will employ noisy alarms.
10	This would correspond to a very high level of security. This server would maintain constant communications to remote security systems reporting transactions, sensor readings, and attempts to circumvent security.

The characterization of security levels described in Table 2 is not intended to be fixed. More important is the idea of having different security levels for different repositories. It is anticipated that new security classes and requirements will evolve according to social situations and changes in technology.

Repository User Interface

A user interface is broadly defined as the mechanism by which a user interacts with a repository in order to invoke transactions to gain access to a digital work, or exercise usage rights. As described above, a repository may be embodied in various forms. The user interface for a repository will differ depending on the particular embodiment. The user interface may be a graphical user interface having icons representing the digital works and the various transactions that may be performed. The user interface may be a generated dialog in which a user is prompted for information.

The user interface itself need not be part of the repository. As a repository may be embedded in some other device, the user interface may merely be a part of the device in which the repository is embedded. For example, the repository could be embedded in a "card" that is inserted into an available slot in a computer system. The user interface may be a combination of a display, keyboard, cursor control device and software executing on the computer system.

At a minimum, the user interface must permit a user to input information such as access requests and alpha numeric data and provide feedback as to transaction status. The user interface will then cause the repository to initiate the suitable transactions to service the request. Other facets of a particular user interface will depend on the functionality that a repository will provide.

CREDIT SERVERS

In the present invention, fees may be associated with the exercise of a right. The requirement for payment of fees is described with each version of a usage right in the usage rights language. The recording and reporting of such fees is performed by the credit server. One of the capabilities enabled by associating fees with rights is the possibility of supporting a wide range of charging models. The simplest model, used by conventional software, is that there is a single fee at the time of purchase, after which the purchaser obtains unlimited rights to use the work as often and for as long as he or she wants. Alternative models, include metered use and variable fees. A single work can have different fees for different uses. For example, viewing a photograph on a display could have different fees than making a hardcopy

or including it in a newly created work. A key to these alternative charging models is to have a low overhead means of establishing fees and accounting for credit on these transactions.

A credit server is a computational system that reliably authorizes and records these transactions so that fees are billed and paid. The credit server reports fees to a billing clearinghouse. The billing clearinghouse manages the financial transactions as they occur. As a result, bills may be generated and accounts reconciled. Preferably, the credit server would store the fee transactions and periodically communicate via a network with the billing clearinghouse for reconciliation. In such an embodiment, communications with the billing clearinghouse would be encrypted for integrity and security reasons. In another embodiment, the credit server acts as a "debit card" where transactions occur in "real-time" against a user account.

A credit server is comprised of memory, a processing means, a clock, and interface means for coupling to a repository and a financial institution (e.g. a modem). The credit server will also need to have security and authentication functionality. These elements are essentially the same elements as those of a repository. Thus, a single device can be both a repository and a credit server, provided that it has the appropriate processing elements for carrying out the corresponding functions and protocols. Typically, however, a credit server would be a card-sized system in the possession of the owner of the credit. The credit server is coupled to a repository and would interact via financial transactions as described below. Interactions with a financial institution may occur via protocols established by the financial institutions themselves.

In the currently preferred embodiment credit servers associated with both the server and the repository report the financial transaction to the billing clearinghouse. For example, when a digital work is copied by one repository to another for a fee, credit servers coupled to each of the repositories will report the transaction to the billing clearinghouse. This is desirable in that it insures that a transaction will be accounted for in the event of some break in the communication between a credit server and the billing clearinghouse. However, some implementations may embody only a single credit server reporting the transaction to minimize transaction processing at the risk of losing some transactions.

USAGE RIGHTS LANGUAGE

The present invention uses statements in a high level "usage rights language" to define rights associated with digital works and their parts. Usage rights statements are interpreted by repositories and are used to determine what transactions can be successfully carried out for a digital work and also to determine parameters for those transactions. For example, sentences in the language determine whether a given digital work can be copied, when and how it can be used, and what fees (if any) are to be charged for that use. Once the usage rights statements are generated, they are encoded in a suitable form for accessing during the processing of transactions.

Defining usage rights in terms of a language in combination with the hierarchical representation of a digital work enables the support of a wide variety of distribution and fee schemes. An example is the ability to attach multiple versions of a right to a work. So a creator may attach a PRINT right to make 5 copies for $10.00 and a PRINT right to make unlimited copies for $100.00. A purchaser may then choose which option best fits his needs. Another example is that rights and fees are additive. So in the case of a composite work, the rights and fees of each of the components works is used in determining the rights and fees for the work as a whole.

The basic contents of a right are illustrated in Figure 14. Referring to Figure 14, a right 1450 has a transactional component 1451 and a specifications component 1452. A right 1450 has a label (e.g. COPY or PRINT) which indicates the use or distribution privileges that are embodied by the right. The transactional component 1451 corresponds to a particular way in which a digital work may be used or distributed. The transactional component 1451 is typically embodied in software instructions in a repository which implement the use or distribution privileges for the right. The specifications components 1452 are used to specify conditions which must be satisfied prior to the right being exercised or to designate various transaction related parameters. In the currently preferred embodiment, these specifications include copy count 1453, Fees and Incentives 1454, Time 1455, Access and Security 1456 and Control 1457. Each of these specifications will be described in greater detail below with respect to the language grammar elements.

The usage rights language is based on the grammar described below. A grammar is a convenient means for defining valid sequence of symbols for a language. In describing the grammar the notation "[a|b|c]" is used to indicate distinct choices among alternatives. In this example, a sentence can have either an "a", "b" or "c". It must include exactly one of them. The braces {} are used to indicate optional items. Note that brackets, bars and braces are used to describe the language of usage rights sentences but do not appear in actual sentences in the language.

In contrast, parentheses are part of the usage rights language. Parentheses are used to group items together in lists. The notation (x^*) is used to indicate a variable length list, that is, a list containing one or more items of type x. The notation $(x)^*$ is used to indicate a variable number of lists containing x.

Keywords in the grammar are words followed by colons. Keywords are a common and very special case in the language. They are often used to indicate a single value, typically an identifier. In many cases, the keyword and the parameter are entirely optional. When a keyword is given, it often takes a single identifier as its value. In some cases,

the keyword takes a list of identifiers.

In the usage rights language, time is specified in an hours:minutes:seconds (or hh:mm:ss) representation. Time zone indicators, e.g. PDT for Pacific Daylight Time, may also be specified. Dates are represented as year/ month/day (or YYYY/MMM/DD). Note that these time and date representations may specify moments in time or units of time Money units are specified in terms of dollars.

Finally, in the usage rights language, various "things" will need to interact with each other. For example, an instance of a usage right may specify a bank account, a digital ticke etc.. Such things need to be identified and are specified herein using the suffix "-ID."

The Usage Rights Grammar is listed in its entirety in Figure 15 and is described below.

Grammar element 1501 **"Digital Work Rights:= (Rights*)"** define the digital wor rights as a set of rights. The set of rights attached to a digital work define how that digital work may be transferred, used, performed or played. A set of rights will attach to the entire digital work and in the case of compound digital works, each of the components of the digita work. The usage rights of components of a digital may be different.

Grammar element 1502 **"Right: = (Right-Code {Copy-Count} {Control-Spec} {Time-Spec} {Access-Spec} {Fee-Spec})"** enumerates the content of a right. Each usage right must specify a right code. Each right may also optionally specify conditions which must be satisfied before the right can be exercised. These conditions are copy count, control, time, access and fee conditions. In the currently preferred embodiment, for the optional elements, the following defaults apply: copy count equals 1, no time limit on the use of the right, no access tests or a security level required to use the right and no fee is required. These conditions will each be described in greater detail below.

It is important to note that a digital work may have multiple versions of a right, each having the same right code. The multiple version would provide alternative conditions and fees for accessing the digital work.

Grammar element 1503 **"Right-Code : = Render-Code I Transport-Code I File-Management-CodeI Derivative-Works- Code Configuration-Code"** distinguishes each of the specific rights into a particular right type (although each right is identified by distinct right codes). In this way, the grammar provides a catalog of possible rights that can be associated with parts of digital works. In the following, rights are divided into categories for convenience in describing them.

Grammar element 1504 **"Render-Code : = [Play:{Player:Player-ID}I Print: {Printer: Printer-ID}]"** lists a category of rights all involving the making of ephemeral, transitory, or non-digital copies of the digital work. After use the copies are erased.

- Play A process of rendering or performing a digital work on some processor. This includes such things as playing digital movies, playing digital music, playing a video game, running a computer program, or displaying a document on a display.
- Print To render the work in a medium that is not further protected by usage rights, such as printing on paper.

Grammar element 1505 **"Transport-Code : = [Copy I Transfer I Loan (Remaining-Rights: Next-Set-of-Rights)] {(Next-Copy-Rights: Next-Set of Rights)}"** lists a category of rights involving the making of persistent, usable copies of the digital work on other repositories. The optional Next-Copy-Rights determine the rights on the work after it is transported. If this is not specified, then the rights on the transported copy are the same as on the original. The optional Remaining-Rights specify the rights that remain with a digital work when it is loaned out. If this is not specified, then the default is that no rights can be exercised when it is loaned out.

- Copy Make a new copy of a work
- Transfer Moving a work from one repository to another.
- Loan Temporarily loaning a copy to another repository for a specified period of time.

Grammar element 1506 **"File-Management-Code: = Backup {Back-Up-Copy-Rights: Next-Set -of Rights}I Restore I Delete I Folder I Directory {Name:Hide-Local I Hide - Remote}{Parts:Hide-Local I Hide-Remote}"** lists a category of rights involving operations for file management, such as the making of backup copies to protect the copy owner against catastrophic equipment failure.

Many software licenses and also copyright law give a copy owner the right to make backup copies to protect against catastrophic failure of equipment. However, the making of uncontrolled backup copies is inherently at odds with the ability to control usage, since an uncontrolled backup copy can be kept and then restored even after the authorized copy was sold.

The File management rights enable the making and restoring of backup copies in a way that respects usage rights, honoring the requirements of both the copy owner and the rights grantor and revenue owner. Backup copies of work descriptions (including usage rights and fee data) can be sent under appropriate protocol and usage rights control to other document repositories of sufficiently high security. Further rights permit organization of digital works into folders

which themselves are treated as digital works and whose contents may be "hidden" from a party seeking to determine the contents of a repository.

- Backup To make a backup copy of a digital work as protection against media failure.
- Restore To restore a backup copy of a digital work.
- Delete To delete or erase a copy of a digital work.
- Folder To create and name folders, and to move files and folders between folders.
- Directory To hide a folder or its contents.

Grammar element 1507 **"Derivative-Works-Code: [Extract I Embed I Edit {Process: Process-ID}] {Next-Copy-Rights : Next-Set-of Rights}"** lists a category of rights involving the use of a digital work to create new works.

- Extract To remove a portion of a work, for the purposes of creating a new work.
- Embed To include a work in an existing work.
- Edit To alter a digital work by copying, selecting and modifying portions of an existing digital work.

Grammar element 1508 **"Configuration-Code: = Install I Uninstall"** lists a category of rights for installing and uninstalling software on a repository (typically a rendering repository.) This would typically occur for the installation of a new type of player within the rendering repository.

- Install: To install new software on a repository.
- Uninstall: To remove existing software from a repository.

Grammar element 1509 **"Next-Set-of-Rights: = {(Add: Set-Of-Rights)} {(Delete: Set-Of-Rights)} {(Replace: Set-Of-Rights)} {(Keep: Set-Of-Rights)}"** defines how rights are carried forward for a copy of a digital work. If the Next-Copy-Rights is not specified, the rights for the next copy are the same as those of the current copy. Otherwise, the set of rights for the next copy can be specified. Versions of rights after Add: are added to the current set of rights. Rights after Delete: are deleted from the current set of rights. If only right codes are listed after Delete:, then all versions of rights with those codes are deleted. Versions of rights after Replace: subsume all versions of rights of the same type in the current set of rights.

If Remaining-Rights is not specified, then there are no rights for the original after all Loan copies are loaned out. If Remaining-Rights is specified, then the Keep: token can be used to simplify the expression of what rights to keep behind. A list of right codes following keep means that all of the versions of those listed rights are kept in the remaining copy. This specification can be overridden by subsequent Delete: or Replace: specifications.

Copy Count Specification

For various transactions, it may be desirable to provide some limit as to the number of "copies" of the work which may be exercised simultaneously for the right. For example, it may be desirable to limit the number of copies of a digital work that may be loaned out at a time or viewed at a time.

Grammar element 1510 **"Copy-Count : = (Copies: positive-integer I 0 I unlimited)"** provides a condition which defines the number of "copies" of a work subject to the right . A copy count can be 0, a fixed number, or unlimited. The copy-count is associated with each right, as opposed to there being just a single copy-count for the digital work. The Copy-Count for a right is decremented each time that a right is exercised. When the Copy-Count equals zero, the right can no longer be exercised. If the Copy-Count is not specified, the default is one.

Control Specification

Rights and fees depend in general on rights granted by the creator as well as further restrictions imposed by later distributors. Control specifications deal with interactions between the creators and their distributors governing the imposition of further restrictions and fees. For example, a distributor of a digital work may not want an end consumer of a digital work to add fees or otherwise profit by commercially exploiting the purchased digital work.

Grammar element 1511 **"Control-Spec : = (Control: {Restrictable I Unrestrictable} {Unchargeable I Chargeable})"** provides a condition to specify the effect of usage rights and fees of parents on the exercise of the right. A digital work is restrictable if higher level d-blocks can impose further restrictions (time specifications and access specifications) on the right. It is unrestrictable if no further restrictions can be imposed. The default setting is restrictable. A right is unchargeable if no more fees can be imposed on the use of the right. It is chargeable if more fees can be imposed. The default is chargeable.

Time Specification

It is often desirable to assign a start date or specify some duration as to when a right may be exercised. Grammar element 1512 **"Time-Spec : = ({Fixed-Interval I Sliding-Interval I Meter-Time} Until: Expiration-Date)"** provides for specification of time conditions on the exercise of a right. Rights may be granted for a specified time. Different kinds of time specifications are appropriate for different kinds of rights. Some rights may be exercised during a fixed and predetermined duration. Some rights may be exercised for an interval that starts the first time that the right is invoked by some transaction. Some rights may be exercised or are charged according to some kind of metered time, which may be split into separate intervals. For example, a right to view a picture for an hour might be split into six ten minute viewings or four fifteen minute viewings or twenty three minute viewings.

The terms "time" and "date" are used synonymously to refer to a moment in time. There are several kinds of time specifications. Each specification represents some limitation on the times over which the usage right applies. The Expiration-Date specifies the moment at which the usage right ends. For example, if the Expiration-Date is "Jan 1, 1995," then the right ends at the first moment of 1995. If the Expiration-Date is specified as *forever*, then the rights are interpreted as continuing without end. If only an expiration date is given, then the right can be exercised as often as desired until the expiration date.

Grammar element 1513 **"Fixed-Interval : = From: Start-Time"** is used to define a predetermined interval that runs from the start time to the expiration date.

Grammar element 1514 **"Sliding-Interval : = Interval: Use-Duration"** is used to define an indeterminate (or "open") start time. It sets limits on a continuous period of time over which the contents are accessible. The period starts on the first access and ends after the duration has passed or the expiration date is reached, whichever comes first. For example, if the right gives 10 hours of continuous access, the use-duration would begin when the first access was made and end 10 hours later.

Grammar element 1515 **"Meter-Time: = Time-Remaining: Remaining-Use"** is used to define a "meter time," that is, a measure of the time that the right is actually exercised. It differs from the Sliding-Interval specification in that the time that the digital work is in use need not be continuous. For example, if the rights guarantee three days of access, those days could be spread out over a month. With this specification, the rights can be exercised until the meter time is exhausted or the expiration date is reached, whichever comes first.

Remaining-Use: = Time-Unit
Start-Time: = Time-Unit
Use-Duration: = Time-Unit
All of the time specifications include time-unit specifications in their ultimate instantiation.

Security Class and Authorization Specification

The present invention provides for various security mechanisms to be introduced into a distribution or use scheme. Grammar element 1516 **"Access-Spec : ({SC: Security-Class} {Authorization: Authorization-ID*} {Other-Authorization: Authorization-ID*} {Ticket: Ticket-ID})"** provides a means for restricting access and transmission. Access specifications can specify a required security class for a repository to exercise a right or a required authorization test that must be satisfied.

The keyword **"SC:"** is used to specify a minimum security level for the repositories involved in the access. If **"SC: "** is not specified, the lowest security level is acceptable.

The optional **"Authorization:"** keyword is used to specify required authorizations on the same repository as the work. The optional **"Other-Authorization:"** keyword is used to specify required authorizations on the other repository in the transaction.

The optional **"Ticket:"** keyword specifies the identity of a ticket required for the transaction. A transaction involving digital tickets must locate an appropriate digital ticket agent who can "punch" or otherwise validate the ticket before the transaction can proceed. Tickets are described in greater detail below.

In a transaction involving a repository and a document server, some usage rights may require that the repository have a particular authorization, that the server have some authorization, or that both repositories have (possibly different) authorizations. Authorizations themselves are digital works (hereinafter referred to as an authorization object) that can be moved between repositories in the same manner as other digital works. Their copying and transferring is subject to the same rights and fees as other digital works. A repository is said to have an authorization if that authorization object is contained within the repository.

In some cases, an authorization may be required from a source other than the document server and repository. An authorization object referenced by an Authorization-ID can contain digital address information to be used to set up a communications link between a repository and the authorization source. These are analogous to phone numbers.

For such access tests, the communication would need to be established and authorization obtained before the right could be exercised.

For one-time usage rights, a variant on this scheme is to have a digital ticket. A ticket is presented to a digital ticket agent, whose type is specified on the ticket. In the simplest case, a certified generic ticket agent, available on all repositories, is available to "punch" the ticket. In other cases, the ticket may contain addressing information for locating a "special" ticket agent. Once a ticket has been punched, it cannot be used again for the same kind of transaction (unless it is unpunched or refreshed in the manner described below.) Punching includes marking the ticket with a timestamp of the date and time it was used. Tickets are digital works and can be copied or transferred between repositories according to their usage rights.

In the currently preferred embodiment, a "punched" ticket becomes "unpunched" or "refreshed" when it is copied or extracted. The Copy and Extract operations save the date and time as a property of the digital ticket. When a ticket agent is given a ticket, it can simply check whether the digital copy was made after the last time that it was punched. Of course, the digital ticket must have the copy or extract usage rights attached thereto.

The capability to unpunch a ticket is inportant in the following cases:

- A digital work is circulated at low cost with a limitation that it can be used only once.
- A digital work is circulated with a ticket that can be used once to give discounts on purchases of other works.
- A digital work is circulated with a ticket (included in the purchase price and possibly embedded in the work) that can be used for a future upgrade.

In each of these cases, if a paid copy is made of the digital work (including the ticket) the new owner would expect to get a fresh (unpunched) ticket, whether the copy seller has used the work or not. In contrast, loaning a work or simply transferring it to another repository should not revitalize the ticket.

Usage Fees and Incentives Specification

The billing for use of a digital work is fundamental to a commercial distribution system. Grammar Element 1517 **"Fee-Spec: = {Scheduled-Discount} Regular-Fee-Spec I Scheduled-Fee-Spec I Markup-Spec"** provides a range of options for billing for the use of digital works.

A key feature of this approach is the development of low-overhead billing for transactions in potentially small amounts. Thus, it becomes feasible to collect fees of only a few cents each for thousands of transactions.

The grammar differentiates between uses where the charge is per use from those where it is metered by the time unit. Transactions can support fees that the user pays for using a digital work as well as incentives paid by the right grantor to users to induce them to use or distribute the digital work.

The optional scheduled discount refers to the rest of the fee specification--discounting it by a percentage over time. If it is not specified, then there is no scheduled discount. Regular fee specifications are constant over time. Scheduled fee specifications give a schedule of dates over which the fee specifications change. Markup specifications are used in d-blocks for adding a percentage to the fees already being charged.

Grammar Element 1518 **"Scheduled-Discount: = (Scheduled-Discount: (Time-Spec Percentage)*)"** A Scheduled-Discount is a essentially a scheduled modifier of any other fee specification for this version of the right of the digital work. (It does not refer to children or parent digital works or to other versions of rights.). It is a list of pairs of times and percentages. The most recent time in the list that has not yet passed at the time of the transaction is the one in effect. The percentage gives the discount percentage. For example, the number 10 refers to a 10% discount.

Grammar Element 1519 **"Regular-Fee-Spec : = ({Fee: I Incentive:} [Per-Use-Spec I Metered-Rate-Spec I Best-Price-Spec I Call-For-Price-Spec] {Min: Money-Unit Per: Time-Spec}{Max: Money-Unit Per: Time-Spec} To: Account-ID)"** provides for several kinds of fee specifications.

Fees are paid by the copy-owner/user to the revenue-owner if **Fee:** is specified. Incentives are paid by the revenue-owner to the user if **Incentive:** is specified. If the **Min:** specification is given, then there is a minimum fee to be charged per time-spec unit for its use. If the **Max:** specification is given, then there is a maximum fee to be charged per time-spec for its use. When **Fee:** is specified, **Account-ID** identifies the account to which the fee is to be paid. When **Incentive:** is specified, **Account-ID** identifies the account from which the fee is to be paid.

Grammar element 1520 **"Per-Use-Spec: = Per-Use: Money-unit"** defines a simple fee to be paid every time the right is exercised. regardless of how much time the transaction takes.

Grammar element 1521 **"Metered-Rate-Spec : = Metered: Money-Unit Per: Time-Spec"** defines a metered-rate fee paid according to how long the right is exercised. Thus, the time it takes to complete the transaction determines the fee.

Grammar element 1522 **"Best-Price-Spec : = Best-Price: Money-unit Max: Money-unit"** is used to specify a best-price that is determined when the account is settled. This specification is to accommodate special deals, rebates,

and pricing that depends on information that is not available to the repository. All fee specifications can be combined with tickets or authorizations that could indicate that the consumer is a wholesaler or that he is a preferred customer, or that the seller be authorized in some way. The amount of money in the **Max:** field is the maximum amount that the use will cost. This is the amount that is tentatively debited from the credit server. However, when the transaction is ultimately reconciled, any excess amount will be returned to the consumer in a separate transaction.

Grammar element 1523 **"Call-For-Price-Spec: = Call-For-Price "** is similar to a **"Best-Price-Spec"** in that it is intended to accommodate cases where prices are dynamic. A **Call-For-Price Spec** requires a communication with a dealer to determine the price. This option cannot be exercised if the repository cannot communicate with a dealer at the time that the right is exercised. It is based on a secure transaction whereby the dealer names a price to exercise the right and passes along a deal certificate which is referenced or included in the billing process.

Grammar element 1524 **"Scheduled-Fee-Spec: = (Schedule: (Time-Spec Regular-Fee-Spec)*)"** is used to provide a schedule of dates over which the fee specifications change. The fee specification with the most recent date not in the future is the one that is in effect. This is similar to but more general than the scheduled discount. It is more general, because it provides a means to vary the fee agreement for each time period.

Grammar element 1525 **"Markup-Spec: = Markup: percentage To: Account-ID"** is provided for adding a percentage to the fees already being charged. For example, a 5% markup means that a fee of 5% of cumulative fee so far will be allocated to the distributor. A markup specification can be applied to all of the other kinds of fee specifications. It is typically used in a shell provided by a distributor. It refers to fees associated with d-blocks that are parts of the current d-block. This might be a convenient specification for use in taxes, or in distributor overhead.

REPOSITORY TRANSACTIONS

When a user requests access to a digital work, the repository will initiate various transactions. The combination of transactions invoked will depend on the specifications assigned for a usage right. There are three basic types of transactions, Session Initiation Transactions, Financial Transactions and Usage Transactions. Generally, session initiation transactions are initiated first to establish a valid session. When a valid session is established, transactions corresponding to the various usage rights are invoked. Finally, request specific transactions are performed.

Transactions occur between two repositories (one acting as a server), between a repository and a document playback platform (e.g. for executing or viewing), between a repository and a credit server or between a repository and an authorization server. When transactions occur between more than one repository, it is assumed that there is a reliable communication channel between the repositories. For example, this could be a TCP/IP channel or any other commercially available channel that has built-in capabilities for detecting and correcting transmission errors. However, it is not assumed that the communication channel is secure. Provisions for security and privacy are part of the requirements for specifying and implementing repositories and thus form the need for various transactions.

Message Transmission

Transactions require that there be some communication between repositories. Communication between repositories occurs in units termed as messages. Because the communication line is assumed to be unsecure, all communications with repositories that are above the lowest security class are encrypted utilizing a public key encryption technique. Public key encryption is a well known technique in the encryption arts. The term key refers to a numeric code that is used with encryption and decryption algorithms. Keys come in pairs, where "writing keys" are used to encrypt data and "checking keys" are used to decrypt data. Both writing and checking keys may be public or private. Public keys are those that are distributed to others. Private keys are maintained in confidence.

Key management and security is instrumental in the success of a public key encryption system. In the currently preferred embodiment, one or more master repositories maintain the keys and create the identification certificates used by the repositories.

When a sending repository transmits a message to a receiving repository, the sending repository encrypts all of its data using the public writing key of the receiving repository. The sending repository includes its name, the name of the receiving repository, a session identifier such as a nonce (described below), and a message counter in each message.

In this way, the communication can only be read (to a high probability) by the receiving repository, which holds the private checking key for decryption. The auxiliary data is used to guard against various replay attacks to security. If messages ever arrive with the wrong counter or an old nonce, the repositories can assume that someone is interfering with communication and the transaction terminated.

The respective public keys for the repositories to be used for encryption are obtained in the registration transaction described below.

Session Initiation Transactions

A usage transaction is carried out in a session between repositories. For usage transactions involving more than one repository. or for financial transactions between a repository and a credit server. a registration transaction is performed. A second transaction termed a login transaction, may also be needed to initiate the session. The goal of the registration transaction is to establish a secure channel between two repositories who know each others identities. As it is assumed that the communication channel between the repositories is reliable but not secure, there is a risk that a non-repository may mimic the protocol in order to gain illegitimate access to a repository.

The registration transaction between two repositories is described with respect to Figures 16 and 17. The steps described are from the perspective of a "repository-1" registering its identity with a "repository-2". The registration must be symmetrical so the same set of steps will be repeated for repository-2 registering its identity with repository-1. Referring to Figure 16, repository-1 first generates an encrypted registration identifier, step 1601 and then generates a registration message, step 1602. A registration message is comprised of an identifier of a master repository, the identification certificate for the repository-1 and an encrypted random registration identifier. The identification certificate is encrypted by the master repository in its private key and attests to the fact that the repository (here repository-1) is a bona fide repository. The identification certificate also contains a public key for the repository. the repository security level and a timestamp (indicating a time after which the certificate is no longer valid.) The registration identifier is a number generated by the repository for this registration. The registration identifier is unique to the session and is encrypted in repository-1's private key. The registration identifier is used to improve security of authentication by detecting certain kinds of communications based attacks. Repository-1 then transmits the registration message to repository-2, step 1603.

Upon receiving the registration message, repository-2 determines if it has the needed public key for the master repository, step 1604. If repository-2 does not have the needed public key to decrypt the identification certificate, the registration transaction terminates in an error, step 1618.

Assuming that repository-2 has the proper public key the identification certificate is decrypted, step 1605. Repository-2 saves the encrypted registration identifier, step 1606, and extracts the repository identifier, step 1607. The extracted repository identifier is checked against a "hotlist" of compromised document repositories, step 1608. In the currently preferred embodiment, each repository will contain "hotlists" of compromised repositories. If the repository is on the "hotlist', the registration transaction terminates in an error per step 1618. Repositories can be removed from the hotlist when their certificates expire, so that the list does not need to grow without bound. Also, by keeping a short list of hotlist certificates that it has previously received, a repository can avoid the work of actually going through the list. These lists would be encrypted by a master repository. A minor variation on the approach to improve efficiency would have the repositories first exchange lists of names of hotlist certificates, ultimately exchanging only those lists that they had not previously received. The "hotlists" are maintained and distributed by Master repositories.

Note that rather than terminating in error, the transaction could request that another registration message be sent based on an identification certificate created by another master repository. This may be repeated until a satisfactory identification certificate is found, or it is determined that trust cannot be established.

Assuming that the repository is not on the hotlist, the repository identification needs to be verified. In other words, repository-2 needs to validate that the repository on the other end is really repository-1. This is termed performance testing and is performed in order to avoid invalid access to the repository via a counterfeit repository replaying a recording of a prior session initiation between repository-1 and repository-2. Performance testing is initiated by repository-2 generating a performance message, step 1609. The performance message consists of a nonce, the names of the respective repositories, the time and the registration identifier received from repository-1. A nonce is a generated message based on some random and variable information (e.g. the time or the temperature.) The nonce is used to check whether repository-1 can actually exhibit correct encrypting of a message using the private keys it claims to have, on a message that it has never seen before. The performance message is encrypted using the public key specified in the registration message of repository-1. The performance message is transmitted to repository-1, step 1610, where it is decrypted by repository-1 using its private key, step 1611. Repository-1 then checks to make sure that the names of the two repositories are correct, step 1612, that the time is accurate, step 1613 and that the registration identifier corresponds to the one it sent, step 1614. If any of these tests fails, the transaction is terminated per step 1616. Assuming that the tests are passed, repository-1 transmits the nonce to repository-2 in the clear, step 1615. Repository-2 then compares the received nonce to the original nonce, step 1617. If they are not identical, the registration transaction terminates in an error per step 1618. If they are the same, the registration transaction has successfully completed.

At this point, assuming that the transaction has not terminated, the repositories exchange messages containing session keys to be used in all communications during the session and synchronize their clocks. Figure 17 illustrates the session information exchange and clock synchronization steps (again from the perspective of repository-1.) Referring to Figure 17, repository-1 creates a session key pair, step 1701. A first key is kept private and is used by repository-1 to encrypt messages. The second key is a public key used by repository-2 to decrypt messages. The

second key is encrypted using the public key of repository-2, step 1702 and is sent to repository-2, step 1703. Upon receipt, repository-2 decrypts the second key, step 1704. The second key is used to decrypt messages in subsequent communications. When each repository has completed this step, they are both convinced that the other repository is bona fide and that they are communicating with the original. Each repository has given the other a key to be used in decrypting further communications during the session. Since that key is itself transmitted in the public key of the receiving repository only it will be able to decrypt the key which is used to decrypt subsequent messages.

After the session information is exchanged, the repositories must synchronize their clocks. Clock synchronization is used by the repositories to establish an agreed upon time base for the financial records of their mutual transactions. Referring back to Figure 17, repository-2 initiates clock synchronization by generating a time stamp exchange message, step 1705, and transmits it to repository- 1, step 1706. Upon receipt, repository-1 generates its own time stamp message, step 1707 and transmits it back to repository-2, step 1708. Repository-2 notes the current time, step 1709 and stores the time received from repository-1, step 1710. The current time is compared to the time received from repository-1, step 1711. The difference is then checked to see if it exceeds a predetermined tolerance (e.g. one minute), step 1712. If it does, repository-2 terminates the transaction as this may indicate tampering with the repository, step 1713. If not repository-2 computes an adjusted time delta, step 1714. The adjusted time delta is the difference between the clock time of repository-2 and the average of the times from repository-1 and repository-2.

To achieve greater accuracy, repository-2 can request the time again up to a fixed number of times (e.g. five times), repeat the clock synchronization steps, and average the results.

A second session initiation transaction is a Login transaction. The Login transaction is used to check the authenticity of a user requesting a transaction. A Login transaction is particularly prudent for the authorization of financial transactions that will be charged to a credit server. The Login transaction involves an interaction between the user at a user interface and the credit server associated with a repository. The information exchanged here is a login string supplied by the repository/credit server to identify itself to the user, and a Personal Identification Number (PIN) provided by the user to identify himself to the credit server. In the event that the user is accessing a credit server on a repository different from the one on which the user interface resides, exchange of the information would be encrypted using the public and private keys of the respective repositories.

Billing Transactions

Billing Transactions are concerned with monetary transactions with a credit server. Billing Transactions are carried out when all other conditions are satisfied and a usage fee is required for granting the request. For the most part, billing transactions are well understood in the state of the art. These transactions are between a repository and a credit server, or between a credit server and a billing clearinghouse. Briefly, the required transactions include the following:

- Registration and LOGIN transactions by which the repository and user establish their bona fides to a credit server. These transactions would be entirely internal in cases where the repository and credit server are implemented as a single system.
- Registration and LOGIN transactions, by which a credit server establishes its bona fides to a billing clearinghouse.
- An Assign-fee transaction to assign a charge. The information in this transaction would include a transaction identifier, the identities of the repositories in the transaction, and a list of charges from the parts of the digital work. If there has been any unusual event in the transaction such as an interruption of communications, that information is included as well.
- A Begin-charges transaction to assign a charge. This transaction is much the same as an assign-fee transaction except that it is used for metered use. It includes the same information as the assign-fee transaction as well as the usage fee information. The credit-server is then responsible for running a clock.
- An End-charges transaction to end a charge for metered use. (In a variation on this approach, the repositories would exchange periodic charge information for each block of time.)
- A report-charges transaction between a personal credit server and a billing clearinghouse. This transaction is invoked at least once per billing period. It is used to pass along information about charges. On debit and credit cards, this transaction would also be used to update balance information and credit limits as needed.

All billing transactions are given a transaction ID and are reported to the credit severs by both the server and the client. This reduces possible loss of billing information if one of the parties to a transaction loses a banking card and provides a check against tampering with the system.

Usage Transactions

After the session initiation transactions have been completed, the usage request may then be processed. To sim-

plify the description of the steps carried out in processing a usage request, the term requester is used to refer to a repository in the requester mode which is initiating a request, and the term server is used to refer to a repository in the server mode and which contains the desired digital work. In many cases such as requests to print or view a work, the requester and server may be the same device and the transactions described in the following would be entirely internal. In such instances, certain transaction steps, such as the registration transaction, need not be performed.

There are some common steps that are part of the semantics of all of the usage rights transactions. These steps are referred to as the common transaction steps. There are two sets -- the "opening" steps and the "closing" steps. For simplicity, these are listed here rather than repeating them in the descriptions of all of the usage rights transactions.

Transactions can refer to a part of a digital work, a complete digital work, or a Digital work containing other digital works. Although not described in detail herein, a transaction may even refer to a folder comprised of a plurality of digital works. The term "work" is used to refer to what ever portion or set of digital works is being accessed.

Many of the steps here involve determining if certain conditions are satisfied. Recall that each usage right may have one or more conditions which must be satisfied before the right can be exercised. Digital works have parts and parts have parts. Different parts can have different rights and fees. Thus, it is necessary to verify that the requirements are met for ALL of the parts that are involved in a transaction For brevity, when reference is made to checking whether the rights exist and conditions for exercising are satisfied, it is meant that all such checking takes place for each of the relevant parts of the work.

Figure 18 illustrates the initial common opening and closing steps for a transaction. At this point it is assumed that registration has occurred and that a "trusted" session is in place. General tests are tests on usage rights associated with the folder containing the work or some containing folder higher in the file system hierarchy. These tests correspond to requirements imposed on the work as a consequence of its being on the particular repository, as opposed to being attached to the work itself. Referring to Figure 18, prior to initiating a usage transaction, the requester performs any general tests that are required before the right associated with the transaction can be exercised, step, 1801. For example, install, uninstall and delete rights may be implemented to require that a requester have an authorization certificate before the right can be exercised. Another example is the requirement that a digital ticket be present and punched before a digital work may be copied to a requester. If any of the general tests fail, the transaction is not initiated, step, 1802. Assuming that such required tests are passed, upon receiving the usage request, the server generates a transaction identifier that is used in records or reports of the transaction, step 1803. The server then checks whether the digital work has been granted the right corresponding to the requested transaction, step 1804. If the digital work has not been granted the right corresponding to the request, the transaction terminates, step 1805. If the digital work has been granted the requested right, the server then determines if the various conditions for exercising the right are satisfied. Time based conditions are examined, step 1806. These conditions are checked by examining the time specification for the the version of the right. If any of the conditions are not satisfied, the transaction terminates per step 1805.

Assuming that the time based conditions are satisfied, the server checks security and access conditions, step 1807. Such security and access conditions are satisfied if: 1) the requester is at the specified security class, or a higher security class, 2) the server satisfies any specified authorization test and 3) the requester satisfies any specified authorization tests and has any required digital tickets. If any of the conditions are not satisfied, the transaction terminates per step 1805.

Assuming that the security and access conditions are all satisfied, the server checks the copy count condition, step 1808. If the copy count equals zero, then the transaction cannot be completed and the transaction terminates per step 1805.

Assuming that the copy count does not equal zero, the server checks if the copies in use for the requested right is greater than or equal to any copy count for the requested right (or relevant parts), step 1809. If the copies in use is greater than or equal to the copy count, this indicates that usage rights for the version of the transaction have been exhausted. Accordingly, the server terminates the transaction, step 1805. If the copy count is less than the copies in use for the transaction the transaction can continue, and the copies in use would be incremented by the number of digital works requested in the transaction, step 1810.

The server then checks if the digital work has a "Loan" access right, step 1811. The "Loan" access right is a special case since remaining rights may be present even though all copies are loaned out. If the digital work has the "Loan" access right, a check is made to see if all copies have been loaned out, step 1812. The number of copies that could be loaned is the sum of the Copy-Counts for all of the versions of the loan right of the digital work. For a composite work, the relevant figure is the minimal such sum of each of the components of the composite work. If all copies have been loaned out, the remaining rights are determined, step 1813. The remaining-rights is determined from the remaining rights specifications from the versions of the Loan right. If there is only one version of the Loan right, then the determination is simple. The remaining rights are the ones specified in that version of the Loan right, or none if Remaining-Rights: is not specified. If there are multiple versions of the Loan right and all copies of all of the versions are loaned out, then the remaining rights is taken as the minimum set (intersection) of remaining rights across all of the versions of the loan right. The server then determines if the requested right is in the set of remaining rights, step 1814. If the

requested right is not in the set of remaining rights, the server terminates the transaction, step 1805.

If Loan is not a usage right for the digital work or if all copies have not been loaned out or the requested right is in the set of remaining rights, fee conditions for the right are then checked, step 1815. This will initiate various financial transactions between the repository and associated credit server. Further, any metering of usage of a digital work will commence. If any financial transaction fails, the transaction terminates per step 1805.

It should be noted that the order in which the conditions are checked need not follow the order of steps 1806-1815.

At this point, right specific steps are now performed and are represented here as step 1816. The right specific steps are described in greater detail below.

The common closing transaction steps are now performed. Each of the closing transaction steps are performed by the server after a successful completion of a transaction. Referring back to Figure 18, the copies in use value for the requested right is decremented by the number of copies involved in the transaction, step 1817. Next, if the right had a metered usage fee specification, the server subtracts the elapsed time from the Remaining-Use-Time associated with the right for every part involved in the transaction, step 1818. Finally, if there are fee specifications associated with the right, the server initiates End-Charge financial transaction to confirm billing, step 1819.

Transmission Protocol

An important area to consider is the transmission of the digital work from the server to the requester. The transmission protocol described herein refers to events occurring after a valid session has been created. The transmission protocol must handle the case of disruption in the communications between the repositories. It is assumed that interference such as injecting noise on the communication channel can be detected by the integrity checks (e.g., parity, checksum, etc.) that are built into the transport protocol and are not discussed in detail herein.

The underlying goal in the transmission protocol is to preclude certain failure modes, such as malicious or accidental interference on the communications channel. Suppose, for example, that a user pulls a card with the credit server at a specific time near the end of a transaction. There should not be a vulnerable time at which "pulling the card" causes the repositories to fail to correctly account for the number of copies of the work that have been created. Restated, there should be no time at which a party can break a connection as a means to avoid payment after using a digital work.

If a transaction is interrupted (and fails), both repositories restore the digital works and accounts to their state prior to the failure, modulo records of the failure itself.

Figure 19 is a state diagram showing steps in the process of transmitting information during a transaction. Each box represents a state of a repository in either the server mode (above the central dotted line 1901) or in the requester mode (below the dotted line 1901). Solid arrows stand for transitions between states. Dashed arrows stand for message communications between the repositories. A dashed message arrow pointing to a solid transition arrow is interpreted as meaning that the transition takes place when the message is received. Unlabeled transition arrows take place unconditionally. Other labels on state transition arrows describe conditions that trigger the transition.

Referring now to Figure 19, the server is initially in a state 1902 where a new transaction is initiated via start message 1903. This message includes transaction information including a transaction identifier and a count of the blocks of data to be transferred. The requester, initially in a wait state 1904 then enters a data wait state 1905.

The server enters a data transmit state 1906 and transmits a block of data 1907 and then enters a wait for acknowledgement state 1908. As the data is received, the requester enters a data receive state 1909 and when the data blocks are completely received it enters an acknowledgement state 1910 and transmits an Acknowledgement message 1911 to the server.

If there are more blocks to send, the server waits until receiving an Acknowledgement message from the requester. When an Acknowledgement message is received it sends the next block to the requester and again waits for acknowledgement. The requester also repeats the same cycle of states.

If the server detects a communications failure before sending the last block, it enters a cancellation state 1912 wherein the transaction is cancelled. Similarly, if the requester detects a communications failure before receiving the last block it enters a cancellation state 1913.

If there are no more blocks to send, the server commits to the transaction and waits for the final Acknowledgement in state 1914. If there is a communications failure before the server receives the final Acknowledgement message, it still commits to the transaction but includes a report about the event to its credit server in state 1915. This report serves two purposes. It will help legitimize any claims by a user of having been billed for receiving digital works that were not completely received. Also it helps to identify repositories and communications lines that have suspicious patterns of use and interruption. The server then enters its completion state 1916.

On the requester side, when there are no more blocks to receive, the requester commits to the transaction in state 1917. If the requester detects a communications failure at this state, it reports the failure to its credit server in state 1918, but still commits to the transaction. When it has committed, it sends an acknowledgement message to the server. The server then enters its completion state 1919.

The key property is that both the server and the requester cancel a transaction if it is interrupted before all of the data blocks are delivered, and commits to it if all of the data blocks have been delivered.

There is a possibility that the server will have sent all of the data blocks (and committed) but the requester will not have received all of them and will cancel the transaction. In this case, both repositories will presumably detect a communications failure and report it to their credit server. This case will probably be rare since it depends on very precise timing of the communications failure. The only consequence will be that the user at the requester repository may want to request a refund from the credit services -- and the case for that refund will be documented by reports by both repositories.

To prevent loss of data, the server should not delete any transferred digital work until receiving the final acknowledgement from the requester. But it also should not use the file. A well known way to deal with this situation is called "two-phase commit" or 2PC.

Two-phase commit works as follows. The first phase works the same as the method described above. The server sends all of the data to the requester. Both repositories mark the transaction (and appropriate files) as uncommitted. The server sends a ready-to-commit message to the requester. The requester sends back an acknowledgement. The server then commits and sends the requester a commit message. When the requester receives the commit message, it commits the file.

If there is a communication failure or other crash, the requester must check back with the server to determine the status of the transaction. The server has the last word on this. The requester may have received all of the data, but if it did not get the final message, it has not committed. The server can go ahead and delete files (except for transaction records) once it commits, since the files are known to have been fully transmitted before starting the 2PC cycle.

There are variations known in the art which can be used to achieve the same effect. For example, the server could use an additional level of encryption when transmitting a work to a client. Only after the client sends a message acknowledging receipt does it send the key. The client then agrees to pay for the digital work. The point of this variation is that it provides a clear audit trail that the client received the work. For trusted systems, however, this variation adds a level of encryption for no real gain in accountability.

The transaction for specific usage rights are now discussed.

The Copy Transaction

A Copy transaction is a request to make one or more independent copies of the work with the same or lesser usage rights. Copy differs from the extraction right discussed later in that it refers to entire digital works or entire folders containing digital works. A copy operation cannot be used to remove a portion of a digital work.

- The requester sends the server a message to initiate the Copy Transaction. This message indicates the work to be copied, the version of the copy right to be used for the transaction, the destination address information (location in a folder) for placing the work, the file data for the work (including its size), and the number of copies requested.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the client according to the transmission protocol. If a Next-Set-Of-Rights has been provided in the version of the right, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted. In any event, the Copy-Count field for the copy of the digital work being sent right is set to the number-of-copies requested.
- The requester records the work contents, data, and usage rights and stores the work. It records the date and time that the copy was made in the properties of the digital work.
- The repositories perform the common closing transaction steps.

The Transfer Transaction

A Transfer transaction is a request to move copies of the work with the same or lesser usage rights to another repository. In contrast with a copy transaction, this results in removing the work copies from the server.

- The requester sends the server a message to initiate the Transfer Transaction. This message indicates the work to be transferred, the version of the transfer right to be used in the transaction, the destination address information for placing the work, the file data for the work, and the number of copies involved.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted. In either case, the Copy-Count field for the transmitted rights are set to the number-of-copies requested.

- The requester records the work contents, data, and usage rights and stores the work.
- The server decrements its copy count by the number of copies involved in the transaction.
- The repositories perform the common closing transaction steps.
- If the number of copies remaining in the server is now zero, it erases the digital work from its memory.

The Loan Transaction

A loan transaction is a mechanism for loaning copies of a digital work. The maximum duration of the loan is determined by an internal parameter of the digital work. Works are automatically returned after a predetermined time period.

- The requester sends the server a message to initiate the Transfer Transaction. This message indicates the work to be loaned, the version of the loan right to be used in the transaction, the destination address information for placing the work, the number of copies involved, the file data for the work, and the period of the loan.
- The server checks the validity of the requested loan period, and ends with an error if the period is not valid. Loans for a loaned copy cannot extend beyond the period of the original loan to the server.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted, as modified to reflect the loan period.
- The requester records the digital work contents, data, usage rights, and loan period and stores the work.
- The server updates the usage rights information in the digital work to reflect the number of copies loaned out.
- The repositories perform the common closing transaction steps.
- The server updates the usage rights data for the digital work. This may preclude use of the work until it is returned from the loan. The user on the requester platform can now use the transferred copies of the digital work. A user accessing the original repository cannot use the digital work , unless there are copies remaining. What happens next depends on the order of events in time.

  Case 1. If the time of the loan period is not yet exhausted and the requester sends the repository a Return message.

  - The return message includes the requester identification, and the transaction ID.
  - The server decrements the copies-in-use field by the number of copies that were returned. (If the number of digital works returned is greater than the number actually borrowed, this is treated as an error.) This step may now make the work available at the server for other users.
  - The requester deactivates its copies and removes the contents from its memory.

  Case 2. If the time of the loan period is exhausted and the requester has not yet sent a Return message.

  - The server decrements the copies-in-use field by the number digital works that were borrowed.
  - The requester automatically deactivates its copies of the digital work. It terminates all current uses and erases the digital work copies from memory. One question is why a requester would ever return a work earlier than the period of the loan, since it would be returned automatically anyway. One reason for early return is that there may be a metered fee which determines the cost of the loan. Returning early may reduce that fee.

The Play Transaction

A play transaction is a request to use the contents of a work. Typically, to "play" a work is to send the digital work through some kind of transducer, such as a speaker or a display device. The request implies the intention that the contents will not be communicated digitally to any other system. For example, they will not be sent to a printer, recorded on any digital medium, retained after the transaction or sent to another repository.

This term "play" is natural for examples like playing music, playing a movie, or playing a video game. The general form of play means that a "player" is used to use the digital work. However, the term play covers all media and kinds of recordings. Thus one would "play" a digital work, meaning, to render it for reading, or play a computer program, meaning to execute it. For a digital ticket the player would be a digital ticket agent.

- The requester sends the server a message to initiate the play transaction. This message indicates the work to be

played, the version of the play right to be used in the transaction, the identity of the player being used, and the file data for the work.

- The server checks the validity of the player identification and the compatibility of the player identification with the player specification in the right. It ends with an error if these are not satisfactory.
- The repositories perform the common opening transaction steps.
- The server and requester read and write the blocks of data as requested by the player according to the transmission protocol. The requester plays the work contents, using the player.
- When the player is finished, the player and the requester remove the contents from their memory.
- The repositories perform the common closing transaction steps.

The Print Transaction

A Print transaction is a request to obtain the contents of a work for the purpose of rendering them on a "printer." We use the term "printer" to include the common case of writing with ink on paper. However, the key aspect of "printing" in our use of the term is that it makes a copy of the digital work in a place outside of the protection of usage rights. As with all rights, this may require particular authorization certificates.

Once a digital work is printed, the publisher and user are bound by whatever copyright laws are in effect. However, printing moves the contents outside the control of repositories. For example, absent any other enforcement mechanisms, once a digital work is printed on paper, it can be copied on ordinary photocopying machines without intervention by a repository to collect usage fees. If the printer to a digital disk is permitted, then that digital copy is outside of the control of usage rights. Both the creator and the user know this, although the creator does not necessarily give tacit consent to such copying, which may violate copyright laws.

- The requester sends the server a message to initiate a Print transaction. This message indicates the work to be played, the identity of the printer being used, the file data for the work, and the number of copies in the request.
- The server checks the validity of the printer identification and the compatibility of the printer identification with the printer specification in the right. It ends with an error if these are not satisfactory.
- The repositories perform the common opening transaction steps.
- The server transmits blocks of data according to the transmission protocol.
- The requester prints the work contents, using the printer.
- When the printer is finished, the printer and the requester remove the contents from their memory.
- The repositories perform the common closing transaction steps.

The Backup Transaction

A Backup transaction is a request to make a backup copy of a digital work, as a protection against media failure. In the context of repositories, secure backup copies differ from other copies in three ways: (1) they are made under the control of a Backup transaction rather than a Copy transaction, (2) they do not count as regular copies, and (3) they are not usable as regular copies. Generally, backup copies are encrypted.

Although backup copies may be transferred or copied, depending on their assigned rights, the only way to make them useful for playing, printing or embedding is to restore them.

The output of a Backup operation is both an encrypted data file that contains the contents and description of a work, and a restoration file with an encryption key for restoring the encrypted contents. In many cases, the encrypted data file would have rights for " printing" it to a disk outside of the protection system, relying just on its encryption for security. Such files could be stored anywhere that was physically safe and convenient. The restoration file would be held in the repository. This file is necessary for the restoration of a backup copy. It may have rights for transfer between repositories.

- The requester sends the server a message to initiate a backup transaction. This message indicates the work to be backed up, the version of the backup right to be used in the transaction, the destination address information for placing the backup copy, the file data for the work.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, a set of default rights for backup files of the original are transmitted by the server.
- The requester records the work contents, data, and usage rights. It then creates a one-time key and encrypts the contents file. It saves the key information in a restoration file.
- The repositories perform the common closing transaction steps.

In some cases, it is convenient to be able to archive the large, encrypted contents file to secure offline storage, such as a magneto-optical storage system or magnetic tape. This creation of a non-repository archive file is as secure as the encryption process. Such non-repository archive storage is considered a form of "printing" and is controlled by a print right with a specified "archive-printer." An archive-printer device is programmed to save the encrypted contents file (but not the description file) offline in such a way that it can be retrieved.

The Restore Transaction

A Restore transaction is a request to convert an encrypted backup copy of a digital work into a usable copy. A restore operation is intended to be used to compensate for catastrophic media failure. Like all usage rights, restoration rights can include fees and access tests including authorization checks.

- The requester sends the server a message to initiate a Restore transaction. This message indicates the work to be restored, the version of the restore right for the transaction, the destination address information for placing the work, and the file data for the work.
- The server verifies that the contents file is available (i.e. a digital work corresponding to the request has been backed-up.) If it is not, it ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The server retrieves the key from the restoration file. It decrypts the work contents, data, and usage rights.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, a set of default rights for backup files of the original are transmitted by the server.
- The requester stores the digital work.
- The repositories perform the common closing transaction steps.

The Delete Transaction

A Delete transaction deletes a digital work or a number of copies of a digital work from a repository. Practically all digital works would have delete rights.

- The requester sends the server a message to initiate a delete transaction. This message indicates the work to be deleted , the version of the delete right for the transaction.
- The repositories perform the common opening transaction steps.
- The server deletes the file, erasing it from the file system.
- The repositories perform the common closing transaction steps.

The Directory Transaction

A Directory transaction is a request for information about folders, digital works, and their parts. This amounts to roughly the same idea as protection codes in a conventional file system like TENEX, except that it is generalized to the full power of the access specifications of the usage rights language.

The Directory transaction has the important role of passing along descriptions of the rights and fees associated with a digital work. When a user wants to exercise a right, the user interface of his repository implicitly makes a directory request to determine the versions of the right that are available. Typically these are presented to the user -- such as with different choices of billing for exercising a right. Thus, many directory transactions are invisible to the user and are exercised as part of the normal process of exercising all rights.

- The requester sends the server a message to initiate a Directory transaction. This message indicates the file or folder that is the root of the directory request and the version of the di rectory right used for the transaction.
- The server verifies that the information is accessible to the requester. In particular, it does not return the names of any files that have a HIDE-NAME status in their directory specifications, and it does not return the parts of any folders or files that have HIDE-PARTS in their specification. If the information is not accessible, the server ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The server sends the requested data to the requester according to the transmission protocol.
- The requester records the data.
- The repositories perform the common closing transaction steps.

The Folder Transaction

A Folder transaction is a request to create or rename a folder, or to move a work between folders. Together with Directory rights. Folder rights control the degree to which organization of a repository can be accessed or modified from another repository.

- The requester sends the server a message to initiate a Folder transaction. This message indicates the folder that is the root of the folder request, the version of the folder right for the transaction, an operation, and data. The operation can be one of create, rename, and move file. The data are the specifications required for the operation, such as a specification of a folder or digital work and a name.
- The repositories perform the common opening transaction steps.
- The server performs the requested operation -- creating a folder, renaming a folder, or moving a work between folders.
- The repositories perform the common closing transaction steps.

The Extract Transaction

A extract transaction is a request to copy a part of a digital work and to create a new work containing it. The extraction operation differs from copying in that it can be used to separate a part of a digital work from d-blocks or shells that place additional restrictions or fees on it. The extraction operation differs from the edit operation in that it does not change the contents of a work, only its embedding in d-blocks. Extraction creates a new digital work.

- The requester sends the server a message to initiate an Extract transaction. This message indicates the part of the work to be extracted, the version of the extract right to be used in the transaction, the destination address information for placing the part as a new work, the file data for the work, and the number of copies involved.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the new work. Otherwise, the rights of the original are transmitted. The Copy-Count field for this right is set to the number-of-copies requested.
- The requester records the contents, data, and usage rights and stores the work. It records the date and time that new work was made in the properties of the work.
- The repositories perform the common closing transaction steps.

The Embed Transaction

An embed transaction is a request to make a digital work become a part of another digital work or to add a shell d-block to enable the adding of fees by a distributor of the work.

- The requester sends the server a message to initiate an Embed transaction. This message indicates the work to be embedded, the version of the embed right to be used in the transaction, the destination address information for placing the part as a a work, the file data for the work, and the number of copies involved.
- The server checks the control specifications for all of the rights in the part and the destination. If they are incompatible, the server ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the new work. Otherwise, the rights of the original are transmitted. The Copy-Count field for this right is set to the number-of-copies requested.
- The requester records the contents, data, and usage rights and embeds the work in the destination file.
- The repositories perform the common closing transaction steps.

The Edit Transaction

An Edit transaction is a request to make a new digital work by copying, selecting and modifying portions of an existing digital work. This operation can actually change the contents of a digital work. The kinds of changes that are permitted depend on the process being used. Like the extraction operation, edit operates on portions of a digital work. In contrast with the extract operation, edit does not affect the rights or location of the work. It only changes the contents. The kinds of changes permitted are determined by the type specification of the processor specified in the rights. In the currently preferred embodiment, an edit transaction changes the work itself and does not make a new work. However,

it would be a reasonable variation to cause a new copy of the work to be made.

- The requester sends the server a message to initiate an Edit transaction. This message indicates the work to be edited, the version of the edit right to be used in the transaction, the file data for the work (including its size), the process-ID for the process, and the number of copies involved.
- The server checks the compatibility of the process-ID to be used by the requester against any process-ID specification in the right. If they are incompatible, it ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The requester uses the process to change the contents of the digital work as desired. (For example, it can select and duplicate parts of it; combine it with other information; or compute functions based on the information. This can amount to editing text, music, or pictures or taking whatever other steps are useful in creating a derivative work.)
- The repositories perform the common closing transaction steps.

The edit transaction is used to cover a wide range of kinds of works. The category describes a process that takes as its input any portion of a digital work and then modifies the input in some way. For example, for text, a process for editing the text would require edit rights. A process for "summarizing" or counting words in the text would also be considered editing. For a music file, processing could involve changing the pitch or tempo, or adding reverberations, or any other audio effect. For digital video works, anything which alters the image would require edit rights. Examples would be colorizing, scaling, extracting still photos, selecting and combining frames into story boards, sharpening with signal processing, and so on.

Some creators may want to protect the authenticity of their works by limiting the kinds of processes that can be performed on them. If there are no edit rights, then no processing is allowed at all. A processor identifier can be included to specify what kind of process is allowed. If no process identifier is specified, then arbitrary processors can be used. For an example of a specific process, a photographer may want to allow use of his photograph but may not want it to be colorized. A musician may want to allow extraction of portions of his work but not changing of the tonality.

Authorization Transactions

There are many ways that authorization transactions can be defined. In the following, our preferred way is to simply define them in terms of other transactions that we already need for repositories. Thus, it is convenient sometimes to speak of "authorization transactions," but they are actually made up of other transactions that repositories already have.

A usage right can specify an authorization-ID, which identifies an authorization object (a digital work in a file of a standard format) that the repository must have and which it must process. The authorization is given to the generic authorization (or ticket) server of the repository which begins to interpret the authorization.

As described earlier, the authorization contains a server identifier, which may just be the generic authorization server or it may be another server. When a remote authorization server is required, it must contain a digital address. It may also contain a digital certificate.

If a remote authorization server is required, then the authorization process first performs the following steps:

- The generic authorization server attempts to set up the communications channel. (If the channel cannot be set up, then authorization fails with an error.)
- When the channel is set up, it performs a registration process with the remote repository. (If registration fails, then the authorization fails with an error.)
- When registration is complete, the generic authorization server invokes a "Play" transaction with the remote repository, supplying the authorization document as the digital work to be played, and the remote authorization server (a program) as the "player." (If the player cannot be found or has some other error, then the authorization fails with an error.)
- The authorization server then "plays" the authorization. This involves decrypting it using either the public key of the master repository that issued the certificate or the session key from the repository that transmitted it. The authorization server then performs various tests. These tests vary according to the authorization server. They include such steps as checking issue and validity dates of the authorization and checking any hot-lists of known invalid authorizations. The authorization server may require carrying out any other transactions on the repository as well, such as checking directories, getting some person to supply a password, or playing some other digital work. It may also invoke some special process for checking information about locations or recent events. The "script" for such steps is contained within the authorization server.
- If all of the required steps are completed satisfactorily, the authorization server completes the transaction normally, signaling that authorization is granted.

The Install Transaction

An Install transaction is a request to install a digital work as runnable software on a repository. In a typical case, the requester repository is a rendering repository and the software would be a new kind or new version of a player. Also in a typical case, the software would be copied to file system of the requester repository before it is installed.

- The requester sends the server an Install message. This message indicates the work to be installed, the version of the Install right being invoked, and the file data for the work (including its size).
- The repositories perform the common opening transaction steps.
- The requester extracts a copy of the digital certificate for the software. If the certificate cannot be found or the master repository for the certificate is not known to the requester, the transaction ends with an error.
- The requester decrypts the digital certificate using the public key of the master repository, recording the identity of the supplier and creator, a key for decrypting the software, the compatibility information, and a tamper-checking code. (This step certifies the software.)
- The requester decrypts the software using the key from the certificate and computes a check code on it using a 1-way hash function. If the check-code does not match the tamper-checking code from the certificate, the installation transaction ends with an error. (This step assures that the contents of the software, including the various scripts, have not been tampered with.)
- The requester retrieves the instructions in the compatibility-checking script and follows them. If the software is not compatible with the repository, the installation transaction ends with an error. (This step checks platform compatibility.)
- The requester retrieves the instructions in the installation script and follows them. If there is an error in this process (such as insufficient resources), then the transaction ends with an error. Note that the installation process puts the runnable software in a place in the repository where it is no longer accessible as a work for exercising any usage rights other than the execution of the software as part of repository operations in carrying out other transactions.
- The repositories perform the common closing transaction steps.

The Uninstall Transaction

An Uninstall transaction is a request to remove software from a repository. Since uncontrolled or incorrect removal of software from a repository could compromise its behavioral integrity, this step is controlled.

- The requester sends the server an Uninstall message. This message indicates the work to be uninstalled, the version of the Uninstall right being invoked, and the file data for the work (including its size).
- The repositories perform the common opening transaction steps.
- The requester extracts a copy of the digital certificate for the software. If the certificate cannot be found or the master repository for the certificate is not known to the requester, the transaction ends with an error.
- The requester checks whether the software is installed. If the software is not installed, the transaction ends with an error.
- The requester decrypts the digital certificate using the public key of the master repository, recording the identity of the supplier and creator, a key for decrypting the software, the compatibility information, and a tamper-checking code. (This step authenticates the certification of the software, including the script for uninstalling it.)
- The requester decrypts the software using the key from the certificate and computes a check code on it using a 1-way hash function. If the check-code does not match the tamper-checking code from the certificate, the installation transaction ends with an error. (This step assures that the contents of the software, including the various scripts, have not been tampered with.)
- The requester retrieves the instructions in the uninstallation script and follows them. If there is an error in this process (such as insufficient resources), then the transaction ends with an error.
- The repositories perform the common closing transaction steps.

Claims

1. A distribution system for distributing digital works, said digital works having one or more usage rights attached thereto, said distribution system comprising:

   a grammar for creating instances of usage rights indicating a manner by which a possessor of an associated digital work may transport said associated digital work;

means for creating usage rights from said grammar;
means for attaching created usage rights to a digital work;
a requester repository for accessing digital works, said requester repository having means for generating usage transactions, each said usage transaction specifying a usage right;
a server repository for storing digital works with attached created usage rights, said server repository having means for processing usage transactions from said requester repository to determine if access to a digital work may be granted.

2. The distribution system as recited in Claim 1 wherein said grammar further specifies a default plurality of conditions for an instance of a usage right, wherein said one or more conditions must be satisfied before said usage right may be exercised.

3. The distribution system as recited in Claim 2 wherein said means for creating usage rights from said grammar is further comprised of means for changing said default plurality of conditions for an instance of a usage right.

4. The distribution system as recited in Claim 1 wherein said digital work is a software program.

5. The distribution system as recited in Claim 1 wherein said grammar is further for creating a first version of a usage right having a first set of conditions and a second version of said usage right having a second set of conditions.

6. A computer based system for controlling distribution and use of digital works comprising:

a usage rights grammar for creating instances of usages rights which define how a digital work may be used or distributed, said usage rights grammar comprising a first plurality of grammar elements for defining transport usage rights and a second plurality of grammar elements for defining rendering usage rights;
means for attaching usage rights to digital works;
a plurality of repositories for storing and exchanging digital works, each of said plurality of repositories comprising :
means for storing digital works and their attached usage rights;
transaction processing means having a requester mode of operation for requesting access to a requested digital work, said request specifying a usage right, and a server mode of operation for processing requests to access said requested digital work based on said usage right specified in said request and the usage rights attached to said requested digital work; and
a coupling means for coupling to another of said plurality of repositories across a communications medium.

7. The computer based system for controlling distribution and use of digital works as recited in Claim 6 wherein said first plurality of grammar elements is comprised of:

a loan grammar element for enabling a digital work to be loaned to another repository;
a copy grammar element for enabling a copy of a digital work to be made and transported to another repository; and
a transfer grammar element for enabling a digital work to be transferred to another repository.

8. The computer based system for controlling distribution and use of digital works as recited in Claim 6 or Claim 7 wherein said second plurality of grammar elements is comprised of:

a play grammar element for enabling a digital work to be rendered on a specified class of player device; and
a print grammar element for enabling a digital work to be printed on a specified class of printer device.

9. The computer based system for controlling distribution and use of digital works as recited in any one of Claims 6 to 8 wherein said grammar comprises one or more further pluralities of grammar elements, for defining file management usage rights, for enabling a digital work to be used in the creation of a new digital work,for enabling the secure installation and uninstallation of digital works comprising of software programs, or for providing a set of creator specified conditions which must be satisfied for each instantiation of a usage right defined by a grammar element.

10. A method for controlling distribution and use of digital works comprising the steps of:

a) creating a set of usage rights from a usage rights grammar, each of said usage rights defining a specific instance of how a digital work may be used or distributed, each of said usage rights specifying one or more conditions which must be satisfied in order for said usage right to be exercised;
b) attaching said set of usage rights to a digital work;
c) storing said digital work and its attached usage rights in a first repository;
d) a second repository initiating a request to access said digital work in said first repository, said request specifying a usage right;
e) said first repository receiving said request from said second repository;
f) said first repository determining if said specified usage right is attached to said digital work;
g) said first repository denying access to said digital work if said identified usage right is not attached to said digital work;
h) if said identified usage right is attached to said digital work, said first repository determining if conditions specified by said usage right are satisfied;
i) if said conditions are not satisfied, said first repository denying access to said digital work;
j) if said conditions are satisfied, said first repository transmitting said digital work to said second repository.

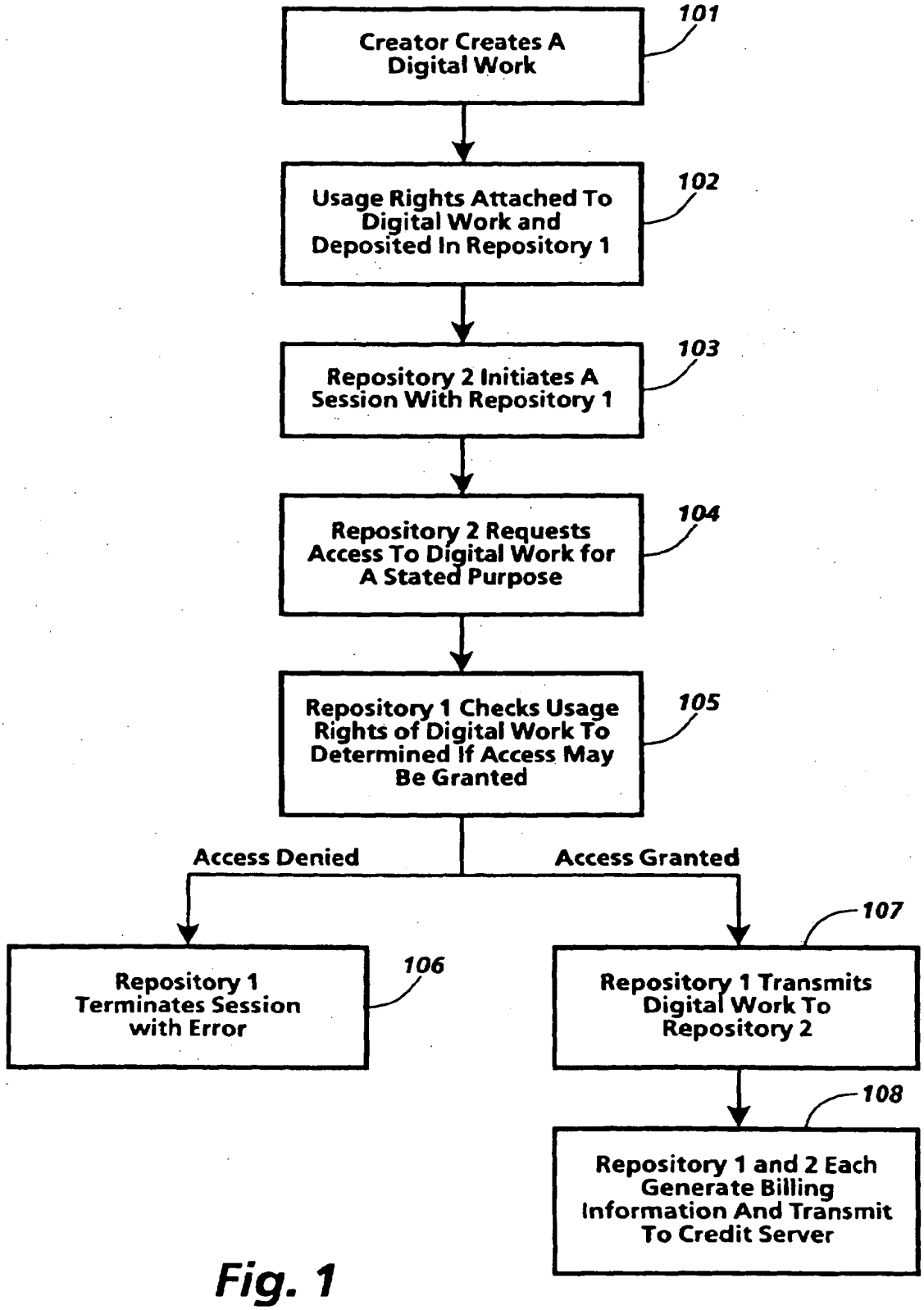


Fig. 1

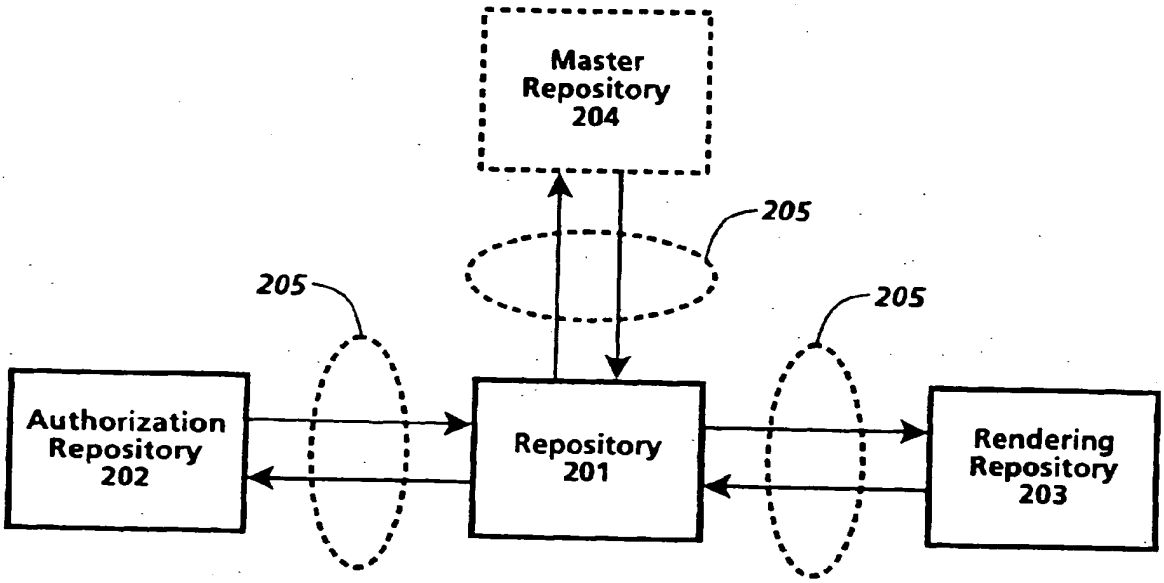


Fig. 2

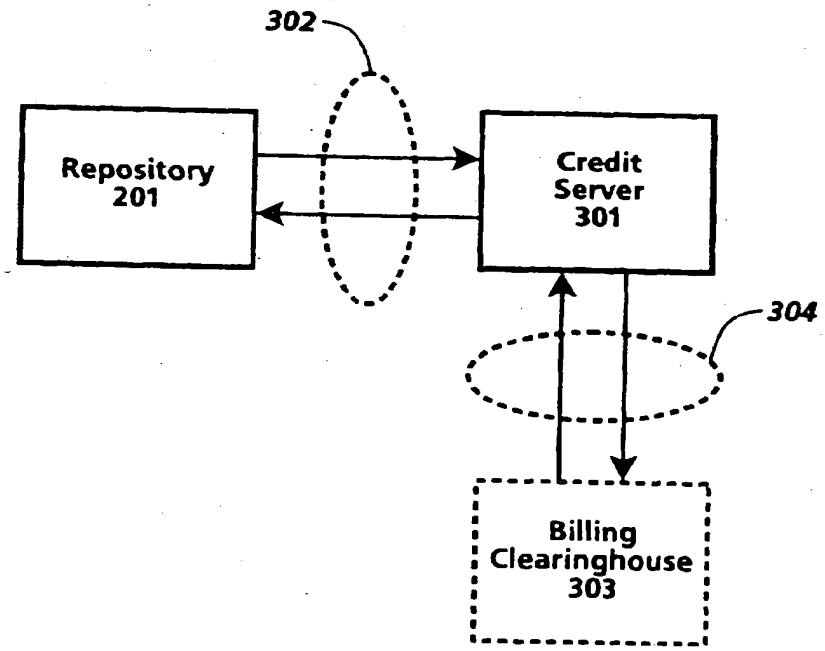


Fig. 3

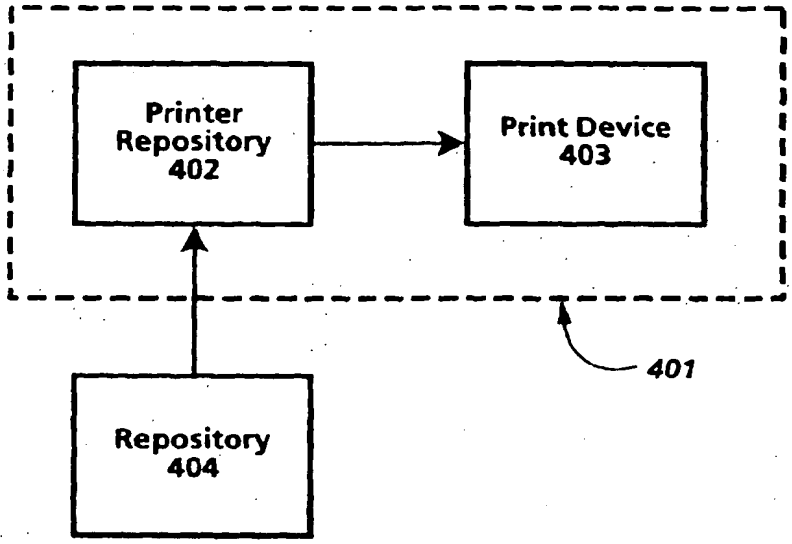


Fig. 4a

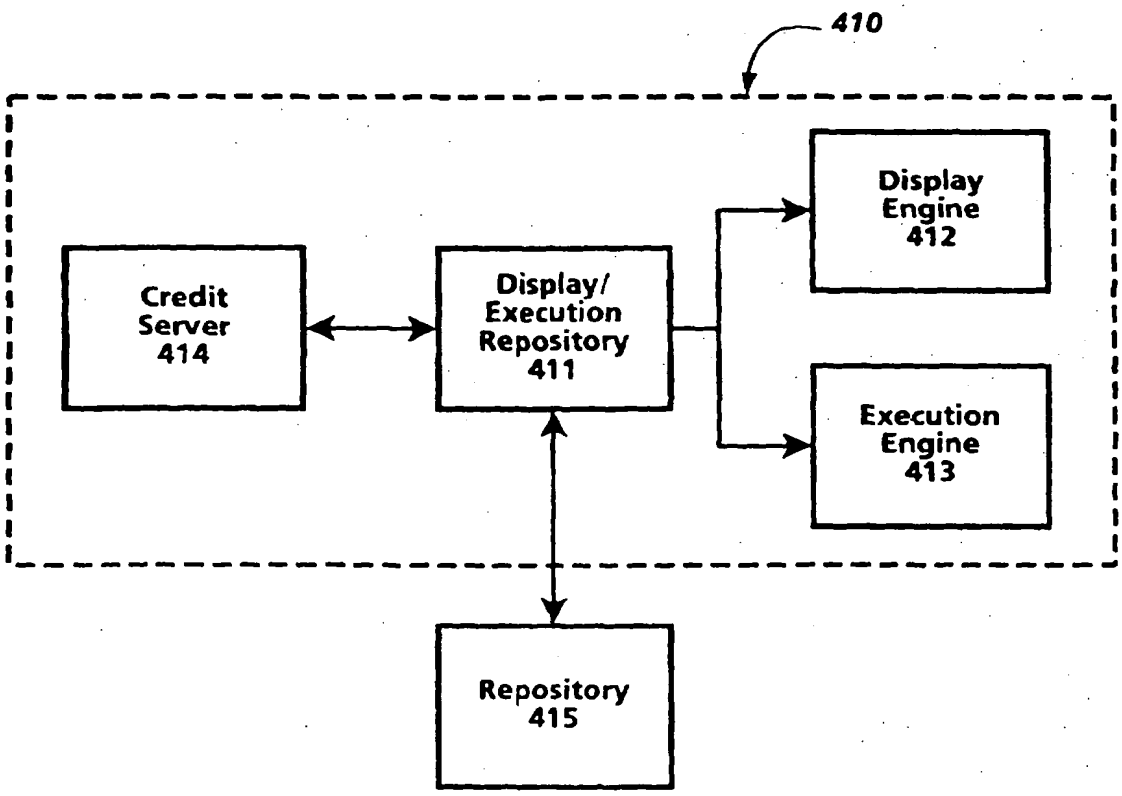


Fig. 4b

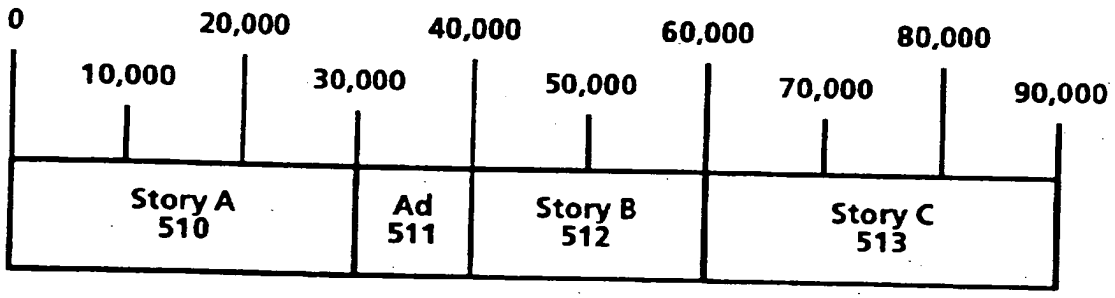


Fig. 5

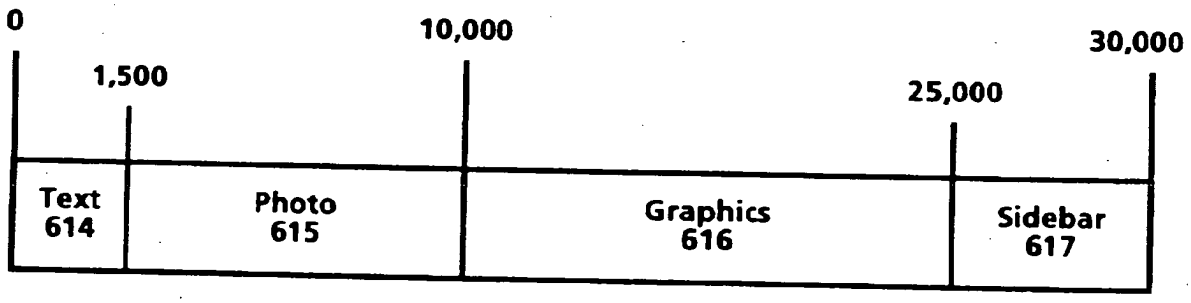


Fig. 6

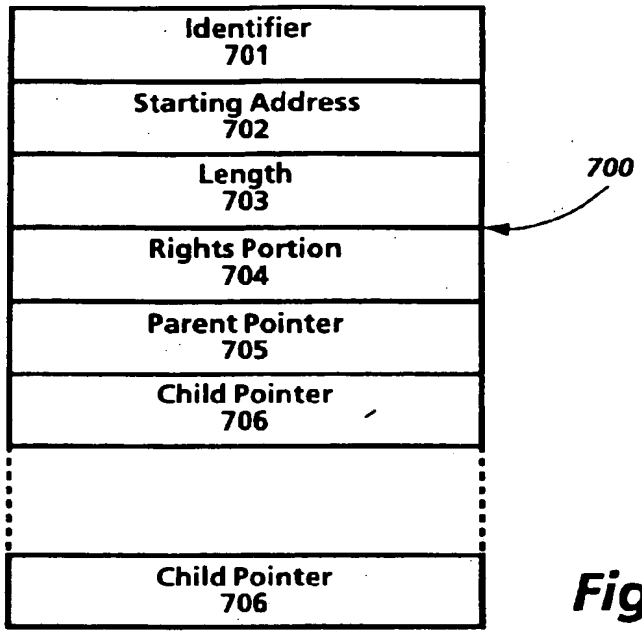


Fig. 7

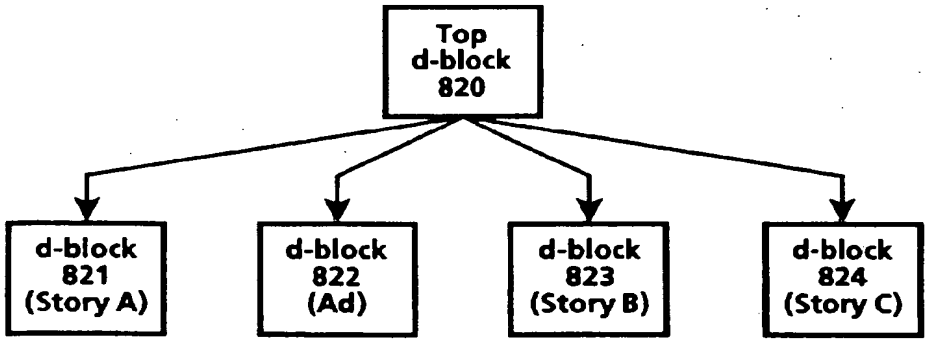


Fig. 8

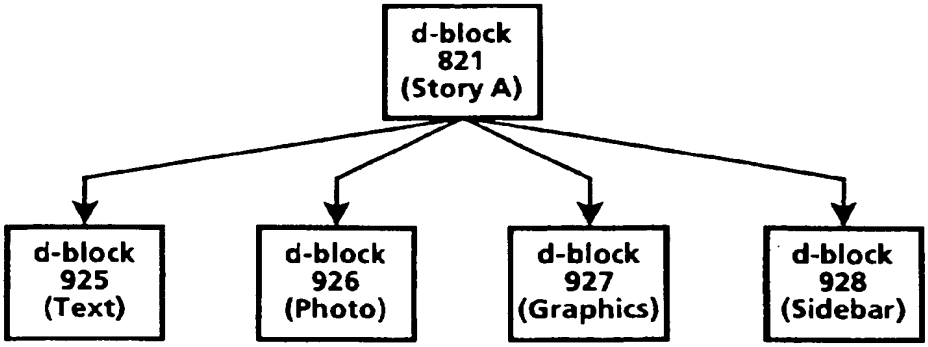


Fig. 9

Fig.10

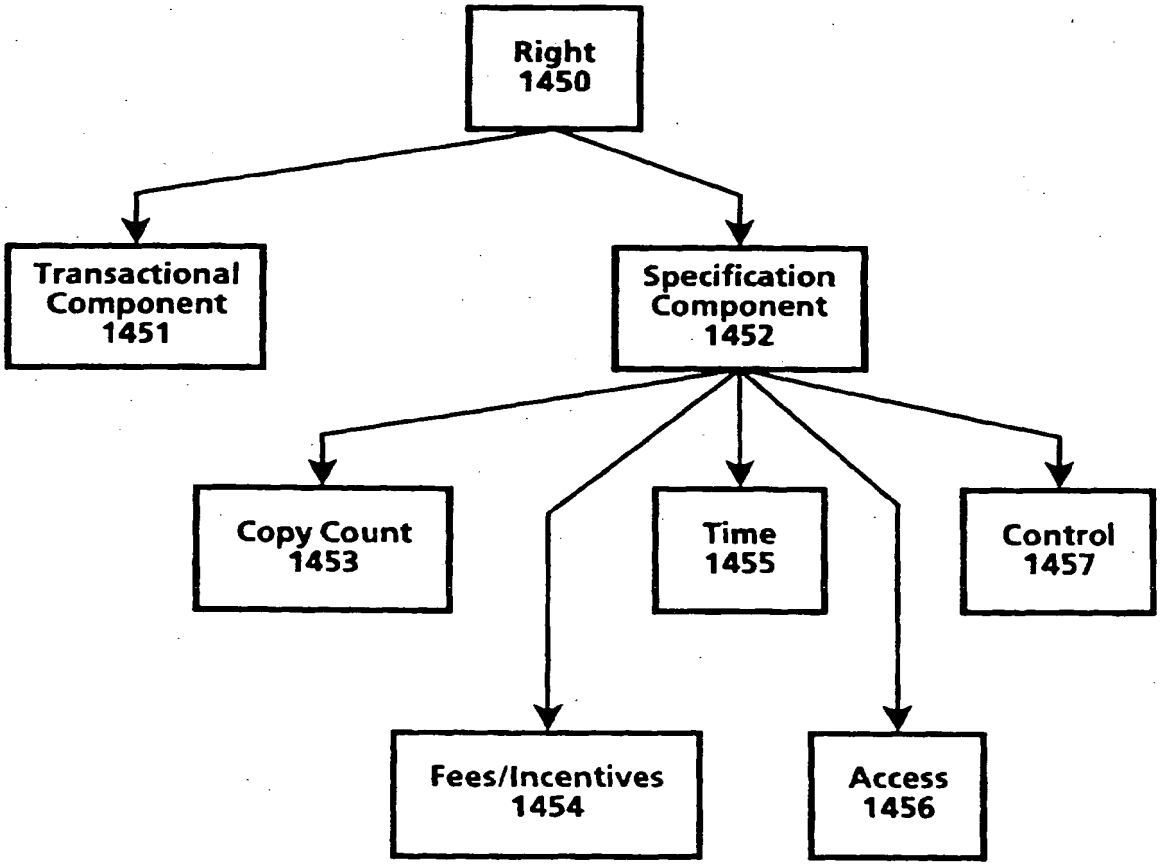


Fig.14

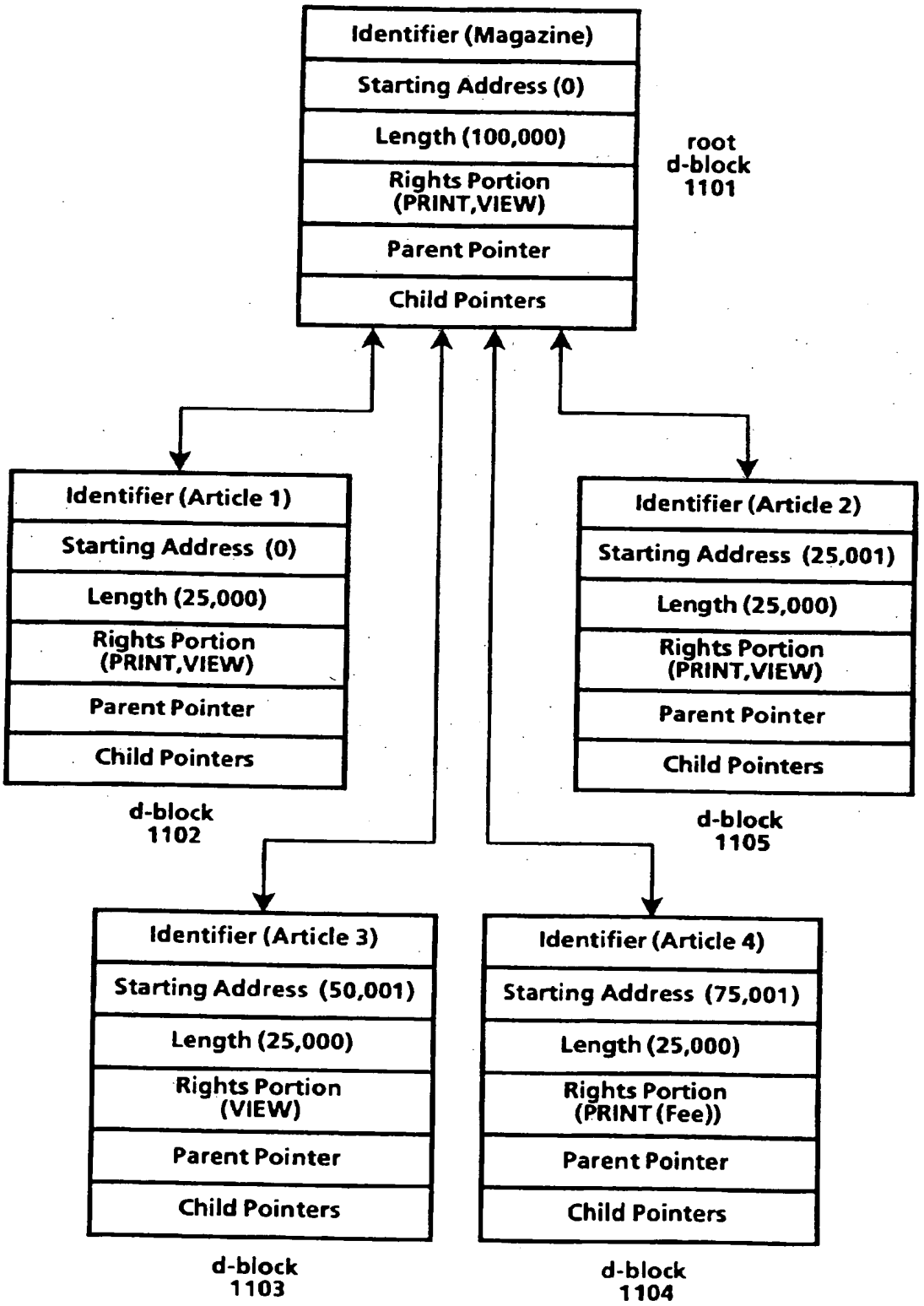


Fig. 11

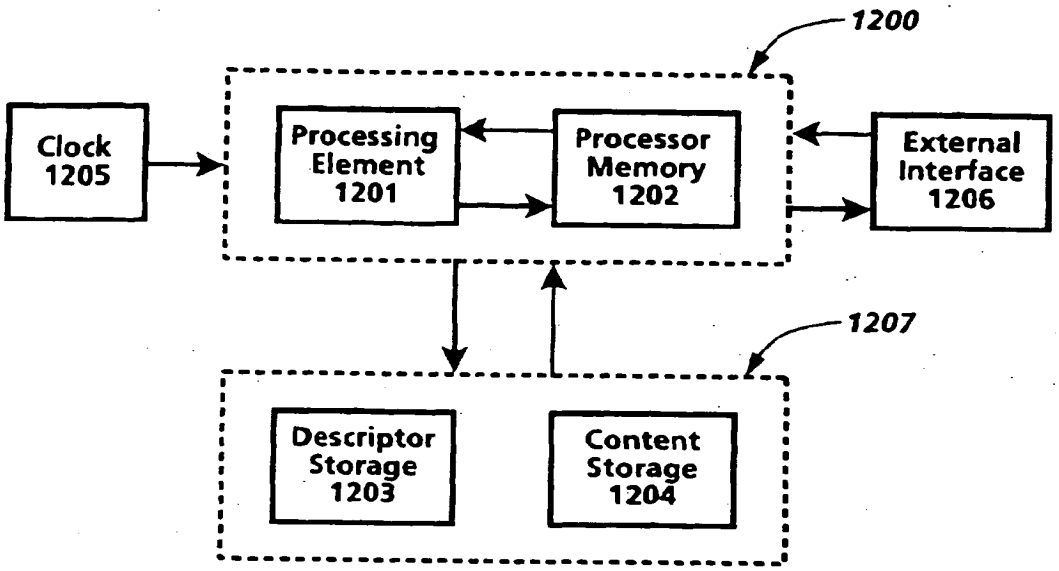


Fig.12

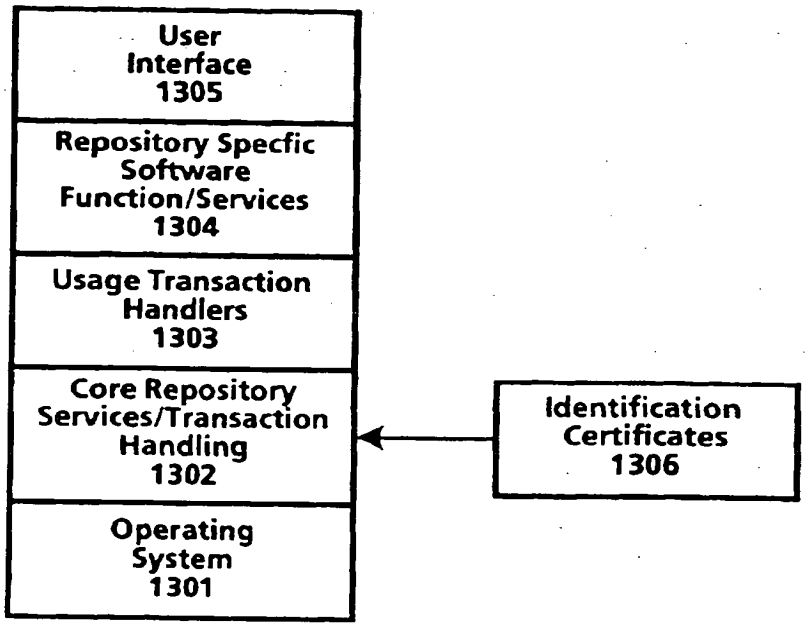


Fig.13

1501 ~ **Digital Work Rights:** = (Rights*)

1502 ~ **Right** : = (Right-Code {Copy-Count} {Control-Spec} {Time-Spec } {Access-Spec} {Fee-Spec})

1503 ~ **Right-Code** : = Render-Code | Transport-Code | File-Management-Code| Derivative-Works- Code | Configuration-Code

1504 ~ **Render-Code** : = [Play : {Player: Player-ID} | Print: {Printer: Printer-ID}]

1505 ~ **Transport-Code** : = [Copy | Transfer | Loan {Remaining-Rights: Next-Set-of-Rights}]{(Next-Copy-Rights: Next-Set-of-Rights)}

1506 ~ **File-Management-Code** : = Backup {Back-Up-Copy-Rights: Next-Set-of-Rights} | Restore | Delete | Folder | Directory {Name: Hide-Local | Hide-Remote} {Parts: Hide-Local | Hide-Remote}

1507 ~ **Derivative-Works-Code** : = [Extract | Embed | Edit{Process: Process-ID}] {Next-Copy-Rights : Next-Set-of Rights}

1508 ~ **Configuration-Code** : = Install | Uninstall

1509 ~ **Next-Set-of-Rights** : = {(Add: Set-Of-Rights)} {(Delete: Set-Of-Rights)} {(Replace: Set-Of-Rights)}{(Keep: Set-Of-Rights)}

1510 ~ **Copy-Count** : = (Copies:positive-integer | 0 | Unlimited)

1511 ~ **Control-Spec** : = (Control: {Restrictable | Unrestrictable} {Unchargeable | Chargeable})

1512 ~ **Time-Spec** : = ({Fixed-Interval | Sliding-Interval | Meter-Time} Until: Expiration-Date)

1513 ~ **Fixed-Interval** : = From: Start-Time

1514 ~ **Sliding-Interval** : = Interval: Use-Duration

1515 ~ **Meter-Time**: = Time-Remaining: Remaining-Use

1516 ~ **Access-Spec** : = ({SC: Security-Class} {Authorization: Authorization-ID*} {Other-Authorization: Authorization-ID*} {Ticket: Ticket-ID})

1517 ~ **Fee-Spec**: = {Scheduled-Discount} Regular-Fee-Spec | Scheduled-Fee-Spec | Markup-Spec

1518 ~ **Scheduled-Discount**: = Scheduled-Discount: (Scheduled-Discount: (Time-Spec Percentage)*)

1519 ~ **Regular-Fee-Spec** : = ({Fee: | Incentive: } [Per-Use-Spec | Metered-Rate-Spec | Best-Price-Spec | Call-For-Price-Spec] {Min: Money-Unit Per: Time-Spec}{Max: Money-Unit Per: Time-Spec} To: Account-ID)

1520 ~ **Per-Use-Spec**: = Per-Use: Money-unit

1521 ~ **Metered-Rate-Spec** : = Metered: Money-Unit Per: Time-Spec

1522 ~ **Best-Price-Spec** : = Best-Price: Money-unit Max: Money-unit

1523 ~ **Call-For-Price-Spec** : = Call-For -Price

1524 ~ **Scheduled-Fee-Spec**: = (Schedule: (Time-Spec Regular-Fee-Spec)*)

1525 ~ **Markup-Spec**: = Markup: percentage To: Account-ID

Fig.15

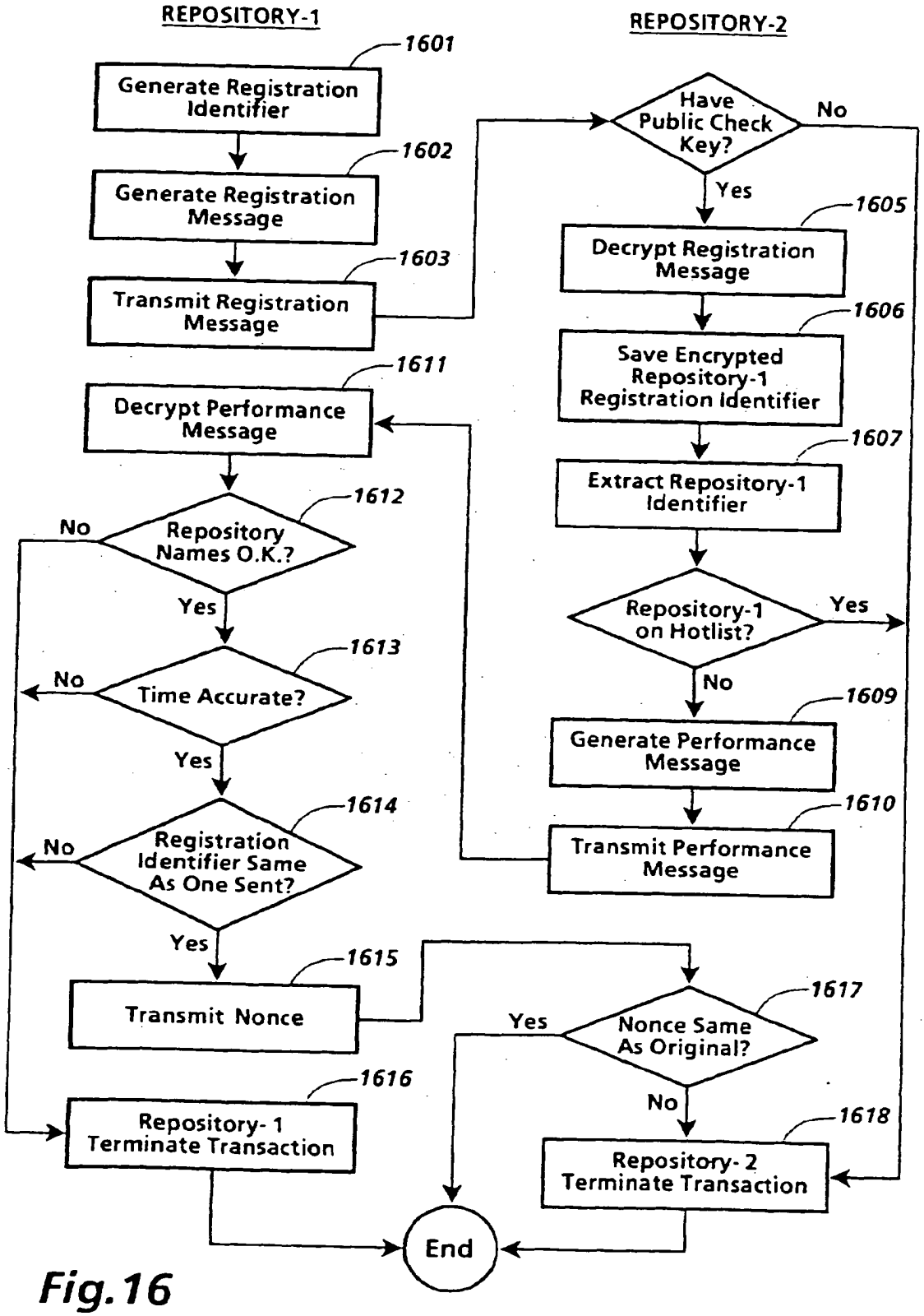


Fig.16

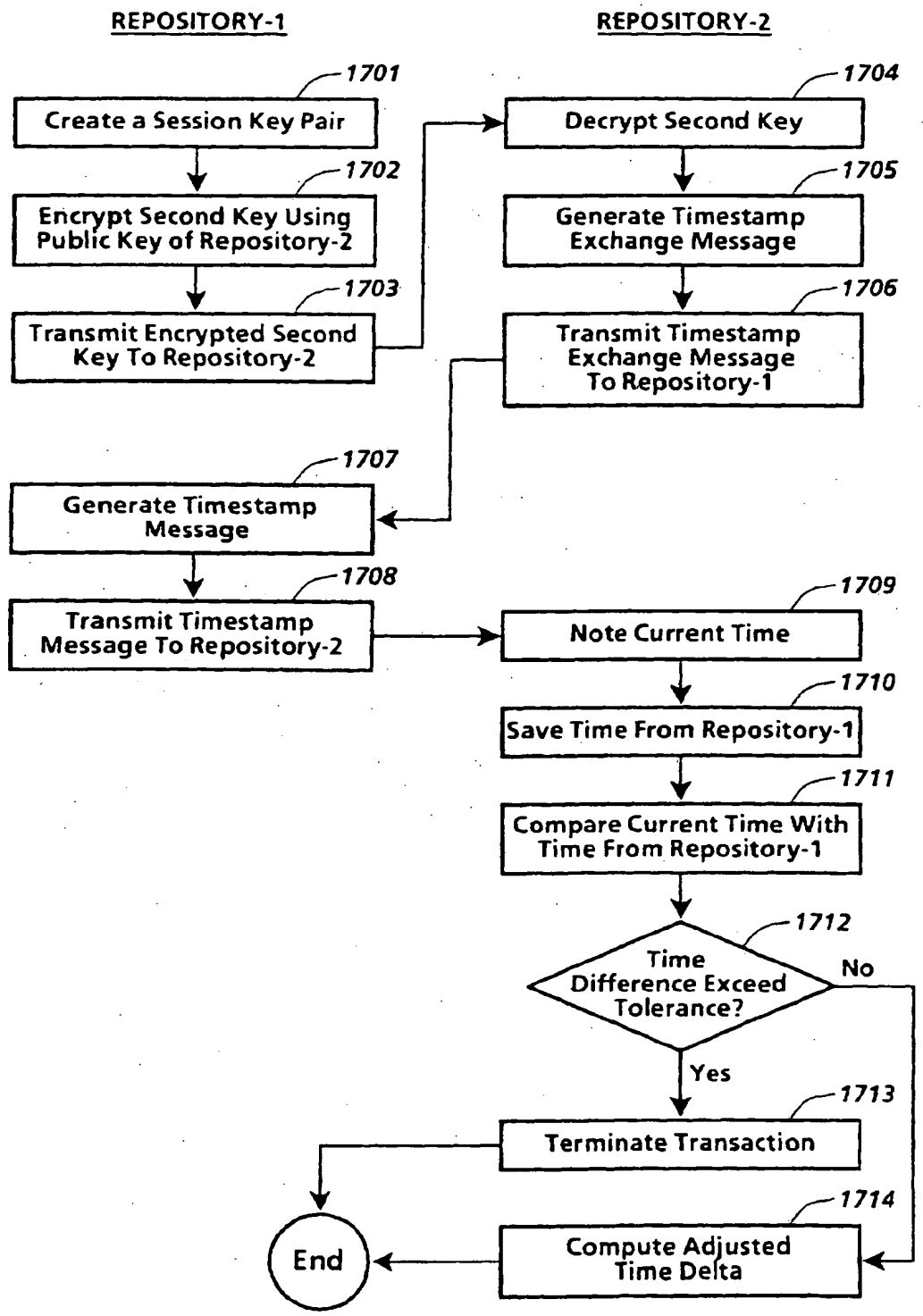


Fig.17

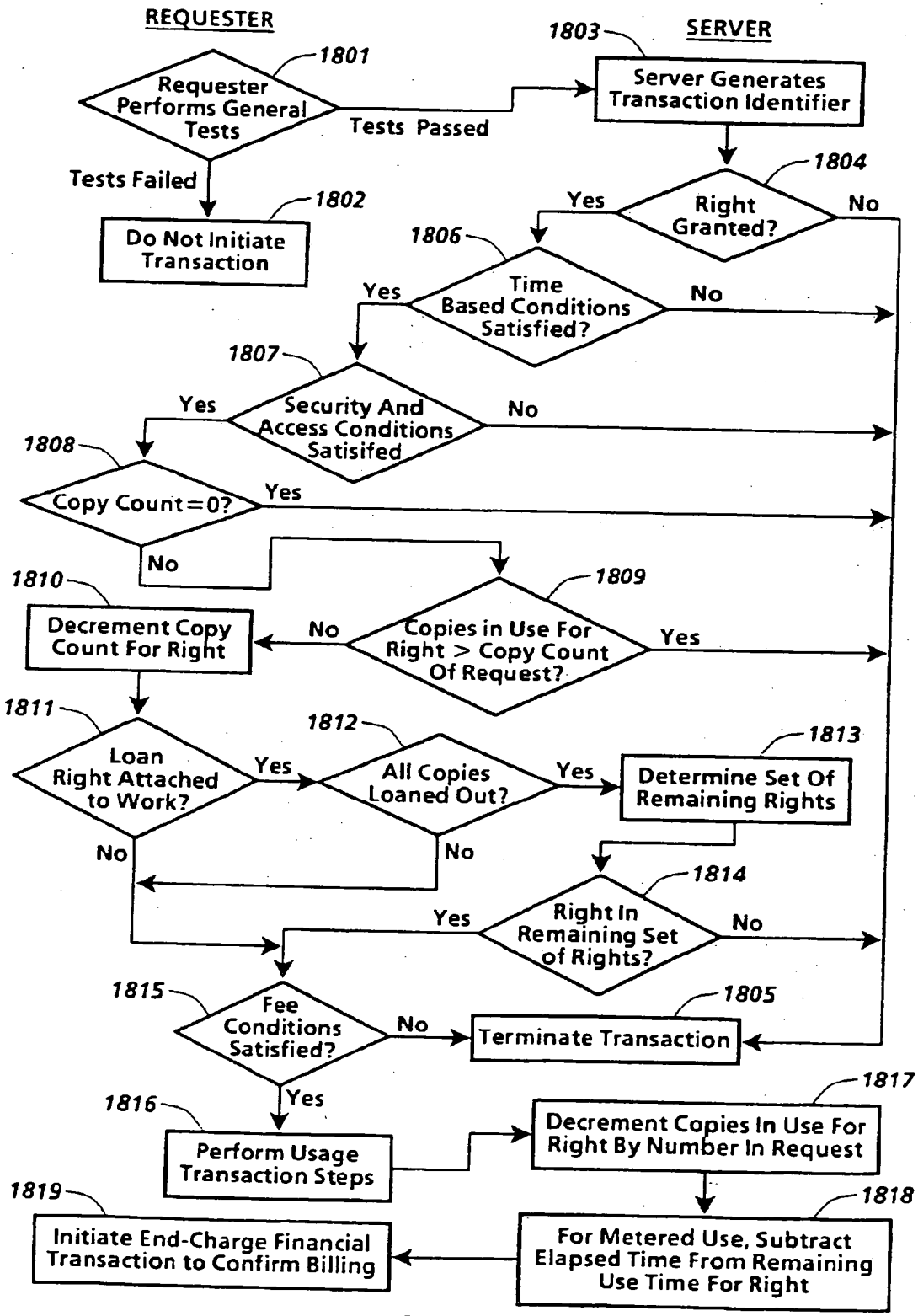


Fig.18

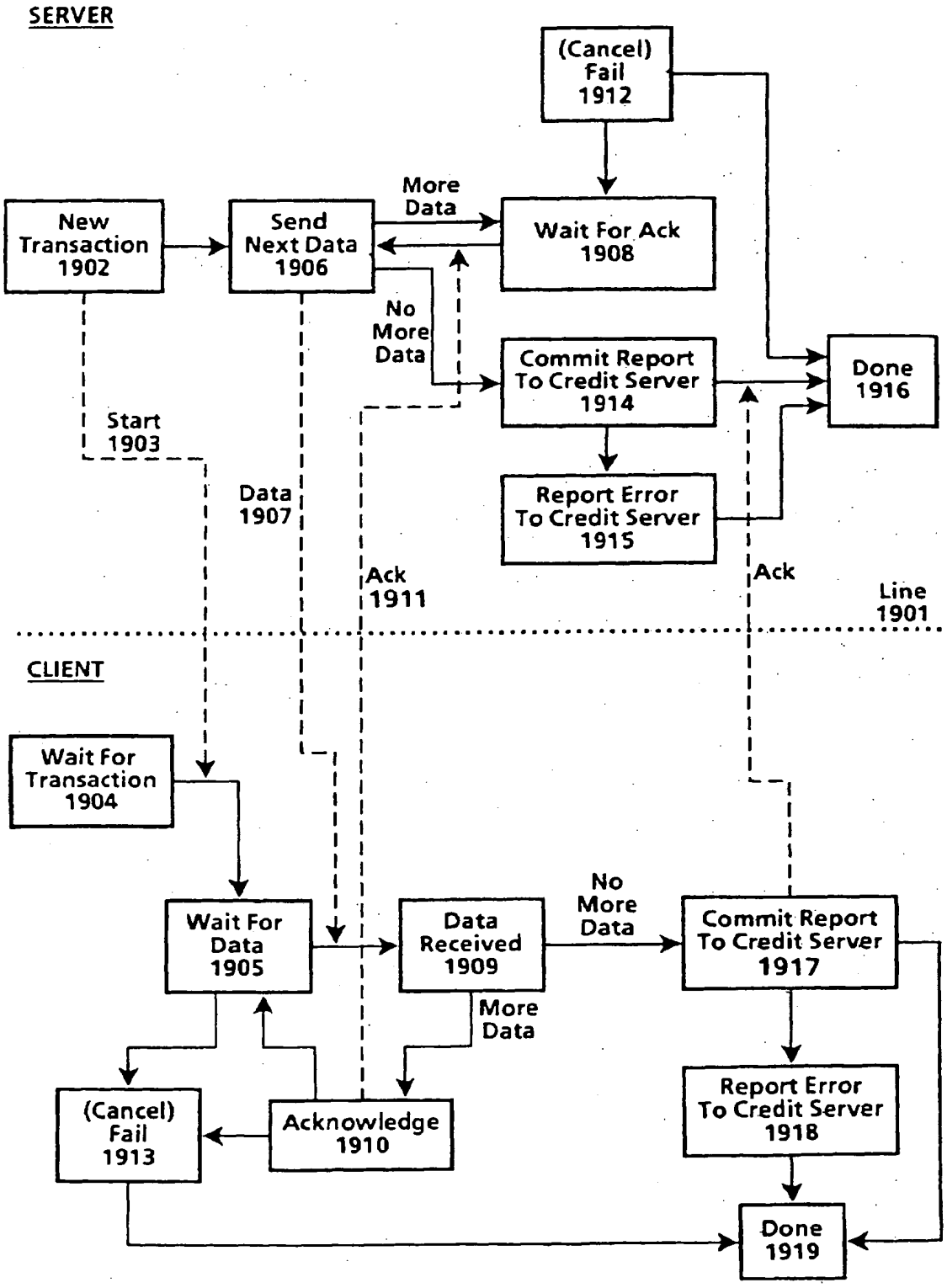


Fig.19

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 95 30 8417

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	WO-A-92 20022 (DIGITAL EQUIPMENT CORP.) * page 45, line 10 - page 80, line 19; figures 1-43 * ---	1,6,10	G06F1/00
A	US-A-5 291 596 (MIITA) * the whole document * ---	1,6,10	
A	GB-A-2 236 604 (SUN MICROSYSTEMS INC) * page 9, line 11 - page 20, line 15 * -----	1,6,10	

TECHNICAL FIELDS SEARCHED (Int.Cl.6)

G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	1 April 1996	Moens, R

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

EP 25661 (4)

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

(11) EP 0 715 243 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
05.06.1996 Bulletin 1996/23

(51) Int Cl.[6]: **G06F 1/00**, G06F 17/60

(21) Application number: **95308414.2**

(22) Date of filing: **23.11.1995**

(84) Designated Contracting States:
DE FR GB

(30) Priority: **23.11.1994 US 344773**

(71) Applicant: **XEROX CORPORATION**
Rochester New York 14644 (US)

(72) Inventors:
- **Stefik, Mark J.**
  Woodside, California 94062 (US)
- **Pirolli, Peter L.T.**
  El Cerrrito, California 94530 (US)
- **Merkle, Ralph C.**
  Sunnyvale, California 94087 (US)

(74) Representative: **Goode, Ian Roy et al**
Rank Xerox Ltd
Patent Department
Parkway
Marlow Buckinghamshire SL7 1YL (GB)

(54) System for controlling the distribution and use of digital works having a fee reporting mechanism

(57) A fee accounting mechanism for reporting fees associated with the distribution and use of digital works. Usage rights and fees are attached to digital works. The usage rights define how the digital work may be used or further distributed. Usage fees are specified as part of a usage right. The digital works and their usage rights and fees are stored in repositories (201). The repositories control access to the digital works. Upon determination that the exercise of a usage right requires a fee, the repository generates a fee reporting transaction (302). Fee reporting is done to a credit server (301). The credit server collects the fee information and periodically transmits it to a billing clearinghouse (303).

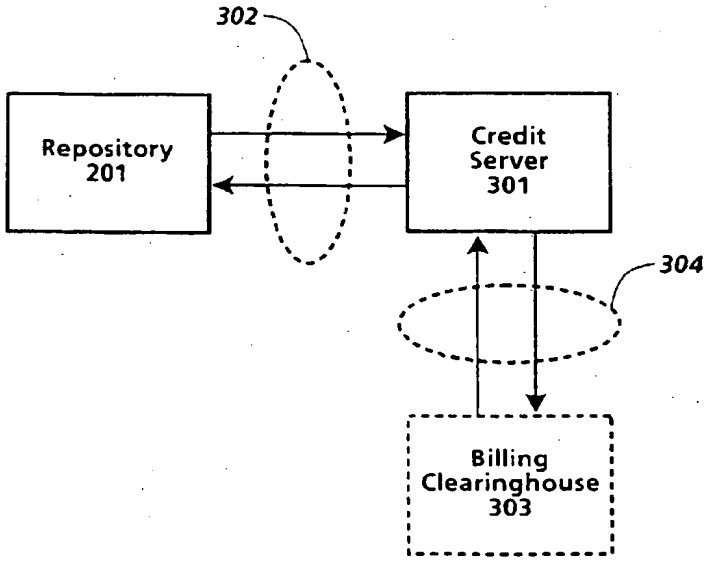


Fig. 3

EP 0 715 243 A1

Printed by Jouve, 75001 PARIS (FR)

Description

The present invention relates to the field of distribution and usage rights enforcement for digitally encoded works.

A fundamental issue facing the publishing and information industries as they consider electronic publishing is how to prevent the unauthorized and unaccounted distribution or usage of electronically published materials. Electronically published materials are typically distributed in a digital form and recreated on a computer based system having the capability to recreate the materials. Audio and video recordings, software, books and multimedia works are all being electronically published. Companies in these industries receive royalties for each accounted for delivery of the materials, e.g. the sale of an audio CD at a retail outlet. Any unaccounted distribution of a work results in an unpaid royalty (e.g. copying the audio recording CD to another digital medium.)

The ease in which electronically published works can be "perfectly" reproduced and distributed is a major concern. The transmission of digital works over networks is commonplace. One such widely used network is the Internet. The Internet is a widespread network facility by which computer users in many universities, corporations and government entities communicate and trade ideas and information. Computer bulletin boards found on the Internet and commercial networks such as CompuServ and Prodigy allow for the posting and retrieving of digital information. Information services such as Dialog and LEXIS/NEXIS provide databases of current information on a wide variety of topics. Another factor which will exacerbate the situation is the development and expansion of the National Information Infrastructure (the NII). It is anticipated that, as the NII grows, the transmission of digital works over networks will increase many times over. It would be desirable to utilize the NII for distribution of digital works without the fear of widespread unauthorized copying.

The most straightforward way to curb unaccounted distribution is to prevent unauthorized copying and transmission. For existing materials that are distributed in digital form, various safeguards are used. In the case of software, copy protection schemes which limit the number of copies that can be made or which corrupt the output when copying is detected have been employed. Another scheme causes software to become disabled after a predetermined period of time has lapsed. A technique used for workstation based software is to require that a special hardware device must be present on the workstation in order for the software to run, e.g., see US-A-4,932,054 entitled "Method and Apparatus for Protecting Computer Software Utilizing Coded Filter Network in Conjunction with an Active Coded Hardware Device." Such devices are provided with the software and are commonly referred to as dongles.

Yet another scheme is to distribute software, but which requires a "key" to enable its use. This is employed in distribution schemes where "demos" of the software are provided on a medium along with the entire product. The demos can be freely used, but in order to use the actual product, the key must be purchased. These schemes do not hinder copying of the software once the key is initially purchased.

It is an object of the present invention to provide an improved system and method for controlling the use and distribution of digital works.

The invention accordingly provides a system and method as claimed in the accompanying claims.

In a system for the control of distribution and use of digital works, a fee reporting mechanism for reporting fees associated with such distribution and use is disclosed. The system includes a means for attaching usage rights to a digital work. The usage rights define how the digital work may be used or further distributed by a possessor of the digital work. Usage fees are specified as part of a usage right. The ability to report usage fees may be a condition to the exercise of a usage right. Further, different fees may be assigned to different usage rights.

The present invention enables various usage fee scenarios to be used. Fees may be assessed on a per use basis, on a metered basis or based on a predetermined schedule. Fees may also be discounted on a predetermined schedule, or they can be marked-up a predetermined percentage (e.g. as a distributor fee). Fee reporting may also be deferred to a later time, to accommodate special deals, rebates or some other external information not yet available.

The present invention supports usage fees in an additive fashion. Usage fees may be reported for a composite digital work, i.e. a digital work comprised of a plurality of discrete digital works each having their own usage rights, and for distributors of digital works. Accordingly, fees to multiple revenue owners can be reported.

Usage fee reporting is done to a credit server. The credit server collects the fee information and periodically transmits it to a billing clearinghouse. Alternatively, the credit server may have a pre-allocated credit which is decremented as fees are incurred. In this alternative embodiment, the credit server would have to be periodically reallocated with credits to enable further use.

A system and method in accordance with the invention will now be described, by way of example, with reference to the accompanying drawings, in which:-

Figure 1 is a flowchart illustrating a simple instantiation of the operation of the currently preferred embodiment of the present invention.

Figure 2 is a block diagram illustrating the various repository types and the repository transaction flow between them in the currently preferred embodiment of the present invention.

Figure 3 is a block diagram of a repository coupled with a credit server in the currently preferred embodiment of

the present invention.

Figures 4a and 4b are examples of rendering systems as may be utilized in the currently preferred embodiment of the present invention.

Figure 5 illustrates a contents file layout for a digital work as may be utilized in the currently preferred embodiment of the present invention.

Figure 6 illustrates a contents file layout for an individual digital work of the digital work of Figure 5 as may be utilized in the currently preferred embodiment of the present invention.

Figure 7 illustrates the components of a description block of the currently preferred embodiment of the present invention.

Figure 8 illustrates a description tree for the contents file layout of the digital work illustrated in Figure 5.

Figure 9 illustrates a portion of a description tree corresponding to the individual digital work illustrated in Figure 6.

Figure 10 illustrates a layout for the rights portion of a description block as may be utilized in the currently preferred embodiment of the present invention.

Figure 11 is a description tree wherein certain d-blocks have PRINT usage rights and is used to illustrate "strict" and "lenient" rules for resolving usage rights conflicts.

Figure 12 is a block diagram of the hardware components of a repository as are utilized in the currently preferred embodiment of the present invention.

Figure 13 is a block diagram of the functional (logical) components of a repository as are utilized in the currently preferred embodiment of the present invention.

Figure 14 is diagram illustrating the basic components of a usage right in the currently preferred embodiment of the present invention.

Figure 15 lists the usage rights grammar of the currently preferred embodiment of the present invention.

Figure 16 is a flowchart illustrating the steps of certificate delivery, hotlist checking and performance testing as performed in a registration transaction as may be performed in the currently preferred embodiment of the present invention.

Figure 17 is a flowchart illustrating the steps of session information exchange and clock synchronization as may be performed in the currently preferred embodiment of the present invention, after each repository in the registration transaction has successfully completed the steps described in Figure 16.

Figure 18 is a flowchart illustrating the basic flow for a usage transaction, including the common opening and closing step, as may be performed in the currently preferred embodiment of the present invention.

Figure 19 is a state diagram of server and client repositories in accordance with a transport protocol followed when moving a digital work from the server to the client repositories, as may be performed in the currently preferred embodiment of the present invention.

OVERVIEW

A system for controlling use and distribution of digital works is disclosed. The present invention is directed to supporting commercial transactions involving digital works.

Herein the terms "digital work", "work" and "content" refer to any work that has been reduced to a digital representation. This would include any audio, video, text, or multimedia work and any accompanying interpreter (e.g. software) that may be required for recreating the work. The term composite work refers to a digital work comprised of a collection of other digital works. The term "usage rights" or "rights" is a term which refers to rights granted to a recipient of a digital work. Generally, these rights define how a digital work can be used and if it can be further distributed. Each usage right may have one or more specified conditions which must be satisfied before the right may be exercised.

Figure 1 is a high level flowchart omitting various details but which demonstrates the basic operation of the present invention. Referring to Figure 1, a creator creates a digital work, step 101. The creator will then determine appropriate usage rights and fees, attach them to the digital work, and store them in Repository 1, step 102. The determination of appropriate usage rights and fees will depend on various economic factors. The digital work remains securely in Repository 1 until a request for access is received. The request for access begins with a session initiation by another repository. Here a Repository 2 initiates a session with Repository 1, step 103. As will be described in greater detail below, this session initiation includes steps which helps to insure that the respective repositories are trustworthy. Assuming that a session can be established, Repository 2 may then request access to the Digital Work for a stated purpose, step 104. The purpose may be, for example, to print the digital work or to obtain a copy of the digital work. The purpose will correspond to a specific usage right. In any event, Repository 1 checks the usage rights associated with the digital work to determine if the access to the digital work may be granted, step 105. The check of the usage rights essentially involves a determination of whether a right associated with the access request has been attached to the digital work and if all conditions associated with the right are satisfied. If the access is denied, repository 1 terminates the session with an error message, step 106. If access is granted, repository 1 transmits the digital work to repository

2, step 107. Once the digital work has been transmitted to repository 2, repository 1 and 2 each generate billing information for the access which is transmitted to a credit server, step 108. Such double billing reporting is done to insure against attempts to circumvent the billing process.

Figure 2 illustrates the basic interactions between repository types in the present invention. As will become apparent from Figure 2, the various repository types will serve different functions. It is fundamental that repositories will share a core set of functionality which will enable secure and trusted communications. Referring to Figure 2, a repository 201 represents the general instance of a repository. The repository 201 has two modes of operation; a server mode and a requester mode. When in the server mode, the repository will be receiving and processing access requests to digital works. When in the requester mode, the repository will be initiating requests to access digital works. Repository 201 is general in the sense that its primary purpose is as an exchange medium for digital works. During the course of operation, the repository 201 may communicate with a plurality of other repositories, namely authorization repository 202, rendering repository 203 and master repository 204. Communication between repositories occurs utilizing a repository transaction protocol 205.

Communication with an authorization repository 202 may occur when a digital work being accessed has a condition requiring an authorization. Conceptually, an authorization is a digital certificate such that possession of the certificate is required to gain access to the digital work. An authorization is itself a digital work that can be moved between repositories and subjected to fees and usage rights conditions. An authorization may be required by both repositories involved in an access to a digital work.

Communication with a rendering repository 203 occurs in connection with the rendering of a digital work. As will be described in greater detail below, a rendering repository is coupled with a rendering device (e.g. a printer device) to comprise a rendering system.

Communication with a master repository 205 occurs in connection with obtaining an identification certificate. Identification certificates are the means by which a repository is identified as "trustworthy". The use of identification certificates is described below with respect to the registration transaction.

Figure 3 illustrates the repository 201 coupled to a credit server 301. The credit server 301 is a device which accumulates billing information for the repository 201. The credit server 301 communicates with repository 201 via billing transactions 302 to record billing transactions. Billing transactions are reported to a billing clearinghouse 303 by the credit server 301 on a periodic basis. The credit server 301 communicates to the billing clearinghouse 303 via clearinghouse transactions 304. The clearinghouse transactions 304 enable a secure and encrypted transmission of information to the billing clearinghouse 303.

RENDERING SYSTEMS

A rendering system is generally defined as a system comprising a repository and a rendering device which can render a digital work into its desired form. Examples of a rendering system may be a computer system, a digital audio system, or a printer. A rendering system has the same security features as a repository. The coupling of a rendering repository with the rendering device may occur in a manner suitable for the type of rendering device.

Figure 4a illustrates a printer as an example of a rendering system. Referring to Figure 4, printer system 401 has contained therein a printer repository 402 and a print device 403. It should be noted that the the dashed line defining printer system 401 defines a secure system boundary. Communications within the boundary are assumed to be secure. Depending on the security level, the boundary also represents a barrier intended to provide physical integrity. The printer repository 402 is an instantiation of the rendering repository 205 of Figure 2. The printer repository 402 will in some instances contain an ephemeral copy of a digital work which remains until it is printed out by the print engine 403. In other instances, the printer repository 402 may contain digital works such as fonts, which will remain and can be billed based on use. This design assures that all communication lines between printers and printing devices are encrypted, unless they are within a physically secure boundary. This design feature eliminates a potential "fault" point through which the digital work could be improperly obtained. The printer device 403 represents the printer components used to create the printed output.

Also illustrated in Figure 4a is the repository 404. The repository 404 is coupled to the printer repository 402. The repository 404 represents an external repository which contains digital works.

Figure 4b is an example of a computer system as a rendering system. A computer system may constitute a "multifunction" device since it may execute digital works (e.g. software programs) and display digital works (e.g. a digitized photograph). Logically, each rendering device can be viewed as having its own repository, although only one physical repository is needed. Referring to Figure 4b, a computer system 410 has contained therein a display/execution repository 411. The display/execution repository 411 is coupled to display device, 412 and execution device 413. The dashed box surrounding the computer system 410 represents a security boundary within which communications are assumed to be secure. The display/execution repository 411 is further coupled to a credit server 414 to report any fees to be billed for access to a digital work and a repository 415 for accessing digital works stored therein.

STRUCTURE OF DIGITAL WORKS

Usage rights are attached directly to digital works. Thus, it is important to understand the structure of a digital work. The structure of a digital work, in particular composite digital works, may be naturally organized into an acyclic structure such as a hierarchy. For example, a magazine has various articles and photographs which may have been created and are owned by different persons. Each of the articles and photographs may represent a node in a hierarchical structure. Consequently, controls, i.e. usage rights, may be placed on each node by the creator. By enabling control and fee billing to be associated with each node, a creator of a work can be assured that the rights and fees are not circumvented.

In the currently preferred embodiment, the file information for a digital work is divided into two files: a "contents" file and a "description tree" file. From the perspective of a repository, the "contents" file is a stream of addressable bytes whose format depends completely on the interpreter used to play, display or print the digital work. The description tree file makes it possible to examine the rights and fees for a work without reference to the content of the digital work. It should be noted that the term description tree as used herein refers to any type of acyclic structure used to represent the relationship between the various components of a digital work.

Figure 5 illustrates the layout of a contents file. Referring to Figure 5, a digital work is comprised of story A 510, advertisement 511, story B 512 and story C 513. It is assumed that the digital work is stored starting at a relative address of 0. Each of the parts of the digital work are stored linearly so that story A 510 is stored at approximately addresses 0-30,000, advertisement 511 at addresses 30,001-40,000, story B 512 at addresses 40,001-60,000 and story C 513 at addresses 60,001-85K. The detail of story A 510 is illustrated in Figure 6. Referring to Figure 6, the story A 510 is further broken down to show text 614 stored at address 0-1500, soldier photo 615 at addresses 1501-10,000, graphics 616 stored at addresses 10,001-25,000 and sidebar 617 stored address 25,001-30,000. Note that the data in the contents file may be compressed (for saving storage) or encrypted (for security).

From Figures 5 and 6 it is readily observed that a digital work can be represented by its component parts as a hierarchy. The description tree for a digital work is comprised of a set of related descriptor blocks (d-blocks). The contents of each d-block is described with respect to Figure 7. Referring to Figure 7, a d-block 700 includes an identifier 701 which is a unique identifier for the work in the repository, a starting address 702 providing the start address of the first byte of the work, a length 703 giving the number of bytes in the work, a rights portion 704 wherein the granted usage rights and their status data are maintained, a parent pointer 705 for pointing to a parent d-block and child pointers 706 for pointing to the child d-blocks. In the currently preferred embodiment, the identifier 701 has two parts. The first part is a unique number assigned to the repository upon manufacture. The second part is a unique number assigned to the work upon creation. The rights portion 704 will contain a data structure, such as a look-up table, wherein the various information associated with a right is maintained. The information required by the respective usage rights is described in more detail below. D-blocks form a strict hierarchy. The top d-block of a work has no parent; all other d-blocks have one parent. The relationship of usage rights between parent and child d-blocks and how conflicts are resolved is described below.

A special type of d-block is a "shell" d-block. A shell d-block adds no new content beyond the content of its parts. A shell d-block is used to add rights and fee information, typically by distributors of digital works.

Figure 8 illustrates a description tree for the digital work of Figure 5. Referring to Figure 8, a top d-block 820 for the digital work points to the various stories and advertisements contained therein. Here, the top d-block 820 points to d-block 821 (representing story A 510), d-block 822 (representing the advertisement 511), d-block 823 (representing story B 512) and and d-block 824 (representing story C 513).

The portion of the description tree for Story A 510 is illustrated in Figure 9. D-block 925 represents text 614, d-block 926 represents photo 615, d-block 927 represents graphics 616 by and d-block 928 represents sidebar 617.

The rights portion 704 of a descriptor block is further illustrated in Figure 10. Figure 10 illustrates a structure which is repeated in the rights portion 704 for each right. Referring to Figure 10, each right will have a right code field 1050 and status information field 1052. The right code field 1050 will contain a unique code assigned to a right. The status information field 1052 will contain information relating to the state of a right and the digital work. Such information is indicated below in Table 1. The rights as stored in the rights portion 704 may typically be in numerical order based on the right code.

TABLE 1

DIGITAL WORK STATE INFORMATION		
Property	**Value**	**Use**
Copies-in-Use	Number	A counter of the number of copies of a work that are in use. Incremented when another copy is used; decremented when use is completed.
Loan-Period	Time-Units	Indicator of the maximum number of time-units that a document can be loaned out
Loaner-Copy	Boolean	Indicator that the current work is a loaned out copy of an authorized digital work.
Remaining-Time	Time-Units	Indicator of the remaining time of use on a metered document right.
Document-Descr	String	A string containing various identifying information about a document. The exact format of this is not specified, but it can include information such as a publisher name, author name, ISBN number, and so on.
Revenue-Owner	RO-Descr	A handle identifying a revenue owner for a digital work. This is used reporting usage fees.
Publication-Date	Date-Descr	The date that the digital work was published.
History-list	History-Rec	A list of events recording the repostories and dates for operations that copy, transfer, backup, or restore a digital work.

The approach for representing digtal works by separating description data from content assumes that parts of a file are contiguous but takes no position on the actual representation of content. In particular, it is neutral to the question of whether content representation may take an object oriented approach. It would be natural to represent content as objects. In principle, it may be convenient to have content objects that include the billing structure and rights information that is represented in the d-blocks. Such variations in the design of the representation are possible and are viable alternatives but may introduce processing overhead, e.g. the interpretation of the objects.

Digital works are stored in a repository as part of a hierarchical file system. Folders (also termed directories and sub-directories) contain the digital works as well as other folders. Digital works and folders in a folder are ordered in alphabetical order. The digital works are typed to reflect how the files are used. Usage rights can be attached to folders so that the folder itself is treated as a digital work. Access to the folder would then be handled in the same fashion as any other digital work As will be described in more detail below, the contents of the folder are subject to their own rights. Moreover, file management rights may be attached to the folder which define how folder contents can be managed.

ATTACHING USAGE RIGHTS TO A DIGITAL WORK

It is fundamental to the present invention that the usage rights are treated as part of the digital work. As the digital work is distributed, the scope of the granted usage rights will remain the same or may be narrowed. For example, when a digital work is transferred from a document server to a repository, the usage rights may include the right to loan a copy for a predetermined period of time (called the original rights). When the repository loans out a copy of the digital work, the usage rights in the loaner copy (called the next set of rights) could be set to prohibit any further rights to loan out the copy. The basic idea is that one cannot grant more rights than they have.

The attachment of usage rights into a digital work may occur in a variety of ways. If the usage rights will be the same for an entire digital work, they could be attached when the digital work is processed for deposit in the digital work server. In the case of a digital work having different usage rights for the various components, this can be done as the digital work is being created. An authoring tool or digital work assembling tool could be utilized which provides for an automated process of attaching the usage rights.

As will be described below, when a digital work is copied, transferred or loaned, a "next set of rights" can be specified. The"next set of rights" will be attached to the digital work as it is transported.

Resolving Conflicting Rights

Because each part of a digital work may have its own usage rights, there will be instances where the rights of a "contained part" are different from its parent or container part. As a result, conflict rules must be established to dictate when and how a right may be exercised. The hierarchical structure of a digital work facilitates the enforcement of such rules. A "strict" rule would be as follows: a right for a part in a digital work is sanctioned if and only if it is sanctioned

for the part, for ancestor d-blocks containing the part and for all descendent d-blocks. By sanctioned, it is meant that (1) each of the respective parts must have the right, and (2) any conditions for exercising the right are satisfied.

It also possible to implement the present invention using a more lenient rule. In the more lenient rule, access to the part may be enabled to the descendent parts which have the right, but access is denied to the descendents which do not.

An example of applying both the strict rule and lenient is illustrated with reference to Figure 11. Referring to Figure 11, a root d-block 1101 has child d-blocks 1102-1105. In this case, root d-block represents a magazine, and each of the child d-blocks 1102-1105 represent articles in the magazine. Suppose that a request is made to PRINT the digital work represented by root d-block 1101 wherein the strict rule is followed. The rights for the root d-block 1101 and child d-blocks 1102-1105 are then examined. Root d-block 1101 and child d-blocks 1102 and 1105 have been granted PRINT rights. Child d-block 1103 has not been granted PRINT rights and child d-block 1104 has PRINT rights conditioned on payment of a usage fee.

Under the strict rule the PRINT right cannot be exercised because the child d-block does not have the PRINT right. Under the lenient rule, the result would be different. The digital works represented by child d-blocks 1102 and 1105 could be printed and the digital work represented by d-block 1104 could be printed so long as the usage fee is paid. Only the digital work represented by d-block 1103 could not be printed. This same result would be accomplished under the strict rule if the requests were directed to each of the individual digital works.

The present invention supports various combinations of allowing and disallowing access. Moreover, as will be described below, the usage rights grammar permits the owner of a digital work to specify if constraints may be imposed on the work by a container part. The manner in which digital works may be sanctioned because of usage rights conflicts would be implementation specific and would depend on the nature of the digital works.

REPOSITORIES

In the description of Figure 2, it was indicated that repositories come in various forms. All repositories provide a core set of services for the transmission of digital works. The manner in which digital works are exchanged is the basis for all transaction between repositories. The various repository types differ in the ultimate functions that they perform. Repositories may be devices themselves, or they may be incorporated into other systems. An example is the rendering repository 203 of Figure 2.

A repository will have associated with it a repository identifier. Typically, the repository identifier would be a unique number assigned to the repository at the time of manufacture. Each repository will also be classified as being in a particular security class. Certain communications and transactions may be conditioned on a repository being in a particular security class. The various security classes are described in greater detail below.

As a prerequisite to operation, a repository will require possession of an identification certificate. Identification certificates are encrypted to prevent forgery and are issued by a Master repository. A master repository plays the role of an authorization agent to enable repositories to receive digital works. Identification certificates must be updated on a periodic basis. Identification certificates are described in greater detail below with respect to the registration transaction.

A repository has both a hardware and functional embodiment. The functional embodiment is typically software executing on the hardware embodiment. Alternatively, the functional embodiment may be embedded in the hardware embodiment such as an Application Specific Integrated Circuit (ASIC) chip.

The hardware embodiment of a repository will be enclosed in a secure housing which if compromised, may cause the repository to be disabled. The basic components of the hardware embodiment of a repository are described with reference to Figure 12. Referring to Figure 12, a repository is comprised of a processing means 1200, storage system 1207, clock 1205 and external interface 1206. The processing means 1200 is comprised of a processor element 1201 and processor memory 1202. The processing means 1201 provides controller, repository transaction and usage rights transaction functions for the repository. Various functions in the operation of the repository such as decryption and/or decompression of digital works and transaction messages are also performed by the processing means 1200. The processor element 1201 may be a microprocessor or other suitable computing component. The processor memory 1202 would typically be further comprised of Read Only Memories (ROM) and Random Access Memories (RAM). Such memories would contain the software instructions utilized by the processor element 1201 in performing the functions of the repository.

The storage system 1207 is further comprised of descriptor storage 1203 and content storage 1204. The description tree storage 1203 will store the description tree for the digital work and the content storage will store the associated content. The description tree storage 1203 and content storage 1204 need not be of the same type of storage medium, nor are they necessarily on the same physical device. So for example, the descriptor storage 1203 may be stored on a solid state storage (for rapid retrieval of the description tree information), while the content storage 1204 may be on a high capacity storage such as an optical disk.

The clock 1205 is used to time-stamp various time based conditions for usage rights or for metering usage fees which may be associated with the digital works. The clock 1205 will have an uninterruptable power supply, e.g. a battery, in order to maintain the integrity of the time-stamps. The external interface means 1206 provides for the signal connection to other repositories and to a credit server. The external interface means 1206 provides for the exchange of signals via such standard interfaces such as RS-232 or Personal Computer Manufacturers Card Industry Association (PCMCIA) standards, or FDDI. The external interface means 1206 may also provide network connectivity.

The functional embodiment of a repository is described with reference to Figure 13. Referring to Figure 13, the functional embodiment is comprised of an operating system 1301, core repository services 1302, usage transaction handlers 1303, repository specific functions, 1304 and a user interface 1305. The operating system 1301 is specific to the repository and would typically depend on the type of processor being used. The operating system 1301 would also provide the basic services for controlling and interfacing between the basic components of the repository.

The core repository services 1302 comprise a set of functions required by each and every repository. The core repository services 1302 include the session initiation transactions which are defined in greater detail below. This set of services also includes a generic ticket agent which is used to "punch" a digital ticket and a generic authorization server for processing authorization specifications. Digital tickets and authorizations are specific mechanisms for controlling the distribution and use of digital works and are described in more detail below. Note that coupled to the core repository services are a plurality of identification certificates 1306. The identification certificates 1306 are required to enable the use of the repository.

The usage transactions handlers 1303 comprise functionality for processing access requests to digital works and for billing fees based on access. The usage transactions supported will be different for each repository type. For example, it may not be necessary for some repositories to handle access requests for digital works.

The repository specific functionality 1304 comprises functionality that is unique to a repository. For example, the master repository has special functionality for issuing digital certificates and maintaining encryption keys. The repository specific functionality 1304 would include the user interface implementation for the repository.

Repository Security Classes

For some digital works the losses caused by any individual instance of unauthorized copying is insignificant and the chief economic concern lies in assuring the convenience of access and low-overhead billing. In such cases, simple and inexpensive handheld repositories and network-based workstations may be suitable repositories, even though the measures and guarantees of security are modest.

At the other extreme, some digital works such as a digital copy of a first run movie or a bearer bond or stock certificate would be of very high value so that it is prudent to employ caution and fairly elaborate security measures to ensure that they are not copied or forged. A repository suitable for holding such a digital work could have elaborate measures for ensuring physical integrity and for verifying authorization before use.

By arranging a universal protocol, all kinds of repositories can communicate with each other in principle. However, creators of some works will want to specify that their works will only be transferred to repositories whose level of security is high enough. For this reason, document repositories have a ranking system for classes and levels of security. The security classes in the currently preferred embodiment are described in Table 2.

TABLE 2

REPOSITORY SECURITY LEVELS	
Level	Description of Security
0	Open system.Document transmission is unencrypted. No digital certificate is required for identification. The security of the system depends mostly on user honesty, since only modest knowledge may be needed to circumvent the security measures. The repositery has no provisions for preventing unauthorized programs from running and accessing or copying files. The system does not prevent the use of removable storage and does not encrypt stored files.
1	Minimal security. Like the previous class except that stored files are minimally encrypted, including ones on removable storage.
2	Basic security. Like the previous class except that special tools and knowledge are required to compromise the programming, the contents of the repository, or the state of the clock. All digital communications are encrypted. A digital certificate is provided as identification. Medium level encryption is used. Repository identification number is unforgeable.

Continuation of the Table on the next page

TABLE 2 (continued)

REPOSITORY SECURITY LEVELS	
Level	Description of Security
3	General security. Like the previous class plus the requirement of special tools are needed to compromise the physical integrity of the repository and that modest encryption is used on all transmissions. Password protection is required to use the local user interface. The digital clock system cannot be reset without authorization. No works would be stored on removable storage. When executing works as programs, it runs them in their own address space and does not give them direct access to any file storage or other memory containing system code or works. They can access works only through the transmission transaction protocol.
4	Like the previous class except that high level encryption is used on all communications. Sensors are used to record attempts at physical and electronic tampering. After such tampering, the repository will not perform other transactions until it has reported such tampering to a designated server.
5	Like the previous class except that if the physical or digital attempts at tampering exceed some preset thresholds that threaten the physical integrity of the repository or the integrity of digital and cryptographic barriers, then the repository will save only document description records of history but will erase or destroy any digital identifiers that could be misused if released to an unscrupulous. It also modifies any certificates of authenticity to indicate that the physical system has been compromised. It also erases the contents of designated documents.
6	Like the previous class except that the repository will attempt wireless communication to report tampering and will employ noisy alarms.
10	This would correspond to a very high level of security. This server would maintain constant communications to remote security systems reporting transactions, sensor readings, and attempts to circumvent security.

The characterization of security levels described in Table 2 is not intended to be fixed. More important is the idea of having different security levels for different repositories. It is anticipated that new security classes and requirements will evolve according to social situations and changes in technology.

Repository User Interface

A user interface is broadly defined as the mechanism by which a user interacts with a repository in order to invoke transactions to gain access to a digital work, or exercise usage rights. As described above, a repository may be embodied in various forms. The user interface for a repository will differ depending on the particular embodiment. The user interface may be a graphical user interface having icons representing the digital works and the various transactions that may be performed. The user interface may be a generated dialog in which a user is prompted for information.

The user interface itself need not be part of the repository. As a repository may be embedded in some other device, the user interface may merely be a part of the device in which the repository is embedded. For example, the repository could be embedded in a "card" that is inserted into an available slot in a computer system. The user interface may be a combination of a display, keyboard, cursor control device and software executing on the computer system.

At a minimum, the user interface must permit a user to input information such as access requests and alpha numeric data and provide feedback as to transaction status. The user interface will then cause the repository to initiate the suitable transactions to service the request. Other facets of a particular user interface will depend on the functionality that a repository will provide.

CREDIT SERVERS

In the present invention, fees may be associated with the exercise of a right. The requirement for payment of fees is described with each version of a usage right in the usage rights language. The recording and reporting of such fees is performed by the credit server. One of the capabilities enabled by associating fees with rights is the possibility of supporting a wide range of charging models. The simplest model, used by conventional software, is that there is a single fee at the time of purchase, after which the purchaser obtains unlimited rights to use the work as often and for as long as he or she wants. Alternative models, include metered use and variable fees. A single work can have different fees for different uses. For example, viewing a photograph on a display could have different fees than making a hardcopy

or including it in a newly created work. A key to these alternative charging models is to have a low overhead means of establishing fees and accounting for credit on these transactions.

A credit server is a computational system that reliably authorizes and records these transactions so that fees are billed and paid. The credit server reports fees to a billing clearinghouse. The billing clearinghouse manages the financial transactions as they occur. As a result, bills may be generated and accounts reconciled. Preferably, the credit server would store the fee transactions and periodically communicate via a network with the billing clearinghouse for reconciliation. In such an embodiment, communications with the billing clearinghouse would be encrypted for integrity and security reasons. In another embodiment, the credit server acts as a "debit card" where transactions occur in "real-time" against a user account.

A credit server is comprised of memory, a processing means, a clock, and interface means for coupling to a repository and a financial institution (e.g. a modem). The credit server will also need to have security and authentication functionality. These elements are essentially the same elements as those of a repository. Thus, a single device can be both a repository and a credit server, provided that it has the appropriate processing elements for carrying out the corresponding functions and protocols. Typically, however, a credit server would be a cardsized system in the possession of the owner of the credit. The credit server is coupled to a repository and would interact via financial transactions as described below. Interactions with a financial institution may occur via protocols established by the financial institutions themselves.

In the currently preferred embodiment credit servers associated with both the server and the repository report the financial transaction to the billing clearinghouse. For example, when a digital work is copied by one repository to another for a fee, credit servers coupled to each of the repositories will report the transaction to the billing clearinghouse. This is desirable in that it insures that a transaction will be accounted for in the event of some break in the communication between a credit server and the billing clearinghouse. However, some implementations may embody only a single credit server reporting the transaction to minimize transaction processing at the risk of losing some transactions.

USAGE RIGHTS LANGUAGE

The present invention uses statements in a high level "usage rights language" to define rights associated with digital works and their parts. Usage rights statements are interpreted by repositories and are used to determine what transactions can be successfully carried out for a digital work and also to determine parameters for those transactions. For example, sentences in the language determine whether a given digital work can be copied, when and how it can be used, and what fees (if any) are to be charged for that use. Once the usage rights statements are generated, they are encoded in a suitable form for accessing during the processing of transactions.

Defining usage rights in terms of a language in combination with the hierarchical representation of a digital work enables the support of a wide variety of distribution and fee schemes. An example is the ability to attach multiple versions of a right to a work. So a creator may attach a PRINT right to make 5 copies for $10.00 and a PRINT right to make unlimited copies for $100.00. A purchaser may then choose which option best fits his needs. Another example is that rights and fees are additive. So in the case of a composite work, the rights and fees of each of the components works is used in determining the rights and fees for the work as a whole.

The basic contents of a right are illustrated in Figure 14. Referring to Figure 14, a right 1450 has a transactional component 1451 and a specifications component 1452. A right 1450 has a label (e.g. COPY or PRINT) which indicates the use or distribution privileges that are embodied by the right. The transactional component 1451 corresponds to a particular way in which a digital work may be used or distributed. The transactional component 1451 is typically embodied in software instructions in a repository which implement the use or distribution privileges for the right. The specifications components 1452 are used to specify conditions which must be satisfied prior to the right being exercised or to designate various transaction related parameters. In the currently preferred embodiment, these specifications include copy count 1453, Fees and Incentives 1454, Time 1455, Access and Security 1456 and Control 1457. Each of these specifications will be described in greater detail below with respect to the language grammar elements.

The usage rights language is based on the grammar described below. A grammar is a convenient means for defining valid sequence of symbols for a language. In describing the grammar the notation "[a|b|c]" is used to indicate distinct choices among alternatives. In this example, a sentence can have either an "a", "b" or "c". It must include exactly one of them. The braces {} are used to indicate optional items. Note that brackets, bars and braces are used to describe the language of usage rights sentences but do not appear in actual sentences in the language.

In contrast, parentheses are part of the usage rights language. Parentheses are used to group items together in lists. The notation (x⁺) is used to indicate a variable length list, that is, a list containing one or more items of type x. The notation (x)* is used to indicate a variable number of lists containing x.

Keywords in the grammar are words followed by colons. Keywords are a common and very special case in the language. They are often used to indicate a single value, typically an identifier. In many cases, the keyword and the parameter are entirely optional. When a keyword is given, it often takes a single identifier as its value. In some cases,

the keyword takes a list of identifiers.

In the usage rights language, time is specified in an hours:minutes:seconds (or hh:mm:ss) representation. Time zone indicators, e.g. PDT for Pacific Daylight Time, may also be specified. Dates are represented as year/ month/day (or YYYY/MMM/DD). Note that these time and date representations may specify moments in time or units of time Money units are specified in terms of dollars.

Finally, in the usage rights language, various "things" will need to interact with each other. For example, an instance of a usage right may specify a bank account, a digital ticket, etc.. Such things need to be identified and are specified herein using the suffix "-ID."

The Usage Rights Grammar is listed in its entirety in Figure 15 and is described below.

Grammar element 1501 **"Digital Work Rights: = (Rights*)"** define the digital work rights as a set of rights. The set of rights attached to a digital work define how that digital work may be transferred, used, performed or played. A set of rights will attach to the entire digital work and in the case of compound digital works, each of the components of the digital work. The usage rights of components of a digital may be different.

Grammar element 1502 **"Right : = (Right-Code {Copy-Count} {Control-Spec} {Time-Spec} {Access-Spec} {Fee-Spec})"** enumerates the content of a right. Each usage right must specify a right code. Each right may also optionally specify conditions which must be satisfied before the right can be exercised. These conditions are copy count, control, time, access and fee conditions. In the currently preferred embodiment, for the optional elements, the following defaults apply: copy count equals 1, no time limit on the use of the right, no access tests or a security level required to use the right and no fee is required. These conditions will each be described in greater detail below.

It is important to note that a digital work may have multiple versions of a right, each having the same right code. The multiple version would provide alternative conditions and fees for accessing the digital work.

Grammar element 1503 **"Right-Code : = Render-Code I Transport-Code I File-Management-Code| Derivative-Works- Code Configuration-Code"** distinguishes each of the specific rights into a particular right type (although each right is identified by distinct right codes). In this way, the grammar provides a catalog of possible rights that can be associated with parts of digital works. In the following, rights are divided into categories for convenience in describing them.

Grammar element 1504 **"Render-Code : = [Play : {Player: Player-ID} I Print: {Printer: Printer-ID}]"** lists a category of rights all involving the making of ephemeral, transitory, or non-digital copies of the digital work. After use the copies are erased.

- Play — A process of rendering or performing a digital work on some processor. This includes such things as playing digital movies, playing digital music, playing a video game, running a computer program, or displaying a document on a display.
- Print — To render the work in a medium that is not further protected by usage rights, such as printing on paper.

Grammar element 1505 **"Transport-Code : = [Copy I Transfer I Loan (Remaining-Rights: Next-Set-of-Rights)] {(Next-Copy-Rights: Next-Set of Rights)}"** lists a category of rights involving the making of persistent, usable copies of the digital work on other repositories. The optional Next-Copy-Rights determine the rights on the work after it is transported. If this is not specified, then the rights on the transported copy are the same as on the original. The optional Remaining-Rights specify the rights that remain with a digital work when it is loaned out. If this is not specified, then the default is that no rights can be exercised when it is loaned out.

- Copy — Make a new copy of a work
- Transfer — Moving a work from one repository to another.
- Loan — Temporarily loaning a copy to another repository for a specified period of time.

Grammar element 1506 **"File-Management-Code : = Backup {Back-Up-Copy-Rights: Next-Set -of Rights}I Restore I Delete I Folder I Directory {Name:Hide-Local I Hide - Remote}{Parts:Hide-Local I Hide-Remote}"** lists a category of rights involving operations for file management, such as the making of backup copies to protect the copy owner against catastrophic equipment failure.

Many software licenses and also copyright law give a copy owner the right to make backup copies to protect against catastrophic failure of equipment. However, the making of uncontrolled backup copies is inherently at odds with the ability to control usage, since an uncontrolled backup copy can be kept and then restored even after the authorized copy was sold.

The File management rights enable the making and restoring of backup copies in a way that respects usage rights, honoring the requirements of both the copy owner and the rights grantor and revenue owner. Backup copies of work descriptions (including usage rights and fee data) can be sent under appropriate protocol and usage rights control to other document repositories of sufficiently high security. Further rights permit organization of digital works into folders

which themselves are treated as digital works and whose contents may be "hidden" from a party seeking to determine the contents of a repository.

- Backup — To make a backup copy of a digital work as protection against media failure.
- Restore — To restore a backup copy of a digital work.
- Delete — To delete or erase a copy of a digital work.
- Folder — To create and name folders, and to move files and folders between folders.
- Directory — To hide a folder or its contents.

Grammar element 1507 **"Derivative-Works-Code : [Extract | Embed | Edit {Process: Process-ID}] {Next-Copy-Rights : Next-Set-of Rights}"** lists a category of rights involving the use of a digital work to create new works.

- Extract — To remove a portion of a work, for the purposes of creating a new work.
- Embed — To include a work in an existing work.
- Edit — To alter a digital work by copying, selecting and modifying portions of an existing digital work.

Grammar element 1508 **"Configuration-Code: = Install | Uninstall"** lists a category of rights for installing and uninstalling software on a repository (typically a rendering repository.) This would typically occur for the installation of a new type of player within the rendering repository.

- Install: — To install new software on a repository.
- Uninstall: — To remove existing software from a repository.

Grammar element 1509 **"Next-Set-of-Rights : = {(Add: Set-Of-Rights)} {(Delete: Set-Of-Rights)} {(Replace: Set-Of-Rights)} {(Keep: Set-Of-Rights)}"** defines how rights are carried forward for a copy of a digital work. If the Next-Copy-Rights is not specified, the rights for the next copy are the same as those of the current copy. Otherwise, the set of rights for the next copy can be specified. Versions of rights after Add: are added to the current set of rights. Rights after Delete: are deleted from the current set of rights. If only right codes are listed after Delete:, then all versions of rights with those codes are deleted. Versions of rights after Replace: subsume all versions of rights of the same type in the current set of rights.

If Remaining-Rights is not specified, then there are no rights for the original after all Loan copies are loaned out. If Remaining-Rights is specified: then the Keep: token can be used to simplify the expression of what rights to keep behind. A list of right codes following keep means that all of the versions of those listed rights are kept in the remaining copy. This specification can be overridden by subsequent Delete: or Replace: specifications.

Copy Count Specification

For various transactions, it may be desirable to provide some limit as to the number of "copies" of the work which may be exercised simultaneously for the right. For example, it may be desirable to limit the number of copies of a digital work that may be loaned out at a time or viewed at a time.

Grammar element 1510 **"Copy-Count : = (Copies: positive-integer | 0 | unlimited)"** provides a condition which defines the number of "copies" of a work subject to the right . A copy count can be 0, a fixed number, or unlimited. The copy-count is associated with each right, as opposed to there being just a single copy-count for the digital work. The Copy-Count for a right is decremented each time that a right is exercised. When the Copy-Count equals zero, the right can no longer be exercised. If the Copy-Count is not specified, the default is one.

Control Specification

Rights and fees depend in general on rights granted by the creator as well as further restrictions imposed by later distributors. Control specifications deal with interactions between the creators and their distributors governing the imposition of further restrictions and fees. For example, a distributor of a digital work may not want an end consumer of a digital work to add fees or otherwise profit by commercially exploiting the purchased digital work.

Grammar element 1511 **"Control-Spec : = (Control: {Restrictable | Unrestrictable} {Unchargeable | Chargeable})"** provides a condition to specify the effect of usage rights and fees of parents on the exercise of the right. A digital work is restrictable if higher level d-blocks can impose further restrictions (time specifications and access specifications) on the right. It is unrestrictable if no further restrictions can be imposed. The default setting is restrictable. A right is unchargeable if no more fees can be imposed on the use of the right. It is chargeable if more fees can be imposed. The default is chargeable.

Time Specification

It is often desirable to assign a start date or specify some duration as to when a right may be exercised. Grammar element 1512 **"Time-Spec : = ({Fixed-Interval | Sliding-Interval | Meter-Time} Until: Expiration-Date)"** provides for specification of time conditions on the exercise of a right. Rights may be granted for a specified time. Different kinds of time specifications are appropriate for different kinds of rights. Some rights may be exercised during a fixed and predetermined duration. Some rights may be exercised for an interval that starts the first time that the right is invoked by some transaction. Some rights may be exercised or are charged according to some kind of metered time, which may be split into separate intervals. For example, a right to view a picture for an hour might be split into six ten minute viewings or four fifteen minute viewings or twenty three minute viewings.

The terms "time" and "date" are used synonymously to refer to a moment in time. There are several kinds of time specifications. Each specification represents some limitation on the times over which the usage right applies. The Expiration-Date specifies the moment at which the usage right ends. For example, if the Expiration-Date is "Jan 1, 1995," then the right ends at the first moment of 1995. If the Expiration-Date is specified as "forever", then the rights are interpreted as continuing without end. If only an expiration date is given, then the right can be exercised as often as desired until the expiration date.

Grammar element 1513 **"Fixed-Interval : = From: Start-Time"** is used to define a predetermined interval that runs from the start time to the expiration date.

Grammar element 1514 **"Sliding-Interval : = Interval: Use-Duration"** is used to define an indeterminate (or "open") start time. It sets limits on a continuous period of time over which the contents are accessible. The period starts on the first access and ends after the duration has passed or the expiration date is reached, whichever comes first. For example, if the right gives 10 hours of continuous access, the use-duration would begin when the first access was made and end 10 hours later.

Grammar element 1515 **"Meter-Time: = Time-Remaining: Remaining-Use"** is used to define a "meter time," that is, a measure of the time that the right is actually exercised. It differs from the Sliding-Interval specification in that the time that the digital work is in use need not be continuous. For example, if the rights guarantee three days of access, those days could be spread out over a month. With this specification, the rights can be exercised until the meter time is exhausted or the expiration date is reached, whichever comes first.

Remaining-Use: = Time-Unit
Start-Time: = Time-Unit
Use-Duration: = Time-Unit
All of the time specifications include time-unit specifications in their ultimate instantiation.

Security Class and Authorization Specification

The present invention provides for various security mechanisms to be introduced into a distribution or use scheme. Grammar element 1516 **"Access-Spec : = ({SC: Security-Class} {Authorization: Authorization-ID*} {Other-Authorization: Authorization-ID*} {Ticket: Ticket-ID})"** provides a means for restricting access and transmission. Access specifications can specify a required security class for a repository to exercise a right or a required authorization test that must be satisfied.

The keyword **"SC:"** is used to specify a minimum security level for the repositories involved in the access. If "SC: " is not specified, the lowest security level is acceptable.

The optional **"Authorization:"** keyword is used to specify required authorizations on the same repository as the work. The optional **"Other-Authorization:"** keyword is used to specify required authorizations on the other repository in the transaction.

The optional **"Ticket:"** keyword specifies the identity of a ticket required for the transaction. A transaction involving digital tickets must locate an appropriate digital ticket agent who can "punch" or otherwise validate the ticket before the transaction can proceed. Tickets are described in greater detail below.

In a transaction involving a repository and a document server, some usage rights may require that the repository have a particular authorization, that the server have some authorization, or that both repositories have (possibly different) authorizations. Authorizations themselves are digital works (hereinafter referred to as an authorization object) that can be moved between repositories in the same manner as other digital works. Their copying and transferring is subject to the same rights and fees as other digital works. A repository is said to have an authorization if that authorization object is contained within the repository.

In some cases, an authorization may be required from a source other than the document server and repository. An authorization object referenced by an Authorization-ID can contain digital address information to be used to set up a communications link between a repository and the authorization source. These are analogous to phone numbers. For such access tests, the communication would need to be established and authorization obtained before the right

could be exercised.

For one-time usage rights, a variant on this scheme is to have a digital ticket. A ticket is presented to a digital ticket agent, whose type is specified on the ticket. In the simplest case, a certified generic ticket agent, available on all repositories, is available to "punch" the ticket. In other cases, the ticket may contain addressing information for locating a "special" ticket agent. Once a ticket has been punched, it cannot be used again for the same kind of transaction (unless it is unpunched or refreshed in the manner described below.) Punching includes marking the ticket with a timestamp of the date and time it was used. Tickets are digital works and can be copied or transferred between repositories according to their usage rights.

In the currently preferred embodiment, a "punched" ticket becomes "unpunched" or "refreshed" when it is copied or extracted. The Copy and Extract operations save the date and time as a property of the digital ticket. When a ticket agent is given a ticket, it can simply check whether the digital copy was made after the last time that it was punched. Of course, the digital ticket must have the copy or extract usage rights attached thereto.

The capability to unpunch a ticket is inportant in the following cases:

- A digital work is circulated at low cost with a limitation that it can be used only once.
- A digital work is circulated with a ticket that can be used once to give discounts on purchases of other works.
- A digital work is circulated with a ticket (included in the purchase price and possibly embedded in the work) that can be used for a future upgrade.

In each of these cases, if a paid copy is made of the digital work (including the ticket) the new owner would expect to get a fresh (unpunched) ticket, whether the copy seller has used the work or not. In contrast, loaning a work or simply transferring it to another repository should not revitalize the ticket.

Usage Fees and Incentives Specification

The billing for use of a digital work is fundamental to a commercial distribution system. Grammar Element 1517 **"Fee-Spec: = {Scheduled-Discount} Regular-Fee-Spec | Scheduled-Fee-Spec | Markup-Spec"** provides a range of options for billing for the use of digital works.

A key feature of this approach is the development of low-overhead billing for transactions in potentially small amounts. Thus, it becomes feasible to collect fees of only a few cents each for thousands of transactions.

The grammar differentiates between uses where the charge is per use from those where it is metered by the time unit. Transactions can support fees that the user pays for using a digital work as well as incentives paid by the right grantor to users to induce them to use or distribute the digital work.

The optional scheduled discount refers to the rest of the fee specification--discounting it by a percentage over time. If it is not specified, then there is no scheduled discount. Regular fee specifications are constant over time. Scheduled fee specifications give a schedule of dates over which the fee specifications change. Markup specifications are used in d-blocks for adding a percentage to the fees already being charged.

Grammar Element 1518 **"Scheduled-Discount: = (Scheduled-Discount: (Time-Spec Percentage)*)"** A Scheduled-Discount is a essentially a scheduled modifier of any other fee specification for this version of the right of the digital work. (It does not refer to children or parent digital works or to other versions of rights.). It is a list of pairs of times and percentages. The most recent time in the list that has not yet passed at the time of the transaction is the one in effect. The percentage gives the discount percentage. For example, the number 10 refers to a 10% discount.

Grammar Element 1519 **"Regular-Fee-Spec : = ({Fee: | Incentive: } [Per-Use-Spec | Metered-Rate-Spec | Best-Price-Spec | Call-For-Price-Spec] {Min: Money-Unit Per: Time-Spec}{Max: Money-Unit Per: Time-Spec} To: Account-ID)"** provides for several kinds of fee specifications.

Fees are paid by the copy-owner/user to the revenue-owner if **Fee**: is specified. Incentives are paid by the revenue-owner to the user if Incentive: is specified. If the **Min**: specification is given, then there is a minimum fee to be charged per time-spec unit for its use. If the **Max**: specification is given, then there is a maximum fee to be charged per time-spec for its use. When **Fee**: is specified, **Account-ID** identifies the account to which the fee is to be paid. When Incentive: is specified, **Account-ID** identifies the account from which the fee is to be paid.

Grammar element 1520 **"Per-Use-Spec: = Per-Use: Money-unit"** defines a simple fee to be paid every time the right is exercised, regardless of how much time the transaction takes.

Grammar element 1521 **"Metered-Rate-Spec := Metered: Money-Unit Per: Time-Spec"** defines a metered-rate fee paid according to how long the right is exercised. Thus, the time it takes to complete the transaction determines the fee.

Grammar element 1522 **"Best-Price-Spec := Best-Price: Money-unit Max: Money-unit"** is used to specify a best-price that is determined when the account is settled. This specification is to accommodate special deals, rebates, and pricing that depends on information that is not available to the repository. All fee specifications can be combined

with tickets or authorizations that could indicate that the consumer is a wholesaler or that he is a preferred customer, or that the seller be authorized in some way. The amount of money in the **Max**: field is the maximum amount that the use will cost. This is the amount that is tentatively debited from the credit server. However, when the transaction is ultimately reconciled, any excess amount will be returned to the consumer in a separate transaction.

Grammar element 1523 **"Call-For-Price-Spec : = Call-For-Price "** is similar to a **"Best-Price-Spec"** in that it is intended to accommodate cases where prices are dynamic. **A Call-For-Price Spec** requires a communication with a dealer to determine the price. This option cannot be exercised if the repository cannot communicate with a dealer at the time that the right is exercised. It is based on a secure transaction whereby the dealer names a price to exercise the right and passes along a deal certificate which is referenced or included in the billing process.

Grammar element 1524 **"Scheduled-Fee-Spec: = (Schedule: (Time-Spec Regular-Fee-Spec)*)"** is used to provide a schedule of dates over which the fee specifications change. The fee specification with the most recent date not in the future is the one that is in effect. This is similar to but more general than the scheduled discount. It is more general, because it provides a means to vary the fee agreement for each time period.

Grammar element 1525 **"Markup-Spec: = Markup: percentage To: Account-ID"** is provided for adding a percentage to the fees already being charged. For example, a 5% markup means that a fee of 5% of cumulative fee so far will be allocated to the distributor. A markup specification can be applied to all of the other kinds of fee specifications. It is typically used in a shell provided by a distributor. It refers to fees associated with d-blocks that are parts of the current d-block. This might be a convenient specification for use in taxes, or in distributor overhead.

REPOSITORY TRANSACTIONS

When a user requests access to a digital work, the repository will initiate various transactions. The combination of transactions invoked will depend on the specifications assigned for a usage right. There are three basic types of transactions, Session Initiation Transactions, Financial Transactions and Usage Transactions. Generally, session initiation transactions are initiated first to establish a valid session. When a valid session is established, transactions corresponding to the various usage rights are invoked. Finally, request specific transactions are performed.

Transactions occur between two repositories (one acting as a server), between a repository and a document playback platform (e.g. for executing or viewing), between a repository and a credit server or between a repository and an authorization server. When transactions occur between more than one repository, it is assumed that there is a reliable communication channel between the repositories. For example, this could be a TCP/IP channel or any other commercially available channel that has built-in capabilities for detecting and correcting transmission errors. However, it is not assumed that the communication channel is secure. Provisions for security and privacy are part of the requirements for specifying and implementing repositories and thus form the need for various transactions.

Message Transmission

Transactions require that there be some communication between repositories. Communication between repositories occurs in units termed as messages. Because the communication line is assumed to be unsecure, all communications with repositories that are above the lowest security class are encrypted utilizing a public key encryption technique. Public key encryption is a well known technique in the encryption arts. The term key refers to a numeric code that is used with encryption and decryption algorithms. Keys come in pairs, where "writing keys" are used to encrypt data and "checking keys" are used to decrypt data. Both writing and checking keys may be public or private. Public keys are those that are distributed to others Private keys are maintained in confidence.

Key management and security is instrumental in the success of a public key encryption system. In the currently preferred embodiment, one or more master repositories maintain the keys and create the identification certificates used by the repositories.

When a sending repository transmits a message to a receiving repository, the sending repository encrypts all of its data using the public writing key of the receiving repository. The sending repository includes its name, the name of the receiving repository, a session identifier such as a nonce (described below), and a message counter in each message.

In this way, the communication can only be read (to a high probability) by the receiving repository, which holds the private checking key for decryption. The auxiliary data is used to guard against various replay attacks to security. If messages ever arrive with the wrong counter or an old nonce, the repositories can assume that someone is interfering with communication and the transaction terminated.

The respective public keys for the repositories to be used for encryption are obtained in the registration transaction described below.

Session Initiation Transactions

A usage transaction is carried out in a session between repositories. For usage transactions involving more than one repository. or for financial transactions between a repository and a credit server, a registration transaction is performed. A second transaction termed a login transaction, may also be needed to initiate the session. The goal of the registration transaction is to establish a secure channel between two repositories who know each others identities. As it is assumed that the communication channel between the repositories is reliable but not secure, there is a risk that a non-repository may mimic the protocol in order to gain illegitimate access to a repository.

The registration transaction between two repositories is described with respect to Figures 16 and 17. The steps described are from the perspective of a "repository-1" registering its identity with a "repository-2". The registration must be symmetrical so the same set of steps will be repeated for repository-2 registering its identity with repository-1. Referring to Figure 16, repository-1 first generates an encrypted registration identifier, step 1601 and then generates a registration message, step 1602. A registration message is comprised of an identifier of a master repository, the identification certificate for the repository-1 and an encrypted random registration identifier. The identification certificate is encrypted by the master repository in its private key and attests to the fact that the repository (here repository-1) is a bona fide repository. The identification certificate also contains a public key for the repository, the repository security level and a timestamp (indicating a time after which the certificate is no longer valid.) The registration identifier is a number generated by the repository for this registration. The registration identifier is unique to the session and is encrypted in repository-1's private key. The registration identifier is used to improve security of authentication by detecting certain kinds of communications based attacks. Repository-1 then transmits the registration message to repository-2, step 1603.

Upon receiving the registration message, repository-2 determines if it has the needed public key for the master repository, step 1604. If repository-2 does not have the needed public key to decrypt the identification certificate, the registration transaction terminates in an error, step 1618.

Assuming that repository-2 has the proper public key the identification certificate is decrypted, step 1605. Repository-2 saves the encrypted registration identifier, step 1606, and extracts the repository identifier, step 1607. The extracted repository identifier is checked against a "hotlist" of compromised document repositories, step 1608. In the currently preferred embodiment, each repository will contain "hotlists" of compromised repositories. If the repository is on the "hotlist', the registration transaction terminates in an error per step 1618. Repositories can be removed from the hotlist when their certificates expire, so that the list does not need to grow without bound. Also, by keeping a short list of hotlist certificates that it has previously received, a repository can avoid the work of actually going through the list. These lists would be encrypted by a master repository. A minor variation on the approach to improve efficiency would have the repositories first exchange lists of names of hotlist certificates, ultimately exchanging only those lists that they had not previously received. The "hotlists" are maintained and distributed by Master repositories.

Note that rather than terminating in error, the transaction could request that another registration message be sent based on an identification certificate created by another master repository. This may be repeated until a satisfactory identification certificate is found, or it is determined that trust cannot be established.

Assuming that the repository is not on the hotlist, the repository identification needs to be verified. In other words, repository-2 needs to validate that the repository on the other end is really repository-1. This is termed performance testing and is performed in order to avoid invalid access to the repository via a counterfeit repository replaying a recording of a prior session initiation between repository-1 and repository-2. Performance testing is initiated by repository-2 generating a performance message, step 1609. The performance message consists of a nonce, the names of the respective repositories, the time and the registration identifier received from repository-1. A nonce is a generated message based on some random and variable information (e.g. the time or the temperature.) The nonce is used to check whether repository-1 can actually exhibit correct encrypting of a message using the private keys it claims to have, on a message that it has never seen before. The performance message is encrypted using the public key specified in the registration message of repository-1. The performance message is transmitted to repository-1, step 1610, where it is decrypted by repository-1 using its private key, step 1611. Repository-1 then checks to make sure that the names of the two repositories are correct, step 1612, that the time is accurate, step 1613 and that the registration identifier corresponds to the one it sent, step 1614. If any of these tests fails, the transaction is terminated per step 1616. Assuming that the tests are passed, repository-1 transmits the nonce to repository-2 in the clear, step 1615. Repository-2 then compares the received nonce to the original nonce, step 1617. If they are not identical, the registration transaction terminates in an error per step 1618. If they are the same, the registration transaction has successfully completed.

At this point, assuming that the transaction has not terminated, the repositories exchange messages containing session keys to be used in all communications during the session and synchronize their clocks. Figure 17 illustrates the session information exchange and clock synchronization steps (again from the perspective of repository-1.) Referring to Figure 17, repository-1 creates a session key pair, step 1701. A first key is kept private and is used by repository-1 to encrypt messages. The second key is a public key used by repository-2 to decrypt messages. The

second key is encrypted using the public key of repository-2, step 1702 and is sent to repository-2, step 1703. Upon receipt, repository-2 decrypts the second key, step 1704. The second key is used to decrypt messages in subsequent communications. When each repository has completed this step, they are both convinced that the other repository is bona fide and that they are communicating with the original. Each repository has given the other a key to be used in decrypting further communications during the session. Since that key is itself transmitted in the public key of the receiving repository only it will be able to decrypt the key which is used to decrypt subsequent messages.

After the session information is exchanged, the repositories must synchronize their clocks Clock synchronization is used by the repositories to establish an agreed upon time base for the financial records of their mutual transactions. Referring back to Figure 17, repository-2 initiates clock synchronization by generating a time stamp exchange message, step 1705, and transmits it to repository- 1, step 1706. Upon receipt, repository-1 generates its own time stamp message, step 1707 and transmits it back to repository-2, step 1708. Repository-2 notes the current time, step 1709 and stores the time received from repository-1, step 1710. The current time is compared to the time received from repository-1, step 1711. The difference is then checked to see if it exceeds a predetermined tolerance (e.g. one minute), step 1712. If it does, repository-2 terminates the transaction as this may indicate tampering with the repository, step 1713. If not repository-2 computes an adjusted time delta, step 1714. The adjusted time delta is the difference between the clock time of repository-2 and the average of the times from repository-1 and repository-2.

To achieve greater accuracy, repository-2 can request the time again up to a fixed number of times (e.g. five times), repeat the clock synchronization steps, and average the results.

A second session initiation transaction is a Login transaction. The Login transaction is used to check the authenticity of a user requesting a transaction. A Login transaction is particularly prudent for the authorization of financial transactions that will be charged to a credit server. The Login transaction involves an interaction between the user at a user interface and the credit server associated with a repository. The information exchanged here is a login string supplied by the repository/credit server to identify itself to the user, and a Personal Identification Number (PIN) provided by the user to identify himself to the credit server. In the event that the user is accessing a credit server on a repository different from the one on which the user interface resides, exchange of the information would be encrypted using the public and private keys of the respective repositories.

Billing Transactions

Billing Transactions are concerned with monetary transactions with a credit server. Billing Transactions are carried out when all other conditions are satisfied and a usage fee is required for granting the request. For the most part, billing transactions are well understood in the state of the art. These transactions are between a repository and a credit server, or between a credit server and a billing clearinghouse. Briefly, the required transactions include the following:

- Registration and LOG IN transactions by which the repository and user establish their bona fides to a credit server. These transactions would be entirely internal in cases where the repository and credit server are implemented as a single system.
- Registration and LOG IN transactions, by which a credit server establishes its bona fides to a billing clearinghouse.
- An Assign-fee transaction to assign a charge. The information in this transaction would include a transaction identifier, the identities of the repositories in the transaction, and a list of charges from the parts of the digital work. If there has been any unusual event in the transaction such as an interruption of communications, that information is included as well.
- A Begin-charges transaction to assign a charge. This transaction is much the same as an assign-fee transaction except that it is used for metered use. It includes the same information as the assign-fee transaction as well as the usage fee information. The credit-server is then responsible for running a clock.
- An End-charges transaction to end a charge for metered use. (In a variation on this approach, the repositories would exchange periodic charge information for each block of time.)
- A report-charges transaction between a personal credit server and a billing clearinghouse. This transaction is invoked at least once per billing period. It is used to pass along information about charges. On debit and credit cards, this transaction would also be used to update balance information and credit limits as needed.

All billing transactions are given a transaction ID and are reported to the credit severs by both the server and the client. This reduces possible loss of billing information if one of the parties to a transaction loses a banking card and provides a check against tampering with the system.

Usage Transactions

After the session initiation transactions have been completed, the usage request may then be processed. To sim-

plify the description of the steps carried out in processing a usage request, the term requester is used to refer to a repository in the requester mode which is initiating a request, and the term server is used to refer to a repository in the server mode and which contains the desired digital work. In many cases such as requests to print or view a work, the requester and server may be the same device and the transactions described in the following would be entirely internal. In such instances, certain transaction steps, such as the registration transaction, need not be performed.

There are some common steps that are part of the semantics of all of the usage rights transactions. These steps are referred to as the common transaction steps. There are two sets --the "opening" steps and the "closing" steps. For simplicity, these are listed here rather than repeating them in the descriptions of all of the usage rights transactions.

Transactions can refer to a part of a digital work, a complete digital work, or a Digital work containing other digital works. Although not described in detail herein, a transaction may even refer to a folder comprised of a plurality of digital works. The term "work" is used to refer to what ever portion or set of digital works is being accessed.

Many of the steps here involve determining if certain conditions are satisfied. Recall that each usage right may have one or more conditions which must be satisfied before the right can be exercised. Digital works have parts and parts have parts. Different parts can have different rights and fees. Thus, it is necessary to verify that the requirements are met for ALL of the parts that are involved in a transaction For brevity, when reference is made to checking whether the rights exist and conditions for exercising are satisfied, it is meant that all such checking takes place for each of the relevant parts of the work.

Figure 18 illustrates the initial common opening and closing steps for a transaction. At this point it is assumed that registration has occurred and that a "trusted" session is in place. General tests are tests on usage rights associated with the folder containing the work or some containing folder higher in the file system hierarchy. These tests correspond to requirements imposed on the work as a consequence of its being on the particular repository, as opposed to being attached to the work itself. Referring to Figure 18, prior to initiating a usage transaction, the requester performs any general tests that are required before the right associated with the transaction can be exercised, step, 1801. For example, install, uninstall and delete rights may be implemented to require that a requester have an authorization certificate before the right can be exercised. Another example is the requirement that a digital ticket be present and punched before a digital work may be copied to a requester. If any of the general tests fail, the transaction is not initiated, step, 1802. Assuming that such required tests are passed; upon receiving the usage request, the server generates a transaction identifier that is used in records or reports of the transaction, step 1803. The server then checks whether the digital work has been granted the right corresponding to the requested transaction, step 1804. If the digital work has not been granted the right corresponding to the request, the transaction terminates, step 1805. If the digital work has been granted the requested right, the server then determines if the various conditions for exercising the right are satisfied. Time based conditions are examined, step 1806. These conditions are checked by examining the time specification for the the version of the right. If any of the conditions are not satisfied, the transaction terminates per step 1805.

Assuming that the time based conditions are satisfied, the server checks security and access conditions, step 1807. Such security and access conditions are satisfied if: 1) the requester is at the specified security class, or a higher security class, 2) the server satisfies any specified authorization test and 3) the requester satisfies any specified authorization tests and has any required digital tickets. If any of the conditions are not satisfied, the transaction terminates per step 1805.

Assuming that the security and access conditions are all satisfied, the server checks the copy count condition, step 1808. If the copy count equals zero, then the transaction cannot be completed and the transaction terminates per step 1805.

Assuming that the copy count does not equal zero, the server checks if the copies in use for the requested right is greater than or equal to any copy count for the requested right (or relevant parts), step 1809. If the copies in use is greater than or equal to the copy count, this indicates that usage rights for the version of the transaction have been exhausted. Accordingly, the server terminates the transaction, step 1805. If the copy count is less than the copies in use for the transaction the transaction can continue, and the copies in use would be incremented by the number of digital works requested in the transaction, step 1810.

The server then checks if the digital work has a "Loan" access right, step 1811. The "Loan" access right is a special case since remaining rights may be present even though all copies are loaned out. If the digital work has the "Loan" access right, a check is made to see if all copies have been loaned out, step 1812. The number of copies that could be loaned is the sum of the Copy-Counts for all of the versions of the loan right of the digital work. For a composite work, the relevant figure is the minimal such sum of each of the components of the composite work. If all copies have been loaned out, the remaining rights are determined, step 1813. The remaining-rights is determined from the remaining rights specifications from the versions of the Loan right. If there is only one version of the Loan right, then the determination is simple. The remaining rights are the ones specified in that version of the Loan right, or none if Remaining-Rights: is not specified. If there are multiple versions of the Loan right and all copies of all of the versions are loaned out, then the remaining rights is taken as the minimum set (intersection) of remaining rights across all of the versions of the loan right. The server then determines if the requested right is in the set of remaining rights, step 1814. If the

requested right is not in the set of remaining rights, the server terminates the transaction, step 1805.

If Loan is not a usage right for the digital work or if all copies have not been loaned out or the requested right is in the set of remaining rights, fee conditions for the right are then checked, step 1815. This will initiate various financial transactions between the repository and associated credit server. Further, any metering of usage of a digital work will commence. If any financial transaction fails, the transaction terminates per step 1805.

It should be noted that the order in which the conditions are checked need not follow the order of steps 1806-1815.

At this point, right specific steps are now performed and are represented here as step 1816. The right specific steps are described in greater detail below.

The common closing transaction steps are now performed. Each of the closing transaction steps are performed by the server after a successful completion of a transaction. Referring back to Figure 18, the copies in use value for the requested right is decremented by the number of copies involved in the transaction, step 1817. Next, if the right had a metered usage fee specification, the server subtracts the elapsed time from the Remaining-Use-Time associated with the right for every part involved in the transaction, step 1818. Finally, if there are fee specifications associated with the right, the server initiates End-Charge financial transaction to confirm billing, step 1819.

Transmission Protocol

An important area to consider is the transmission of the digital work from the server to the requester. The transmission protocol described herein refers to events occurring after a valid session has been created. The transmission protocol must handle the case of disruption in the communications between the repositories. It is assumed that interference such as injecting noise on the communication channel can be detected by the integrity checks (e.g., parity, checksum, etc.) that are built into the transport protocol and are not discussed in detail herein.

The underlying goal in the transmission protocol is to preclude certain failure modes, such as malicious or accidental interference on the communications channel. Suppose, for example, that a user pulls a card with the credit server at a specific time near the end of a transaction. There should not be a vulnerable time at which "pulling the card" causes the repositories to fail to correctly account for the number of copies of the work that have been created. Restated, there should be no time at which a party can break a connection as a means to avoid payment after using a digital work.

If a transaction is interrupted (and fails), both repositories restore the digital works and accounts to their state prior to the failure, modulo records of the failure itself.

Figure 19 is a state diagram showing steps in the process of transmitting information during a transaction. Each box represents a state of a repository in either the server mode (above the central dotted line 1901) or in the requester mode (below the dotted line 1901). Solid arrows stand for transitions between states. Dashed arrows stand for message communications between the repositories. A dashed message arrow pointing to a solid transition arrow is interpreted as meaning that the transition takes place when the message is received. Unlabeled transition arrows take place unconditionally. Other labels on state transition arrows describe conditions that trigger the transition.

Referring now to Figure 19, the server is initially in a state 1902 where a new transaction is initiated via start message 1903. This message includes transaction information including a transaction identifier and a count of the blocks of data to be transferred. The requester, initially in a wait state 1904 then enters a data wait state 1905.

The server enters a data transmit state 1906 and transmits a block of data 1907 and then enters a wait for acknowledgement state 1908. As the data is received, the requester enters a data receive state 1909 and when the data blocks are completely received it enters an acknowledgement state 1910 and transmits an Acknowledgement message 1911 to the server.

If there are more blocks to send, the server waits until receiving an Acknowledgement message from the requester. When an Acknowledgement message is received it sends the next block to the requester and again waits for acknowledgement. The requester also repeats the same cycle of states.

If the server detects a communications failure before sending the last block, it enters a cancellation state 1912 wherein the transaction is cancelled. Similarly, if the requester detects a communications failure before receiving the last block it enters a cancellation state 1913.

If there are no more blocks to send, the server commits to the transaction and waits for the final Acknowledgement in state 1914. If there is a communications failure before the server receives the final Acknowledgement message, it still commits to the transaction but includes a report about the event to its credit server in state 1915. This report serves two purposes. It will help legitimize any claims by a user of having been billed for receiving digital works that were not completely received. Also it helps to identify repositories and communications lines that have suspicious patterns of use and interruption. The server then enters its completion state 1916.

On the requester side, when there are no more blocks to receive, the requester commits to the transaction in state 1917. If the requester detects a communications failure at this state, it reports the failure to its credit server in state 1918, but still commits to the transaction. When it has committed, it sends an acknowledgement message to the server. The server then enters its completion state 1919

The key property is that both the server and the requester cancel a transaction if it is interrupted before all of the data blocks are delivered, and commits to it if all of the data blocks have been delivered.

There is a possibility that the server will have sent all of the data blocks (and committed) but the requester will not have received all of them and will cancel the transaction. In this case. both repositories will presumably detect a communications failure and report it to their credit server. This case will probably be rare since it depends on very precise timing of the communications failure. The only consequence will be that the user at the requester repository may want to request a refund from the credit services -- and the case for that refund will be documented by reports by both repositories.

To prevent loss of data, the server should not delete any transferred digital work until receiving the final acknowledgement from the requester. But it also should not use the file. A well known way to deal with this situation is called "two-phase commit" or 2PC.

Two-phase commit works as follows. The first phase works the same as the method described above. The server sends all of the data to the requester. Both repositories mark the transaction (and appropriate files) as uncommitted. The server sends a ready-to-commit message to the requester. The requester sends back an acknowledgement. The server then commits and sends the requester a commit message. When the requester receives the commit message, it commits the file.

If there is a communication failure or other crash, the requester must check back with the server to determine the status of the transaction. The server has the last word on this. The requester may have received all of the data, but if it did not get the final message, it has not committed. The server can go ahead and delete files (except for transaction records) once it commits, since the files are known to have been fully transmitted before starting the 2PC cycle.

There are variations known in the art which can be used to achieve the same effect. For example, the server could use an additional level of encryption when transmitting a work to a client. Only after the client sends a message acknowledging receipt does it send the key. The client then agrees to pay for the digital work. The point of this variation is that it provides a clear audit trail that the client received the work. For trusted systems, however, this variation adds a level of encryption for no real gain in accountability.

The transaction for specific usage rights are now discussed.

The Copy Transaction

A Copy transaction is a request to make one or more independent copies of the work with the same or lesser usage rights. Copy differs from the extraction right discussed later in that it refers to entire digital works or entire folders containing digital works. A copy operation cannot be used to remove a portion of a digital work.

- The requester sends the server a message to initiate the Copy Transaction. This message indicates the work to be copied, the version of the copy right to be used for the transaction, the destination address information (location in a folder) for placing the work, the file data for the work (including its size), and the number of copies requested.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the client according to the transmission protocol. If a Next-Set-Of-Rights has been provided in the version of the right, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted. In any event, the Copy-Count field for the copy of the digital work being sent right is set to the number-of-copies requested.
- The requester records the work contents, data, and usage rights and stores the work. It records the date and time that the copy was made in the properties of the digital work.
- The repositories perform the common closing transaction steps.

The Transfer Transaction

A Transfer transaction is a request to move copies of the work with the same or lesser usage rights to another repository. In contrast with a copy transaction, this results in removing the work copies from the server.

- The requester sends the server a message to initiate the Transfer Transaction. This message indicates the work to be transferred, the version of the transfer right to be used in the transaction, the destination address information for placing the work, the file data for the work, and the number of copies involved.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted. In either case, the Copy-Count field for the transmitted rights are set to the number-of-copies requested.

- The requester records the work contents, data, and usage rights and stores the work.
- The server decrements its copy count by the number of copies involved in the transaction.
- The repositories perform the common closing transaction steps.
- If the number of copies remaining in the server is now zero, it erases the digital work from its memory.

The Loan Transaction

A loan transaction is a mechanism for loaning copies of a digital work. The maximum duration of the loan is determined by an internal parameter of the digital work. Works are automatically returned after a predetermined time period.

- The requester sends the server a message to initiate the Transfer Transaction. This message indicates the work to be loaned, the version of the loan right to be used in the transaction, the destination address information for placing the work, the number of copies involved, the file data for the work, and the period of the loan.
- The server checks the validity of the requested loan period, and ends with an error if the period is not valid. Loans for a loaned copy cannot extend beyond the period of the original loan to the server.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted, as modified to reflect the loan period.
- The requester records the digital work contents, data, usage rights, and loan period and stores the work.
- The server updates the usage rights information in the digital work to reflect the number of copies loaned out.
- The repositories perform the common closing transaction steps.
- The server updates the usage rights data for the digital work. This may preclude use of the work until it is returned from the loan. The user on the requester platform can now use the transferred copies of the digital work. A user accessing the original repository cannot use the digital work , unless there are copies remaining. What happens next depends on the order of events in time.

Case 1. If the time of the loan period is not yet exhausted and the requester sends the repository a Return message.

- The return message includes the requester identification, and the transaction ID.
- The server decrements the copies-in-use field by the number of copies that were returned. (If the number of digital works returned is greater than the number actually borrowed, this is treated as an error.) This step may now make the work available at the server for other users.
- The requester deactivates its copies and removes the contents from its memory.

Case 2. If the time of the loan period is exhausted and the requester has not yet sent a Return message.

- The server decrements the copies-in-use field by the number digital works that were borrowed.
- The requester automatically deactivates its copies of the digital work. It terminates all current uses and erases the digital work copies from memory. One question is why a requester would ever return a work earlier than the period of the loan, since it would be returned automatically anyway. One reason for early return is that there may be a metered fee which determines the cost of the loan. Returning early may reduce that fee.

The Play Transaction

A play transaction is a request to use the contents of a work. Typically, to "play" a work is to send the digital work through some kind of transducer, such as a speaker or a display device. The request implies the intention that the contents will not be communicated digitally to any other system. For example, they will not be sent to a printer, recorded on any digital medium, retained after the transaction or sent to another repository.

This term "play" is natural for examples like playing music, playing a movie, or playing a video game. The general form of play means that a "player" is used to use the digital work. However, the term play covers all media and kinds of recordings. Thus one would "play" a digital work, meaning, to render it for reading, or play a computer program, meaning to execute it. For a digital ticket the player would be a digital ticket agent.

- The requester sends the server a message to initiate the play transaction. This message indicates the work to be played, the version of the play right to be used in the transaction, the identity of the player being used, and the file data for the work.

- The server checks the validity of the player identification and the compatibility of the player identification with the player specification in the right. It ends with an error if these are not satisfactory.
- The repositories perform the common opening transaction steps.
- The server and requester read and write the blocks of data as requested by the player according to the transmission protocol. The requester plays the work contents, using the player.
- When the player is finished, the player and the requester remove the contents from their memory.
- The repositories perform the common closing transaction steps.

The Print Transaction

A Print transaction is a request to obtain the contents of a work for the purpose of rendering them on a "printer." We use the term "printer" to include the common case of writing with ink on paper. However, the key aspect of "printing" in our use of the term is that it makes a copy of the digital work in a place outside of the protection of usage rights. As with all rights, this may require particular authorization certificates.

Once a digital work is printed, the publisher and user are bound by whatever copyright laws are in effect. However, printing moves the contents outside the control of repositories. For example, absent any other enforcement mechanisms, once a digital work is printed on paper, it can be copied on ordinary photocopying machines without intervention by a repository to collect usage fees. If the printer to a digital disk is permitted, then that digital copy is outside of the control of usage rights. Both the creator and the user know this, although the creator does not necessarily give tacit consent to such copying, which may violate copyright laws.

- The requester sends the server a message to initiate a Print transaction. This message indicates the work to be played, the identity of the printer being used, the file data for the work, and the number of copies in the request.
- The server checks the validity of the printer identification and the compatibility of the printer identification with the printer specification in the right. It ends with an error if these are not satisfactory.
- The repositories perform the common opening transaction steps.
- The server transmits blocks of data according to the transmission protocol.
- The requester prints the work contents, using the printer.
- When the printer is finished, the printer and the requester remove the contents from their memory.
- The repositories perform the common closing transaction steps.

The Backup Transaction

A Backup transaction is a request to make a backup copy of a digital work, as a protection against media failure. In the context of repositories, secure backup copies differ from other copies in three ways: (1) they are made under the control of a Backup transaction rather than a Copy transaction, (2) they do not count as regular copies, and (3) they are not usable as regular copies. Generally, backup copies are encrypted.

Although backup copies may be transferred or copied, depending on their assigned rights, the only way to make them useful for playing, printing or embedding is to restore them.

The output of a Backup operation is both an encrypted data file that contains the contents and description of a work, and a restoration file with an encryption key for restoring the encrypted contents. In many cases, the encrypted data file would have rights for "printing" it to a disk outside of the protection system, relying just on its encryption for security. Such files could be stored anywhere that was physically safe and convenient. The restoration file would be held in the repository. This file is necessary for the restoration of a backup copy. It may have rights for transfer between repositories.

- The requester sends the server a message to initiate a backup transaction. This message indicates the work to be backed up, the version of the backup right to be used in the transaction, the destination address information for placing the backup copy, the file data for the work.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, a set of default rights for backup files of the original are transmitted by the server.
- The requester records the work contents, data, and usage rights. It then creates a one-time key and encrypts the contents file. It saves the key information in a restoration file.
- The repositories perform the common closing transaction steps.

In some cases, it is convenient to be able to archive the large, encrypted contents file to secure offline storage,

such as a magneto-optical storage system or magnetic tape. This creation of a non-repository archive file is as secure as the encryption process. Such non-repository archive storage is considered a form of "printing" and is controlled by a print right with a specified "archive-printer." An archive-printer device is programmed to save the encrypted contents file (but not the description file) offline in such a way that it can be retrieved.

The Restore Transaction

A Restore transaction is a request to convert an encrypted backup copy of a digital work into a usable copy. A restore operation is intended to be used to compensate for catastrophic media failure. Like all usage rights, restoration rights can include fees and access tests including authorization checks.

- The requester sends the server a message to initiate a Restore transaction. This message indicates the work to be restored, the version of the restore right for the transaction, the destination address information for placing the work, and the file data for the work.
- The server verifies that the contents file is available (i.e. a digital work corresponding to the request has been backed-up.) If it is not, it ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The server retrieves the key from the restoration file. It decrypts the work contents, data, and usage rights.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, a set of default rights for backup files of the original are transmitted by the server.
- The requester stores the digital work.
- The repositories perform the common closing transaction steps.

The Delete Transaction

A Delete transaction deletes a digital work or a number of copies of a digital work from a repository. Practically all digital works would have delete rights.

- The requester sends the server a message to initiate a delete transaction. This message indicates the work to be deleted, the version of the delete right for the transaction.
- The repositories perform the common opening transaction steps.
- The server deletes the file, erasing it from the file system.
- The repositories perform the common closing transaction steps.

The Directory Transaction

A Directory transaction is a request for information about folders, digital works, and their parts. This amounts to roughly the same idea as protection codes in a conventional file system like TENEX, except that it is generalized to the full power of the access specifications of the usage rights language.

The Directory transaction has the important role of passing along descriptions of the rights and fees associated with a digital work. When a user wants to exercise a right, the user interface of his repository implicitly makes a directory request to determine the versions of the right that are available. Typically these are presented to the user -- such as with different choices of billing for exercising a right. Thus, many directory transactions are invisible to the user and are exercised as part of the normal process of exercising all rights.

- The requester sends the server a message to initiate a Directory transaction. This message indicates the file or folder that is the root of the directory request and the version of the directory right used for the transaction.
- The server verifies that the information is accessible to the requester. In particular, it does not return the names of any files that have a HIDE-NAME status in their directory specifications, and it does not return the parts of any folders or files that have HIDE-PARTS in their specification. If the information is not accessible, the server ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The server sends the requested data to the requester according to the transmission protocol.
- The requester records the data.
- The repositories perform the common closing transaction steps.

The Folder Transaction

A Folder transaction is a request to create or rename a folder, or to move a work between folders. Together with Directory rights. Folder rights control the degree to which organization of a repository can be accessed or modified from another repository.

- The requester sends the server a message to initiate a Folder transaction. This message indicates the folder that is the root of the folder request, the version of the folder right for the transaction, an operation, and data. The operation can be one of create, rename, and move file. The data are the specifications required for the operation, such as a specification of a folder or digital work and a name.
- The repositories perform the common opening transaction steps.
- The server performs the requested operation -- creating a folder, renaming a folder, or moving a work between folders.
- The repositories perform the common closing transaction steps.

The Extract Transaction

A extract transaction is a request to copy a part of a digital work and to create a new work containing it. The extraction operation differs from copying in that it can be used to separate a part of a digital work from d-blocks or shells that place additional restrictions or fees on it. The extraction operation differs from the edit operation in that it does not change the contents of a work, only its embedding in d-blocks. Extraction creates a new digital work.

- The requester sends the server a message to initiate an Extract transaction. This message indicates the part of the work to be extracted, the version of the extract right to be used in the transaction, the destination address information for placing the part as a new work, the file data for the work, and the number of copies involved.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the new work. Otherwise, the rights of the original are transmitted. The Copy-Count field for this right is set to the number-of-copies requested.
- The requester records the contents, data, and usage rights and stores the work. It records the date and time that new work was made in the properties of the work.
- The repositories perform the common closing transaction steps.

The Embed Transaction

An embed transaction is a request to make a digital work become a part of another digital work or to add a shell d-block to enable the adding of fees by a distributor of the work.

- The requester sends the server a message to initiate an Embed transaction. This message indicates the work to be embedded, the version of the embed right to be used in the transaction, the destination address information for placing the part as a a work, the file data for the work, and the number of copies involved.
- The server checks the control specifications for all of the rights in the part and the destination. If they are incompatible, the server ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the new work. Otherwise, the rights of the original are transmitted. The Copy-Count field for this right is set to the number-of-copies requested.
- The requester records the contents, data, and usage rights and embeds the work in the destination file.
- The repositories perform the common closing transaction steps.

The Edit Transaction

An Edit transaction is a request to make a new digital work by copying, selecting and modifying portions of an existing digital work. This operation can actually change the contents of a digital work. The kinds of changes that are permitted depend on the process being used. Like the extraction operation, edit operates on portions of a digital work. In contrast with the extract operation, edit does not affect the rights or location of the work. It only changes the contents. The kinds of changes permitted are determined by the type specification of the processor specified in the rights. In the currently preferred embodiment, an edit transaction changes the work itself and does not make a new work. However,

it would be a reasonable variation to cause a new copy of the work to be made.

- The requester sends the server a message to initiate an Edit transaction. This message indicates the work to be edited, the version of the edit right to be used in the transaction, the file data for the work (including its size), the process-ID for the process, and the number of copies involved.
- The server checks the compatibility of the process-ID to be used by the requester against any process-ID specification in the right. If they are incompatible, it ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The requester uses the process to change the contents of the digital work as desired. (For example, it can select and duplicate parts of it; combine it with other information; or compute functions based on the information. This can amount to editing text, music, or pictures or taking whatever other steps are useful in creating a derivative work.)
- The repositories perform the common closing transaction steps.

The edit transaction is used to cover a wide range of kinds of works. The category describes a process that takes as its input any portion of a digital work and then modifies the input in some way. For example, for text, a process for editing the text would require edit rights. A process for "summarizing" or counting words in the text would also be considered editing. For a music file, processing could involve changing the pitch or tempo, or adding reverberations, or any other audio effect. For digital video works, anything which alters the image would require edit rights. Examples would be colorizing, scaling, extracting still photos, selecting and combining frames into story boards, sharpening with signal processing, and so on.

Some creators may want to protect the authenticity of their works by limiting the kinds of processes that can be performed on them. If there are no edit rights, then no processing is allowed at all. A processor identifier can be included to specify what kind of process is allowed. If no process identifier is specified, then arbitrary processors can be used. For an example of a specific process, a photographer may want to allow use of his photograph but may not want it to be colorized. A musician may want to allow extraction of portions of his work but not changing of the tonality.

Authorization Transactions

There are many ways that authorization transactions can be defined. In the following, our preferred way is to simply define them in terms of other transactions that we already need for repositories. Thus, it is convenient sometimes to speak of "authorization transactions," but they are actually made up of other transactions that repositories already have.

A usage right can specify an authorization-ID, which identifies an authorization object (a digital work in a file of a standard format) that the repository must have and which it must process. The authorization is given to the generic authorization (or ticket) server of the repository which begins to interpret the authorization.

As described earlier, the authorization contains a server identifier, which may just be the generic authorization server or it may be another server. When a remote authorization server is required, it must contain a digital address. It may also contain a digital certificate.

If a remote authorization server is required, then the authorization process first performs the following steps:

- The generic authorization server attempts to set up the communications channel. (If the channel cannot be set up, then authorization fails with an error.)
- When the channel is set up, it performs a registration process with the remote repository. (If registration fails, then the authorization fails with an error.)
- When registration is complete, the generic authorization server invokes a "Play" transaction with the remote repository, supplying the authorization document as the digital work to be played, and the remote authorization server (a program) as the "player." (If the player cannot be found or has some other error, then the authorization fails with an error.)
- The authorization server then "plays" the authorization. This involves decrypting it using either the public key of the master repository that issued the certificate or the session key from the repository that transmitted it. The authorization server then performs various tests. These tests vary according to the authorization server. They include such steps as checking issue and validity dates of the authorization and checking any hot-lists of known invalid authorizations. The authorization server may require carrying out any other transactions on the repository as well, such as checking directories, getting some person to supply a password, or playing some other digital work. It may also invoke some special process for checking information about locations or recent events. The "script" for such steps is contained within the authorization server.
- If all of the required steps are completed satisfactorily, the authorization server completes the transaction normally, signaling that authorization is granted.

The Install Transaction

An Install transaction is a request to install a digital work as runnable software on a repository. In a typical case, the requester repository is a rendering repository and the software would be a new kind or new version of a player. Also in a typical case, the software would be copied to file system of the requester repository before it is installed.

- The requester sends the server an Install message. This message indicates the work to be installed, the version of the Install right being invoked, and the file data for the work (including its size).
- The repositories perform the common opening transaction steps.
- The requester extracts a copy of the digital certificate for the software. If the certificate cannot be found or the master repository for the certificate is not known to the requester, the transaction ends with an error.
- The requester decrypts the digital certificate using the public key of the master repository, recording the identity of the supplier and creator, a key for decrypting the software, the compatibility information, and a tamper-checking code. (This step certifies the software.)
- The requester decrypts the software using the key from the certificate and computes a check code on it using a 1-way hash function. If the check-code does not match the tamper-checking code from the certificate, the installation transaction ends with an error. (This step assures that the contents of the software, including the various scripts, have not been tampered with.)
- The requester retrieves the instructions in the compatibility-checking script and follows them. If the software is not compatible with the repository, the installation transaction ends with an error. (This step checks platform compatibility.)
- The requester retrieves the instructions in the installation script and follows them. If there is an error in this process (such as insufficient resources), then the transaction ends with an error. Note that the installation process puts the runnable software in a place in the repository where it is no longer accessible as a work for exercising any usage rights other than the execution of the software as part of repository operations in carrying out other transactions.
- The repositories perform the common closing transaction steps.

The Uninstall Transaction

An Uninstall transaction is a request to remove software from a repository. Since uncontrolled or incorrect removal of software from a repository could compromise its behavioral integrity, this step is controlled.

- The requester sends the server an Uninstall message. This message indicates the work to be uninstalled, the version of the Uninstall right being invoked, and the file data for the work (including its size).
- The repositories perform the common opening transaction steps.
- The requester extracts a copy of the digital certificate for the software. If the certificate cannot be found or the master repository for the certificate is not known to the requester, the transaction ends with an error.
- The requester checks whether the software is installed. If the software is not installed, the transaction ends with an error.
- The requester decrypts the digital certificate using the public key of the master repository, recording the identity of the supplier and creator, a key for decrypting the software, the compatibility information, and a tamper-checking code. (This step authenticates the certification of the software, including the script for uninstalling it.)
- The requester decrypts the software using the key from the certificate and computes a check code on it using a 1-way hash function. If the check-code does not match the tamper-checking code from the certificate, the installation transaction ends with an error. (This step assures that the contents of the software, including the various scripts, have not been tampered with.)
- The requester retrieves the instructions in the uninstallation script and follows them. If there is an error in this process (such as insufficient resources), then the transaction ends with an error.
- The repositories perform the common closing transaction steps.

Claims

1. A system for controlling the distribution and use of digital works having a mechanism for reporting fees based on the distribution and use of digital works, said system comprising:

   means for attaching usage rights to a digital work, each of said usage rights specifying how a digital work may be used or distributed, each of said usage rights specifying usage fee information, said usage fee information

comprising a fee type and fee parameters which define a fee to be paid in connection with the exercise of said usage right;
a communication medium for coupling repositories to enable communication between repositories; and
a plurality of repositories, each of said repositories comprising:
an external interface for removably coupling to said communications medium;
storage means for storing digital works having attached usage rights and fees;
requesting means for generating a request to access a digital work stored in another of said plurality of repositories, said request indicating a particular usage right; and
processing means for processing requests to access digital works stored in said storage means and for generating fee transactions when a request indicates a usage right that is attached to a digital work and said usage right specifies usage fee information;
each of said plurality of repositories being removably coupled to a credit server, said credit server being arranged for recording fee transactions from said repository and subsequently reporting said fee transactions to a billing clearinghouse.

2. The fee reporting system as recited in Claim 1 wherein said fee type of said fee information is a metered use fee, a per use fee, a best price fee, a scheduled fee,or a mark-up fee.

3. A method for reporting fees associated with the distribution and use of digital works in a system for controlling the distribution and use of digital works, said method comprising the steps of:

a) attaching one or more usage rights to a digital work, each of said one or more usage rights comprising an indicator of how said digital work may be distributed or used and a usage fee to be paid upon exercise of said right;
b) storing said digital work and attached one or more usage rights in a server repository, said server repository controlling access to said digital work;
c) said server repository receiving a request to access said digital work from a requesting repository;
d) said server repository identifying a usage right associated with said access request;
e) said server repository determining if said identified usage right is the same as one of said one or more usage rights attached to said digital work;
f) if said identified usage right is not the same as any one of said one or more usage rights attached to said digital work, said server repository denying access to said digital work;
g) if said usage right is included with said digital work, said server repository determining if a usage fee is associated with the exercise of said usage right;
h) if a usage fee is associated with usage right, said server repository calculating said usage fee;
i) said server repository transmitting a first assign fee transaction identifying said requesting repository as a payer for said usage fee to a first credit server;
j) said requesting repository transmitting a second assign fee transaction identifying said requesting repository as a payer for said usage fee to a second credit server;
k) said server repository transmitting said digital work to said requesting repository;
l) said server repository transmitting a first confirm fee transaction to said first credit server; and
m) said requesting repository transmitting a second confirm fee transaction to said second credit server.

4. The method as recited in Claim 3 wherein said digital work is comprised of a plurality of independent digital works and said step of said server calculating said usage fee is further comprised of the step of reporting the usage fees for each of the plurality of independent digital works.

5. A method for reporting fees associated with the distribution and use of digital works in a system for controlling the distribution and use of digital works, said method comprising the steps of:

a) attaching one or more usage rights to a digital work, each of said one or more usage rights comprising an indicator of how said digital work may be distributed or used and a usage fee to be paid for exercise of said right;
b) storing said digital work and said attached one or more usage rights in a server repository, said server repository controlling access to said digital work;
c) said server repository receiving a request to access said digital work from a requesting repository;
d) said server repository identifying a usage right associated with said access request;
e) said server repository determining if said digital work has attached thereto said identified usage right;
f) if said identified usage right is not attached to said digital work, said server repository denying access to

said digital work;
g) if said usage right is attached to said digital work, said server repository determining if a usage fee is associated with the exercise of said usage right;
h) if a usage fee is associated with said usage right, said server repository determining a fee type;
i) said server repository transmitting a first fee transaction identifying said requesting repository as a payee for said usage fee to a credit server, said first fee transaction being dependent on said determined fee type; and
k) said server repository transmitting said digital work to said requesting repository.

6. A system for controlling the distribution and utilization of digital works having a mechanism for reporting usage fees, said system comprising:

digital works comprising a first part for storing the digitally encoded data corresponding to a digital work and a second part for storing usage rights and fees for said digital work, said usage rights specifying how a digital work may be used or distributed and said usage fees specifying a fee to be paid in connection with the exercise of a corresponding usage right;
a plurality of repositories, each of said repositories comprising:
communication means for communicating with another of said plurality of repositories;
storage means for storing digital works;
requesting means for generating a request to access a digital work stored in another of said plurality of repositories, said request indicating a particular usage right;
processing means for processing requests to access digital works stored in said storage means and granting access when said particular usage right corresponds to a stored usage right stored in said digital work, said processing means generating fee transactions when said access is granted and said stored usage right specifies a fee;
each of said plurality of repositories being removably coupled to a credit server, said credit server being arranged for recording fee transactions from said repository and subsequently reporting said fee transactions to a billing clearinghouse.

7. The system as recited in Claim 6 wherein said storage means is further comprised of a first storage device for storing said first part of said digital work and a second storage device for storing said second part of said digital work.

8. A method for reporting fees associated with use of rendering digital works by a rendering device in a system for controlling the rendering of digital works by a rendering system, said rendering system comprised of a rendering repository and a rendering device, said rendering device utilizing a rendering digital work for rendering a digital work, said method comprising the steps of:

a) storing a first digital work in a server repository, said digital work specifying a first usage fee to be reported for a use of said first digital work;
b) storing a rendering digital work in said rendering repository, said first rendering digital work specifying a second usage fee to be reported for a use of said rendering digital work;
c) said server repository receiving a request to use said first digital work from said rendering repository;
d) said server repository determining if said request may be granted;
e) if said server repository determines that said request may not be granted, said server repository denying access to said first digital work;
f) if said server repository determines that said request may be granted, said server repository transmitting said digital work to said rendering repository;
g) said server repository transmitting a first fee transaction identifying said rendering repository as a payee for said first usage fee for use of said first digital work to a first credit server;
h) said rendering device rendering said first digital work using said rendering digital work; and
i) said rendering repository transmitting a second fee transaction identifying said rendering repository as a payee for said second usage fee for use of said rendering digital work to a second credit server.

9. The method as recited in Claim 8 further comprising the step of said rendering repository transmitting a third fee transaction identifying said rendering repository as a payee for said first usage fee for use of said first digital work to said second credit server.

10. The method as recited in Claim 9 wherein said rendering digital work is a set of coded rendering instructions for controlling said rendering device.

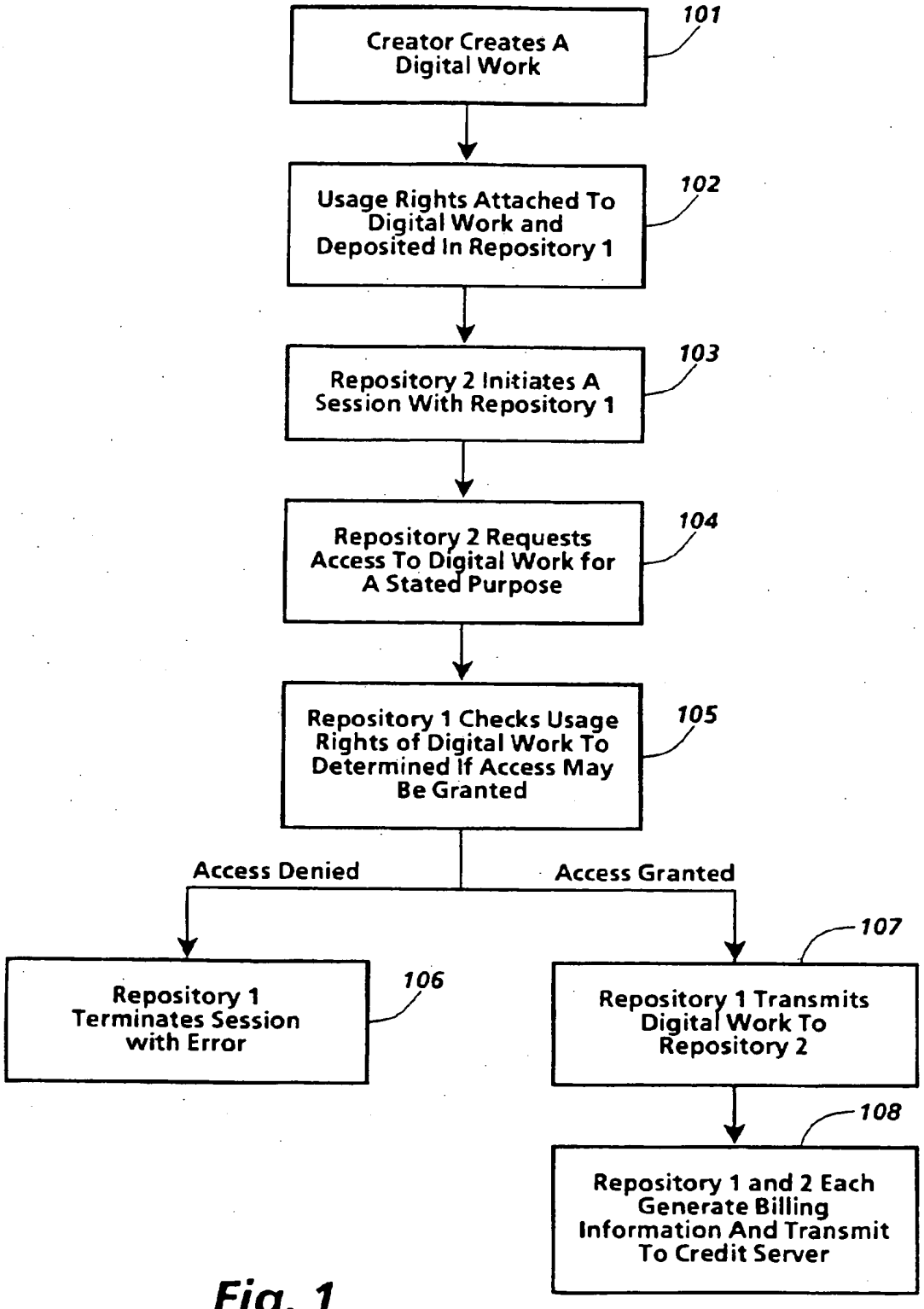


Fig. 1

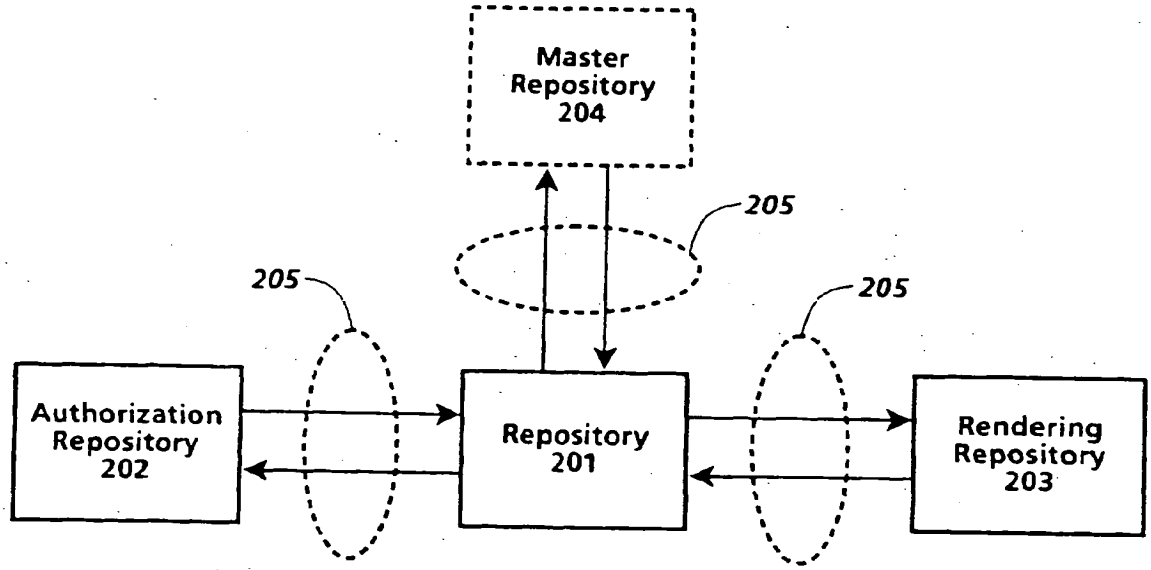


Fig. 2

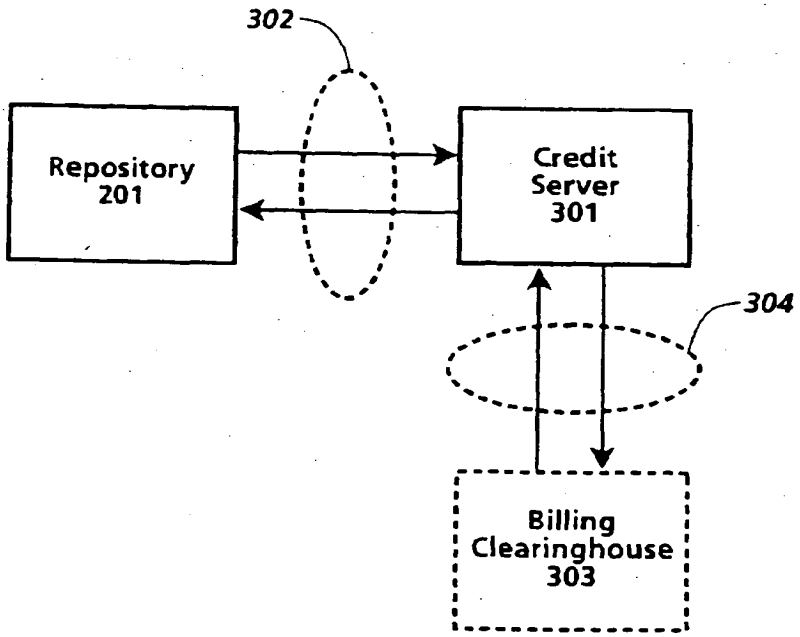


Fig. 3

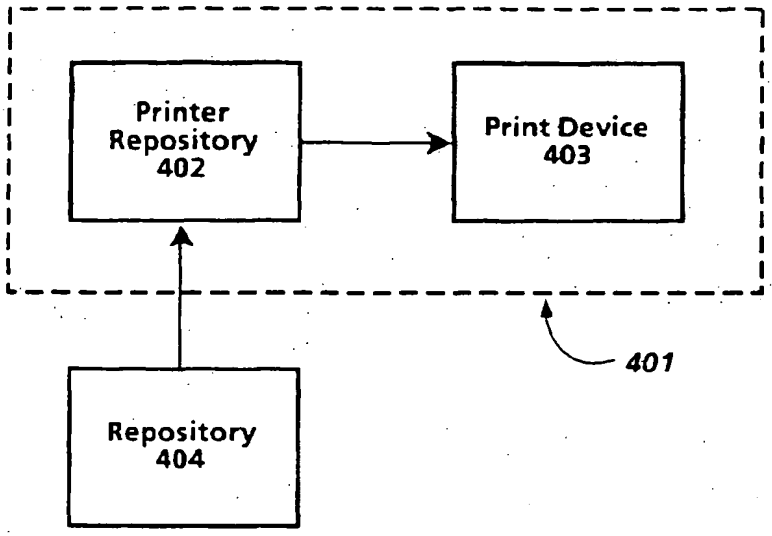


Fig. 4a

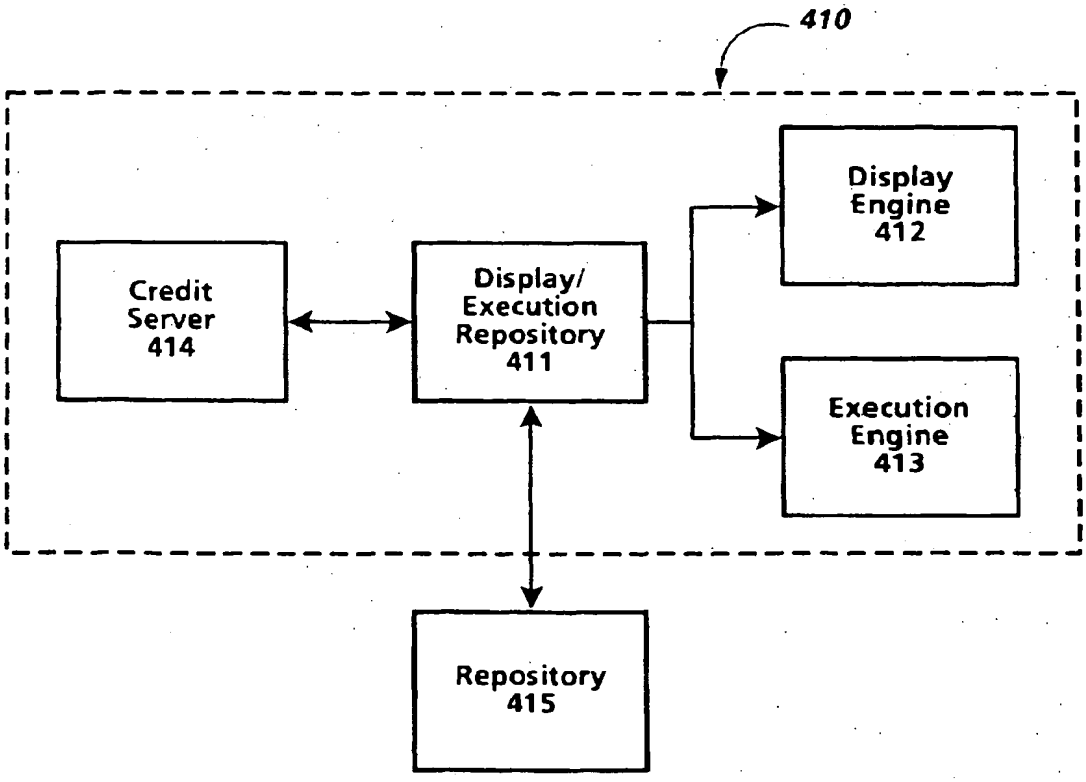


Fig. 4b

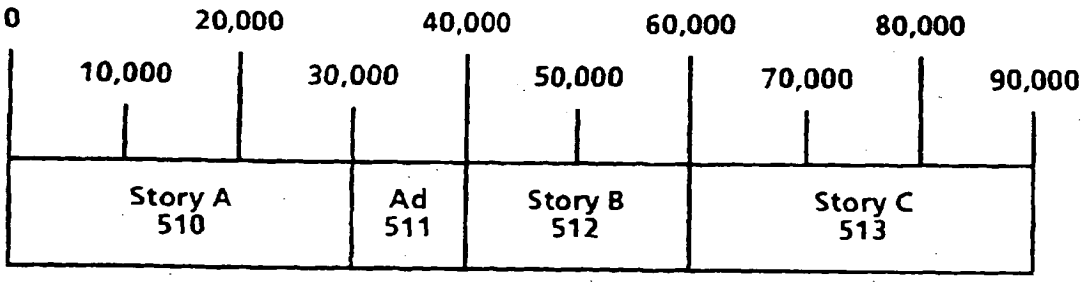


Fig. 5

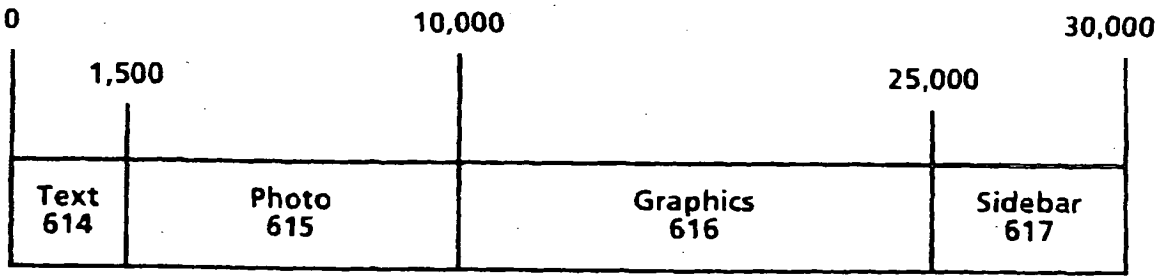


Fig. 6

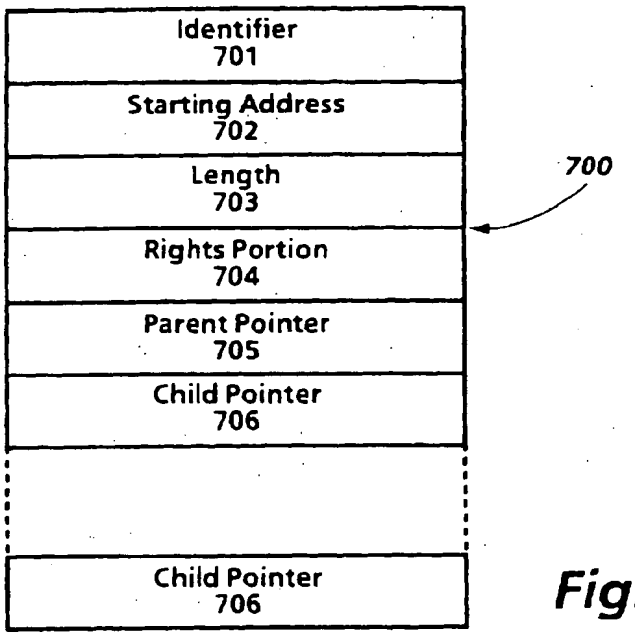


Fig. 7

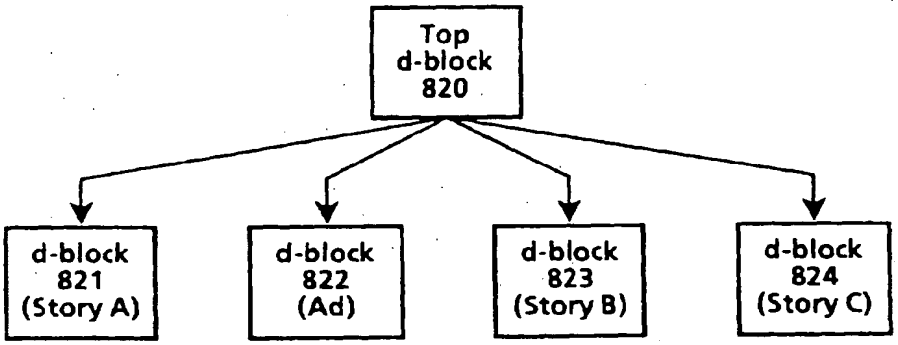


Fig. 8

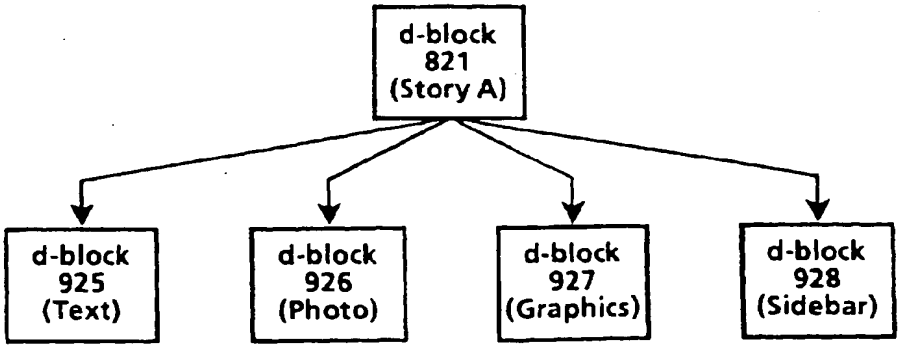


Fig. 9

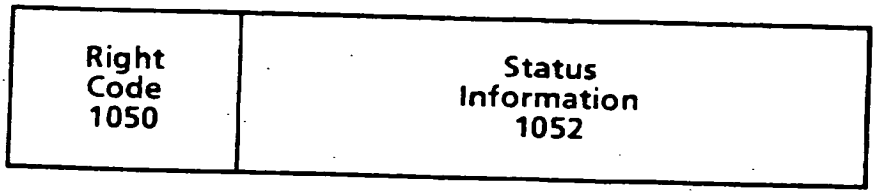


Fig.10

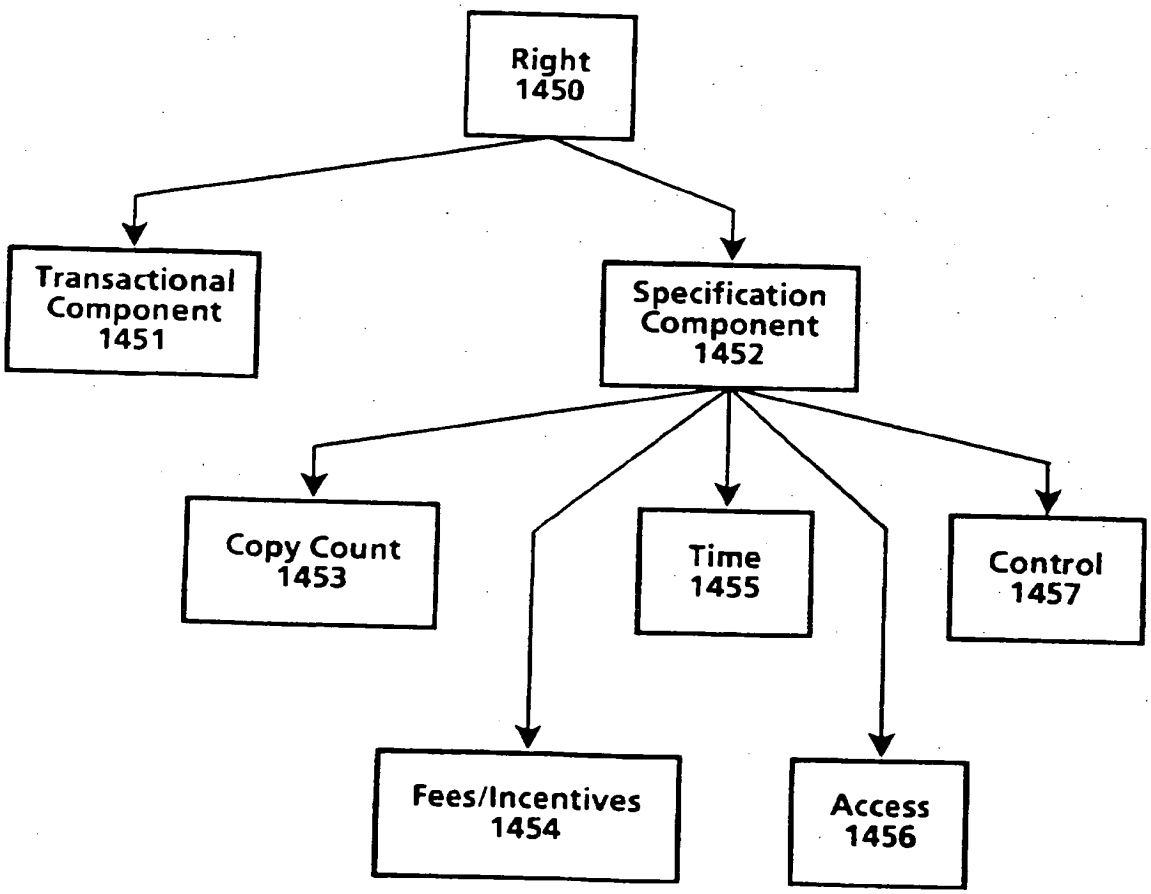


Fig.14

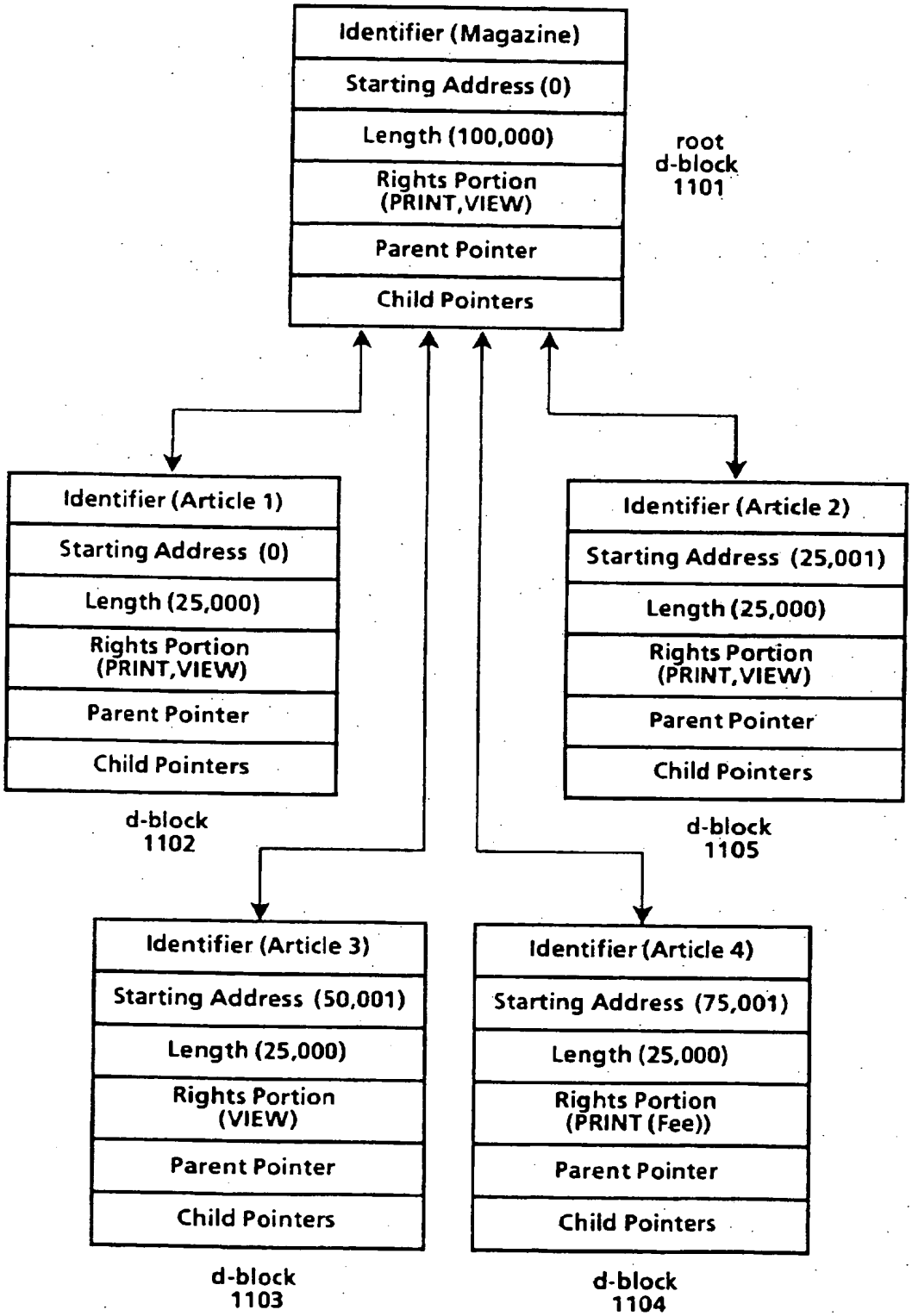


Fig.11

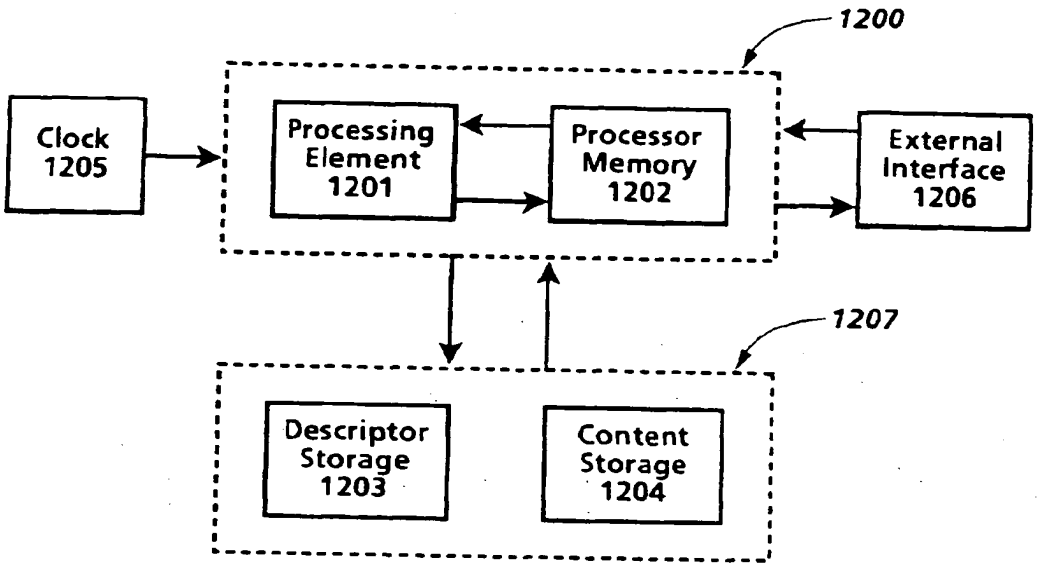


Fig.12

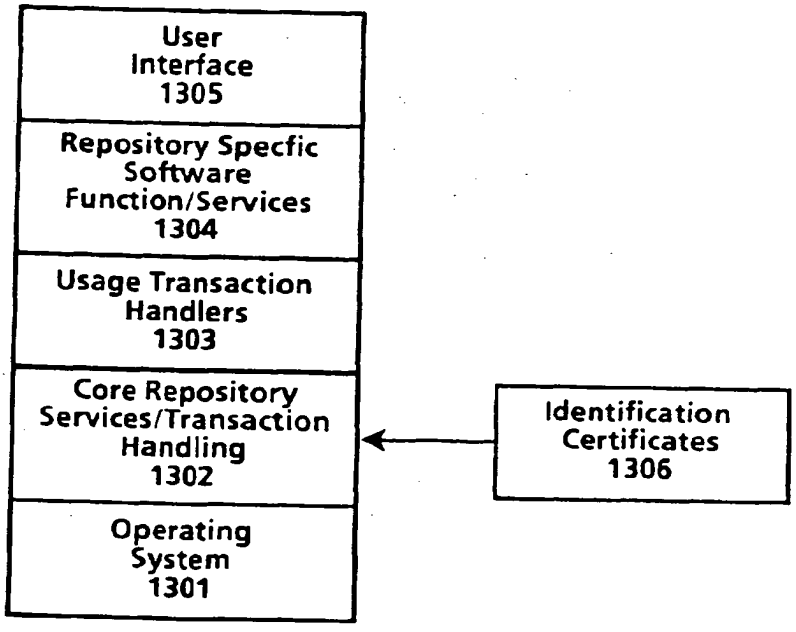


Fig.13

```
1501 ~Digital Work Rights: = (Rights*)
1502 ~Right : = (Right-Code {Copy-Count} {Control-Spec} {Time-Spec }
        {Access-Spec} {Fee-Spec})
   1503 ~Right-Code : = Render-Code | Transport-Code | File-Management-
          Code| Derivative-Works- Code | Configuration-Code
      1504 ~Render-Code : = [ Play : {Player: Player-ID} | Print: {Printer: Printer-ID}]
      1505 ~Transport-Code : = {Copy | Transfer | Loan {Remaining-Rights:
           Next-Set-of-Rights}]{(Next-Copy-Rights: Next-Set-of-Rights)}
      1506 ~File-Management-Code  : = Backup {Back-Up-Copy-Rights:
                                  Next-Set-of-Rights} | Restore | Delete | Folder
                                  | Directory {Name: Hide-Local | Hide-Remote}
                                  {Parts: Hide-Local | Hide-Remote}
      1507 ~Derivative-Works-Code : =  [Extract | Embed | Edit{Process:
                                  Process-ID}] {Next-Copy-Rights :
                                  Next-Set-of Rights}
      1508 ~Configuration-Code : = Install | Uninstall
      1509 ~Next-Set-of-Rights : = {(Add: Set-Of-Rights)} {(Delete:
           Set-Of-Rights)} {(Replace: Set-Of-Rights )}{(Keep: Set-Of-Rights )}
1510 ~Copy-Count : = (Copies:positive-integer | 0 | Unlimited)
1511 ~Control-Spec : = (Control: {Restrictable | Unrestrictable}
                    {Unchargeable | Chargeable})
1512 ~Time-Spec : = ({Fixed-Interval | Sliding-Interval | Meter-Time}
                  Until: Expiration-Date)
   1513 ~Fixed-Interval : = From: Start-Time
   1514 ~Sliding-Interval : = Interval: Use-Duration
   1515 ~Meter-Time: = Time-Remaining: Remaining-Use
1516 ~Access-Spec : = ({SC: Security-Class} {Authorization: Authorization-ID*}
        {Other-Authorization: Authorization-ID*} {Ticket: Ticket-ID})
1517 ~Fee-Spec: = {Scheduled-Discount} Regular-Fee-Spec | Scheduled-Fee-Spec |
                  Markup-Spec
   1518 ~Scheduled-Discount: = Scheduled-Discount: (Scheduled-Discount:
                              (Time-Spec Percentage)*)
   1519 ~Regular-Fee-Spec : = ({Fee: | Incentive: } [Per-Use-Spec | Metered-Rate-
                          Spec | Best-Price-Spec | Call-For-Price-Spec]
                          {Min: Money-Unit Per: Time-Spec}{Max:
                          Money-Unit Per: Time-Spec} To: Account-ID)
      1520 ~Per-Use-Spec: = Per-Use: Money-unit
      1521 ~Metered-Rate-Spec : = Metered: Money-Unit Per: Time-Spec
      1522 ~Best-Price-Spec : = Best-Price: Money-unit Max: Money-unit
      1523 ~Call-For-Price-Spec : = Call-For -Price
   1524 ~Scheduled-Fee-Spec: = (Schedule: (Time-Spec Regular-Fee-Spec)* )
   1525 ~Markup-Spec: = Markup: percentage To: Account-ID
```

Fig.15

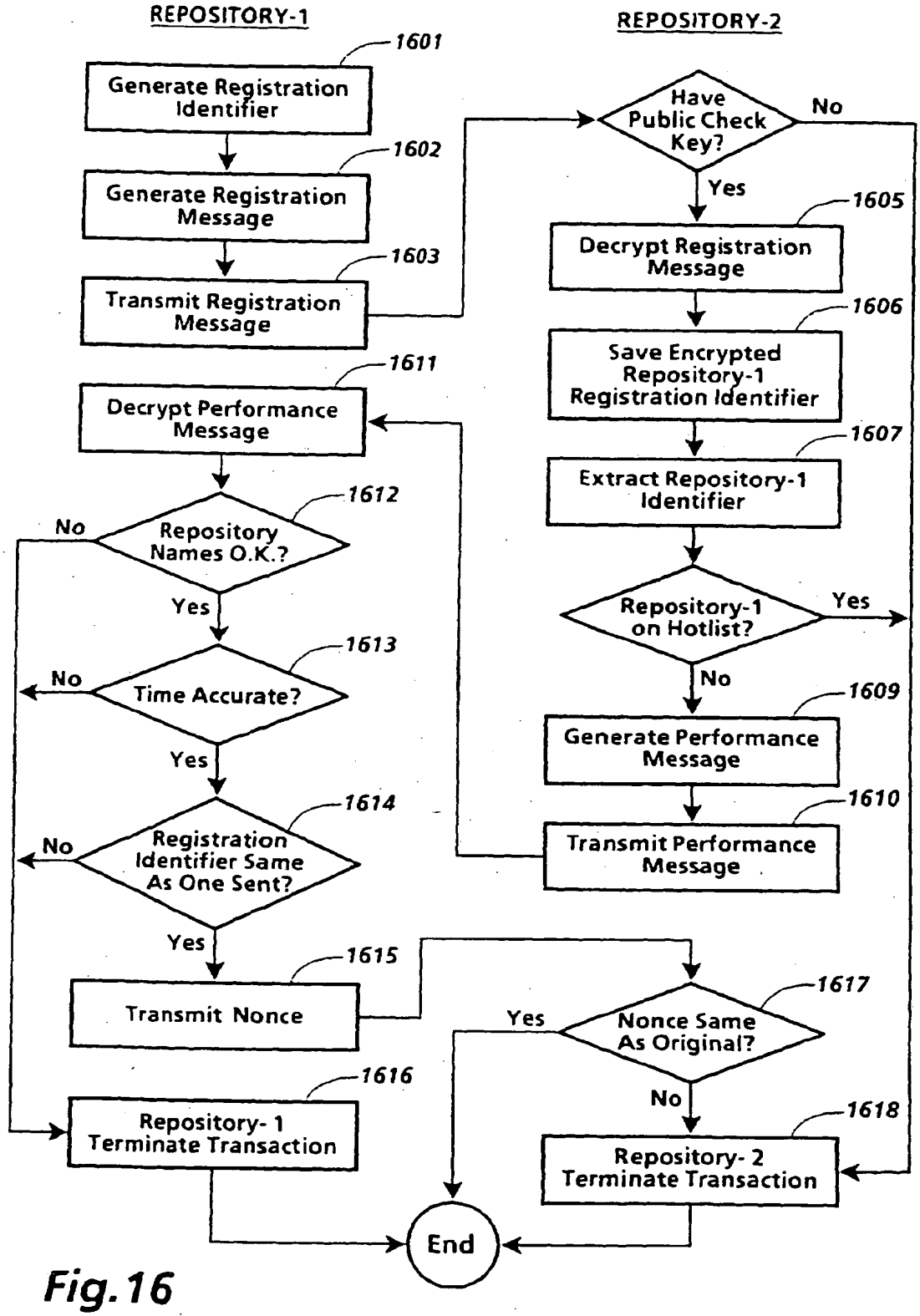


Fig.16

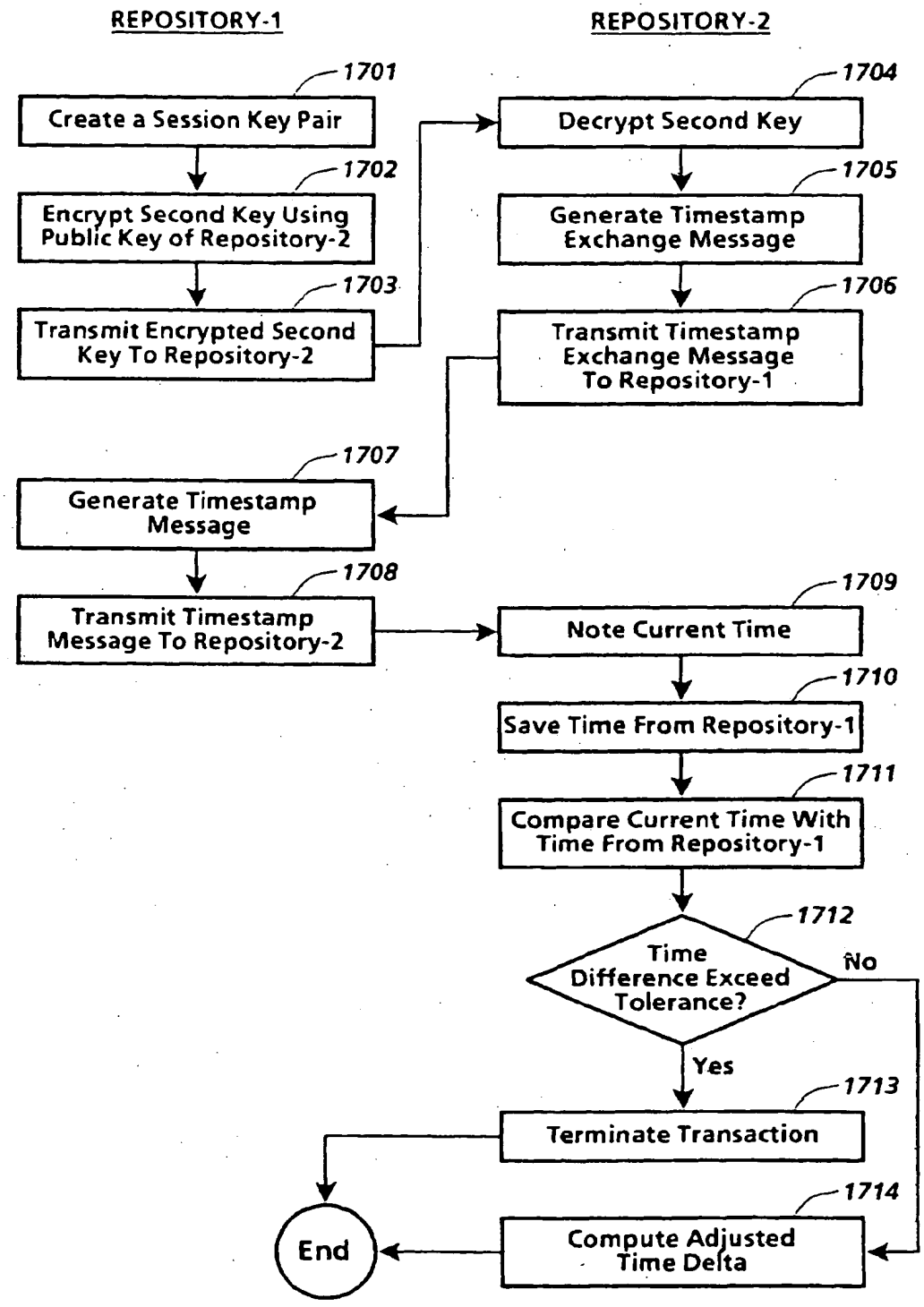


Fig.17

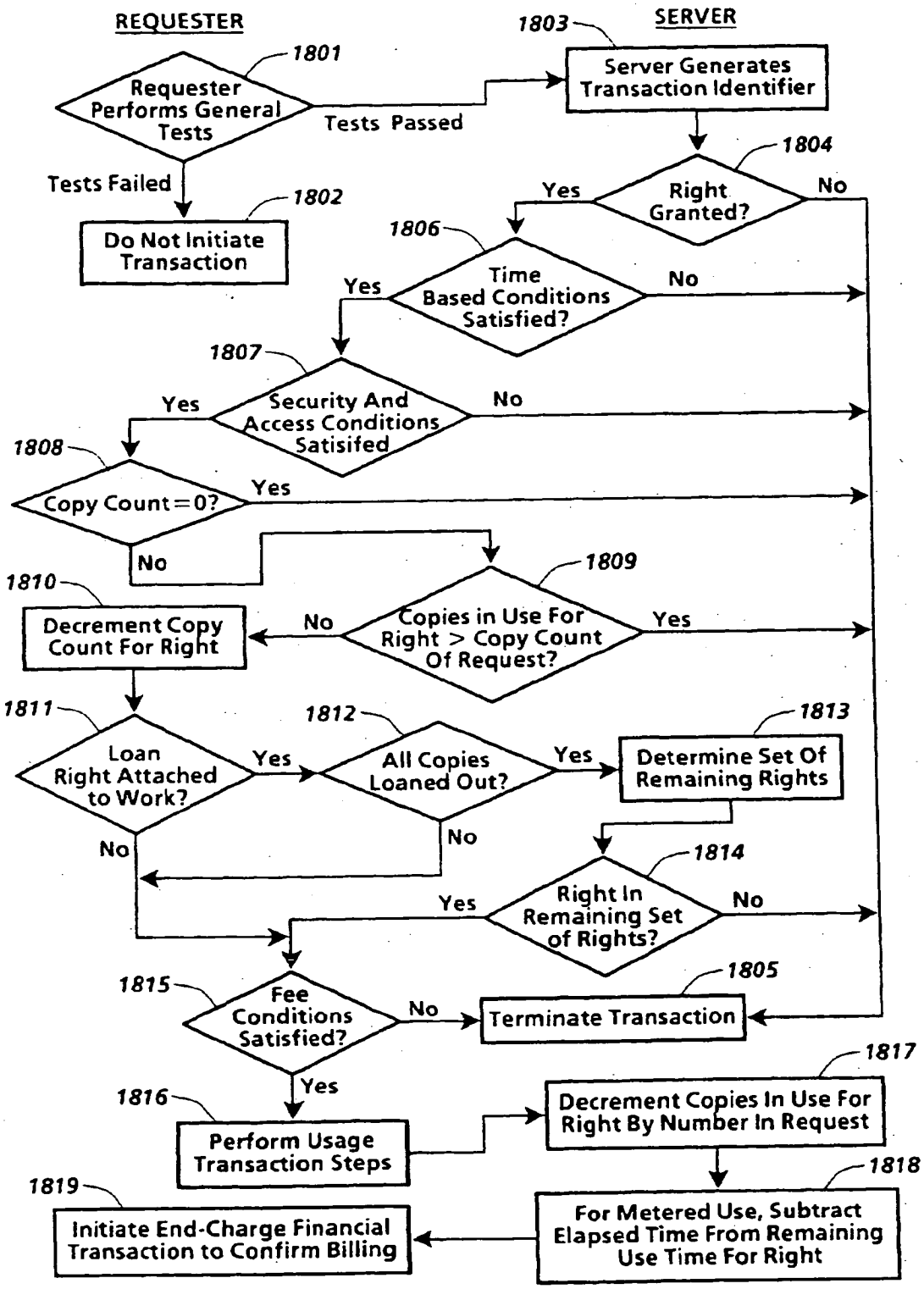


Fig.18

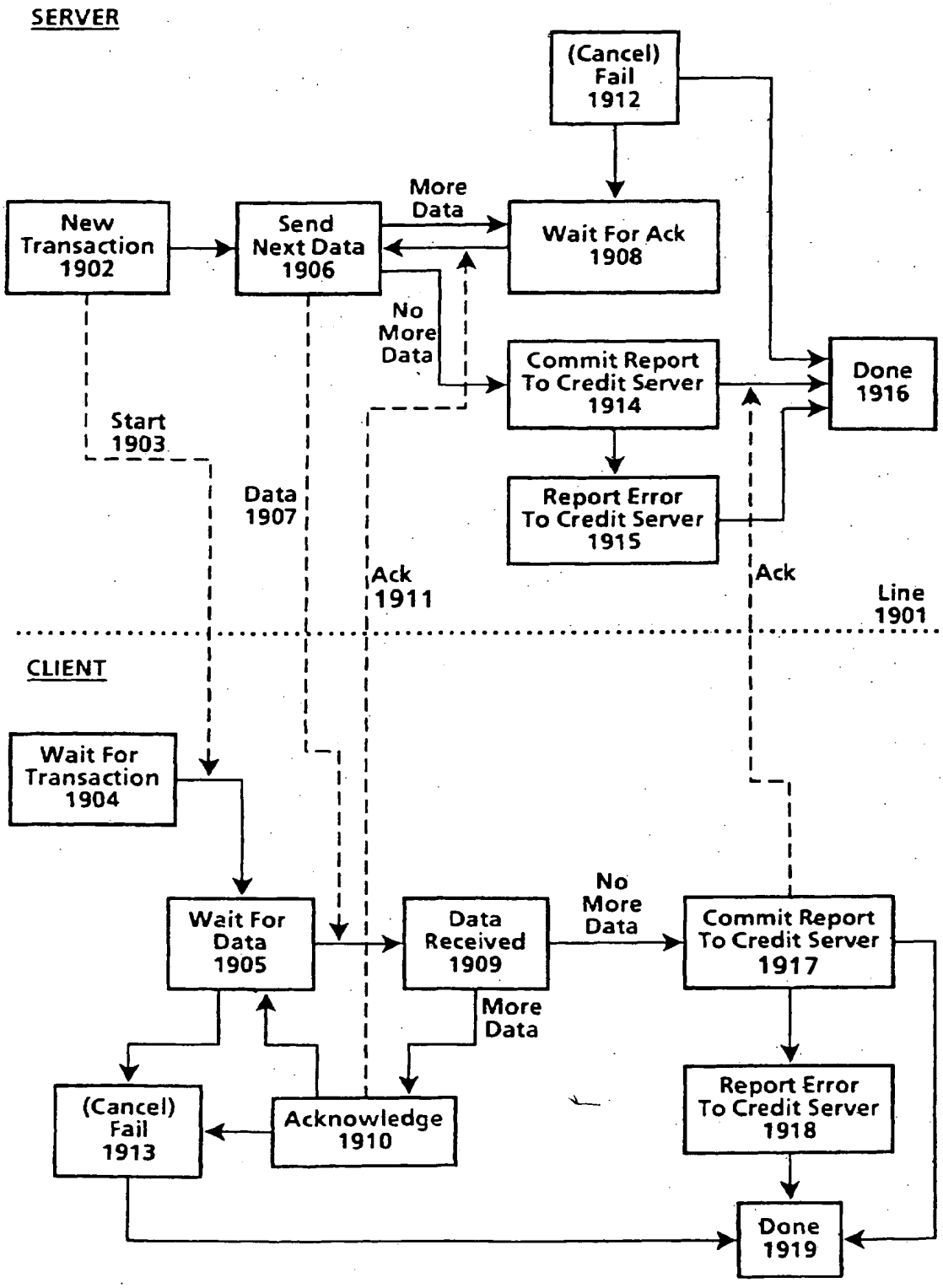


Fig.19

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 95 30 8414

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	WO-A-92 20022 (DIGITAL EQUIPMENT CORP.) * page 45, line 10 - page 64, line 17 *	1,3,5,6,8	G06F1/00 G06F17/60
A	TRANSACTIONS OF THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGINEERS OF JAPAN, vol. E73, no. 7, July 1990 TOKYO JP, pages 1133-1146, XP 000159229 MORI ET AL. 'SUPERDISTRIBUTION: THE CONCEPT AND THE ARCHITECTURE' * page 1135, left column, line 17 - page 1136, left column, line 40 *	1,3,5,6,8	
A	US-A-5 291 596 (MITA) * the whole document *	1,3,5,6,8	
A	GB-A-2 236 604 (SUN MICROSYSTEMS INC) * page 9, line 11 - page 20, line 15 *	1,3,5,6,8	

TECHNICAL FIELDS SEARCHED (Int.Cl.6)

G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	1 April 1996	Moens, R

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

IPW

PATENT
Attorney Docket No.: 111325-104 (230300)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: Xin **WANG**, *et al.*	)	Examiner: Unassigned
Application No.: 10/162,212	)	Group Art Unit: 3621
Filed: June 5, 2002	)	
For: RIGHTS OFFERING AND GRANTING	)	

Commissioner of Patents
U.S. Patent and Trademark Office
220 20th Street S.
Customer Window
Crystal Plaza Two, Lobby, Room 1B03
Arlington, VA 22202

Sir:

INFORMATION DISCLOSURE STATEMENT UNDER 37 C.F.R. § 1.97 (b)

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 listed documents were cited in two communications from the European Patent Office in two counterpart foreign applications. The Search Reports were dated April 26, 2004 and June 11, 2004, which is less than three months ago, therefore no fee or certification is required under 37 C.F.R. § 1.97(b). Enclosed are copies of the European Search Reports.

Enclosed herewith is a Form PTO-1449 listing the art cited in the European Search Report for the above captioned application. It is requested that the accompanying PTO-1449 be considered and made of record in the above-identified 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 (111325-104/230300).

Respectfully submitted,

NIXON PEABODY, LLP

By: _____

Marc S. Kaufman
Registration No.: 35,212

Dated: **July 8, 2004**

NIXON PEABODY LLP
Customer No.: 22204
401 9th Street, N.W., Suite 900
Washington, DC 20004-2128
Telephone: (202) 585-8000
FAX: (202) 585-8080

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(use as many sheets as necessary)

Sheet 1 of 1

Complete if Known	
Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin **WANG**, *et al.*
Art Unit	3621
Examiner Name	Unassigned
Attorney Docket Number	111325-104 (230300)

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	U.S. Patent Document: Number - Kind Code[2] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US – 6,112,239	August 29, 2000	Kenner, et al.	
		US – 5,764,807	June 9, 1998	Pearlman, et al.	
		US – 5,991,306	November 23, 1999	Burns, et al.	
		US – 5,848,154	December 8, 1998	Nishio, et al.	
		US - 4,740,890	April 26, 1988	William	
		US - 5,386,369	January 31, 1995	Christiano	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document: Country Code[3] Number[4] Kind Code[5] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Application of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]
		EP 1 041 823 A2	October 4, 2000			
		WO 00/73922 A2	December 7, 2000			
		WO 01/24530 A2	April 5, 2001			
		EP 0 332 304 A3	September 13, 1989			
		EP 0 731 404 A1	September 9, 1996			
		WO 00/59152	October 5, 2000			
		EP 0 818 748 A2	January 14, 1998			

OTHER PRIOR ART – NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		European Search Report dated June 11, 2004	
		European Search Report dated April 26, 2004	

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference 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 form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

[1] Applicant's unique citation designation number (optional). [2] Applicant is to place a check mark here if English language Translation is attached.

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

(11) EP 0 818 748 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
14.01.1998 Bulletin 1998/03

(51) Int Cl.[6]: **G06F 17/60**

(21) Application number: **97304946.3**

(22) Date of filing: **07.07.1997**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: **08.07.1996 JP 178130/96**
21.05.1997 JP 130626/97

(71) Applicant: **Murakoshi, Hiromasa**
Koriyama-shi, Fukushima, 963 (JP)

(72) Inventor: **Kanno, Kazuhiro**
Koriyama-shi, Fukushima, 963-02 (JP)

(74) Representative:
Cross, Rupert Edward Blount et al
BOULT WADE TENNANT,
27 Furnival Street
London EC4A 1PQ (GB)

(54) Software management system and method

(57) An operation management system for managing the operation of a managed software product. When a management target function is executed, reference is made to a battery value and, if the value is zero or greater, the function is allowed to be executed. The battery value is decremented as the function is executed. A charge value is supplied on a charge disk, such as a floppy disk, to allow the user to increase the battery value and to extend the usage period of the managed software product. The charge value may be supplied over a communication line.

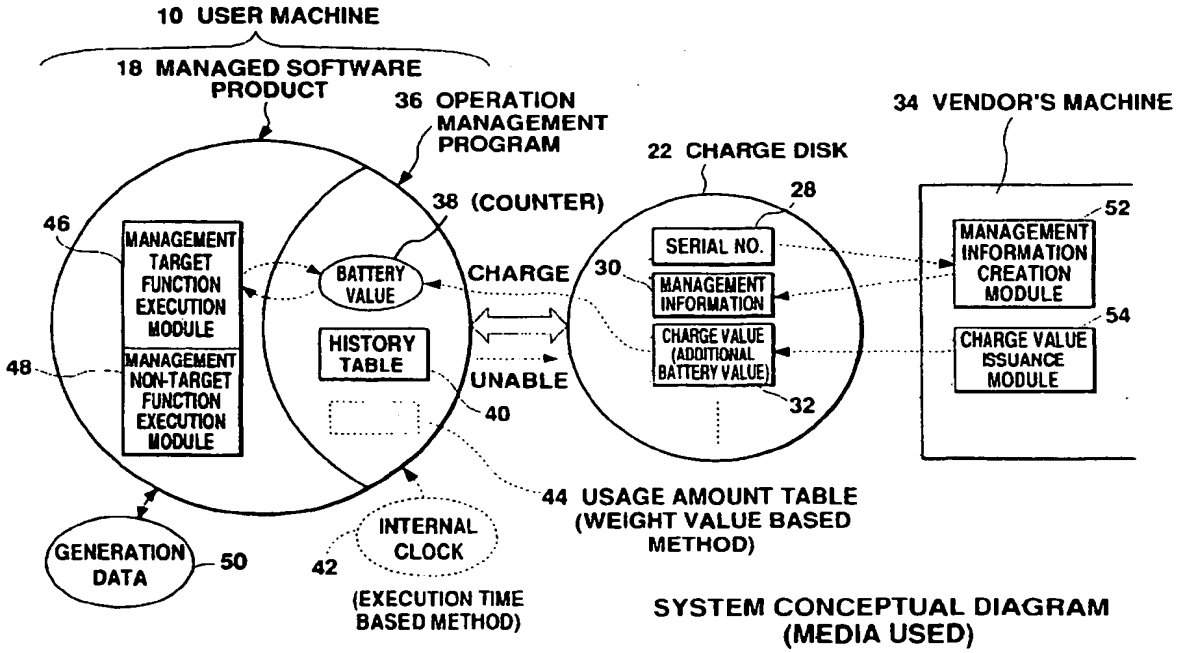


Fig. 3

EP 0 818 748 A2

Description

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates to an operation management system and an operation management method, and more particularly to software operation management or execution management.

Description of the Related Art

As computers and computer use become more common, more advanced technology is introduced and a variety of software products are developed for use in various fields. However, in many cases, the user finds it difficult to select a product from among a variety of software products that seem to meet the user's requirements; often, the user cannot find the best tool for his needs.

To reduce such a risk, a service has been available that supplies the user with a trial-use software product free of charge. However, most of these trial-use software products contain only function descriptions or provide the user with limited functions (e.g., save function and/or output function is/are not included). This makes it difficult for the user to evaluate the actual product (all the functions) correctly.

A sales system which charges the user according to how long the user actually uses a software product (including a trial use) would allow him to buy the product anytime he wants, to fully evaluate the product, and to precisely determine the requirements for continued use (including payment for it). Many users would find this type of sales system appealing and economical.

In Japanese Patent Laid-Open Publication No. Sho 59-41061 and Japanese Patent Laid-Open Publication No. Sho 63-153633, a system is disclosed that automatically prevents a program from being used when the usage count reaches a specified value. In Japanese Patent Laid-Open Publication No. Hei 1-147622 a system is disclosed which accumulates program execution time (total program execution time) and prevents the program from being used when the accumulation time reaches a specified amount. However, these systems do not disclose means for extending the program usage period. Japanese Patent Laid-Open Publication No. Hei 5-134949 discloses a system in which a program and expiry of the program are downloaded from a host computer to a user computer via a communication line. Also disclosed is a system in which a new expiry of the program is downloaded from the host computer to the user computer in order to update the expiry. However, the system only measures the execution time taken for executing the entire program, and does not include any means for changing the expiry on the user computer.

In Japanese Patent Laid-Open Publication No. Hei 7-234785, a system is disclosed that relates to a software rental system. This system connects a computer in a rental company to a user computer on which a rental software product is running over a communication line. When the time elapsed from the rental start time reaches the rental limit time, the system makes the program unavailable for use. (For example, the program is deleted.) To allow the user to update the rental period, the rental company sends a rental period extension program to the user's computer over a communication line. The user runs this program to extend the rental period of the program. A drawback of this system is that the user must pay for the software product regardless of whether the user has used it frequently or not. This means that the amount of money the user has to pay depends, not on how often he has used it, but on how long he has used it.

In Japanese Patent Laid-Open Publication No. Hei 7-244585, a system is disclosed that manages the program usage period. This system assigns a usage limit date to a program and, when the current date becomes greater than the limit date, the program product is made unavailable. To extend the usage limit date, the system reads update limit data from a recording medium containing that data and re-assigns a usage limit date based on the update limit data. This system is not reasonable because the amount of money the user has to pay does not depend on whether or not the user actually uses the program.

For example, during execution of a Computer Aided Design (CAD) software product, the user often spends much time thinking without entering data. In the system disclosed by the above mentioned Japanese Patent Laid-Open Publication No. Hei 7-234785 or Japanese Patent Laid-Open Publication No. Hei 7-244585, the user must pay for this thinking time. This places unwanted pressure on the user, especially when he must think carefully during program execution.

SUMMARY OF THE INVENTION

The present invention seeks to solve the problems associated with the art described above. In view of the foregoing, it is an object of the present invention to provide an operation management system and method which reasonably manage the operation of a managed software product.

It is another object of the present invention to provide an operation management system and method which levy a charge according to the actual usage amount of the managed software product (or the amount of the result generated by the managed software product).

It is still another object of the present invention to provide an operation management system and method which manage the operation according to the property of each function of the managed software product.

(1) To achieve the above objects, an operation manage-

ment system for managing the operation of a managed software product according to the present invention comprises: battery value management means for decrementing a battery value according to the operation amount of the managed software product; operation limit means for limiting the operation of the managed software product when the battery value has decreased to a specified limit value; and charge means for adding a charge value to the current battery value when the charge value is entered from external means.

The "battery value" mentioned above is a "virtual battery" which drives a managed software product. This battery value is preferably the value of a counter.

The battery value management means decrement the battery value according to the operation amount of the managed software product. When the battery value has reached a specified limit value (for example, 0), the operation limit means limit all of or a part of the operation of the managed software product. Upon receiving a charge value (additional battery value) from the external means, the charge means add the received value to the current battery value, thus extending the operation period. That is, the battery value is incremented, just as a battery is charged, to allow the continued use of the managed software product.

The managed software product described above is preferably a packaged application software program including a CAD program, game program, video program, language processor, music program, communication program, or a measurement program.

The battery value management means, operation management means, and charge means described above should be implemented preferably as software programs (management software programs) that run on a computer. The managed software product and the management software product may be separate, or the whole or a part of the management software product may be included in the managed software product.

A system according to the present invention is implemented on a general-purpose computer or special-purpose computer having such peripheral units as a disk drive, display, and input unit. The external means described above include recording media such as a magnetic disk or an optical disk and other host computers connected over a network.

(2) An operation management system according to the present invention may be applied to an application software product sales system. The following explains an example:

A vendor sells an application software product containing the operation management program according to the present invention. The operation management program has a battery value defined as the initial value. In addition to this product, the vendor sells recording media containing charge values (e.g., floppy disk (FD)). In this case, it is desirable that a variety of recording media, each containing a unique charge value, be supplied.

On the other hand, a user who bought the application software product may use the product until the battery value reaches zero. This allows the user to fully evaluate and examine the product. A user who wants to use the product after the battery value becomes zero must buy a recording medium containing a charge value to charge the battery. This enables him to add a charge value to the battery value and to use the product continuously.

If the specifications of the application software product do not satisfy the user's request, the user does not buy the recording medium. This prevents additional charges and reduces the cost to the user.

Considering an increase in the sales profit in recording media that will be produced in the future, a combination of a managed software product and the operation management program will lower prices significantly. The operation management system according to the present invention will increase the profits of both the user and the vendor, making it possible to build a very reasonable, economical system.

(3) In a preferred embodiment of the present invention, the battery value management means calculate the operation amount of each function of the managed software product, and subtracts a value corresponding to the operation amount from the battery value.

A continuous decrease in the battery value during execution of a managed software product, as in a conventional system, decrements the value even when the user is idle (input wait time), which places pressure on the user.

Calculating the operation amount of each function during execution of a managed software product, as in a system according to the present invention, decreases the battery value only when the managed software product is actually used, enabling the user to do operation without having to worry about time elapsed while thinking.

(4) In a preferred embodiment of the present invention, function category determination means are also available which determine if an execution instruction from the user activates a management target function or a management non-target function. And, the battery value management means decrement the battery value only when the management target function is executed.

For example, with the data generation function defined as a management target function and with other functions as management non-target functions, a cost can be levied only when new data are generated.

(5) In a preferred embodiment of the present invention, the battery value management means have a weight table containing an operation amount weight value for each of the management target functions. When any of the management target functions is executed, the battery value management means decrement the battery value by the weight value corresponding to the management target function.

In a preferred embodiment of the present invention,

the battery value management means measure the execution time of each of the management target functions and decrement the battery value by the value corresponding to the execution time.

This weight value system is able to calculate the operation amount regardless of the computer speed, which may differ among computers. In addition, by measuring time in this manner, the execution time is directly monitored and therefore the operation mount becomes proportional to the CPU load.

(6) In a preferred embodiment of the present invention, the operation limit means prevent only the management target functions from being executed when the battery value has decreased to a specified limit value; management non-target functions are executed.

For example, forcing a game program used at home to terminate when the battery value has reached a specified value does not cause a serious problem.

However, for a CAD program used in an office, forced termination when the battery value has reached a specified value may make already-produced data unavailable, possibly interrupting a job. Therefore, considering user's advantage and convenience, the embodiment keeps some functions operable even when the battery value has reached a specified value.

(7) A preferred embodiment of the present invention has remainder warning means for issuing a remainder warning message when the battery value has decremented to a specified warning value because a sudden inoperable condition in the managed software product without prior notice may cause the user unexpected damage. The remainder warning means alert the user to that condition before it occurs. In other words, the warning message prompts the user to determine whether to charge the battery value.

A preferred embodiment of the present invention has remainder display means for displaying the battery value on the screen during execution of the managed software product. This remainder display information keeps the user informed of the amount by which the managed software product will be able to continue operation without being charged.

It is also possible to program the system so that, upon detecting that the battery value has been charged to a specified value, the system can automatically disable operation management through the battery value to allow the user to use the product indefinitely.

(8) To achieve the above objects, a method for managing the operation of a managed software product according to the present invention comprises: a count value management step for changing a count value according to the operation amount of the managed software product; an operation limit step for limiting the operation of the managed software product when the count value has reached a specified limit value; and a charge step for charging the current count value or the limit value when a charge value is entered from external means.

The above count value is incremented or decremented according to the operation amount of the managed software product. When the count value is incremented, a charge value is added to the limit value; when the count value is decremented, a charge value is added to the current count value. In either case, the usage period is extended by charging the battery value.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagram showing a user machine used in the operation management system according to the present invention.

Fig. 2 is a diagram showing the data structure of a charge disk.

Fig. 3 is a diagram showing the concept of the operation management system according to the present invention.

Fig. 4 is a diagram showing an example of the history table.

Fig. 5 is a diagram showing an example of the usage amount table.

Fig. 6 is a flowchart showing the processing of the system when a management target function is executed in the execution time based method.

Fig. 7 is a flowchart showing the processing of the system when a management target function is executed in the weight value based method.

Fig. 8 is a flowchart showing the charge disk read processing.

Fig. 9 is a flowchart showing the charge processing.

Fig. 10 is a diagram showing a user machine used in another embodiment.

Fig. 11 is a diagram showing the structure of data sent from the host machine to a user machine.

Fig. 12 is a diagram showing the concept of the system in another embodiment.

Fig. 13 is a diagram showing an example of the user registration table.

Fig. 14 is a flowchart showing the operation of the user machine and a user machine in another embodiment.

Fig. 15 is a diagram showing another configuration of the system.

Fig. 16 is a diagram showing an example of an application according to the present invention.

Fig. 17 is a flowchart showing the function category determination processing.

DESCRIPTION OF PREFERRED EMBODIMENTS

Fig. 1 shows a user machine 10. This user machine 10 is a computer which executes various types of application programs under control of the operation system (OS). The user machine 10 is composed of a system unit 12, display 14, keyboard (not shown in the figure), output unit (not shown in the figure) such as a printer or plotter, and so forth. The system unit 12 contains a CD-ROM disk drive 16 which accesses a CD-ROM and

reads data from it and a floppy disk drive 20 which accesses a floppy disk (FD) and reads data from it.

The CD-ROM shown in Fig. 1 contains a managed software product 18. In this embodiment, the managed software product 18, such as a CAD software product, has an operation management program built in. The operation management program, designed for managing the operation of the managed software product 18, manages the operation using a "battery value" which will be described below. In the example shown in Fig. 1, the managed software product 18 is installed from the CD-ROM to the user machine 10; it may be installed from any other recording medium or via a communication line.

A charge disk 22, containing specified data (including a charge value) on a floppy disk, functions as a battery value charger. Inserting this charge disk 22 into the floppy disk drive 20 causes a charge value to be read and enables the user to extend the allowable operation period of the managed software product 18. In this embodiment, several charge disks 22, each containing a unique charge value, are supplied to allow the user to select or buy a desired charge disk 22 to add a desired charge value to the battery value.

The managed software product 18 and the charge disk 22 are usually supplied from the same vendor. In this embodiment, the managed software product 18 includes the operation management program. Of course, the managed software product 18 and the operation management program may be separately loaded into the user machine 10.

In Fig. 1, the display 14 has a remainder information area 24 where remainder information is displayed and a remainder warning area 26 where a warning message is displayed when the remainder drops below the specified amount. These areas will be described later.

Fig. 2 shows the data structure of the charge disk 22. As shown in Fig. 2, the charge disk 22 contains a serial number 28, management information 30, and charge value (additional battery value) 32. The serial number 28 is a unique identification number that is assigned when the floppy disk is formatted. Usually, this number is not copied when the disk is copied. The management information 30 is created when the serial number 28 is encrypted. This management information 30 is copied when the disk is copied. Therefore, when the disk is copied illegally, the serial number 28 and the management information 30 do not match, thereby making it easy to determine that the disk is copied illegally. Of course, any other conventional security system may also be used instead of this method.

The charge value 32 is an additional charge value to be added to the battery value that is decremented as the user uses the managed software product 18. Charging the battery value with this charge value enables the user to extend the usage period.

When the battery value is managed in the "execution time based method" in which the battery value is decremented by the execution time of each function, an additional time is recorded as the charge value 32. On the other hand, when the battery value is managed in the "weight value based method" in which the battery value is decremented by the weight value of each function, the additional value is recorded as the charge value 32. These methods will be described in more detail later.

Although a floppy disk is used as the charge disk 22 in the embodiment shown in Fig. 1, other types of recording media may also be used. Also, as shown in another embodiment that will be explained later, a charge value may be sent over a communication line.

Fig. 3 shows the concept of the operation management system which uses the charge disk 22. The system is composed primarily of the user machine 10, charge disk 22, and vendor's machine 34. In this embodiment, the managed software product 18 including the operation management program 36 is installed in the user machine 10.

The charge disk 22 is generated on the vendor's machine 34 owned by the vendor which sold the managed software product 18. More specifically, the vendor's machine 34 has two software modules: the management information creation module 52 and the charge value issuance module 54. The management information creation module 52 encrypts the serial number 28 recorded on the charge disk 22, and writes the resulting management information 30 back onto the charge disk 22. Note that the operation management program 36, which contains the encryption condition or the decryption condition, can check whether or not the serial number 28 agrees with the management information 30. The charge value issuance module 54 records the charge value 32, which has been set by the vendor, onto the charge disk 22. In the execution time based method, the charge value 32 is recorded, for example, as 100 hours, 200 hours, or 500 hours. Note that the operation management program 36 contains an initial battery value (for example, 100 hours).

The operation management program 36 has a counter 38 which decrements the battery value (battery value management function). In this embodiment, the operation management program 36 decrements the counter 38 each time a "management target function" provided by the managed software product 18 is executed. When the battery value, i.e., the counter value, has decremented to the limit value of 0, the operation management program 36 prevents management target functions from being executed. That is, in this embodiment, when the battery value has reached a specified limit value, the execution of the managed software product 18 is limited and, when the battery value is charged with the charge value 32 contained on the charge disk 22, the charge value is added to the battery value and the resulting value is used as a new battery value. The usage period of the managed software product 18 is thus extended.

A history table 40 managed by the operation man-

agement program 36 contains history information on charge values recorded on the charge disk 22. Fig. 4 shows an example. As shown in Fig. 4, the history table 40 is composed of three columns: FD serial number column 40A, charge data/time column 40B, and charge value column 40C. The table may have other columns as necessary.

Referring to Fig. 3 again, the following explains how the battery value is managed. When the battery value is managed in the "execution time based method" described above, the execution time of each management target function, measured based on the internal clock 42, is subtracted from the battery value. On the other hand, when the "weight value based method" described above is used, the battery value is managed based on the usage amount table 44. Fig. 5 shows an example of the usage amount table 44. In this embodiment, the table contains entries, each consisting of a function name 44A and the corresponding usage amount 44B. It should be noted that each usage amount is used as a weight value. For example, a weight value is pre-defined according to the processing time of each function. Therefore, when a management target function is executed, the corresponding usage amount (weight value) is subtracted from the battery value.

The managed software product 18 shown in Fig. 3 has many user interface programs as well as many internal functions and common functions used by the programs. These functions are classified roughly into two: management target functions and management non-target functions. Whenever the managed software product 18 attempts to execute a management target function, the operation management program 36 references the battery value and, when it is zero or greater, allows the managed software product 18 to execute that function. When the managed software product 18 attempts to execute a management non-target function, the operation management program 36 does not check the battery value. For example, when input/output function for processing generated data 50 from the managed software product 18 is defined as a management non-target function, the input/output processing is always executed on the generated data 50, even if the usage period of the managed software product 18 has expired. This ensures that the generated data 50 are always processed, thus protecting user assets. Examples of management non-target functions include the data display function, data print function, and data plotter output function.

Management target functions include the data generation function. For example, when the managed software product is a CAD software product, the data generation function includes the straight-line drawing function, curved-line drawing function, circle drawing function, area fill-in function, area hatching function, and character insertion function.

Fig. 3 conceptually shows management target function execution module 46 which executes management target functions and management non-target function execution module 48 which executes management non-target functions. In this embodiment, the battery value is decremented only when a management target function is activated. Note that the battery may be decremented when both a management target function and a management non-target function are activated.

In addition to the data described above, the charge disk 22 may contain other types of data. For example, it may contain the name of the managed software product 18 which accepts a charge value. In this case, the name of the managed software product 18 is used as follows. When the charge disk 22 is read, the operation management program 36 checks whether or not the name of the managed software recorded on the charge disk 22 matches that of the managed software product 18 installed in the user machine 10 and, only when they match, accepts the charge value 32.

The battery value described above is stored on the hard disk and then copied into the computer's RAM. The battery value in the RAM is decremented whenever a management target function is executed. Also, at an interval or as necessary, the battery value in the RAM replaces the battery value on the hard disk. This means that, even when the computer fails, the battery value is not erased. The battery value may also be maintained in some other way.

Fig. 17 is a flowchart showing how the operation management program operates when it accepts an instruction requesting the execution of a managed software product function. The following explains this processing in more detail.

Upon receiving from a user an instruction requesting the execution of a function of the managed software product while the managed software product is in execution (S601), the operation management program checks whether the requested function is a management target function or a management non-target function (S602). When the function is a management target function (S603), the operation management program performs the processing shown in Fig. 6 or Fig. 7 (S604). When the function is a management non-target function (S603), the program executes the function immediately. (S605). This processing is repeated whenever an execution instruction is received.

Next, referring to Fig. 3, the execution of a management target function in the execution time based method is explained with the use of Fig. 6.

When the user requests the execution of a management target function while the managed software product 18 shown in Fig. 3 is in execution, the routine shown in Fig. 6 is started. First, the management target function execution module 46 or the operation management program 36 reads the battery value to check if it is greater than zero. If the battery value is zero or less, the routine is terminated. That is, the requested management target function cannot be started. Note that a management non-target function is started even if the battery value is

zero.

In S102, the routine gets the start time from the internal clock 42 before starting the requested management target function and, in S103, starts the management target function. In S104, the routine gets the end time from the internal clock 42 and, in S105, subtracts the start time from the end time to calculate the processing time (execution time) of the processing executed in S103.

In S106, the routine subtracts the processing time calculated in S105 from the battery value. In S107, the routine checks if the resulting battery value is equal to or less than the warning value and, if so, displays a message in the remainder warning area 26 shown in Fig. 1. If the resulting battery value is greater than the warning value, the routine does not display the message. As shown in Fig. 1, the remainder information area 24 is displayed during execution of the managed software product 18 (see Fig. 1) to allow the user to check the remaining amount. This helps the user determine how long he can execute the managed software product 18.

Fig. 7 shows the processing of a management target function in the weight value based method.

When the execution of a management target function is requested as described above, the routine references the battery value in S201 to check if it is equal to or greater than 0. If it is, the routine executes the requested management target function in S202 and, in S203, references the usage amount table 44 shown in Fig. 5 to find the usage amount (weight value) of the executed management target function. Then, in S204, the routine subtracts the processing amount found in S203 from the battery value to find a new battery value. In S205, the routine checks if the battery value is less than the warning value and, if so, displays a message in the remainder warning area 26 in S206.

The "execution time based method" shown in Fig. 6 allows the user to manage operation using a physical amount that is easy to understand. In addition, the user can manage operation in a relatively simple configuration. On the other hand, the "weight value based method" shown in Fig. 7 gives the user the same result regardless of the CPU speed of the user's machine.

Next, referring to Fig. 3, the charge disk 22 read processing is explained with the use of Fig. 8.

This processing is started when the charge disk 22 is inserted into the floppy disk drive 20 as shown in Fig. 1. The routine reads the serial number in S301, and the management information in S302, both from the charge disk 22. In S303, the routine encrypts the serial number according to the encryption condition, or decrypts the management information according to the decryption condition, and comparos the serial number with the management information. This comparison determines whether or not the charge disk 22 is legal. For example, when the disk is illegally copied, the management information 30 is copied, but the serial number 28 is not copied but replaced. This results in a mismatch between the serial number 28 and the management information 30, thereby making it possible to find an illegal copy.

In S304, the routine checks if the charge disk 22 is valid and, if it is not valid, terminates processing in S308. If it is valid, the routine references the history table 40, containing past charge history data, in S305 to check the validity of the charge value 32 recorded on the charge disk 22. To do so, the routine first checks to see if the serial number 28 of the charge disk 22 is in the history table 40. If the serial number is found, the routine takes the following steps to check if the charge value 32 recorded on the charge disk 22 is valid. The routine finds the charge value initially recorded on the charge disk 22 and, from that initial value, subtracts the actual charge value to find the remainder. The next time the battery value is charged, the routine compares the remainder with the charge value currently recorded on the charge disk. If the charge value on the charge disk 22 is greater than the remainder, the routine determines in S306 that the charge disk is not valid and terminates processing in S308. If the routine finds that the charge value 32 on the charge disk 22 is valid, it performs the charge processing, shown in Fig. 9, in S307.

Fig. 9 shows an example of charge processing. In S401, the routine references the counter 38 to read the current battery value and, in S402, reads the charge value from the charge disk 22. In S403, the routine asks the user to type an actual charge value that does not exceed the charge value 32 recorded on the charge disk 22. The user types the charge value, for example, from the keyboard. In S404, the routine checks that the specified charge value is less than the charge value on the charge disk 22. If the specified charge value is greater than the charge value on the charge disk 22, the routine asks the user to retype the charge value.

In S405, the routine adds the specified charge value to the battery value, thus charging the battery value. In S406, the routine subtracts the specified charge value from the initial charge value and writes the resulting value on the charge disk 22 as a new charge value 32 . If the initial charge value 32 is exhausted, the routine writes the value of 0 on the charge disk 22 to virtually erase the charge value. The value of 0 prevents the charge disk 22 from being re-used. In S407, a record relating to the charge processing is added to the history table 40.

In the above embodiment, the user specifies an actual charge value. Instead of having the user specify a value, a pre-defined charge value may be added to the battery value at that time.

Fig. 10 shows another embodiment according to the present invention. In the embodiment described above, the battery value is charged using a recording medium. In this embodiment, the battery value is charged via a communication line 60. For the same components as those used in the above embodiment, the same numbers are assigned and their descriptions are omitted.

The user machine 10 in Fig. 10 is connected to the

host machine 62 via the communication line 60. From this host machine 62, send data 64 shown in Fig. 11 are sent to the user machine 10 to charge the battery value.

In Fig. 11, address information 68 specifies the address of the user machine 10. Management information 70 is created by encrypting the serial number on the recording medium containing the managed software product 18. A charge value 72, a value to be added to the battery value as with the above embodiment, is an additional period of time in the execution time based method, and is an additional amount in the weight value based method.

Fig. 12 illustrates the system concept of this embodiment.

As described above, the user machine 10 is connected to the host machine 62 via the communication line 60. That is, this host machine 62 is connected to each of a number of user machines 10 for integrated operation management. This host machine 62 has a management information creation module 76, charge value issuance module 78, user registration table 80, and billing module 82. The management information creation module 76 creates the management information 70 shown in Fig. 11, and the charge value issuance module 78 issues a charge value 72 in response to a request from the user machine 10. As shown in Fig. 13, the user registration table 80 is composed primarily of the user ID column 80A, user name column 80B, and request charge value column 80C. The billing module 82 references the user registration table 80 to automatically issue a bill for a requested amount whenever a charge value is issued, or at some specified interval.

Next, referring to Fig. 12, the operation of this embodiment is explained with the use of Fig. 14. The operation of the user machine 10 is shown in the left side of Fig. 14, while that of the host machine 62 is shown on the right.

First, in S501 and S502, the user machine 10 is connected to the host machine 62 via a communication line. In S503, the user machine 10 generates a request for a charge value that will be sent to the host machine 62. In this case, the request contains at least the serial number of the CD-ROM containing the managed software product 18 and information on the charge value. In S504, the user machine sends the request to the host machine and, in S505, the host machine receives the request.

In S506, the host machine checks the user registration table 80. If the host machine finds, in S507, that the requesting user is registered in the host machine 62, the management information creation module 76 creates management information based on the serial number in S508, and the charge value issuance module 78 generates a charge value in response to the request from the user. In S509, the host machine 62 sends the management information and the charge value to the user machine 10 as the send data 64 shown in Fig. 11. In S510, the user machine 10 receives the send data 64. In S511 and S512, the user machine 10 and the host machine 62 are disconnected.

In S513, the operation management program 36 compares the serial number 74 with the management information 70 to check to see if the data received by the user machine 10 are valid. This prevents the user from illegally charging the battery value. If it is found in S514 that the send data are valid, the charge processing is performed in S515. This charge processing is the same as that in Fig. 9.

As shown in Fig. 12, this embodiment may also use the execution time based method or the weight value based method in order to manage the battery value.

Although the battery value is charged over a communication line such as a telephone line in the above embodiment, it may also be charged over a communication satellite (satellite line).

In the above embodiments, the operation management program 36 is included in the managed software product 18. Of course, an external program can manage the operation of the managed software product 18. Fig. 15 shows the concept of such an embodiment.

As shown in Fig. 15, the operation system (OS) 83 is located between the hardware 81 and each of application programs 84, 86, and 88. The operation management program 36 according to the present invention may be located between the operation system 83 and the application program 84.

Operation management program 36 therefore functions as an interface program. Messages are exchanged between the operation management program 36 and the application program 84 according to some specific rule. Messages are also exchanged between the operation management program 36 and the operation system 83 according to a specific rule.

To execute a management target function in this configuration, the operation management program 36 references the battery value when it receives an execution request from the application program 84. If the battery value is not zero, the operation management program 36 sends an instruction to the operation system 83 while simultaneously decrementing the battery value by a value corresponding to the function. If the battery value is zero, the operation management program 36 sends a message back to the application program 84, indicating that the instruction cannot be executed.

To execute a management non-target function, the operation management program 36 does not reference the battery value when it receives an execution request from the application program 84 but instead sends the instruction directly to the operation system 83.

The battery value is decremented as management target functions are executed. Charging the battery value allows the user to extend the usage period of the application program 84, which may be supplied separately from the application program 84.

In the above embodiments, one operation management program manages one operation management program. It is also possible for one operation manage-

ment program to manage several application programs.

Fig. 16 shows an application of the present invention. The system shown in Fig. 16 is composed of one host machine 90 and several user machines 92. Within each user machine 92 are a managed software product 18 and the operation management program 36, which, in turn, contains the counter 38 where the battery value to be decremented is stored. In other words, the operation of the managed software product 18 is controlled by the value stored in the counter 38. To execute the managed software product 18 in this system, it is necessary to insert a battery disk 96 into the user machine 92 and to move the battery value from the battery disk 96 into the counter 38. The battery value is decremented as the operation of the managed software product 18 proceeds. When the user finishes the managed software product 18, a sequence of operations are executed to move the current counter value from the counter 38 to the battery disk 96. This initializes the counter 38 to zero just as it was before the battery disk 96 was inserted.

The host machine 90 has several disk drives into which a battery disk 96 is inserted to read the battery value that was returned to the battery disk 96. This host machine 90 is also used to charge the battery value on the battery disk 96.

Integrated management of the battery values on several battery disks 96 through the host machine 90 brings a benefit of integrally managing several managed software products 18.

This type of system may be used, for example, in a school or a business where many computers are installed. With an individual carrying his or her own portable battery disk 96, it is possible to check and control the software usage amount of each person. In this case, either the "execution time based method" or the "weight value based method" may be used.

Claims

1. An operation management system for managing the operation of a managed software product, comprising:

   battery value management means for decrementing a battery value according to the operation amount of said managed software product;
   operation limit means for limiting the operation of said managed software product when said battery value has decremented to a specified limit value; and
   charge means for adding a charge value to the current battery value when the charge value is entered from external means.

2. An operation management system according to claim 1, wherein said battery value management means find the operation amount for each execution of a function owned by said managed software product and subtract a value corresponding to said operation amount from said battery value.

3. An operation management system according to claim 2, further comprising:

   function category determination means for determining if a function to which an execution instruction is issued is a management target function or a management non-target function, wherein said battery value management means decrement said battery value only when said management target function is executed.

4. An operation management system according to claim 3, wherein

   said battery value management means has a weight table containing pairs of said management target function and a weight value representing said operation amount thereof, and
   said battery value management means subtract a weight value corresponding to said management target function from said battery value when said management target function is executed.

5. An operation management system according to claim 3, wherein, when said management target function is executed, said battery value management means measure the execution time and subtracts the execution time from said battery value.

6. An operation management system according to claim 3, wherein said operation limit means prevent said management target function from being executed but allows said management non-target function to be executed when said battery value has reached a limit value.

7. An operation management system according to claim 3, wherein said managed software product has a data generation function and a data output function and wherein said function category determination means determine said data generation function as said management target function and determine said data output function as said management non-target function.

8. An operation management system according to claim 1, further comprising remainder warning means for issuing a remainder warning when said battery value has decremented to a warning value.

9. An operation management system according to claim 1, further comprising remainder display

means for displaying said battery value during execution of said managed software product.

10. An operation management system for managing the operation of a managed software product, comprising:

battery value management means for decrementing a battery value according to the operation amount of said managed software product;
operation limit means for limiting the operation of said managed software product when said battery value has decremented to a specified limit value;
read means for reading a charge value from a recording medium containing the charge value thereon; and
charge means for adding said charge value to the current battery value.

11. An operation management system according to claim 10, further comprising erase means for erasing the charge value from said recording medium after said charge value is added.

12. An operation management system according to claim 10, further comprising:

specification means for allowing a user to specify an actual charge value by which the current battery value is to be actually charged, the actual charge value not exceeding the charge value recorded on said recording medium; and
rewrite means for rewriting the charge value on said recording medium with a remainder value after said actual charge value is added to the current battery value.

13. An operation management system according to claim 10, in which said recording medium contains not only said charge value, but also the identification number of the recording medium and management information generated through encryption of the identification number, said operation management system further comprising:

validity determination means for comparing said identification number with said management information considering the condition of said encryption to determine the validity of said recording medium.

14. An operation management system comprising:

a managed machine containing a managed software product; and
a managing machine connected to said managed machine with a communication line, wherein
said managed machine comprises:
battery value management means for decrementing a battery value according to the operation amount of said managed software product;
operation limit means for limiting the operation of said managed software product when said battery value has decremented to a specified limit value;
charge value receive means for receiving a charge value from said managing machine; and
charge means for adding said charge value to the current battery value, and wherein
said managing machine comprises:
charge value send means for sending said charge value to said managed machine.

15. An operation management system according to claim 14, wherein said managed machine further comprises:

notification means for notifying said managing machine of the identification number of a portable recording medium initially containing said managed software product; and
validity determination means for comparing management information sent from said managing machine with said identification number to determine the validity of the recording medium; and wherein said managing machine further comprises:
management information creation means for creating said management information generated by encrypting said notified identification number and for sending the management information to said managed machine.

16. An operation management system comprising:

at least one managed machine containing a managed software product; and
a managing machine for managing the operation of said managed machine, wherein said managed machine comprises:
a counter containing a battery value changing according to the operation amount of said managed software product;
first charge means for reading a battery value from a portable recording medium to store the battery value into said counter; and
first return means for writing the current battery value on said recording medium; and wherein,
said managing machine comprises:
second charge means for writing said battery value on said recording medium; and
second return means for reading said battery value from said recording medium.

17. An operation management method comprising:

a count value management step for changing a count value according to the operation amount of a managed software product;
an operation limit step for limiting the operation of said managed software product when said count value has reached a specified limit value; and
a charge step for charging the current count value or said limit value when a charge value is entered from external means.

18. A medium containing a management software product for managing the operation of a managed software product, wherein said managed software product and said management software product are executed on computers, said management software product comprising:

a module for changing a count value according to the operation amount of said managed software product;
a module for limiting the operation of said managed software product when said count value has reached a specified limit value; and
a module for charging the current count value or said limit value when a charge value is entered from external means.

19. A medium containing a charge value read by a management software product for use in managing the operation of a managed software product, wherein said managed software product and said management software product are executed on computers, said management software product comprising:

a module for changing a count value according to the operation amount of said managed software product;
a module for limiting the operation of said managed software product when said count value has reached a specified limit value; and
a module for charging the current count value or said limit value when said charge value is entered.

20. A computer system having an interface software product between an operation system and at least one application software product, wherein said interface software product comprises:

a module for changing a count value according to the operation amount of said application software product;
a module for limiting the operation of said application software product when said count value has reached a specified limit value; and
a module for charging the current count value or said limit value when a charge value is entered from external means.

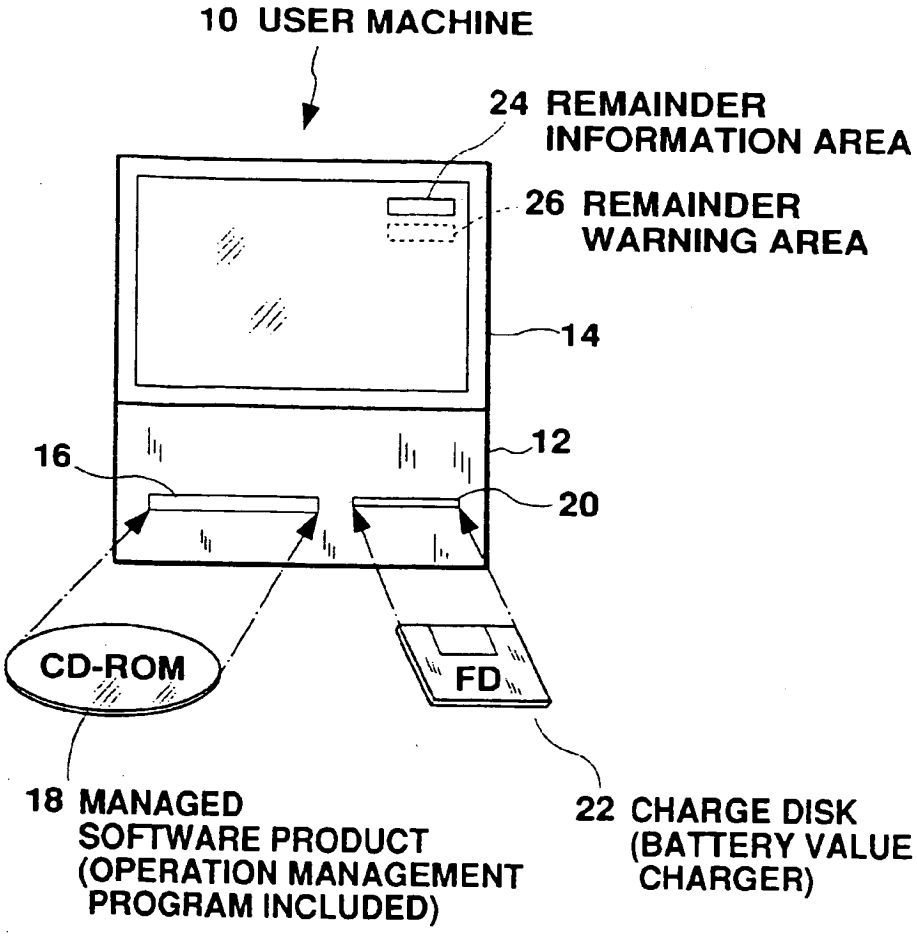


Fig. 1

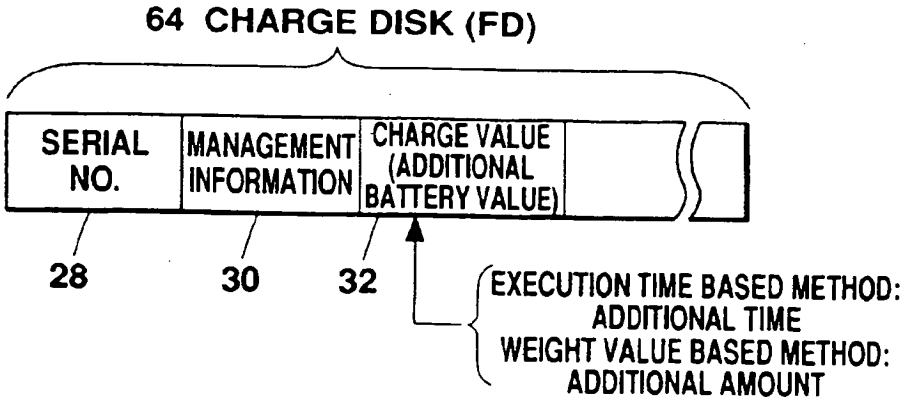


Fig. 2

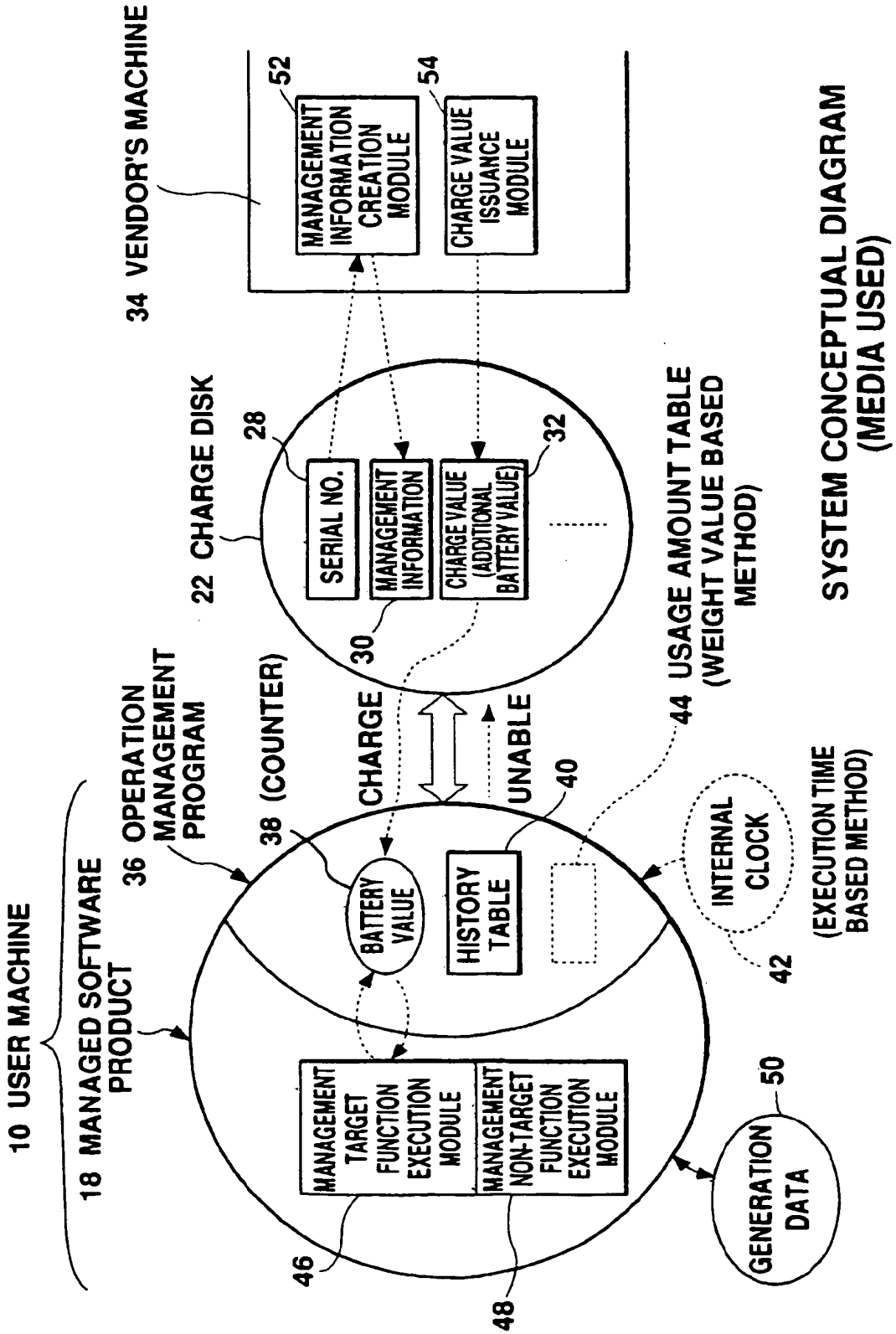


Fig. 3

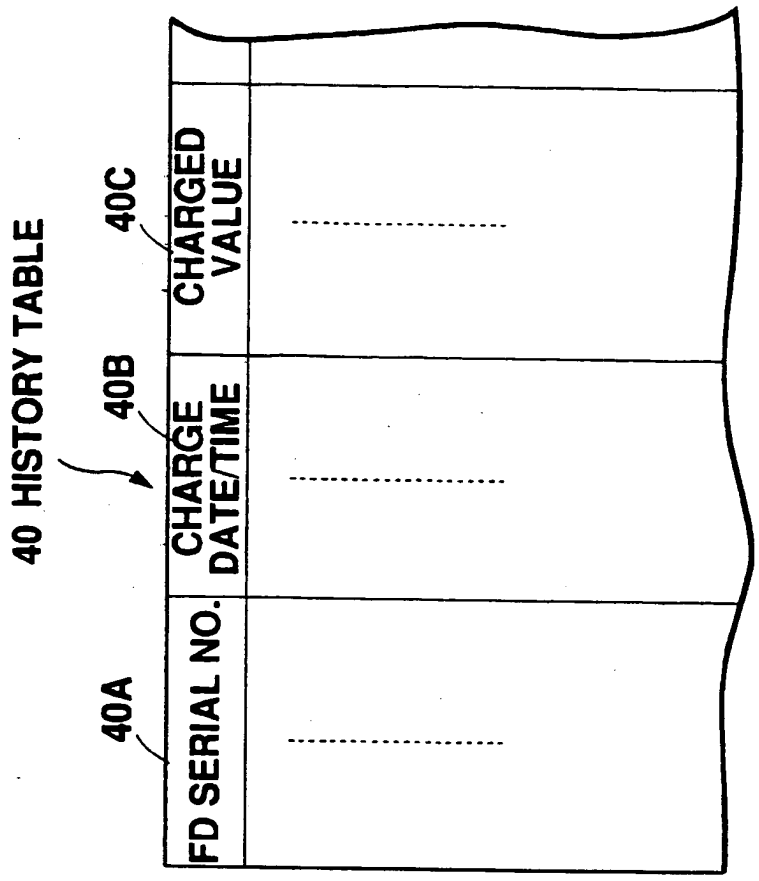


Fig. 5

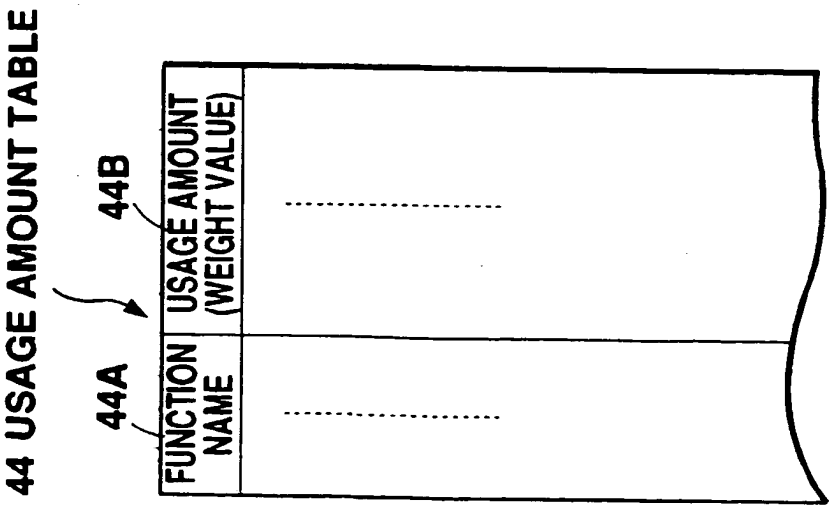


Fig. 4

MANAGEMENT TARGET FUNCTION EXECUTION
(EXECUTION TIME BASED METHOD)

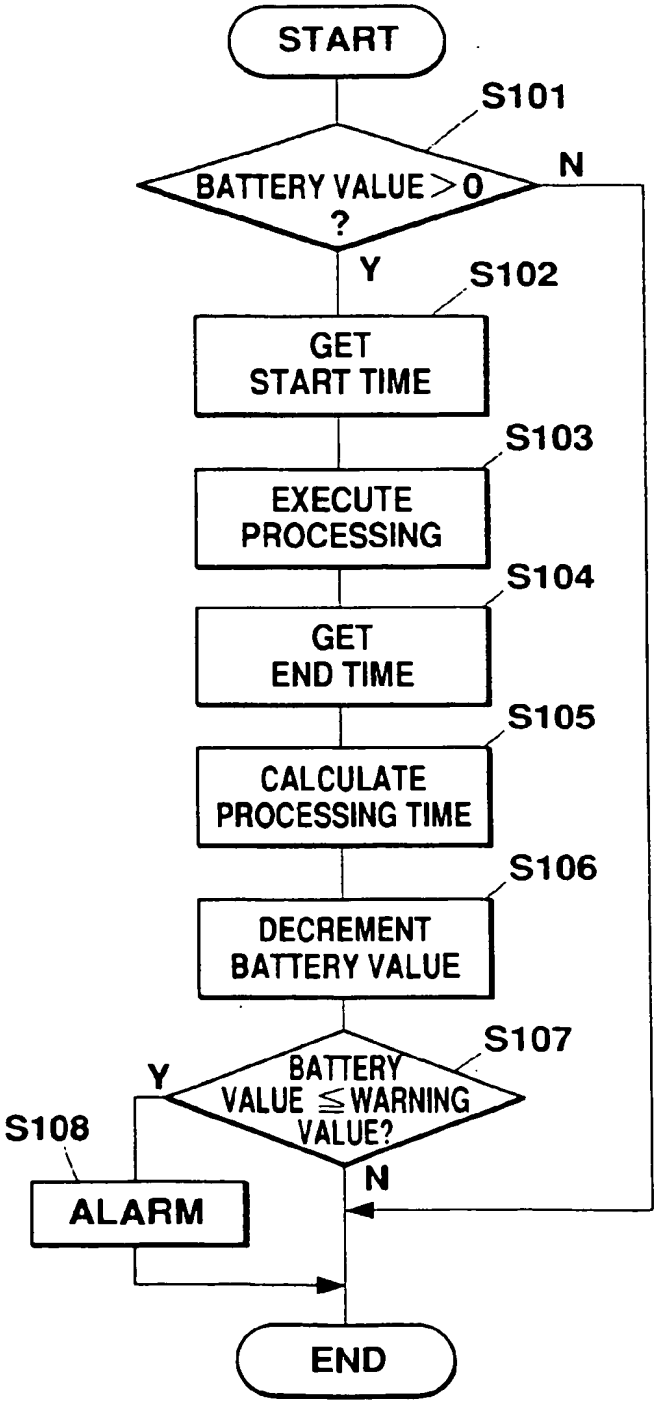


Fig. 6

MANAGEMENT TARGET FUNCTION EXECUTION (WEIGHT VALUE BASED METHOD)

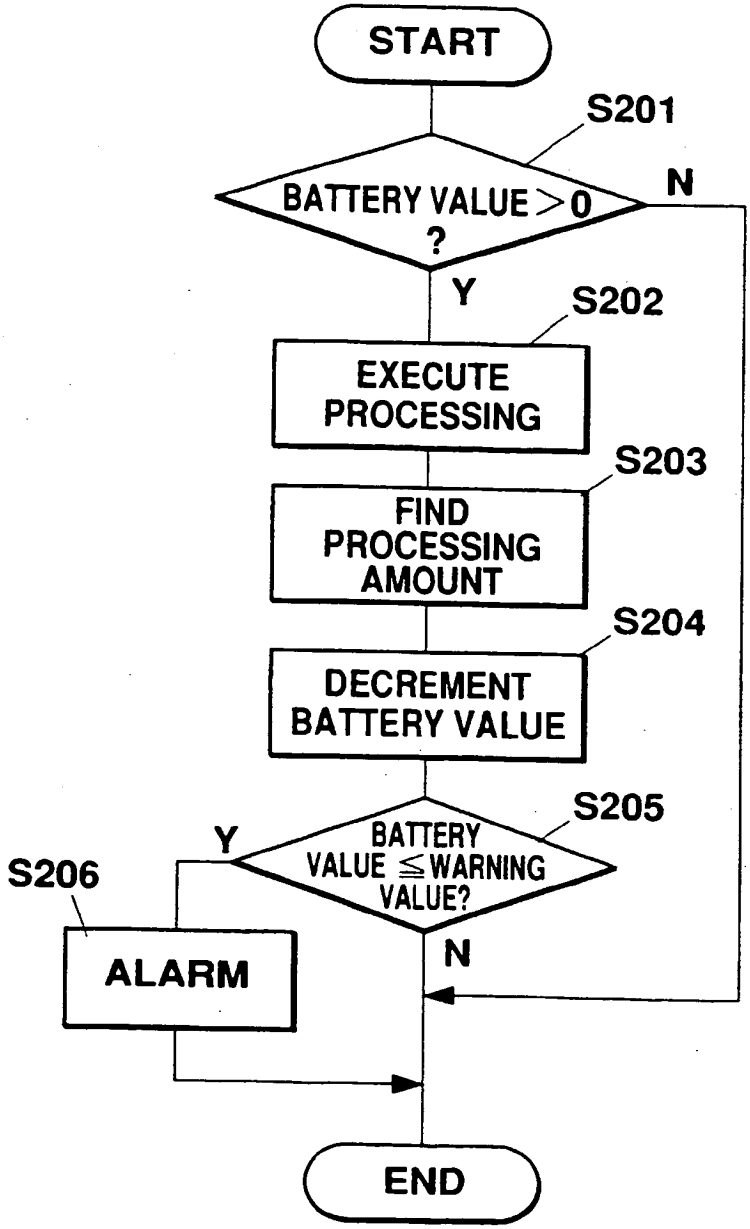


Fig. 7

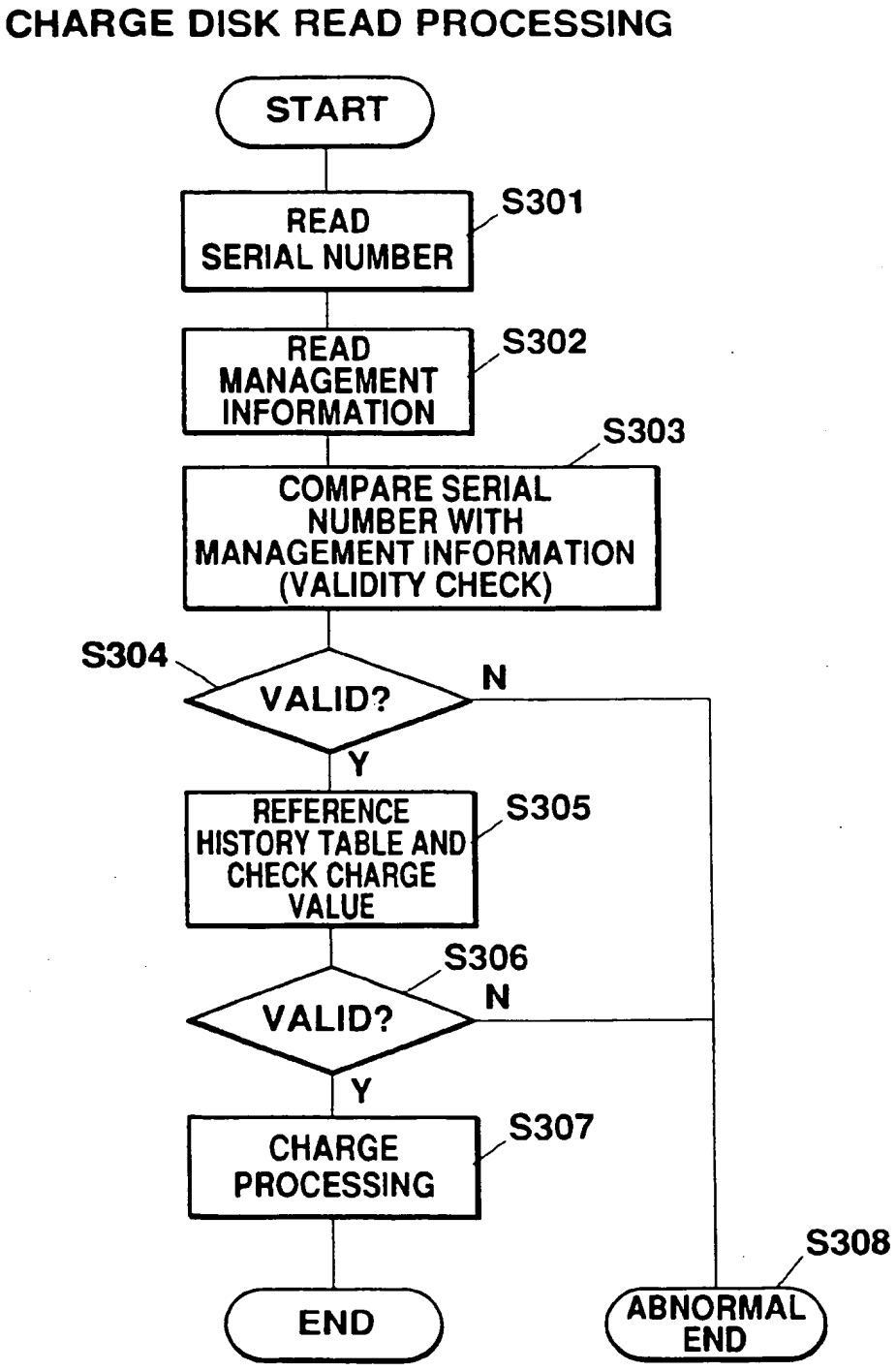


Fig. 8

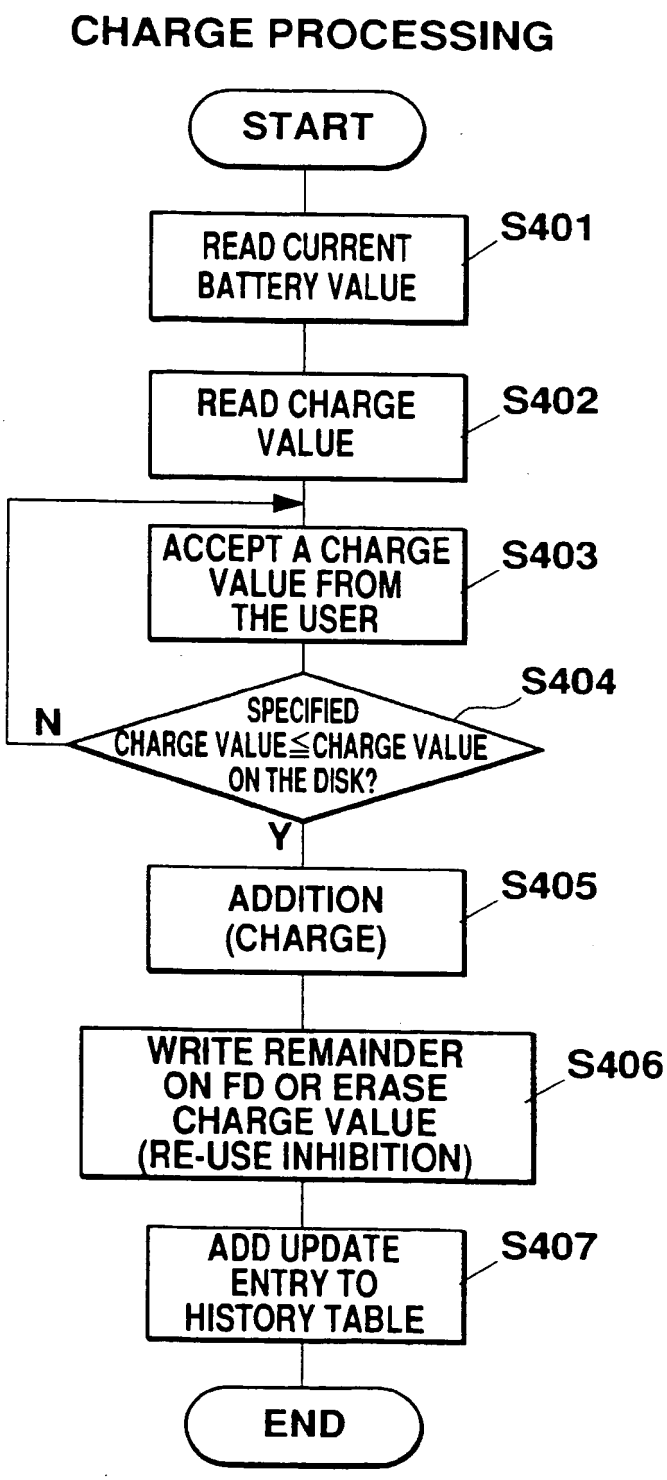


Fig. 9

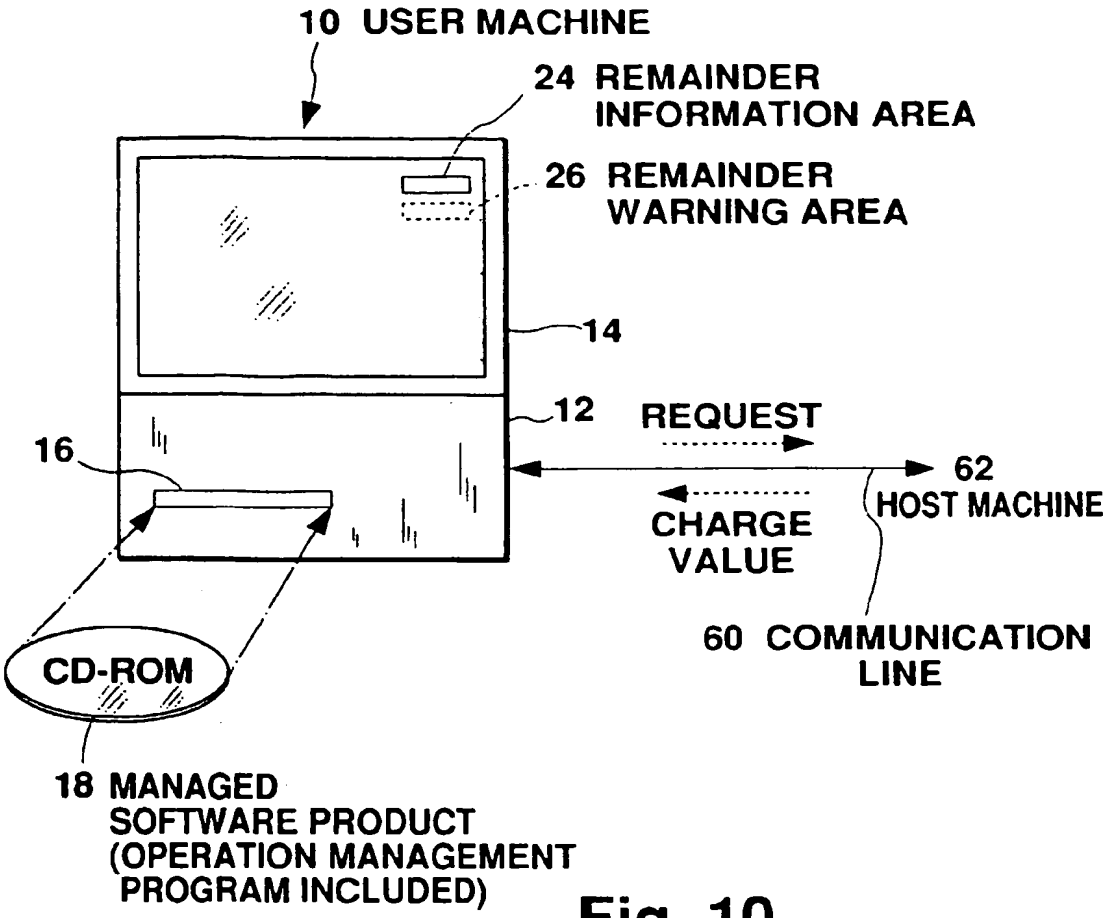


Fig. 10

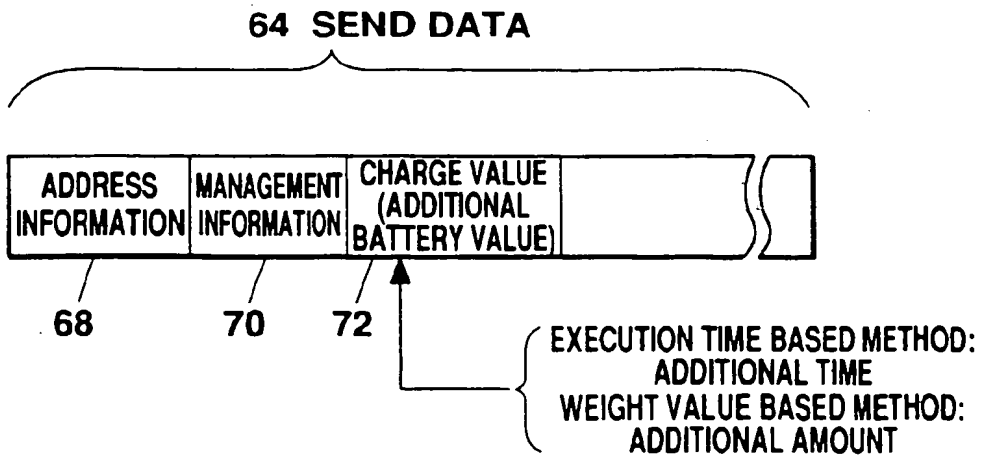
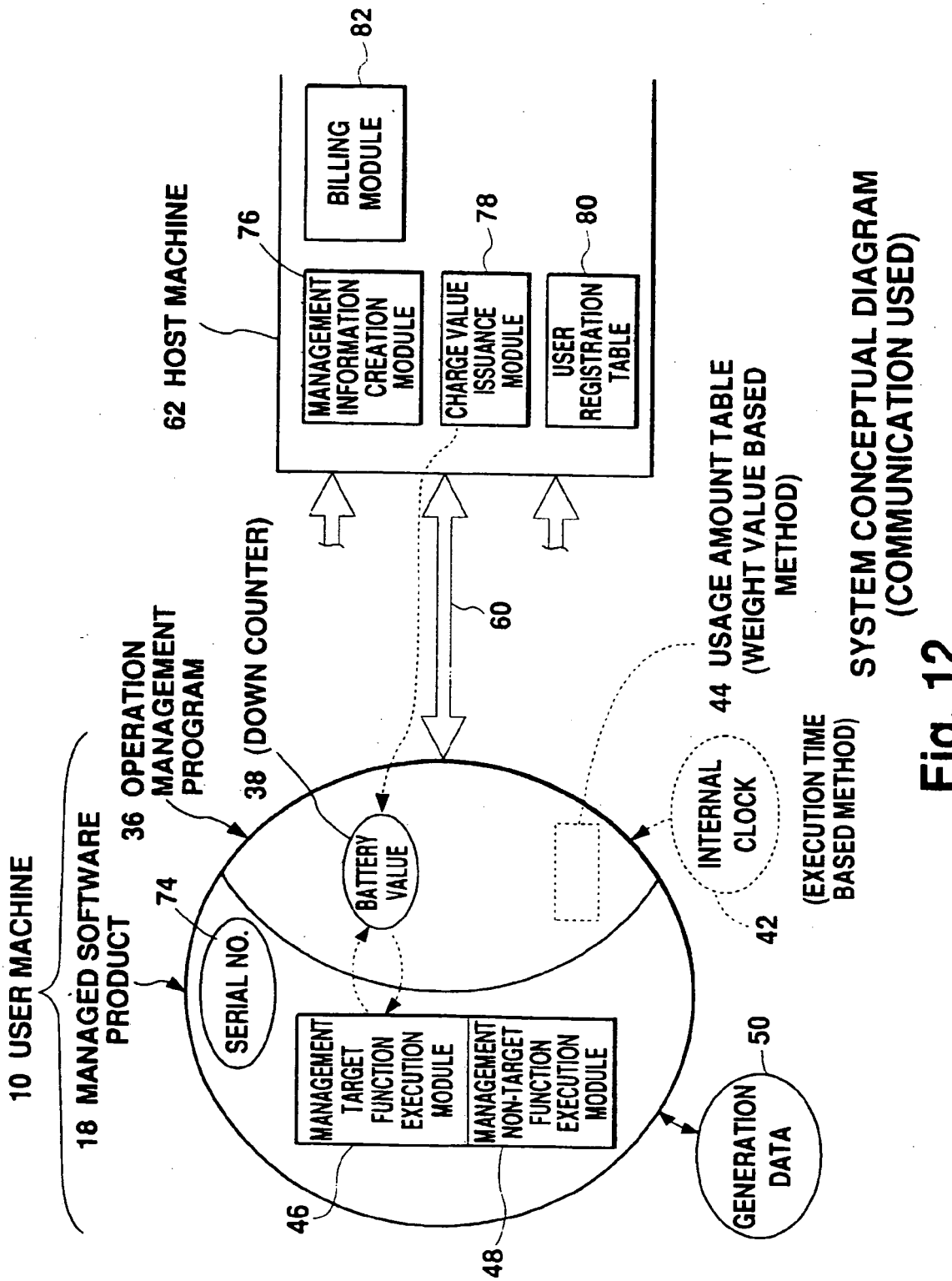


Fig. 11



SYSTEM CONCEPTUAL DIAGRAM (COMMUNICATION USED)

Fig. 12

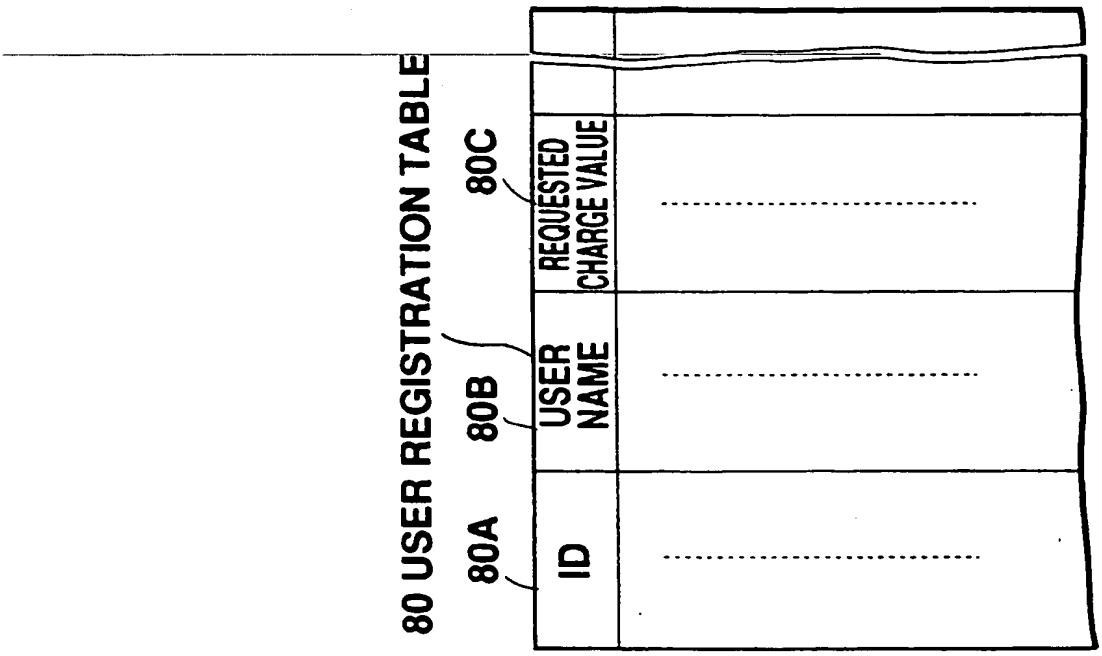


Fig. 13

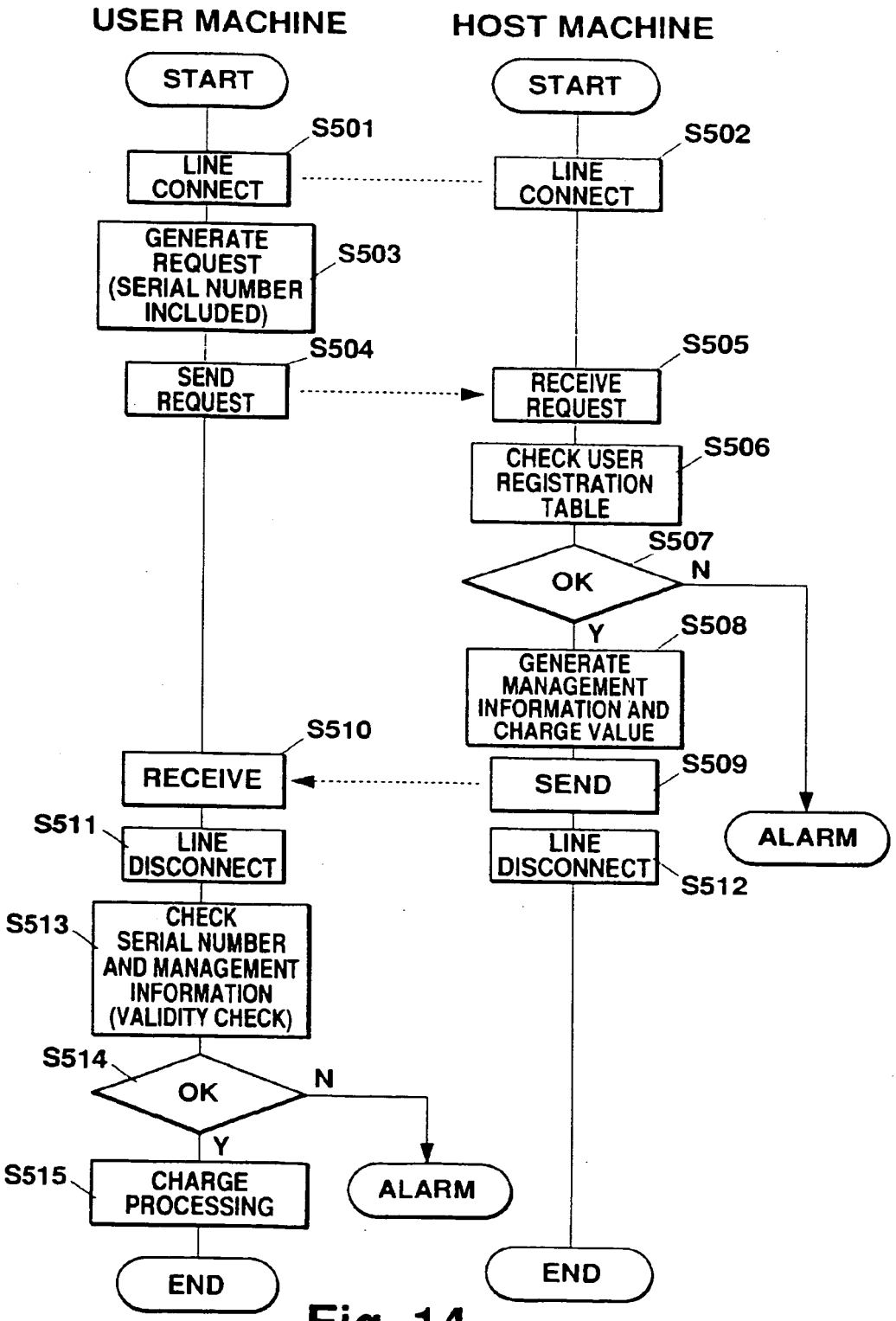


Fig. 14

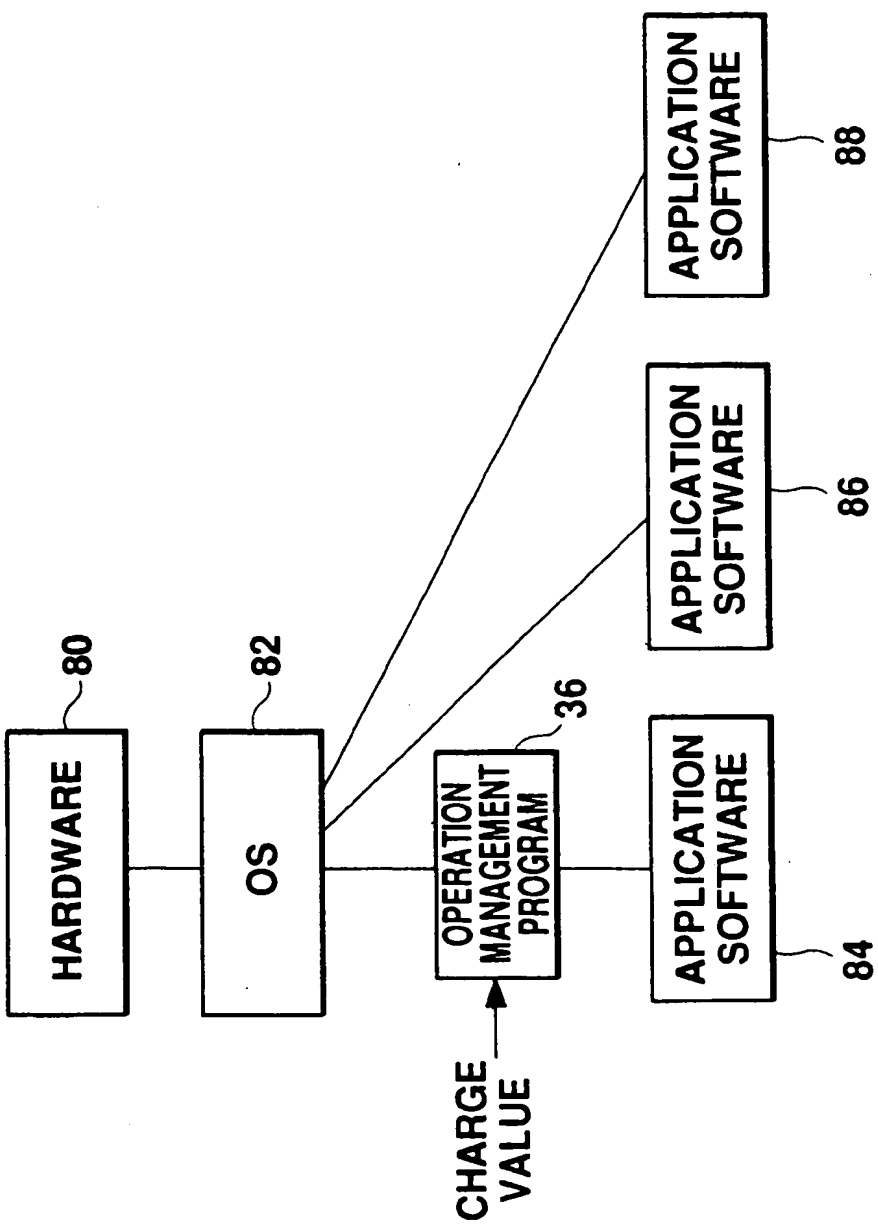


Fig. 15

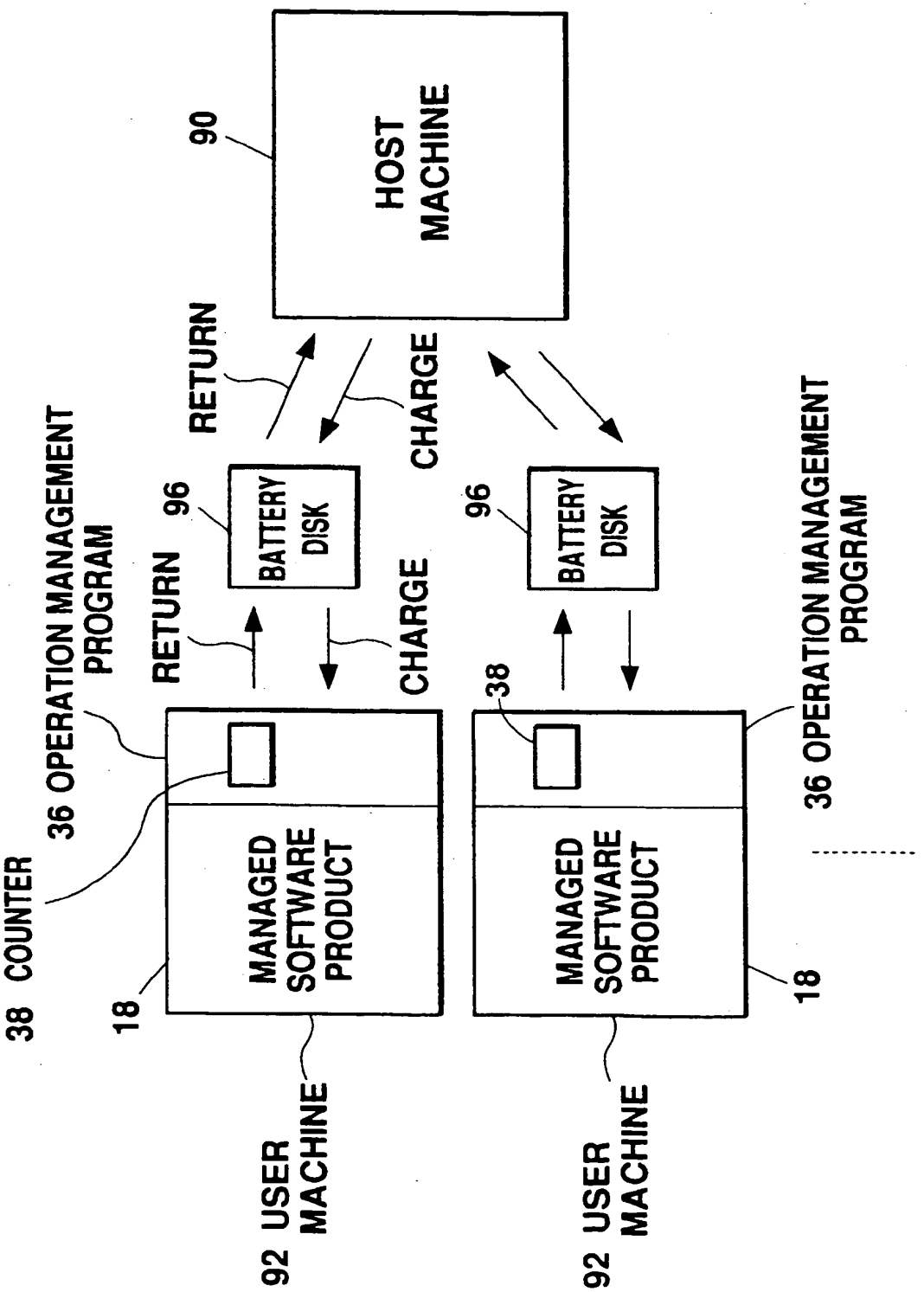


Fig. 16

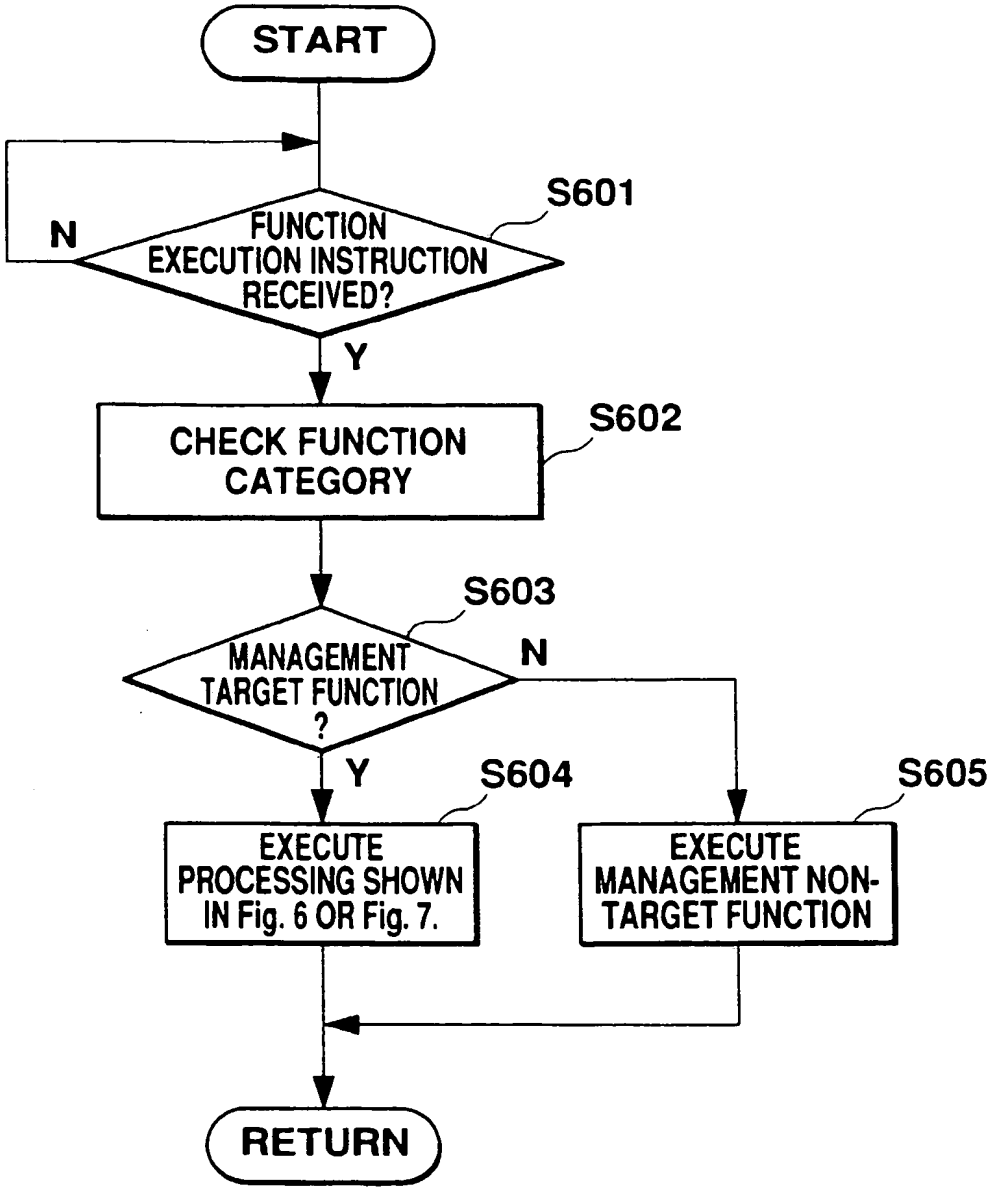


Fig. 17

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

(11) EP 0 818 748 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
15.11.2000 Bulletin 2000/46

(43) Date of publication A2:
14.01.1998 Bulletin 1998/03

(21) Application number: **97304946.3**

(22) Date of filing: **07.07.1997**

(51) Int Cl.[7]: **G06F 17/60**, G06F 1/00

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: **08.07.1996 JP 17813096**
21.05.1997 JP 13062697

(71) Applicant: **Murakoshi, Hiromasa**
Koriyama-shi, Fukushima, 963 (JP)

(72) Inventor: **Kanno, Kazuhiro**
Koriyama-shi, Fukushima, 963-02 (JP)

(74) Representative:
Cross, Rupert Edward Blount et al
BOULT WADE TENNANT,
Verulam Gardens
70 Gray's Inn Road
London WC1X 8BT (GB)

(54) Software management system and method

(57) An operation management system for managing the operation of a managed software product. When a management target function is executed, reference is made to a battery value and, if the value is zero or greater, the function is allowed to be executed. The battery value is decremented as the function is executed. A charge value is supplied on a charge disk, such as a floppy disk, to allow the user to increase the battery value and to extend the usage period of the managed software product. The charge value may be supplied over a communication line.

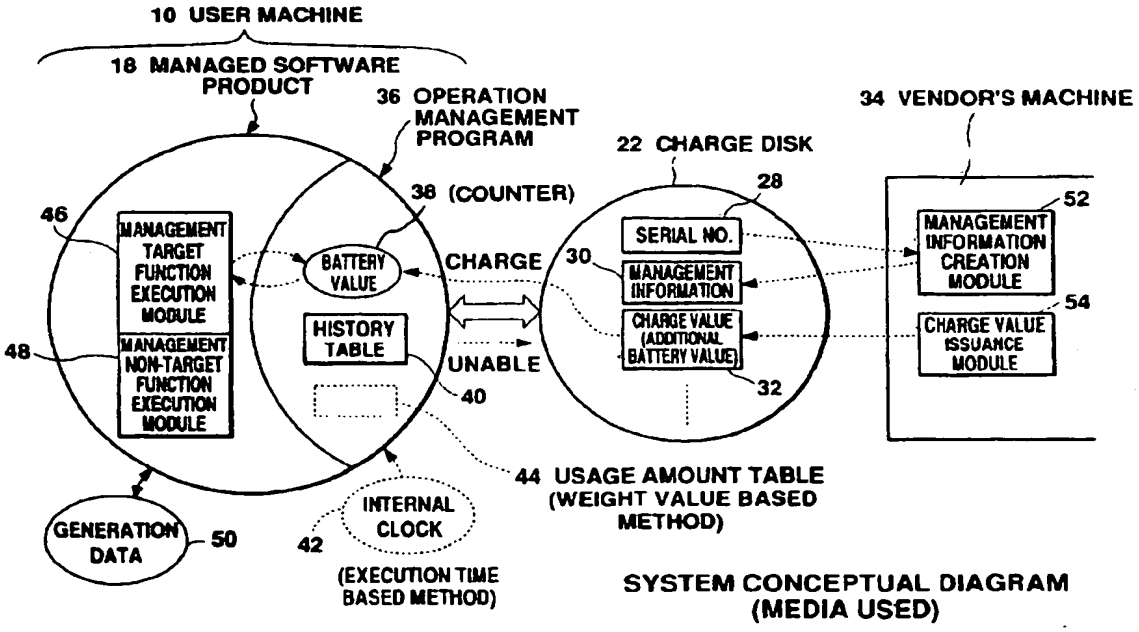


Fig. 3

EP 0 818 748 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 97 30 4946

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl6)
X	US 5 047 928 A (WIEDEMER JOHN D) 10 September 1991 (1991-09-10) * abstract; claims 1-5; figures 1,2 * * column 2, line 35 - column 3, line 6 * * column 4, line 22 - column 14, line 16 * ---	1-20	G06F17/60 G06F1/00
X	US 5 410 598 A (SHEAR VICTOR H) 25 April 1995 (1995-04-25) * abstract; claims 1-12; figures 1-6 * * column 3, line 5 - column 5, line 42 * * column 9, line 25 - line 46 * ---	1-20	
A	FR 2 697 358 A (GENTRALP INTERNATIONAL BV) 29 April 1994 (1994-04-29) * abstract; claims 1-3; figure 2 * * page 1, line 29 - page 2, line 38 * * page 5, line 24 - line 39 * ---	1-20	
A	EP 0 679 979 A (IBM) 2 November 1995 (1995-11-02) * abstract; figures 1,4,7,15 * * column 15, line 39 - column 17, line 42 * -----	1-20	

TECHNICAL FIELDS SEARCHED (Int.Cl6)
G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	27 September 2000	Gardiner, A

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)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 97 30 4946

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

27-09-2000

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5047928 A	10-09-1991	US 4796181 A	03-01-1989
		CA 1281418 A	12-03-1991
		EP 0265183 A	27-04-1988
		JP 63191228 A	08-08-1988
		US 5155680 A	13-10-1992
US 5410598 A	25-04-1995	US 5272750 A	21-12-1993
		US 5050213 A	17-09-1991
		US 4977594 A	11-12-1990
		US 4827508 A	02-05-1989
		AT 133305 T	15-02-1996
		DE 3751678 D	29-02-1996
		DE 3751678 T	14-11-1996
		EP 0329681 A	30-08-1989
		WO 8802960 A	21-04-1988
FR 2697358 A	29-04-1994	NONE	
EP 0679979 A	02-11-1995	US 5689560 A	18-11-1997
		AU 1485695 A	02-11-1995
		BR 9501522 A	21-11-1995
		CA 2145925 A,C	26-10-1995
		CN 1115059 A	17-01-1996
		JP 7295803 A	10-11-1995
		KR 200444 B	15-06-1999

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

THIS PAGE BLANK (USPTO)

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

EP 25660 (2)

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04L 9/00 — A2

(11) International Publication Number: **WO 00/59152**

(43) International Publication Date: 5 October 2000 (05.10.00)

(21) International Application Number: PCT/US00/04983

(22) International Filing Date: 25 February 2000 (25.02.00)

(30) Priority Data:

60/126,614	27 March 1999 (27.03.99)	US
09/290,363	12 April 1999 (12.04.99)	US
09/482,928	13 January 2000 (13.01.00)	US

(71) Applicant: MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, WA 98052 (US).

(72) Inventors: BLINN, Arnold, N.; 9401 NE 27th Street, Bellevue, WA 98004 (US). JONES, Thomas, C.; 23617 NE 6th Street, Redmond, WA 98053–3618 (US).

(74) Agents: ROCCI, Steven, J. et al.; Woodcock Washburn Kurtz Mackiewicz & Norris LLP, 46th floor, One Liberty Place, Philadelphia, PA 19103 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, 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, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, 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), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

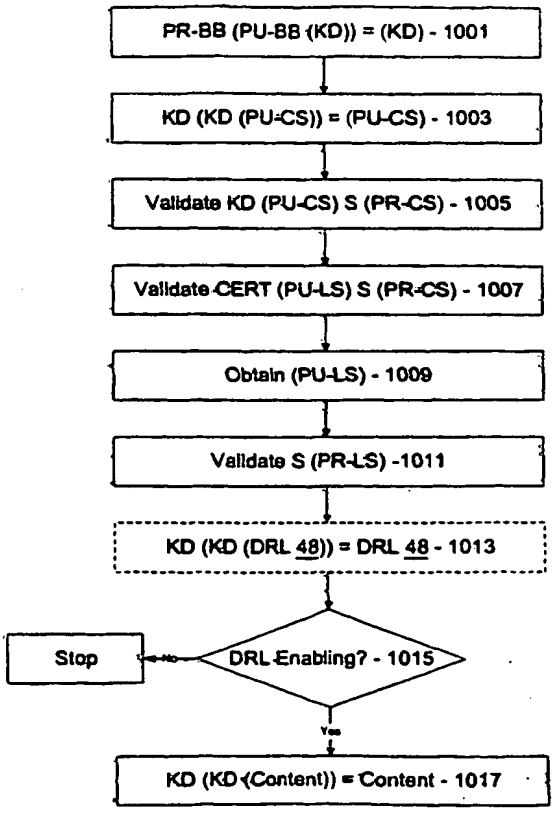
Published

Without international search report and to be republished upon receipt of that report.

(54) Title: METHOD FOR INTERDEPENDENTLY VALIDATING A DIGITAL CONTENT PACKAGE AND A CORRESPONDING DIGITAL LICENSE

(57) Abstract

A method is disclosed for a device to interdependently validate a digital content package having a piece of digital content in an encrypted form, and a corresponding digital license for rendering the digital content. A first key is derived from a source available to the device, and a first digital signature is obtained from the digital content package. The first key is applied to the first digital signature to validate the first digital signature and the digital content package. A second key is derived based on the first digital signature, and a second digital signature is obtained from the license. The second key is applied to the second digital signature to validate the second digital signature and the license.



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		

METHOD FOR INTERDEPENDENTLY VALIDATING A DIGITAL CONTENT PACKAGE AND A CORRESPONDING DIGITAL LICENSE

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. Patent Application No. 09/290,363, filed April 12, 1999 and entitled "ENFORCEMENT ARCHITECTURE AND METHOD FOR DIGITAL RIGHTS MANAGEMENT", and claims the benefit of U.S. Provisional Application No. 60/21,614, filed March 27, 1999 and entitled "ENFORCEMENT ARCHITECTURE AND METHOD FOR DIGITAL RIGHTS MANAGEMENT", both of which are hereby incorporated by reference.

TECHNICAL FIELD

The present invention relates to an architecture for enforcing rights in digital content. More specifically, the present invention relates to such an enforcement architecture that allows access to encrypted digital content only in accordance with parameters specified by license rights acquired by a user of the digital content.

BACKGROUND OF THE INVENTION

Digital rights management and enforcement is highly desirable in connection with digital content such as digital audio, digital video, digital text, digital data, digital multimedia, etc., where such digital content is to be distributed to users. Typical modes of distribution include tangible devices such as a magnetic (floppy) disk, a magnetic tape, an optical (compact) disk (CD), etc., and intangible media such as an electronic bulletin board, an electronic network, the Internet, etc. Upon being received by the user, such user renders or 'plays' the digital content with the aid of an appropriate rendering device such as a media player on a personal computer or the like.

Typically, a content owner or rights-owner, such as an author, a publisher, a broadcaster, etc. (hereinafter "content owner"), wishes to distribute such digital content to a user or recipient in exchange for a license fee or some other consideration. Such content owner, given the choice, would likely wish to restrict what

the user can do with such distributed digital content. For example, the content owner would like to restrict the user from copying and re-distributing such content to a second user, at least in a manner that denies the content owner a license fee from such second user.

In addition, the content owner may wish to provide the user with the flexibility to purchase different types of use licenses at different license fees, while at the same time holding the user to the terms of whatever type of license is in fact purchased. For example, the content owner may wish to allow distributed digital content to be played only a limited number of times, only for a certain total time, only on a certain type of machine, only on a certain type of media player, only by a certain type of user, etc.

However, after distribution has occurred, such content owner has very little if any control over the digital content. This is especially problematic in view of the fact that practically every new or recent personal computer includes the software and hardware necessary to make an exact digital copy of such digital content, and to download such exact digital copy to a write-able magnetic or optical disk, or to send such exact digital copy over a network such as the Internet to any destination.

Of course, as part of the legitimate transaction where the license fee was obtained, the content owner may require the user of the digital content to promise not to re-distribute such digital content. However, such a promise is easily made and easily broken. A content owner may attempt to prevent such re-distribution through any of several known security devices, usually involving encryption and decryption. However, there is likely very little that prevents a mildly determined user from decrypting encrypted digital content, saving such digital content in an un-encrypted form, and then re-distributing same.

A need exists, then, for providing an enforcement architecture and method that allows the controlled rendering or playing of arbitrary forms of digital content, where such control is flexible and definable by the content owner of such digital content. A need also exists for providing a controlled rendering environment

on a computing device such as a personal computer, where the rendering environment includes at least a portion of such enforcement architecture. Such controlled rendering environment allows that the digital content will only be rendered as specified by the content owner, even though the digital content is to be rendered on a computing device which is not under the control of the content owner.

Further, a need exists for a trusted component running on the computing device, where the trusted component enforces the rights of the content owner on such computing device in connection with a piece of digital content, even against attempts by the user of such computing device to access such digital content in ways not permitted by the content owner. As but one example, such a trusted software component prevents a user of the computing device from making a copy of such digital content, except as otherwise allowed for by the content owner thereof.

SUMMARY OF THE INVENTION

The aforementioned needs are satisfied at least in part by an enforcement architecture and method for digital rights management, where the architecture and method enforce rights in protected (secure) digital content available on a medium such as the Internet, an optical disk, etc. For purposes of making content available, the architecture includes a content server from which the digital content is accessible over the Internet or the like in an encrypted form. The content server may also supply the encrypted digital content for recording on an optical disk or the like, wherein the encrypted digital content may be distributed on the optical disk itself. At the content server, the digital content is encrypted using an encryption key, and public / private key techniques are employed to bind the digital content with a digital license at the user's computing device or client machine.

When a user attempts to render the digital content on a computing device, the rendering application invokes a Digital Rights Management (DRM) system on such user's computing device. If the user is attempting to render the digital content for the first time, the DRM system either directs the user to a license server to obtain a license to render such digital content in the manner sought, or transparently obtains

such license from such license server without any action necessary on the part of the user. The license includes:

- a decryption key (KD) that decrypts the encrypted digital content;
- a description of the rights (play, copy, etc.) conferred by the license and related conditions (begin date, expiration date, number of plays, etc.), where such description is in a digitally readable form; and
- a digital signature that ensures the integrity of the license.

The user cannot decrypt and render the encrypted digital content without obtaining such a license from the license server. The obtained license is stored in a license store in the user's computing device.

Importantly, the license server only issues a license to a DRM system that is 'trusted' (i.e., that can authenticate itself). To implement 'trust', the DRM system is equipped with a 'black box' that performs decryption and encryption functions for such DRM system. The black box includes a public / private key pair, a version number and a unique signature, all as provided by an approved certifying authority. The public key is made available to the license server for purposes of encrypting portions of the issued license, thereby binding such license to such black box. The private key is available to the black box only, and not to the user or anyone else, for purposes of decrypting information encrypted with the corresponding public key. The DRM system is initially provided with a black box with a public / private key pair, and the user is prompted to download from a black box server an updated secure black box when the user first requests a license. The black box server provides the updated black box, along with a unique public/private key pair. Such updated black box is written in unique executable code that will run only on the user's computing device, and is re-updated on a regular basis. When a user requests a license, the client machine sends the black box public key, version number, and signature to the license server, and such license server issues a license only if the version number is current and the signature is valid. A license request also includes an identification of the digital content for which a license is requested and a key ID that identifies the

decryption key associated with the requested digital content. The license server uses the black box public key to encrypt the decryption key, and the decryption key to encrypt the license terms, then downloads the encrypted decryption key and encrypted license terms to the user's computing device along with a license signature.

Once the downloaded license has been stored in the DRM system license store, the user can render the digital content according to the rights conferred by the license and specified in the license terms. When a request is made to render the digital content, the black box is caused to decrypt the decryption key and license terms, and a DRM system license evaluator evaluates such license terms. The black box decrypts the encrypted digital content only if the license evaluation results in a decision that the requestor is allowed to play such content. The decrypted content is provided to the rendering application for rendering.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing summary, as well as the following detailed description of the embodiments of the present invention, will be better understood when read in conjunction with the appended drawings. For the purpose of illustrating the invention, there are shown in the drawings embodiments which are presently preferred. As should be understood, however, the invention is not limited to the precise arrangements and instrumentalities shown. In the drawings:

Fig. 1 is a block diagram showing an enforcement architecture in accordance with one embodiment of the present invention;

Fig. 2 is a block diagram of the authoring tool of the architecture of Fig. 1 in accordance with one embodiment of the present invention;

Fig. 3 is a block diagram of a digital content package having digital content for use in connection with the architecture of Fig. 1 in accordance with one embodiment of the present invention;

Fig. 4 is a block diagram of the user's computing device of Fig. 1 in accordance with one embodiment of the present invention;

Figs. 5A and 5B are flow diagrams showing the steps performed in connection with the Digital Rights Management (DRM) system of the computing device of Fig. 4 to render content in accordance with one embodiment of the present invention;

Fig. 6 is a flow diagram showing the steps performed in connection with the DRM system of Fig. 4 to determine whether any valid, enabling licenses are present in accordance with one embodiment of the present invention;

Fig. 7 is a flow diagram showing the steps performed in connection with the DRM system of Fig. 4 to obtain a license in accordance with one embodiment of the present invention;

Fig. 8 is a block diagram of a digital license for use in connection with the architecture of Fig. 1 in accordance with one embodiment of the present invention;

Fig. 9 is a flow diagram showing the steps performed in connection with the DRM system of Fig. 4 to obtain a new black box in accordance with one embodiment of the present invention;

Fig. 10 is a flow diagram showing the key transaction steps performed in connection with the DRM system of Fig. 4 to validate a license and a piece of digital content and render the content in accordance with one embodiment of the present invention;

Fig. 11 is a block diagram showing the license evaluator of Fig. 4 along with a Digital Rights License (DRL) of a license and a language engine for interpreting the DRL in accordance with one embodiment of the present invention; and

Fig. 12 is a block diagram representing a general purpose computer system in which aspects of the present invention and/or portions thereof may be incorporated.

Detailed Description of the Invention

Referring to the drawings in details, wherein like numerals are used to indicate like elements throughout, there is shown in Fig. 1 an enforcement architecture 10 in accordance with one embodiment of the present invention. Overall, the enforcement architecture 10 allows an owner of digital content 12 to specify license rules that must be satisfied before such digital content 12 is allowed to be rendered on a user's computing device 14. Such license rules are embodied within a digital license 16 that the user / user's computing device 14 (hereinafter, such terms are interchangeable unless circumstances require otherwise) must obtain from the content owner or an agent thereof. The digital content 12 is distributed in an encrypted form, and may be distributed freely and widely. Preferably, the decrypting key (KD) for decrypting the digital content 12 is included with the license 16.

COMPUTER ENVIRONMENT

Fig. 12 and the following discussion are intended to provide a brief general description of a suitable computing environment in which the present invention and/or portions thereof may be implemented. Although not required, the invention is described in the general context of computer-executable instructions, such as program modules, being executed by a computer, such as a client workstation or a server. Generally, program modules include routines, programs, objects, components, data structures and the like that perform particular tasks or implement particular abstract data types. Moreover, it should be appreciated that the invention and/or portions thereof may be practiced with other computer system configurations, including hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

As shown in Fig. 12, an exemplary general purpose computing system

includes a conventional personal computer 120 or the like, including a processing unit 121, a system memory 122, and a system bus 18 that couples various system components including the system memory to the processing unit 121. The system bus 18 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory includes read-only memory (ROM) 19 and random access memory (RAM) 20. A basic input/output system 21 (BIOS), containing the basic routines that help to transfer information between elements within the personal computer 120, such as during start-up, is stored in ROM 19.

The personal computer 120 may further include a hard disk drive 22 for reading from and writing to a hard disk (not shown), a magnetic disk drive 128 for reading from or writing to a removable magnetic disk 129, and an optical disk drive 25 for reading from or writing to a removable optical disk 131 such as a CD-ROM or other optical media. The hard disk drive 22, magnetic disk drive 128, and optical disk drive 25 are connected to the system bus 18 by a hard disk drive interface 27, a magnetic disk drive interface 28, and an optical drive interface 29, respectively. The drives and their associated computer-readable media provide non-volatile storage of computer readable instructions, data structures, program modules and other data for the personal computer 20.

Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 129, and a removable optical disk 131, it should be appreciated that other types of computer readable media which can store data that is accessible by a computer may also be used in the exemplary operating environment. Such other types of media include a magnetic cassette, a flash memory card, a digital video disk, a Bernoulli cartridge, a random access memory (RAM), a read-only memory (ROM), and the like.

A number of program modules may be stored on the hard disk, magnetic disk 129, optical disk 131, ROM 19 or RAM 20, including an operating system 30, one or more application programs 136, other program modules 137 and

program data 138. A user may enter commands and information into the personal computer 120 through input devices such as a keyboard 35 and pointing device 142. Other input devices (not shown) may include a microphone, joystick, game pad, satellite disk, scanner, or the like. These and other input devices are often connected to the processing unit 121 through a serial port interface 41 that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, game port, or universal serial bus (USB). A monitor 42 or other type of display device is also connected to the system bus 18 via an interface, such as a video adapter 148. In addition to the monitor 42, a personal computer typically includes other peripheral output devices (not shown), such as speakers and printers. The exemplary system of Fig. 12 also includes a host adapter 50, a Small Computer System Interface (SCSI) bus 156, and an external storage device 162 connected to the SCSI bus 156.

The personal computer 120 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 149. The remote computer 149 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the personal computer 120, although only a memory storage device 150 has been illustrated in Fig. 12. The logical connections depicted in Fig. 12 include a local area network (LAN) 46 and a wide area network (WAN) 47. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet.

When used in a LAN networking environment, the personal computer 120 is connected to the LAN 46 through a network interface or adapter 48. When used in a WAN networking environment, the personal computer 120 typically includes a modem 49 or other means for establishing communications over the wide area network 47, such as the Internet. The modem 49, which may be internal or external, is connected to the system bus 18 via the serial port interface 41. In a networked environment, program modules depicted relative to the personal computer 120, or portions thereof, may be stored in the remote memory storage device. It will be

appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

ARCHITECTURE

Referring again to Fig. 1, in one embodiment of the present invention, the architecture 10 includes an authoring tool 18, a content-key database 20, a content server 22, a license server 24, and a black box server 26, as well as the aforementioned user's computing device 14.

ARCHITECTURE - Authoring Tool 18

The authoring tool 18 is employed by a content owner to package a piece of digital content 12 into a form that is amenable for use in connection with the architecture 10 of the present invention. In particular, the content owner provides the authoring tool 18 with the digital content 12, instructions and/or rules that are to accompany the digital content 12, and instructions and/or rules as to how the digital content 12 is to be packaged. The authoring tool 18 then produces a digital content package 12p having the digital content 12 encrypted according to an encryption / decryption key, and the instructions and/or rules that accompany the digital content 12.

In one embodiment of the present invention, the authoring tool 18 is instructed to serially produce several different digital content 12 packages 12p, each having the same digital content 12 encrypted according to a different encryption / decryption key. As should be understood, having several different packages 12p with the same digital content 12 may be useful for tracking the distribution of such packages 12p / content 12 (hereinafter simply "digital content 12", unless circumstances require otherwise). Such distribution tracking is not ordinarily necessary, but may be used by an investigative authority in cases where the digital content 12 has been illegally sold or broadcast.

In one embodiment of the present invention, the encryption / decryption key that encrypts the digital content 12 is a symmetric key, in that the encryption key is also the decryption key (KD). As will be discussed below in more detail, such decryption key (KD) is delivered to a user's computing device 14 in a hidden form as

part of a license 16 for such digital content 12. Preferably, each piece of digital content 12 is provided with a content ID (or each package 12p is provided with a package ID), each decryption key (KD) has a key ID, and the authoring tool 18 causes the decryption key (KD), key ID, and content ID (or package ID) for each piece of digital content 12 (or each package 12p) to be stored in the content-key database 20. In addition, license data regarding the types of licenses 16 to be issued for the digital content 12 and the terms and conditions for each type of license 16 may be stored in the content-key database 20, or else in another database (not shown). Preferably, the license data can be modified by the content owner at a later time as circumstances and market conditions may require.

In use, the authoring tool 18 is supplied with information including, among other things:

- the digital content 12 to be packaged;
- the type and parameters of watermarking and/or fingerprinting to be employed, if any;
- the type and parameters of data compression to be employed, if any;
- the type and parameters of encryption to be employed;
- the type and parameters of serialization to be employed, if any; and
- the instructions and/or rules that are to accompany the digital content 12.

As is known, a watermark is a hidden, computer-readable signal that is added to the digital content 12 as an identifier. A fingerprint is a watermark that is different for each instance. As should be understood, an instance is a version of the digital content 12 that is unique. Multiple copies of any instance may be made, and any copy is of a particular instance. When a specific instance of digital content 12 is illegally sold or broadcast, an investigative authority can perhaps identify suspects according to the watermark / fingerprint added to such digital content 12.

Data compression may be performed according to any appropriate compression algorithm without departing from the spirit and scope of the present

invention. For example, the .mp3 or .wav compression algorithm may be employed. Of course, the digital content 12 may already be in a compressed state, in which case no additional compression is necessary.

The instructions and/or rules that are to accompany the digital content 12 may include practically any appropriate instructions, rules, or other information without departing from the spirit and scope of the present invention. As will be discussed below, such accompanying instructions / rules / information are primarily employed by the user and the user's computing device 14 to obtain a license 16 to render the digital content 12. Accordingly, such accompanying instructions / rules / information may include an appropriately formatted license acquisition script or the like, as will be described in more detail below. In addition, or in the alternative, such accompanying instructions / rules / information may include 'preview' information designed to provide a user with a preview of the digital content 12.

With the supplied information, the authoring tool 18 then produces one or more packages 12p corresponding to the digital content 12. Each package 12p may then be stored on the content server 22 for distribution to the world.

In one embodiment of the present invention, and referring now to Fig. 2, the authoring tool 18 is a dynamic authoring tool 18 that receives input parameters which can be specified and operated on. Accordingly, such authoring tool 18 can rapidly produce multiple variations of package 12p for multiple pieces of digital content 12. Preferably, the input parameters are embodied in the form of a dictionary 28, as shown, where the dictionary 28 includes such parameters as:

- the name of the input file 29a having the digital content 12;
- the type of encoding that is to take place
- the encryption / decryption key (KD) to be employed,
- the accompanying instructions / rules / information ('header information') to be packaged with the digital content 12 in the package 12p.
- the type of muxing that is to occur: and

- the name of the output file 29b to which the package 12p based on the digital content 12 is to be written.

As should be understood, such dictionary 28 is easily and quickly modifiable by an operator of the authoring tool 18 (human or machine), and therefore the type of authoring performed by the authoring tool 18 is likewise easily and quickly modifiable in a dynamic manner. In one embodiment of the present invention, the authoring tool 18 includes an operator interface (not shown) displayable on a computer screen to a human operator. Accordingly, such operator may modify the dictionary 28 by way of the interface, and further may be appropriately aided and/or restricted in modifying the dictionary 28 by way of the interface.

In the authoring tool 18, and as seen in Fig. 2, a source filter 18a receives the name of the input file 29a having the digital content 12 from the dictionary 28, and retrieves such digital content 12 from such input file and places the digital content 12 into a memory 29c such as a RAM or the like. An encoding filter 18b then performs encoding on the digital content 12 in the memory 29c to transfer the file from the input format to the output format according to the type of encoding specified in the dictionary 28 (i.e., .wav to .asp, .mp3 to .asp, etc.), and places the encoded digital content 12 in the memory 29c. As shown, the digital content 12 to be packaged (music, e.g.) is received in a compressed format such as the .wav or .mp3 format, and is transformed into a format such as the .asp (active streaming protocol) format. Of course, other input and output formats may be employed without departing from the spirit and scope of the present invention.

Thereafter, an encryption filter 18c encrypts the encoded digital content 12 in the memory 29c according to the encryption / decryption key (KD) specified in the dictionary 28, and places the encrypted digital content 12 in the memory 29c. A header filter 18d then adds the header information specified in the dictionary 28 to the encrypted digital content 12 in the memory 29c.

As should be understood, depending on the situation, the package 12p may include multiple streams of temporally aligned digital content 12 (one stream

being shown in Fig. 2), where such multiple streams are multiplexed (i.e., 'muxed'). Accordingly, a mux filter 18e performs muxing on the header information and encrypted digital content 12 in the memory 29c according to the type of muxing specified in the dictionary 28, and places the result in the memory 29c. A file writer filter 18f then retrieves the result from the memory 29c and writes such result to the output file 29b specified in the dictionary 28 as the package 12p.

It should be noted that in certain circumstances, the type of encoding to be performed will not normally change. Since the type of muxing typically is based on the type of encoding, it is likewise the case that the type of muxing will not normally change, either. If this is in fact the case, the dictionary 28 need not include parameters on the type of encoding and/or the type of muxing. Instead, it is only necessary that the type of encoding be 'hardwired' into the encoding filter and/or that the type of muxing be 'hardwired' into the mux filter. Of course, as circumstance require, the authoring tool 18 may not include all of the aforementioned filters, or may include other filters, and any included filter may be hardwired or may perform its function according to parameters specified in the dictionary 28, all without departing from the spirit and scope of the present invention.

Preferably, the authoring tool 18 is implemented on an appropriate computer, processor, or other computing machine by way of appropriate software. The structure and operation of such machine and such software should be apparent based on the disclosure herein and therefore do not require any detailed discussion in the present disclosure.

ARCHITECTURE - Content Server 22

Referring again to Fig. 1, in one embodiment of the present invention, the content server 22 distributes or otherwise makes available for retrieval the packages 12p produced by the authoring tool 18. Such packages 12p may be distributed as requested by the content server 22 by way of any appropriate distribution channel without departing from the spirit and scope of the present invention. For example, such distribution channel may be the Internet or another network, an electronic bulletin

board, electronic mail, or the like. In addition, the content server 22 may be employed to copy the packages 12p onto magnetic or optical disks or other storage devices, and such storage devices may then be distributed.

It will be appreciated that the content server 22 distributes packages 12p without regard to any trust or security issues. As discussed below, such issues are dealt with in connection with the license server 24 and the relationship between such license server 24 and the user's computing device 14. In one embodiment of the present invention, the content server 22 freely releases and distributes packages 12p having digital content 12 to any distributee requesting same. However, the content server 22 may also release and distribute such packages 12p in a restricted manner without departing from the spirit and scope of the present invention. For example, the content server 22 may first require payment of a pre-determined distribution fee prior to distribution, or may require that a distributee identify itself, or may indeed make a determination of whether distribution is to occur based on an identification of the distributee.

In addition, the content server 22 may be employed to perform inventory management by controlling the authoring tool 18 to generate a number of different packages 12p in advance to meet an anticipated demand. For example, the server could generate 100 packages 12p based on the same digital content 12, and serve each package 12p 10 times. As supplies of packages 12p dwindle to 20, for example, the content server 22 may then direct the authoring tool 18 to generate 80 additional packages 12p, again for example.

Preferably, the content server 22 in the architecture 10 has a unique public / private key pair (PU-CS, PR-CS) that is employed as part of the process of evaluating a license 16 and obtaining a decryption key (KD) for decrypting corresponding digital content 12, as will be explained in more detail below. As is known, a public / private key pair is an asymmetric key, in that what is encrypted in one of the keys in the key pair can only be decrypted by the other of the keys in the key pair. In a public / private key pair encryption system, the public key may be made

known to the world, but the private key should always be held in confidence by the owner of such private key. Accordingly, if the content server 22 encrypts data with its private key (PR-CS), it can send the encrypted data out into the world with its public key (PU-CS) for decryption purposes. Correspondingly, if an external device wants to send data to the content server 22 so that only such content server 22 can decrypt such data, such external device must first obtain the public key of the content server 22 (PU-CS) and then must encrypt the data with such public key. Accordingly, the content server 22 (and only the content server 22) can then employ its private key (PR-CS) to decrypt such encrypted data.

As with the authoring tool 18, the content server 22 is implemented on an appropriate computer, processor, or other computing machine by way of appropriate software. The structure and operation of such machine and such software should be apparent based on the disclosure herein and therefore do not require any detailed discussion in the present disclosure. Moreover, in one embodiment of the present invention, the authoring tool 18 and the content server 22 may reside on a single computer, processor, or other computing machine, each in a separate work space. It should be recognized, moreover, that the content server 22 may in certain circumstances include the authoring tool 18 and/or perform the functions of the authoring tool 18, as discussed above.

Structure of Digital Content Package 12p

Referring now to Fig. 3, in one embodiment of the present invention, the digital content package 12p as distributed by the content server 22 includes:

- the digital content 12 encrypted with the encryption / decryption key (KD), as was discussed above (i.e., (KD(CONTENT)));
- the content ID (or package ID) of such digital content 12 (or package 12p);
- the key ID of the decryption key (KD);
- license acquisition information, preferably in an un-encrypted form; and

- the key KD encrypting the content server 22 public key (PU-CS), signed by the content server 22 private key (PR-CS) (i.e., (KD (PU-CS) S (PR-CS))).

With regard to (KD (PU-CS) S (PR-CS)), it is to be understood that such item is to be used in connection with validating the digital content 12 and/or package 12p, as will be explained below. Unlike a certificate with a digital signature (see below), the key (PU-CS) is not necessary to get at (KD (PU-CS)). Instead, the key (PU-CS) is obtained merely by applying the decryption key (KD). Once so obtained, such key (PU-CS) may be employed to test the validity of the signature (S (PR-CS)).

It should also be understood that for such package 12p to be constructed by the authoring tool 18, such authoring tool 18 must already possess the license acquisition information and (KD (PU-CS) S (PR-CS)), presumably as header information supplied by the dictionary 28. Moreover, the authoring tool 18 and the content server 22 must presumably interact to construct (KD (PU-CS) S (PR-CS)). Such interaction may for example include the steps of:

- the content server 22 sending (PU-CS) to the authoring tool 18;
- the authoring tool 18 encrypting (PU-CS) with (KD) to produce (KD (PU-CS));
- the authoring tool 18 sending (KD (PU-CS)) to the content server 22;
- the content server 22 signing (KD (PU-CS)) with (PR-CS) to produce (KD (PU-CS) S (PR-CS)); and
- the content server 22 sending (KD (PU-CS) S (PR-CS)) to the authoring tool 18.

ARCHITECTURE - License Server 24

Referring again to Fig. 1, in one embodiment of the present invention, the license server 24 performs the functions of receiving a request for a license 16 from a user's computing device 14 in connection with a piece of digital content 12,

determining whether the user's computing device 14 can be trusted to honor an issued license 16, negotiating such a license 16, constructing such license 16, and transmitting such license 16 to the user's computing device 14. Preferably, such transmitted license 16 includes the decryption key (KD) for decrypting the digital content 12. Such license server 24 and such functions will be explained in more detail below. Preferably, and like the content server 22, the license server 24 in the architecture 10 has a unique public / private key pair (PU-LS, PR-LS) that is employed as part of the process of evaluating a license 16 and obtaining a decryption key (KD) for decrypting corresponding digital content 12, as will be explained in more detail below.

As with the authoring tool 18 and the content server 22, the license server 24 is implemented on an appropriate computer, processor, or other computing machine by way of appropriate software. The structure and operation of such machine and such software should be apparent based on the disclosure herein and therefore do not require any detailed discussion in the present disclosure. Moreover, in one embodiment of the present invention the authoring tool 18 and/or the content server 22 may reside on a single computer, processor, or other computing machine together with the license server 24, each in a separate work space.

In one embodiment of the present invention, prior to issuance of a license 16, the license server 24 and the content server 22 enter into an agency agreement or the like, wherein the license server 24 in effect agrees to be the licensing authority for at least a portion of the digital content 12 distributed by the content server 22. As should be understood, one content server 22 may enter into an agency agreement or the like with several license servers 24, and/or one license server 24 may enter into an agency agreement or the like with several content servers 22, all without departing from the spirit and scope of the present invention.

Preferably, the license server 24 can show to the world that it does in fact have the authority to issue a license 16 for digital content 12 distributed by the content server 22. To do so, it is preferable that the license server 24 send to the content server 22 the license server 24 public key (PU-LS), and that the content server

22 then send to the license server 24 a digital certificate containing PU-LS as the contents signed by the content server 22 private key (CERT (PU-LS) S (PR-CS)). As should be understood, the contents (PU-LS) in such certificate can only be accessed with the content server 22 public key (PU-CS). As should also be understood, in general, a digital signature of underlying data is an encrypted form of such data, and will not match such data when decrypted if such data has been adulterated or otherwise modified.

As a licensing authority in connection with a piece of digital content 12, and as part of the licensing function, the license server 24 must have access to the decryption key (KD) for such digital content 12. Accordingly, it is preferable that license server 24 have access to the content-key database 20 that has the decryption key (KD), key ID, and content ID (or package ID) for such digital content 12 (or package 12p).

ARCHITECTURE - Black Box Server 26

Still referring to Fig. 1, in one embodiment of the present invention, the black box server 26 performs the functions of installing and/or upgrading a new black box 30 in a user's computing device 14. As will be explained in more detail below, the black box 30 performs encryption and decryption functions for the user's computing device 14. As will also be explained in more detail below, the black box 30 is intended to be secure and protected from attack. Such security and protection is provided, at least in part, by upgrading the black box 30 to a new version as necessary by way of the black box server 26, as will be explained in more detail below.

As with the authoring tool 18, the content server 22, and the license server 24, the black box server 26 is implemented on an appropriate computer, processor, or other computing machine by way of appropriate software. The structure and operation of such machine and such software should be apparent based on the disclosure herein and therefore do not require any detailed discussion in the present disclosure. Moreover, in one embodiment of the present invention the license server 24, the authoring tool 18, and/or the content server 22 may reside on a single computer,

processor, or other computing machine together with the black box server 26, each in a separate work space. Note, though, that for security purposes, it may be wise to have the black box server 26 on a separate machine.

ARCHITECTURE - User's Computing Device 14

Referring now to Fig. 4, in one embodiment of the present invention, the user's computing device 14 is a personal computer or the like, having elements including a keyboard, a mouse, a screen, a processor, RAM, ROM, a hard drive, a floppy drive, a CD player, and/or the like. However, the user's computing device 14 may also be a dedicated viewing device such as a television or monitor, a dedicated audio device such as a stereo or other music player, a dedicated printer, or the like, among other things, all without departing from the spirit and scope of the present invention.

The content owner for a piece of digital content 12 must trust that the user's computing device 14 will abide by the rules specified by such content owner, i.e. that the digital content 12 will not be rendered unless the user obtains a license 16 that permits the rendering in the manner sought. Preferably, then, the user's computing device 14 must provide a trusted component or mechanism 32 that can satisfy to the content owner that such computing device 14 will not render the digital content 12 except according to the license rules embodied in the license 16 associated with the digital content 12 and obtained by the user.

Here, the trusted mechanism 32 is a Digital Rights Management (DRM) system 32 that is enabled when a user requests that a piece of digital content 12 be rendered, that determines whether the user has a license 16 to render the digital content 12 in the manner sought, that effectuates obtaining such a license 16 if necessary, that determines whether the user has the right to play the digital content 12 according to the license 16, and that decrypts the digital content 12 for rendering purposes if in fact the user has such right according to such license 16. The contents and function of the DRM system 32 on the user's computing device 14 and in connection with the architecture 10 are described below.

DRM SYSTEM 32

The DRM system 32 performs four main functions with the architecture 10 disclosed herein: (1) content acquisition, (2) license acquisition, (3) content rendering, and (4) black box 30 installation / update. Preferably, any of the functions can be performed at any time, although it is recognized that some of the functions already require that digital content 12 be acquired.

DRM SYSTEM 32 - Content Acquisition

Acquisition of digital content 12 by a user and/or the user's computing device 14 is typically a relatively straight-forward matter and generally involves placing a file having encrypted digital content 12 on the user's computing device 14. Of course, to work with the architecture 10 and the DRM system 32 disclosed herein, it is necessary that the encrypted digital content 12 be in a form that is amenable to such architecture 10 and DRM system 32, such as the digital package 12p as will be described below.

As should be understood, the digital content 12 may be obtained in any manner from a content server 22, either directly or indirectly, without departing from the spirit and scope of the present invention. For example, such digital content 12 may be downloaded from a network such as the Internet, located on an obtained optical or magnetic disk or the like, received as part of an E-mail message or the like, or downloaded from an electronic bulletin board or the like.

Such digital content 12, once obtained, is preferably stored in a manner such that the obtained digital content 12 is accessible by a rendering application 34 (to be described below) running on the computing device 14, and by the DRM system 32. For example, the digital content 12 may be placed as a file on a hard drive (not shown) of the user's computing device 14, or on a network server (not shown) accessible to the computing device 14. In the case where the digital content 12 is obtained on an optical or magnetic disk or the like, it may only be necessary that such disk be present in an appropriate drive (not shown) coupled to the user's computing device 14.

In the present invention, it is not envisioned that any special tools are

necessary to acquire digital content 12, either from the content server 22 as a direct distribution source or from some intermediary as an indirect distribution source. That is, it is preferable that digital content 12 be as easily acquired as any other data file. However, the DRM system 32 and/or the rendering application 34 may include an interface (not shown) designed to assist the user in obtaining digital content 12 . For example, the interface may include a web browser especially designed to search for digital content 12, links to pre-defined Internet web sites that are known to be sources of digital content 12, and the like.

DRM SYSTEM 32 - Content Rendering, Part 1

Referring now to Fig. 5A, in one embodiment of the present invention, assuming the encrypted digital content 12 has been distributed to and received by a user and placed by the user on the computing device 14 in the form of a stored file, the user will attempt to render the digital content 12 by executing some variation on a render command (step 501). For example, such render command may be embodied as a request to 'play' or 'open' the digital content 12. In some computing environments, such as for example the "MICROSOFT WINDOWS" operating system, distributed by MICROSOFT Corporation of Redmond, Washington, such play or open command may be as simple as 'clicking' on an icon representative of the digital content 12. Of course, other embodiments of such render command may be employed without departing from the spirit and scope of the present invention. In general, such render command may be considered to be executed whenever a user directs that a file having digital content 12 be opened, run, executed, and/or the like.

Importantly, and in addition, such render command may be embodied as a request to copy the digital content 12 to another form, such as to a printed form, a visual form, an audio form, etc. As should be understood, the same digital content 12 may be rendered in one form, such as on a computer screen, and then in another form, such as a printed document. In the present invention, each type of rendering is performed only if the user has the right to do so, as will be explained below.

In one embodiment of the present invention, the digital content 12 is in

the form of a digital file having a file name ending with an extension, and the computing device 14 can determine based on such extension to start a particular kind of rendering application 34. For example, if the file name extension indicates that the digital content 12 is a text file, the rendering application 34 is some form of word processor such as the "MICROSOFT WORD", distributed by MICROSOFT Corporation of Redmond, Washington. Likewise, if the file name extension indicates that the digital content 12 is an audio, video, and/or multimedia file, the rendering application 34 is some form of multimedia player, such as "MICROSOFT MEDIA PLAYER", also distributed by MICROSOFT Corporation of Redmond, Washington.

Of course, other methods of determining a rendering application may be employed without departing from the spirit and scope of the present invention. As but one example, the digital content 12 may contain meta-data in an un-encrypted form (i.e., the aforementioned header information), where the meta-data includes information on the type of rendering application 34 necessary to render such digital content 12.

Preferably, such rendering application 34 examines the digital content 12 associated with the file name and determines whether such digital content 12 is encrypted in a rights-protected form (steps 503, 505). If not protected, the digital content 12 may be rendered without further ado (step 507). If protected, the rendering application 34 determines from the encrypted digital content 12 that the DRM system 32 is necessary to play such digital content 12. Accordingly, such rendering application 34 directs the user's computing device 14 to run the DRM system 32 thereon (step 509). Such rendering application 34 then calls such DRM system 32 to decrypt the digital content 12 (step 511). As will be discussed in more detail below, the DRM system 32 in fact decrypts the digital content 12 only if the user has a valid license 16 for such digital content 12 and the right to play the digital content 12 according to the license rules in the valid license 16. Preferably, once the DRM system 32 has been called by the rendering application 34, such DRM system 32 assumes control from the rendering application 34, at least for purposes of determining whether

the user has a right to play such digital content 12 (step 513).

DRM System 32 Components

In one embodiment of the present invention, and referring again to Fig. 4, the DRM system 32 includes a license evaluator 36, the black box 30, a license store 38, and a state store 40.

DRM System 32 Components - License Evaluator 36

The license evaluator 36 locates one or more licenses 16 that correspond to the requested digital content 12, determines whether such licenses 16 are valid, reviews the license rules in such valid licenses 16, and determines based on the reviewed license rules whether the requesting user has the right to render the requested digital content 12 in the manner sought, among other things. As should be understood, the license evaluator 36 is a trusted component in the DRM system 32. In the present disclosure, to be 'trusted' means that the license server 24 (or any other trusting element) is satisfied that the trusted element will carry out the wishes of the owner of the digital content 12 according to the rights description in the license 16, and that a user cannot easily alter such trusted element for any purpose, nefarious or otherwise.

The license evaluator 36 has to be trusted in order to ensure that such license evaluator 36 will in fact evaluate a license 16 properly, and to ensure that such license evaluator 36 has not been adulterated or otherwise modified by a user for the purpose of bypassing actual evaluation of a license 16. Accordingly, the license evaluator 36 is run in a protected or shrouded environment such that the user is denied access to such license evaluator 36. Other protective measures may of course be employed in connection with the license evaluator 36 without departing from the spirit and scope of the present invention.

DRM System 32 Components - Black Box 30

Primarily, and as was discussed above, the black box 30 performs encryption and decryption functions in the DRM system 32. In particular, the black box 30 works in conjunction with the license evaluator 36 to decrypt and encrypt certain information as part of the license evaluation function. In addition, once the

license evaluator 36 determines that a user does in fact have the right to render the requested digital content 12 in the manner sought, the black box 30 is provided with a decryption key (KD) for such digital content 12, and performs the function of decrypting such digital content 12 based on such decryption key (KD).

The black box 30 is also a trusted component in the DRM system 32. In particular, the license server 24 must trust that the black box 30 will perform the decryption function only in accordance with the license rules in the license 16, and also trust that such black box 30 will not operate should it become adulterated or otherwise modified by a user for the nefarious purpose of bypassing actual evaluation of a license 16. Accordingly, the black box 30 is also run in a protected or shrouded environment such that the user is denied access to such black box 30. Again, other protective measures may be employed in connection with the black box 30 without departing from the spirit and scope of the present invention. Preferably, and like the content server 22 and license server 24, the black box 30 in the DRM system 32 has a unique public / private key pair (PU-BB, PR-BB) that is employed as part of the process of evaluating the license 16 and obtaining a decryption key (KD) for decrypting the digital content 12, as will be described in more detail below.

DRM System 32 Components - License Store 38

The license store 38 stores licenses 16 received by the DRM system 32 for corresponding digital content 12. The license store 38 itself need not be trusted since the license store 38 merely stores licenses 16, each of which already has trust components built thereinto, as will be described below. In one embodiment of the present invention, the license store 38 is merely a sub-directory of a drive such as a hard disk drive or a network drive. However, the license store 38 may be embodied in any other form without departing from the spirit and scope of the present invention, so long as such license store 38 performs the function of storing licenses 16 in a location relatively convenient to the DRM system 32.

DRM System 32 Components - State Store 40

The state store 40 performs the function of maintaining state

information corresponding to licenses 16 presently or formerly in the license store 38. Such state information is created by the DRM system 32 and stored in the state store 40 as necessary. For example, if a particular license 16 only allows a pre-determined number of renderings of a piece of corresponding digital content 12, the state store 40 maintains state information on how many renderings have in fact taken place in connection with such license 16. The state store 40 continues to maintain state information on licenses 16 that are no longer in the license store 38 to avoid the situation where it would otherwise be advantageous to delete a license 16 from the license store 38 and then obtain an identical license 16 in an attempt to delete the corresponding state information from the state store 40.

The state store 40 also has to be trusted in order to ensure that the information stored therein is not reset to a state more favorable to a user. Accordingly, the state store 40 is likewise run in a protected or shrouded environment such that the user is denied access to such state store 40. Once again, other protective measures may of course be employed in connection with the state store 40 without departing from the spirit and scope of the present invention. For example, the state store 40 may be stored by the DRM system 32 on the computing device 14 in an encrypted form.

DRM SYSTEM 32 - Content Rendering, Part 2

Referring again to Fig. 5A, and again discussing content rendering in one embodiment of the present invention, once the DRM system 32 has assumed control from the calling rendering application 34, such DRM system 32 then begins the process of determining whether the user has a right to render the requested digital content 12 in the manner sought. In particular, the DRM system 32 either locates a valid, enabling license 16 in the license store (steps 515, 517) or attempts to acquire a valid, enabling license 16 from the license server 24 (i.e. performs the license acquisition function as discussed below and as shown in Fig. 7).

As a first step, and referring now to Fig. 6, the license evaluator 36 of such DRM system 32 checks the license store 38 for the presence of one or more received licenses 16 that correspond to the digital content 12 (step 601). Typically, the

license 16 is in the form of a digital file, as will be discussed below, although it will be recognized that the license 16 may also be in other forms without departing from the spirit and scope of the present invention. Typically, the user will receive the digital content 12 without such license 16, although it will likewise be recognized that the digital content 12 may be received with a corresponding license 16 without departing from the spirit and scope of the present invention.

As was discussed above in connection with Fig. 3, each piece of digital content 12 is in a package 12p with a content ID (or package ID) identifying such digital content 12 (or package 12p), and a key ID identifying the decryption key (KD) that will decrypt the encrypted digital content 12. Preferably, the content ID (or package ID) and the key ID are in an un-encrypted form. Accordingly, and in particular, based on the content ID of the digital content 12, the license evaluator 36 looks for any license 16 in the license store 38 that contains an identification of applicability to such content ID. Note that multiple such licenses 16 may be found, especially if the owner of the digital content 12 has specified several different kinds of licenses 16 for such digital content 12, and the user has obtained multiple ones of such licenses 16. If in fact the license evaluator 36 does not find in the license store 38 any license 16 corresponding to the requested digital content 12, the DRM system 32 may then perform the function of license acquisition (step 519 of Fig. 5), to be described below.

Assume now that the DRM system 32 has been requested to render a piece of digital content 12, and one or more licenses 16 corresponding thereto are present in the license store 38. In one embodiment of the present invention, then, the license evaluator 36 of the DRM system 32 proceeds to determine for each such license 16 whether such license 16 itself is valid (steps 603 and 605 of Fig. 6). Preferably, and in particular, each license 16 includes a digital signature 26 based on the content 28 of the license 16. As should be understood, the digital signature 26 will not match the license 16 if the content 28 has been adulterated or otherwise modified. Thus, the license evaluator 36 can determine based on the digital signature 26 whether the

content 28 is in the form that it was received from the license server 24 (i.e., is valid). If no valid license 16 is found in the license store 38, the DRM system 32 may then perform the license acquisition function described below to obtain such a valid license 16.

Assuming that one or more valid licenses 16 are found, for each valid license 16, the license evaluator 36 of the DRM system 32 next determines whether such valid license 16 gives the user the right to render the corresponding digital content 12 in the manner desired (i.e., is enabling) (steps 607 and 609). In particular, the license evaluator 36 determines whether the requesting user has the right to play the requested digital content 12 based on the rights description in each license 16 and based on what the user is attempting to do with the digital content 12. For example, such rights description may allow the user to render the digital content 12 into a sound, but not into a decrypted digital copy.

As should be understood, the rights description in each license 16 specifies whether the user has rights to play the digital content 12 based on any of several factors, including who the user is, where the user is located, what type of computing device 14 the user is using, what rendering application 34 is calling the DRM system 32, the date, the time, etc. In addition, the rights description may limit the license 16 to a pre-determined number of plays, or pre-determined play time, for example. In such case, the DRM system 32 must refer to any state information with regard to the license 16, (i.e., how many times the digital content 12 has been rendered, the total amount of time the digital content 12 has been rendered, etc.), where such state information is stored in the state store 40 of the DRM system 32 on the user's computing device 14.

Accordingly, the license evaluator 36 of the DRM system 32 reviews the rights description of each valid license 16 to determine whether such valid license 16 confers the rights sought to the user. In doing so, the license evaluator 36 may have to refer to other data local to the user's computing device 14 to perform a determination of whether the user has the rights sought. As seen in Fig. 4, such data

may include an identification 42 of the user's computing device (machine) 14 and particular aspects thereof, an identification 44 of the user and particular aspects thereof, an identification of the rendering application 34 and particular aspects thereof, a system clock 46, and the like. If no valid license 16 is found that provides the user with the right to render the digital content 12 in the manner sought, the DRM system 32 may then perform the license acquisition function described below to obtain such a license 16, if in fact such a license 16 is obtainable.

Of course, in some instances the user cannot obtain the right to render the digital content 12 in the manner requested, because the content owner of such digital content 12 has in effect directed that such right not be granted. For example, the content owner of such digital content 12 may have directed that no license 16 be granted to allow a user to print a text document, or to copy a multimedia presentation into an un-encrypted form. In one embodiment of the present invention, the digital content 12 includes data on what rights are available upon purchase of a license 16, and types of licenses 16 available. However, it will be recognized that the content owner of a piece of digital content 12 may at any time change the rights currently available for such digital content 12 by changing the licenses 16 available for such digital content 12.

DRM SYSTEM 32 - License Acquisition

Referring now to Fig. 7, if in fact the license evaluator 36 does not find in the license store 38 any valid, enabling license 16 corresponding to the requested digital content 12, the DRM system 32 may then perform the function of license acquisition. As shown in Fig. 3, each piece of digital content 12 is packaged with information in an un-encrypted form regarding how to obtain a license 16 for rendering such digital content 12 (i.e., license acquisition information).

In one embodiment of the present invention, such license acquisition information may include (among other things) types of licenses 16 available, and one or more Internet web sites or other site information at which one or more appropriate license servers 24 may be accessed, where each such license server 24 is in fact capable

of issuing a license 16 corresponding to the digital content 12. Of course, the license 16 may be obtained in other manners without departing from the spirit and scope of the present invention. For example, the license 16 may be obtained from a license server 24 at an electronic bulletin board, or even in person or via regular mail in the form of a file on a magnetic or optical disk or the like.

Assuming that the location for obtaining a license 16 is in fact a license server 24 on a network, the license evaluator 36 then establishes a network connection to such license server 24 based on the web site or other site information, and then sends a request for a license 16 from such connected license server 24 (steps 701, 703). In particular, once the DRM system 32 has contacted the license server 24, such DRM system 32 transmits appropriate license request information 36 to such license server 24. In one embodiment of the present invention, such license 16 request information 36 may include:

- the public key of the black box 30 of the DRM system 32 (PU-BB);
- the version number of the black box 30 of the DRM system 32;
- a certificate with a digital signature from a certifying authority certifying the black box 30 (where the certificate may in fact include the aforementioned public key and version number of the black box 30);
- the content ID (or package ID) that identifies the digital content 12 (or package 12p);
- the key ID that identifies the decryption key (KD) for decrypting the digital content 12;
- the type of license 16 requested (if in fact multiple types are available);
- the type of rendering application 34 that requested rendering of the digital content 12;

and/or the like, among other things. Of course, greater or lessor amounts of license 16 request information 36 may be transmitted to the license server 24 by the DRM system

32 without departing from the spirit and scope of the present invention. For example, information on the type of rendering application 34 may not be necessary, while additional information about the user and/or the user's computing device 14 may be necessary.

Once the license server 24 has received the license 16 request information 36 from the DRM system 32, the license server 24 may then perform several checks for trust / authentication and for other purposes. In one embodiment of the present invention, such license server 24 checks the certificate with the digital signature of the certifying authority to determine whether such has been adulterated or otherwise modified (steps 705, 707). If so, the license server 24 refuses to grant any license 16 based on the request information 36. The license server 24 may also maintain a list of known 'bad' users and/or user's computing devices 14, and may refuse to grant any license 16 based on a request from any such bad user and/or bad user's computing device 14 on the list. Such 'bad' list may be compiled in any appropriate manner without departing from the spirit and scope of the present invention.

Based on the received request and the information associated therewith, and particularly based on the content ID (or package ID) in the license request information, the license server 24 can interrogate the content-key database 20 (Fig. 1) and locate a record corresponding to the digital content 12 (or package 12p) that is the basis of the request. As was discussed above, such record contains the decryption key (KD), key ID, and content ID for such digital content 12. In addition, such record may contain license data regarding the types of licenses 16 to be issued for the digital content 12 and the terms and conditions for each type of license 16. Alternatively, such record may include a pointer, link, or reference to a location having such additional information.

As mentioned above, multiple types of licenses 16 may be available. For example, for a relatively small license fee, a license 16 allowing a limited number of renderings may be available. For a relatively greater license fee, a license 16

allowing unlimited renderings until an expiration date may be available. For a still greater license fee, a license 16 allowing unlimited renderings without any expiration date may be available. Practically any type of license 16 having any kind of license terms may be devised and issued by the license server 24 without departing from the spirit and scope of the present invention.

In one embodiment of the present invention, the request for a license 16 is accomplished with the aid of a web page or the like as transmitted from the license server 24 to the user's computing device 14. Preferably, such web page includes information on all types of licenses 16 available from the license server 24 for the digital content 12 that is the basis of the license 16 request.

In one embodiment of the present invention, prior to issuing a license 16, the license server 24 checks the version number of the black box 30 to determine whether such black box 30 is relatively current (steps 709, 711). As should be understood, the black box 30 is intended to be secure and protected from attacks from a user with nefarious purposes (i.e., to improperly render digital content 12 without a license 16, or outside the terms of a corresponding license 16). However, it is to be recognized that no system and no software device is in fact totally secure from such an attack.

As should also be understood, if the black box 30 is relatively current, i.e., has been obtained or updated relatively recently, it is less likely that such black box 30 has been successfully attacked by such a nefarious user. Preferably, and as a matter of trust, if the license server 24 receives a license request with request information 36 including a black box 30 version number that is not relatively current, such license server 24 refuses to issue the requested license 16 until the corresponding black box 30 is upgraded to a current version, as will be described below. Put simply, the license server 24 will not trust such black box 30 unless such black box 30 is relatively current.

In the context of the black box 30 of the present invention, the term 'current' or 'relatively current' may have any appropriate meaning without departing

from the spirit and scope of the present invention, consistent with the function of providing trust in the black box 30 based on the age or use thereof. For example, 'current' may be defined according to age (i.e., less than one month old). As an alternative example, 'current' may be defined based on a number of times that the black box 30 has decrypted digital content 12 (i.e., less than 200 instances of decryption). Moreover, 'current' may be based on policy as set by each license server 24, where one license server 24 may define 'current' differently from another license server 24, and a license server 24 may further define 'current' differently depending on the digital content 12 for which a license 16 is requested, or depending on the type of license 16 requested, among other things.

Assuming that the license server 24 is satisfied from the version number of a black box 30 or other indicia thereof that such black box 30 is current, the license server 24 then proceeds to negotiate terms and conditions for the license 16 with the user (step 713). Alternatively, the license server 24 negotiates the license 16 with the user, then satisfies itself from the version number of the black box 30 that such black box 30 is current (i.e., performs step 713, then step 711). Of course, the amount of negotiation varies depending on the type of license 16 to be issued, and other factors. For example, if the license server 24 is merely issuing a paid-up unlimited use license 16, very little need be negotiated. On the other hand, if the license 16 is to be based on such items as varying values, sliding scales, break points, and other details, such items and details may need to be worked out between the license server 24 and the user before the license 16 can be issued.

As should be understood, depending on the circumstances, the license negotiation may require that the user provide further information to the license server 24 (for example, information on the user, the user's computing device 14, etc.). Importantly, the license negotiation may also require that the user and the license server 24 determine a mutually acceptable payment instrument (a credit account, a debit account, a mailed check, etc.) and/or payment method (paid-up immediately, spread over a period of time, etc.), among other things.

Once all the terms of the license 16 have been negotiated and agreed to by both the license server 24 and user (step 715), a digital license 16 is generated by the license server 24 (step 719), where such generated license 16 is based at least in part on the license request, the black box 30 public key (PU-BB), and the decryption key (KD) for the digital content 12 that is the basis of the request as obtained from the content-key database 20. In one embodiment of the present invention, and as seen in Fig. 8, the generated license 16 includes:

- the content ID of the digital content 12 to which the license 16 applies;
- a Digital Rights License (DRL) 48 (i.e., the rights description or actual terms and conditions of the license 16 written in a predetermined form that the license evaluator 36 can interrogate), perhaps encrypted with the decryption key (KD) (i.e., KD (DRL));
- the decryption key (KD) for the digital content 12 encrypted with the black box 30 public key (PU-BB) as receive in the license request (i.e.,(PU-BB (KD));
- a digital signature from the license server 24 (without any attached certificate) based on (KD (DRL)) and (PU-BB (KD)) and encrypted with the license server 24 private key (i.e., (S (PR-LS))); and
- the certificate that the license server 24 obtained previously from the content server 22, such certificate indicating that the license server 24 has the authority from the content server 22 to issue the license 16 (i.e., (CERT (PU-LS) S (PR-CS))).

As should be understood, the aforementioned elements and perhaps others are packaged into a digital file or some other appropriate form. As should also be understood, if the DRL 48 or (PU-BB (KD)) in the license 16 should become adulterated or otherwise modified, the digital signature (S (PR-LS)) in the license 16 will not match and therefore will not validate such license 16. For this reason, the DRL 48 need not necessarily be in an encrypted form (i.e., (KD(DRL)) as mentioned

above), although such encrypted form may in some instances be desirable and therefore may be employed without departing from the spirit and scope of the present invention.

Once the digital license 16 has been prepared, such license 16 is then issued to the requestor (i.e., the DRM system 32 on the user's computing device 14) (step 719 of Fig. 7). Preferably, the license 16 is transmitted over the same path through which the request therefor was made (i.e., the Internet or another network), although another path may be employed without departing from the spirit and scope of the present invention. Upon receipt, the requesting DRM system 32 preferably automatically places the received digital license 16 in the license store 38 (step 721).

It is to be understood that a user's computing device 14 may on occasion malfunction, and licenses 16 stored in the license store 38 of the DRM system 32 on such user's computing device 14 may become irretrievably lost. Accordingly, it is preferable that the license server 24 maintain a database 50 of issued licenses 16 (Fig. 1), and that such license server 24 provide a user with a copy or re-issue (hereinafter 're-issue') of an issued license 16 if the user is in fact entitled to such re-issue. In the aforementioned case where licenses 16 are irretrievably lost, it is also likely the case that state information stored in the state store 40 and corresponding to such licenses 16 is also lost. Such lost state information should be taken into account when re-issuing a license 16. For example, a fixed number of renderings license 16 might legitimately be re-issued in a pro-rated form after a relatively short period of time, and not re-issued at all after a relatively longer period of time.

DRM SYSTEM 32 - Installation/Upgrade of Black Box 30

As was discussed above, as part of the function of acquiring a license 16, the license server 24 may deny a request for a license 16 from a user if the user's computing device 14 has a DRM system 32 with a black box 30 that is not relatively current, i.e., has a relatively old version number. In such case, it is preferable that the black box 30 of such DRM system 32 be upgraded so that the license acquisition function can then proceed. Of course, the black box 30 may be upgraded at other times

without departing from the spirit and scope of the present invention.

Preferably, as part of the process of installing the DRM system 32 on a user's computing device 14, a non-unique 'lite' version of a black box 30 is provided. Such 'lite' black box 30 is then upgraded to a unique regular version prior to rendering a piece of digital content 12. As should be understood, if each black box 30 in each DRM system 32 is unique, a security breach into one black box 30 cannot easily be replicated with any other black box 30.

Referring now to Fig. 9, the DRM system 32 obtains the unique black box 30 by requesting same from a black box server 26 or the like (as was discussed above and as shown in Fig. 1) (step 901). Typically, such request is made by way of the Internet, although other means of access may be employed without departing from the spirit and scope of the present invention. For example, the connection to a black box server 26 may be a direct connection, either locally or remotely. An upgrade from one unique non-lite black box 30 to another unique non-lite black box 30 may also be requested by the DRM system 32 at any time, such as for example a time when a license server 24 deems the black box 30 not current, as was discussed above.

Thereafter, the black box server 26 generates a new unique black box 30 (step 903). As seen in Fig. 3, each new black box 30 is provided with a version number and a certificate with a digital signature from a certifying authority. As was discussed above in connection with the license acquisition function, the version number of the black box 30 indicates the relative age and/or use thereof. The certificate with the digital signature from the certifying authority, also discussed above in connection with the license acquisition function, is a proffer or vouching mechanism from the certifying authority that a license server 24 should trust the black box 30. Of course, the license server 24 must trust the certifying authority to issue such a certificate for a black box 30 that is in fact trustworthy. It may be the case, in fact, that the license server 24 does not trust a particular certifying authority, and refuses to honor any certificate issued by such certifying authority. Trust may not occur, for example, if a particular certifying authority is found to be engaging in a pattern of

improperly issuing certificates.

Preferably, and as was discussed above, the black box server 26 includes a new unique public / private key pair (PU-BB, PR-BB) with the newly generated unique black box 30 (step 903 of Fig. 9). Preferably, the private key for the black box 30 (PR-BB) is accessible only to such black box 30, and is hidden from and inaccessible by the remainder of the world, including the computing device 14 having the DRM system 32 with such black box 30, and the user thereof.

Most any hiding scheme may be employed without departing from the spirit and scope of the present invention, so long as such hiding scheme in fact performs the function of hiding the private key (PR-BB) from the world. As but one example, the private key (PR-BB) may be split into several sub-components, and each sub-component may be encrypted uniquely and stored in a different location. In such a situation, it is preferable that such sub-components are never assembled in full to produce the entire private key (PR-BB).

In one embodiment of the present invention, such private key (PR-BB) is encrypted according to code-based encryption techniques. In particular, in such embodiment, the actual software code of the black box 30 (or other software code) is employed as encrypting key(s). Accordingly, if the code of the black box 30 (or the other software code) becomes adulterated or otherwise modified, for example by a user with nefarious purposes, such private key (PR-BB) cannot be decrypted.

Although each new black box 30 is delivered with a new public / private key pair (PU-BB, PR-BB), such new black box 30 is also preferably given access to old public / private key pairs from old black boxes 30 previously delivered to the DRM system 32 on the user's computing device 14 (step 905). Accordingly, the upgraded black box 30 can still employ the old key pairs to access older digital content 12 and older corresponding licenses 16 that were generated according to such old key pairs, as will be discussed in more detail below.

Preferably, the upgraded black box 30 delivered by the black box server 26 is tightly tied to or associated with the user's computing device 14. Accordingly,

the upgraded black box 30 cannot be operably transferred among multiple computing devices 14 for nefarious purposes or otherwise. In one embodiment of the present invention, as part of the request for the black box 30 (step 901) the DRM system 32 provides hardware information unique to such DRM system 32 and/or unique to the user's computing device 14 to the black box server 26, and the black box server 26 generates a black box 30 for the DRM system 32 based in part on such provided hardware information. Such generated upgraded black box 30 is then delivered to and installed in the DRM system 32 on the user's computing device 14 (steps 907, 909). If the upgraded black box 30 is then somehow transferred to another computing device 14, the transferred black box 30 recognizes that it is not intended for such other computing device 14, and does not allow any requested rendering to proceed on such other computing device 14.

Once the new black box 30 is installed in the DRM system 32, such DRM system 32 can proceed with a license acquisition function or with any other function.

DRM SYSTEM 32 - Content Rendering, Part 3

Referring now to Fig. 5B, and assuming, now, that the license evaluator 36 has found at least one valid license 16 and that at least one of such valid licenses 16 provides the user with the rights necessary to render the corresponding digital content 12 in the manner sought (i.e., is enabling), the license evaluator 36 then selects one of such licenses 16 for further use (step 519). Specifically, to render the requested digital content 12, the license evaluator 36 and the black box 30 in combination obtain the decryption key (KD) from such license 16, and the black box 30 employs such decryption key (KD) to decrypt the digital content 12. In one embodiment of the present invention, and as was discussed above, the decryption key (KD) as obtained from the license 16 is encrypted with the black box 30 public key (PU-BB(KD)), and the black box 30 decrypts such encrypted decryption key with its private key (PR-BB) to produce the decryption key (KD) (steps 521, 523). However, other methods of obtaining the decryption key (KD) for the digital content 12 may be employed without

departing from the spirit and scope of the present invention.

Once the black box 30 has the decryption key (KD) for the digital content 12 and permission from the license evaluator 36 to render the digital content 12, control may be returned to the rendering application 34 (steps 525, 527). In one embodiment of the present invention, the rendering application 34 then calls the DRM system 32 / black box 30 and directs at least a portion of the encrypted digital content 12 to the black box 30 for decryption according to the decryption key (KD) (step 529). The black box 30 decrypts the digital content 12 based upon the decryption key (KD) for the digital content 12, and then the black box 30 returns the decrypted digital content 12 to the rendering application 34 for actual rendering (steps 533, 535). The rendering application 34 may either send a portion of the encrypted digital content 12 or the entire digital content 12 to the black box 30 for decryption based on the decryption key (KD) for such digital content 12 without departing from the spirit and scope of the present invention.

Preferably, when the rendering application 34 sends digital content 12 to the black box 30 for decryption, the black box 30 and/or the DRM system 32 authenticates such rendering application 34 to ensure that it is in fact the same rendering application 34 that initially requested the DRM system 32 to run (step 531). Otherwise, the potential exists that rendering approval may be obtained improperly by basing the rendering request on one type of rendering application 34 and in fact rendering with another type of rendering application 34. Assuming the authentication is successful and the digital content 12 is decrypted by the black box 30, the rendering application 34 may then render the decrypted digital content 12 (steps 533, 535).

Sequence of Key Transactions

Referring now to Fig. 10, in one embodiment of the present invention, a sequence of key transactions is performed to obtain the decryption key (KD) and evaluate a license 16 for a requested piece of digital content 12 (i.e., to perform steps 515-523 of Figs. 5A and 5B). Mainly, in such sequence, the DRM system 32 obtains the decryption key (KD) from the license 16, uses information obtained from the

license 16 and the digital content 12 to authenticate or ensure the validity of both, and then determines whether the license 16 in fact provides the right to render the digital content 12 in the manner sought. If so, the digital content 12 may be rendered.

Bearing in mind that each license 16 for the digital content 12, as seen in Fig. 8, includes:

- the content ID of the digital content 12 to which the license 16 applies;
- the Digital Rights License (DRL) 48, perhaps encrypted with the decryption key (KD) (i.e., KD (DRL));
- the decryption key (KD) for the digital content 12 encrypted with the black box 30 public key (PU-BB) (i.e.,(PU-BB (KD));
- the digital signature from the license server 24 based on (KD (DRL)) and (PU-BB (KD)) and encrypted with the license server 24 private key (i.e., (S (PR-LS))); and
- the certificate that the license server 24 obtained previously from the content server 22 (i.e., (CERT (PU-LS) S (PR-CS))),

and also bearing in mind that the package 12p having the digital content 12, as seen in Fig. 3, includes:

- the content ID of such digital content 12;
- the digital content 12 encrypted by KD (i.e., (KD(CONTENT)));
- a license acquisition script that is not encrypted; and
- the key KD encrypting the content server 22 public key (PU-CS), signed by the content server 22 private key (PR-CS) (i.e., (KD (PU-CS) S (PR-CS))),

in one embodiment of the present invention, the specific sequence of key transactions that are performed with regard to a specific one of the licenses 16 for the digital content 12 is as follows:

1. Based on (PU-BB (KD)) from the license 16, the black box 30 of the DRM system 32 on the user's computing device 14 applies its private key (PR-

BB) to obtain (KD) (step 1001). (PR-BB (PU-BB (KD)) = (KD)). Note, importantly, that the black box 30 could then proceed to employ KD to decrypt the digital content 12 without any further ado. However, and also importantly, the license server 24 trusts the black box 30 not to do so. Such trust was established at the time such license server 24 issued the license 16 based on the certificate from the certifying authority vouching for the trustworthiness of such black box 30. Accordingly, despite the black box 30 obtaining the decryption key (KD) as an initial step rather than a final step, the DRM system 32 continues to perform all license 16 validation and evaluation functions, as described below.

2. Based on (KD (PU-CS) S (PR-CS)) from the digital content 12, the black box 30 applies the newly obtained decryption key (KD) to obtain (PU-CS) (step 1003). (KD (KD (PU-CS)) = (PU-CS)). Additionally, the black box 30 can apply (PU-CS) as against the signature (S (PR-CS)) to satisfy itself that such signature and such digital content 12 / package 12p is valid (step 1005). If not valid, the process is halted and access to the digital content 12 is denied.

3. Based on (CERT (PU-LS) S (PR-CS)) from the license 16, the black box 30 applies the newly obtained content server 22 public key (PU-CS) to satisfy itself that the certificate is valid (step 1007), signifying that the license server 24 that issued the license 16 had the authority from the content server 22 to do so, and then examines the certificate contents to obtain (PU-LS) (step 1009). If not valid, the process is halted and access to the digital content 12 based on the license 16 is denied.

4. Based on (S (PR-LS)) from the license 16, the black box 30 applies the newly obtained license server 24 public key (PU-LS) to satisfy itself that the license 16 is valid (step 1011). If not valid, the process is halted and access to the digital content 12 based on the license 16 is denied.

5. Assuming all validation steps are successful, and that the DRL 48 in the license 16 is in fact encrypted with the decryption key (KD), the license evaluator 36 then applies the already-obtained decryption key (KD) to (KD(DRL)) as obtained from the license 16 to obtain the license terms from the license 16 (i.e., the

DRL 48) (step 1013). Of course, if the DRL 48 in the license 16 is not in fact encrypted with the decryption key (KD), step 1013 may be omitted. The license evaluator 36 then evaluates / interrogates the DRL 48 and determines whether the user's computing device 14 has the right based on the DRL 48 in the license 16 to render the corresponding digital content 12 in the manner sought (i.e., whether the DRL 48 is enabling) (step 1015). If the license evaluator 36 determines that such right does not exist, the process is halted and access to the digital content 12 based on the license 16 is denied.

6. Finally, assuming evaluation of the license 16 results in a positive determination that the user's computing device 14 has the right based on the DRL 48 terms to render the corresponding digital content 12 in the manner sought, the license evaluator 36 informs the black box 30 that such black box 30 can render the corresponding digital content 12 according to the decryption key (KD). The black box 30 thereafter applies the decryption key (KD) to decrypt the digital content 12 from the package 12p (i.e., (KD(KD(CONTENT)) = (CONTENT)) (step 1017).

It is important to note that the above-specified series of steps represents an alternating or 'ping-ponging' between the license 16 and the digital content 12. Such ping-ponging ensures that the digital content 12 is tightly bound to the license 16, in that the validation and evaluation process can only occur if both the digital content 12 and license 16 are present in a properly issued and valid form. In addition, since the same decryption key (KD) is needed to get the content server 22 public key (PU-CS) from the license 16 and the digital content 12 from the package 12p in a decrypted form (and perhaps the license terms (DRL 48) from the license 16 in a decrypted form), such items are also tightly bound. Signature validation also ensures that the digital content 12 and the license 16 are in the same form as issued from the content server 22 and the license server 24, respectively. Accordingly, it is difficult if not impossible to decrypt the digital content 12 by bypassing the license server 24, and also difficult if not impossible to alter and then decrypt the digital content 12 or the license 16.

In one embodiment of the present invention, signature verification, and especially signature verification of the license 16, is alternately performed as follows. Rather than having a signature encrypted by the private key of the license server 16 (PR-LS), as is seen in Fig. 8, each license 16 has a signature encrypted by a private root key (PR-R) (not shown), where the black box 30 of each DRM system 32 includes a public root key (PU-R) (also not shown) corresponding to the private root key (PR-R). The private root key (PR-R) is known only to a root entity, and a license server 24 can only issue licenses 16 if such license server 24 has arranged with the root entity to issue licenses 16.

In particular, in such embodiment:

1. the license server 24 provides its public key (PU-LS) to the root entity;
2. the root entity returns the license server public key (PU-LS) to such license server 24 encrypted with the private root key (PR-R) (i.e., (CERT (PU-LS) S (PR-R))); and
3. the license server 24 then issues a license 16 with a signature encrypted with the license server private key (S (PR-LS)), and also attaches to the license the certificate from the root entity (CERT (PU-LS) S (PR-R)).

For a DRM system 18 to validate such issued license 16, then, the DRM system 18:

1. applies the public root key (PU-R) to the attached certificate (CERT (PU-LS) S (PR-R)) to obtain the license server public key (PU-LS); and
2. applies the obtained license server public key (PU-LS) to the signature of the license 16 (S (PR-LS).

Importantly, it should be recognized that just as the root entity gave the license server 24 permission to issue licenses 16 by providing the certificate (CERT (PU-LS) S (PR-R)) to such license server 24, such license server 24 can provide a

similar certificate to a second license server 24 (i.e., (CERT (PU-LS2) S (PR-LS1)), thereby allowing the second license server to also issue licenses 16. As should now be evident, a license 16 issued by the second license server would include a first certificate (CERT (PU-LS1) S (PR-R)) and a second certificate (CERT (PU-LS2) S (PR-LS1)). Likewise, such license 16 is validated by following the chain through the first and second certificates. Of course, additional links in the chain may be added and traversed.

One advantage of the aforementioned signature verification process is that the root entity may periodically change the private root key (PR-R), thereby likewise periodically requiring each license server 24 to obtain a new certificate (CERT (PU-LS) S (PR-R)). Importantly, as a requirement for obtaining such new certificate, each license server may be required to upgrade itself. As with the black box 30, if a license server 24 is relatively current, i.e., has been upgraded relatively recently, it is less likely that license server 24 has been successfully attacked. Accordingly, as a matter of trust, each license server 24 is preferably required to be upgraded periodically via an appropriate upgrade trigger mechanism such as the signature verification process. Of course, other upgrade mechanisms may be employed without departing from the spirit and scope of the present invention.

Of course, if the private root key (PR-R) is changed, then the public root key (PU-R) in each DRM system 18 must also be changed. Such change may for example take place during a normal black box 30 upgrade, or in fact may require that a black box 30 upgrade take place. Although a changed public root key (PU-R) may potentially interfere with signature validation for an older license 16 issued based on an older private root key (PR-R), such interference may be minimized by requiring that an upgraded black box 30 remember all old public root keys (PU-R). Alternatively, such interference may be minimized by requiring signature verification for a license 16 only once, for example the first time such license 16 is evaluated by the license evaluator 36 of a DRM system 18. In such case, state information on whether signature verification has taken place should be compiled, and such state information

should be stored in the state store 40 of the DRM system 18.

Digital Rights License 48

In the present invention, the license evaluator 36 evaluates a Digital Rights License (DRL) 48 as the rights description or terms of a license 16 to determine if such DRL 48 allows rendering of a corresponding piece of digital content 12 in the manner sought. In one embodiment of the present invention, the DRL 48 may be written by a licensor (i.e., the content owner) in any DRL language.

As should be understood, there are a multitude of ways to specify a DRL 48. Accordingly, a high degree of flexibility must be allowed for in any DRL language. However, it is impractical to specify all aspects of a DRL 48 in a particular license language, and it is highly unlikely that the author of such a language can appreciate all possible licensing aspects that a particular digital licensor may desire. Moreover, a highly sophisticated license language may be unnecessary and even a hindrance for a licensor providing a relatively simple DRL 48. Nevertheless, a licensor should not be unnecessarily restricted in how to specify a DRL 48. At the same time, the license evaluator 36 should always be able to get answers from a DRL 48 regarding a number of specific license questions.

In the present invention, and referring now to Fig. 11, a DRL 48 can be specified in any license language, but includes a language identifier or tag 54. The license evaluator 36 evaluating the license 16, then, performs the preliminary step of reviewing the language tag 54 to identify such language, and then selects an appropriate license language engine 52 for accessing the license 16 in such identified language. As should be understood, such license language engine 52 must be present and accessible to the license evaluator 36. If not present, the language tag 54 and/or the DRL 48 preferably includes a location 56 (typically a web site) for obtaining such language engine 52.

Typically, the language engine 52 is in the form of an executable file or set of files that reside in a memory of the user's computing device 14, such as a hard drive. The language engine 52 assists the license evaluator 36 to directly interrogate

the DRL 48, the license evaluator 36 interrogates the DRL 48 indirectly via the language engine 48 acting as an intermediary, or the like. When executed, the language engine 52 runs in a work space in a memory of the user's computing device 14, such as RAM. However, any other form of language engine 52 may be employed without departing from the spirit and scope of the present invention.

Preferably, any language engine 52 and any DRL language supports at least a number of specific license questions that the license evaluator 36 expects to be answered by any DRL 48, as will be discussed below. Accordingly, the license evaluator 36 is not tied to any particular DRL language; a DRL 48 may be written in any appropriate DRL language; and a DRL 48 specified in a new license language can be employed by an existing license evaluator 36 by having such license evaluator 36 obtain a corresponding new language engine 52.

DRL Languages

Two examples of DRL languages, as embodied in respective DRLs 48, are provided below. The first, 'simple' DRL 48 is written in a DRL language that specifies license attributes, while the second 'script' DRL 48 is written in a DRL language that can perform functions according to the script specified in the DRL 48. While written in a DRL language, the meaning of each line of code should be apparent based on the linguistics thereof and/or on the attribute description chart that follows:

Simple DRL 48:

```
<LICENSE>
    <DATA>
        <NAME>Beastie Boy's Play</NAME>
        <ID>39384</ID>
        <DESCRIPTION>Play the song 3 times</DESCRIPTION>
        <TERMS></TERMS>
        <VALIDITY>
            <NOTBEFORE>19980102 23:20:14Z</NOTBEFORE>
            <NOTAFTER>19980102 23:20:14Z</NOTAFTER>
        </VALIDITY>
        <ISSUEDDATE>19980102 23:20:14Z</ISSUEDDATE>
        <LICENSORSITE>http://www.foo.com</LICENSORSITE>
```

```
<CONTENT>
        <NAME>Beastie Boy's</NAME>
        <ID>392</ID>
        <KEYID>39292</KEYID>
        <TYPE>MS Encrypted ASF 2.0</TTYPE>
</CONTENT>
<OWNER>
        <ID>939KDKD393KD</ID>
        <NAME>Universal</NAME>
        <PUBLICKEY></PUBLICKEY>
</OWNER>
<LICENSEE>
        <NAME>Arnold</NAME>
        <ID>939KDKD393KD</ID>
        <PUBLICKEY></PUBLICKEY>
</LICENSEE>
<PRINCIPAL TYPE='AND'>
        <PRINCIPAL TYPE='OR'>
                <PRINCIPAL>
                        <TYPE>x86Computer</TYPE>
                        <ID>3939292939d9e939</ID>
                        <NAME>Personal Computer</NAME>
                        <AUTHTYPE>Intel Authenticated Boot PC
                        SHA-1 DSA512</AUTHTYPE>
                        <AUTHDATA>29293939</AUTHDATA>
                </PRINCIPAL>
                <PRINCIPAL>
                        <TYPE>Application</TYPE>
                        <ID>2939495939292</ID>
                        <NAME>Window's Media Player</NAME>
                        <AUTHTYPE>Authenticode SHA-
                        1</AUTHTYPE>
                        <AUTHDATA>93939</AUTHDATA>
                </PRINCIPAL>
        </PRINCIPAL>
        <PRINCIPAL>
                <TYPE>Person</TYPE>
                <ID>39299482010</ID>
                <NAME>Arnold Blinn</NAME>
                <AUTHTYPE>Authenticate user</AUTHTYPE>
                <AUTHDATA>\\redmond\arnoldb</AUTHDATA>
        </PRINCIPAL>
</PRINCIPAL>
```

```
                <DRLTYPE>Simple</DRLTYPE>  [the language tag 54]
                <DRLDATA>
                        <START>19980102 23:20:14Z</START>
                        <END>19980102 23:20:14Z</END>
                        <COUNT>3</COUNT>
                        <ACTION>PLAY</ACTION>
                </DRLDATA>
                <ENABLINGBITS>aaaabbbbccccdddd</ENABLINGBITS>
        </DATA>
        <SIGNATURE>
        <SIGNERNAME>Universal</SIGNERNAME>
                <SIGNERID>9382ABK3939DKD</SIGNERID>
                <HASHALGORITHMID>MD5</HASHALGORITHMID>
                <SIGNALGORITHMID>RSA 128</SIGNALGORITHMID>
                <SIGNATURE>xxxyyyxxxyyyxxxyyy</SIGNATURE>
                <SIGNERPUBLICKEY></SIGNERPUBLICKEY>
                <CONTENTSIGNEDSIGNERPUBLICKEY></CONTENTSIGNEDSI
                GNERPUBLICKEY>
        </SIGNATURE>
</LICENSE>
```

Script DRL 48:

```
<LICENSE>
        <DATA>
                <NAME>Beastie Boy's Play</NAME>
                <ID>39384</ID>
                <DESCRIPTION>Play the song unlimited</DESCRIPTION>
                <TERMS></TERMS>
                <VALIDITY>
                        <NOTBEFORE>19980102 23:20:14Z</NOTBEFORE>
                        <NOTAFTER>19980102 23:20:14Z</NOTAFTER>
                </VALIDITY>
                <ISSUEDDATE>19980102 23:20:14Z</ISSUEDDATE>
                <LICENSORSITE>http://www.foo.com</LICENSORSITE>
                <CONTENT>
                        <NAME>Beastie Boy's</NAME
                        <ID>392</ID>
                        <KEYID>39292</KEYID>
                        <TYPE>MS Encrypted ASF 2.0</TTYPE>
                </CONTENT>
                <OWNER>
                        <ID>939KDKD393KD</ID>
```

```
                <NAME>Universal</NAME>
                <PUBLICKEY></PUBLICKEY>
            </OWNER>
            <LICENSEE>
                <NAME>Arnold</NAME>
                <ID>939KDKD393KD</ID>
                <PUBLICKEY></PUBLICKEY>
            </LICENSEE>
            <DRLTYPE>Script</DRLTYPE>    [the language tag 54]
            <DRLDATA>
                function on_enable(action. args) as boolean
                    result = False
                    if action = "PLAY" then
                        result = True
                    end if
                    on_action = False
                end function
                . . .
            </DRLDATA>
        </DATA>
        <SIGNATURE>
            <SIGNERNAME>Universal</SIGNERNAME>
            <SIGNERID>9382</SIGNERID>
            <SIGNERPUBLICKEY></SIGNERPUBLICKEY>
            <HASHID>MD5</HASHID>
            <SIGNID>RSA 128</SIGNID>
            <SIGNATURE>xxxyyyxxxyyyxxxyyy</SIGNATURE>
            <CONTENTSIGNEDSIGNERPUBLICKEY></CONTENTSIGNEDSI
            GNERPUBLICKEY>
        </SIGNATURE>
</LICENSE>
```

In the two DRLs 48 specified above. the attributes listed have the following descriptions and data types:

Attribute	Description	Data Type
Id	ID of the license	GUID
Name	Name of the license	String
Content Id	ID of the content	GUID
Content Key Id	ID for the encryption key of the content	GUID
Content Name	Name of the content	String
Content Type	Type of the content	String

Owner Id	ID of the owner of the content	GUID
Owner Name	Name of the owner of the content	String
Owner Public Key	Public key for owner of content. This is a base-64 encoded public key for the owner of the content.	String
Licensee Id	Id of the person getting license. It may be null.	GUID
Licensee Name	Name of the person getting license. It may be null.	String
Licensee Public Key	Public key of the licensee. This is the base-64 encoded public key of the licensee. It may be null.	String
Description	Simple human readable description of the license	String
Terms	Legal terms of the license. This may be a pointer to a web page containing legal prose.	String
Validity Not After	Validity period of license expiration	Date
Validity Not Before	Validity period of license start	Date
Issued Date	Date the license was issued	Date
DRL Type	Type of the DRL. Example include "SIMPLE" or "SCRIPT"	String
DRL Data	Data specific to the DRL	String
Enabling Bits	These are the bits that enable access to the actual content. The interpretation of these bits is up to the application, but typically this will be the private key for decryption of the content. This data will be base-64 encoded. Note that these bits are encrypted using the public key of the individual machine.	String
Signer Id	ID of person signing license	GUID
Signer Name	Name of person signing license	String
Signer Public Key	Public key for person signing license. This is the base-64 encode public key for the signer.	String
Content Signed Signer Public Key	Public key for person signing the license that has been signed by the content server private key. The public key to verify this signature will be encrypted in the content. This is base-64 encoded.	String

Hash Alg Id	Algorithm used to generate hash. This is a string, such as "MD5".	String
Signature Alg Id	Algorithm used to generate signature. This is a string, such as "RSA 128".	String
Signature	Signature of the data. This is base-64 encoded data.	String

Methods

As was discussed above, it is preferable that any language engine 52 and any DRL language support at least a number of specific license questions that the digital license evaluator 36 expects to be answered by any DRL 48. Recognizing such supported questions may include any questions without departing from the spirit and scope of the present invention, and consistent with the terminology employed in the two DRL 48 examples above, in one embodiment of the present invention, such supported questions or 'methods' include 'access methods', 'DRL methods', and 'enabling use methods', as follows:

Access Methods

Access methods are used to query a DRL 48 for top-level attributes.

VARIANT QueryAttribute (BSTR key)

Valid keys include License.Name, License.Id, Content.Name, Content.Id, Content.Type, Owner.Name, Owner.Id, Owner.PublicKey, Licensee.Name, Licensee.Id, Licensee.PublicKey, Description, and Terms, each returning a BSTR variant; and Issued, Validity.Start and Validity.End, each returning a Date Variant.

DRL Methods

The implementation of the following DRL methods varies from DRL 48 to DRL 48. Many of the DRL methods contain a variant parameter labeled 'data' which is intended for communicating more advanced information with a DRL 48. It

is present largely for future expandability.

Boolean IsActivated(Variant data)

This method returns a Boolean indicating whether the DRL 48 / license 16 is activated. An example of an activated license 16 is a limited operation license 16 that upon first play is active for only 48 hours.

Activate(Variant data)

This method is used to activate a license 16. Once a license 16 is activated, it cannot be deactivated.

Variant QueryDRL(Variant data)

This method is used to communicate with a more advanced DRL 48. It is largely about future expandability of the DRL 48 feature set.

Variant GetExpires(BSTR action, Variant data)

This method returns the expiration date of a license 16 with regard to the passed-in action. If the return value is NULL, the license 16 is assumed to never expire or does not yet have an expiration date because it hasn't been activated, or the like.

Variant GetCount(BSTR action, Variant data)

This method returns the number of operations of the passed-in action that are left. If NULL is returned, the operation can be performed an unlimited number of times.

Boolean IsEnabled(BSTR action, Variant data)

This method indicates whether the license 16 supports the requested action at the present time.

Boolean IsSunk(BSTR action, Variant data)

This method indicates whether the license 16 has been paid for. A license 16 that is paid for up front would return TRUE, while a license 16 that is not paid for up front, such as a license 16 that collects payments as it is used, would return FALSE.

Enabling Use Methods.

These methods are employed to enable a license 16 for use in decrypting content.

Boolean Validate (BSTR key)

This method is used to validate a license 16. The passed-in key is the black box 30 public key (PU-BB) encrypted by the decryption key (KD) for the corresponding digital content 12 (i.e.,(KD(PU-BB))) for use in validation of the signature of the license 16. A return value of TRUE indicates that the license 16 is valid. A return value of FALSE indicates invalid.

int OpenLicense 16(BSTR action, BSTR key. Variant data)

This method is used to get ready to access the decrypted enabling bits. The passed-in key is (KD(PU-BB)) as described above. A return value of 0 indicates success. Other return values can be defined.

BSTR GetDecryptedEnablingBits (BSTR action. Variant data)

Variant GetDecryptedEnablingBitsAsBinary (BSTR action. Variant Data)

These methods are used to access the enabling bits in decrypted form. If this is not successful for any of a number of reasons, a null string or null variant is returned.

void CloseLicense 16 (BSTR action. Variant data)

This method is used to unlock access to the enabling bits for performing the passed-in action. If this is not successful for any of a number of reasons, a null string is returned.

Heuristics

As was discussed above, if multiple licenses 16 are present for the same piece of digital content 12, one of the licenses 16 must be chosen for further use. Using the above methods, the following heuristics could be implemented to make such choice. In particular, to perform an action (say "PLAY") on a piece of digital content 12, the following steps could be performed:

1. Get all licenses 16 that apply to the particular piece of digital content 12.

2. Eliminate each license 16 that does not enable the action by calling the IsEnabled function on such license 16.

3. Eliminate each license 16 that is not active by calling IsActivated on such license 16.

4. Eliminate each license 16 that is not paid for up front by calling IsSunk on such license 16.

5. If any license 16 is left, use it. Use an unlimited-number-of-plays license 16 before using a limited-number-of-plays license 16, especially if the unlimited-number-of-plays license 16 has an expiration date. At any time, the user should be allowed to select a specific license 16 that has already been acquired, even if the choice is not cost-effective. Accordingly, the user can select a license 16 based on criteria that are perhaps not apparent to the DRM system 32.

6. If there are no licenses 16 left, return status so indicating. The user would then be given the option of:

using a license 16 that is not paid for up front, if available;

activating a license 16, if available; and/or

performing license acquisition from a license server 24.

CONCLUSION

The programming necessary to effectuate the processes performed in connection with the present invention is relatively straight-forward and should be

apparent to the relevant programming public. Accordingly, such programming is not attached hereto. Any particular programming, then, may be employed to effectuate the present invention without departing from the spirit and scope thereof.

In the foregoing description, it can be seen that the present invention comprises a new and useful enforcement architecture 10 that allows the controlled rendering or playing of arbitrary forms of digital content 12, where such control is flexible and definable by the content owner of such digital content 12. Also, the present invention comprises a new useful controlled rendering environment that renders digital content 12 only as specified by the content owner, even though the digital content 12 is to be rendered on a computing device 14 which is not under the control of the content owner. Further, the present invention comprises a trusted component that enforces the rights of the content owner on such computing device 14 in connection with a piece of digital content 12, even against attempts by the user of such computing device 14 to access such digital content 12 in ways not permitted by the content owner.

It should be appreciated that changes could be made to the embodiments described above without departing from the inventive concepts thereof. It should be understood, therefore, that this invention is not limited to the particular embodiments disclosed, but it is intended to cover modifications within the spirit and scope of the present invention as defined by the appended claims.

CLAIMS

1. A method for a device to interdependently validate:

a digital content package having a piece of digital content in an encrypted form; and

a corresponding digital license for rendering the digital content.

the method comprising:

deriving a first key from a source available to the device;

obtaining a first digital signature from the digital content package;

applying the first key to the first digital signature to validate the first digital signature and the digital content package:

deriving a second key based on the first digital signature;

obtaining a second digital signature from the license; and

applying the second key to the second digital signature to validate the second digital signature and the license.

2. The method of claim 1 wherein deriving the first key comprises:

obtaining a first encrypted key from the license;

applying a key available to the device to the first encrypted key to decrypt the first encrypted key;

obtaining a second encrypted key from the digital content; and

applying the decrypted first encrypted key to the second encrypted key to produce the first key.

3. The method of claim 2 wherein the encrypted digital content is decryptable according to a decryption key (KD), and wherein the first encrypted key is the decryption key (KD) encrypted with the device public key (PU-D) (i.e.,(PU-D (KD))).

4. The method of claim 2 wherein the device has a public key (PU-D) and a private key (PR-D), and wherein the key available to the device is (PR-D).

5. The method of claim 2 wherein the encrypted digital content is decryptable according to a decryption key (KD), wherein the digital content package is provided by a content provider having a public key (PU-C) and a private key (PR-C), and wherein the second encrypted key is the content provider public key (PU-C) encrypted with the decryption key (KD) (i.e., KD (PU-C)).

6. The method of claim 2 wherein the second encrypted key is the basis for the first digital signature.

7. The method of claim 1 wherein deriving the second key comprises:

obtaining a signed certificate from the license, the signed certificate having contents therein; and

applying the first key to the signature of the signed certificate to produce the contents of the certificate and also to validate the signature.

8. The method of claim 7 wherein the digital license is provided by a license provider having a public key (PU-L) and a private key (PR-L), and wherein the contents of the certificate is (PU-L).

9. The method of claim 8 wherein the digital content package is provided by a content provider having a public key (PU-C) and a private key (PR-C), and wherein the signed certificate is a certificate containing the license provider public key (PU-L) and signed by the content provider private key (PR-C) (i.e., (CERT (PU-L) S (PR-C))).

10. The method of claim 8 wherein the digital content package is provided by a content provider authorized by a root source to provide the package, wherein the root source has a public key (PU-R) and a private key (PR-R) and wherein the signed certificate is a certificate containing the license provider public key (PU-L) and signed by the root source private key (PR-R) (i.e., (CERT (PU-L) S (PR-R))).

11. The method of claim 1 wherein the digital content package is provided by a content provider having a public key (PU-C) and a private key (PR-C), and wherein the first key is (PU-C).

12. The method of claim 11 wherein the encrypted digital content is decryptable according to a decryption key (KD), and wherein the first digital signature is based on the content provider public key (PU-C) encrypted with the decryption key (KD) and

is signed by the content provider private key (PR-C) (i.e., (KD (PU-C) S (PR-C))).

13. The method of claim 12 wherein deriving (PU-C) comprises:

deriving (KD) from a source available to the device;

applying (KD) to (KD (PU-C) S (PR-C)) to produce (PU-C).

14. The method of claim 13 wherein the device has a public key (PU-D) and a private key (PR-D), wherein the license has the decryption key (KD) encrypted with the device public key (PU-D) (i.e.,(PU-D (KD))), and wherein deriving (KD) comprises:

obtaining (PU-D (KD)) from the license;

applying (PR-D) to (PU-D (KD)) to produce (KD).

15 The method of claim 14 wherein the license has a license rights description specifying terms and conditions that must be satisfied before the digital content may be rendered, the license rights description being encrypted with the decryption key (KD) (i.e., (KD (DRL))), the method further comprising applying (KD) to (KD(DRL)) to obtain the license terms and conditions.

16. The method of claim 14 wherein the license has a license rights description specifying terms and conditions that must be satisfied before the digital content may be rendered, the method further comprising:

evaluating the license terms and conditions to determine whether the digital content is permitted to be rendered in the manner sought;

if so, applying (KD) to the encrypted digital content to decrypt such encrypted digital content; and

rendering the decrypted digital content.

17. The method of claim 11 wherein the encrypted digital content package is provided by a content provider authorized by a root source to provide the package, wherein the root source has a public key (PU-R) and a private key (PR-R) and wherein the first digital signature is a signed certificate containing the content provider public key (PU-C) and signed by the root source private key (PR-R) (i.e., (CERT (PU-C) S (PR-R))).

18. The method of claim 1 wherein the digital license is provided by a license provider having a public key (PU-L) and a private key (PR-L). and wherein the second key is (PU-L).

19. The method of claim 18 wherein the second digital signature is a digital signature encrypted with the license provider private key (i.e., (S (PR-L))).

20. The method of claim 19 wherein the digital content package is provided by a content provider having a public key (PU-C) and a private key (PR-C). wherein the

license has a certificate containing the license provider public key (PU-L) and signed by the content provider private key (PR-C) (i.e., (CERT (PU-L) S (PR-C))), and wherein deriving (PU-L) comprises:

deriving (PU-C) from a source available to the device;

obtaining (CERT (PU-L) S (PR-C)) from the license; and

applying (PU-C) to (CERT (PU-L) S (PR-C)) to validate (CERT (PU-L) S (PR-C)), to produce (PU-L) and also to validate the content provider.

21. The method of claim 20 wherein the encrypted digital content is decryptable according to a decryption key (KD), wherein the first digital signature is based on the content provider public key (PU-C) encrypted with the decryption key (KD) and is signed by the content provider private key (PR-C) (i.e., (KD (PU-C) S (PR-C))), and wherein deriving (PU-C) comprises:

deriving (KD) from a source available to the device;

applying (KD) to (KD (PU-C) S (PR-C)) to produce (PU-C).

22. The method of claim 21 wherein the device has a public key (PU-D) and a private key (PR-D), wherein the license has the decryption key (KD) encrypted with the device public key (PU-D) (i.e.,(PU-D (KD))), and wherein deriving (KD) comprises:

obtaining (PU-D (KD)) from the license;

applying (PR-D) to (PU-D (KD)) to produce (KD).

23. The method of claim 22 wherein the license has a license rights description specifying terms and conditions that must be satisfied before the digital content may be rendered, the license rights description being encrypted with the decryption key (KD) (i.e., (KD (DRL))), the method further comprising applying (KD) to (KD(DRL)) to obtain the license terms and conditions.

24. The method of claim 22 wherein the license has a license rights description specifying terms and conditions that must be satisfied before the digital content may be rendered, the method further comprising:

evaluating the license terms and conditions to determine whether the digital content is permitted to be rendered in the manner sought;

if so, applying (KD) to the encrypted digital content to decrypt such encrypted digital content; and

rendering the decrypted digital content.

25. A method for a device to interdependently validate a piece of digital content and a corresponding digital license for rendering the digital content, the digital content being encrypted, the encrypted digital content being decryptable according to a decryption key (KD) and being packaged in a digital content package, the digital content package being provided by a content provider having a public key (PU-C) and a private key (PR-C), the digital license being provided by a license provider having

a public key (PU-L) and a private key (PR-L), the device having a public key (PU-D) and a private key (PR-D), the digital content package comprising:

the encrypted digital content: and

the content provider public key (PU-C) encrypted with the decryption key (KD) and signed by the content provider private key (PR-C) (i.e., (KD (PU-C) S (PR-C))):

the digital license comprising:

the decryption key (KD) encrypted with the device public key (PU-D) (i.e.,(PU-D (KD))):

a digital signature from the license provider (without any attached certificate) based on (KD (DRL)) and (PU-D (KD)) and encrypted with the license provider private key (i.e., (S (PR-L))); and

a certificate containing the license provider public key (PU-L) and signed by the content provider private key (PR-C) (i.e., (CERT (PU-L) S (PR-C))));

the method comprising:

obtaining (PU-D (KD)) from the license:

applying (PR-D) to (PU-D (KD)) to produce (KD);

obtaining (KD (PU-C) S (PR-C)) from the digital content package;

applying (KD) to (KD (PU-C) S (PR-C)) to produce (PU-C);

applying (PU-C) to (S (PR-C)) to validate (KD (PU-C) S (PR-C)), thereby validating the digital content package:

obtaining (CERT (PU-L) S (PR-C)) from the license;

applying (PU-C) to (CERT (PU-L) S (PR-C)) to validate (CERT (PU-L) S (PR-C)), thereby validating the content provider, and also to obtain (PU-L);

obtaining (S (PR-L)) from the license; and

applying (PU-L) to (S (PR-L)), thereby validating the license.

26. The method of claim 25 wherein the digital content package further comprises a content / package ID identifying one of the digital content and the digital content package, and wherein the license further comprises the content / package ID of the corresponding digital content / digital content package, the method further comprising ensuring that the content / package ID of the license in fact corresponds to the content / package ID of the digital content / digital content package.

27. The method of claim 25 wherein the license further comprises a license rights description (DRL) specifying terms and conditions that must be satisfied before the digital content may be rendered, the method further comprising;

evaluating the license terms and conditions to determine whether the digital content is permitted to be rendered in the manner sought;

if so, applying (KD) to the encrypted digital content to decrypt such encrypted digital content; and

rendering the decrypted digital content.

28. The method of claim 27 wherein the license rights description is encrypted with the decryption key (KD) (i.e., (KD (DRL))), the method further comprising applying (KD) to (KD (DRL)) to obtain the license terms and conditions.

29. A computer-readable medium having computer-executable instructions for performing a method for a device to interdependently validate:

a digital content package having a piece of digital content in an encrypted form; and

a corresponding digital license for rendering the digital content, the method comprising:

deriving a first key from a source available to the device;

obtaining a first digital signature from the digital content package;

applying the first key to the first digital signature to validate the first digital signature and the digital content package;

deriving a second key based on the first digital signature;

obtaining a second digital signature from the license; and

applying the second key to the second digital signature to validate the second digital signature and the license.

30. The method of claim 28 wherein deriving the first key comprises:

obtaining a first encrypted key from the license;

applying a key available to the device to the first encrypted key to

decrypt the first encrypted key:

obtaining a second encrypted key from the digital content; and

applying the decrypted first encrypted key to the second encrypted key to produce the first key.

31. The method of claim 30 wherein the encrypted digital content is decryptable according to a decryption key (KD), and wherein the first encrypted key is the decryption key (KD) encrypted with the device public key (PU-D) (i.e.,(PU-D (KD))).

32. The method of claim 30 wherein the device has a public key (PU-D) and a private key (PR-D), and wherein the key available to the device is (PR-D).

33. The method of claim 30 wherein the encrypted digital content is decryptable according to a decryption key (KD), wherein the digital content package is provided by a content provider having a public key (PU-C) and a private key (PR-C), and wherein the second encrypted key is the content provider public key (PU-C) encrypted with the decryption key (KD) (i.e., KD (PU-C)).

34. The method of claim 30 wherein the second encrypted key is the basis for the first digital signature.

35. The method of claim 29 wherein deriving the second key comprises:

obtaining a signed certificate from the license, the signed certificate having contents therein; and

applying the first key to the signature of the signed certificate to produce the contents of the certificate and also to validate the signature.

36. The method of claim 35 wherein the digital license is provided by a license provider having a public key (PU-L) and a private key (PR-L), and wherein the contents of the certificate is (PU-L).

37. The method of claim 36 wherein the digital content package is provided by a content provider having a public key (PU-C) and a private key (PR-C), and wherein the signed certificate is a certificate containing the license provider public key (PU-L) and signed by the content provider private key (PR-C) (i.e., (CERT (PU-L) S (PR-C))).

38. The method of claim 36 wherein the digital content package is provided by a content provider authorized by a root source to provide the package, wherein the root source has a public key (PU-R) and a private key (PR-R) and wherein the signed certificate is a certificate containing the license provider public key (PU-L) and signed by the root source private key (PR-R) (i.e., (CERT (PU-L) S (PR-R))).

39. The method of claim 29 wherein the digital content package is provided by a content provider having a public key (PU-C) and a private key (PR-C), and wherein

the first key is (PU-C).

40. The method of claim 39 wherein the encrypted digital content is decryptable according to a decryption key (KD), and wherein the first digital signature is based on the content provider public key (PU-C) encrypted with the decryption key (KD) and is signed by the content provider private key (PR-C) (i.e., (KD (PU-C) S (PR-C))).

41. The method of claim 40 wherein deriving (PU-C) comprises:

deriving (KD) from a source available to the device;

applying (KD) to (KD (PU-C) S (PR-C)) to produce (PU-C).

42. The method of claim 41 wherein the device has a public key (PU-D) and a private key (PR-D), wherein the license has the decryption key (KD) encrypted with the device public key (PU-D) (i.e.,(PU-D (KD))), and wherein deriving (KD) comprises:

obtaining (PU-D (KD)) from the license;

applying (PR-D) to (PU-D (KD)) to produce (KD).

43. The method of claim 42 wherein the license has a license rights description specifying terms and conditions that must be satisfied before the digital content may be rendered, the license rights description being encrypted with the decryption key (KD) (i.e., (KD (DRL))), the method further comprising applying (KD) to (KD(DRL))

to obtain the license terms and conditions.

44. The method of claim 42 wherein the license has a license rights description specifying terms and conditions that must be satisfied before the digital content may be rendered, the method further comprising:

evaluating the license terms and conditions to determine whether the digital content is permitted to be rendered in the manner sought;

if so, applying (KD) to the encrypted digital content to decrypt such encrypted digital content; and

rendering the decrypted digital content.

45. The method of claim 39 wherein the encrypted digital content package is provided by a content provider authorized by a root source to provide the package, wherein the root source has a public key (PU-R) and a private key (PR-R) and wherein the first digital signature is a signed certificate containing the content provider public key (PU-C) and signed by the root source private key (PR-R) (i.e., (CERT (PU-C) S (PR-R))).

46. The method of claim 29 wherein the digital license is provided by a license provider having a public key (PU-L) and a private key (PR-L), and wherein the second key is (PU-L).

47. The method of claim 46 wherein the second digital signature is a digital signature encrypted with the license provider private key (i.e., (S (PR-L))).

48. The method of claim 47 wherein the digital content package is provided by a content provider having a public key (PU-C) and a private key (PR-C), wherein the license has a certificate containing the license provider public key (PU-L) and signed by the content provider private key (PR-C) (i.e., (CERT (PU-L) S (PR-C))), and wherein deriving (PU-L) comprises:

deriving (PU-C) from a source available to the device;

obtaining (CERT (PU-L) S (PR-C)) from the license; and

applying (PU-C) to (CERT (PU-L) S (PR-C)) to validate (CERT (PU-L) S (PR-C)), to produce (PU-L) and also to validate the content provider.

49. The method of claim 48 wherein the encrypted digital content is decryptable according to a decryption key (KD), wherein the first digital signature is based on the content provider public key (PU-C) encrypted with the decryption key (KD) and is signed by the content provider private key (PR-C) (i.e., (KD (PU-C) S (PR-C))), and wherein deriving (PU-C) comprises:

deriving (KD) from a source available to the device;

applying (KD) to (KD (PU-C) S (PR-C)) to produce (PU-C).

50. The method of claim 49 wherein the device has a public key (PU-D) and a

private key (PR-D), wherein the license has the decryption key (KD) encrypted with the device public key (PU-D) (i.e.,(PU-D (KD))), and wherein deriving (KD) comprises:

obtaining (PU-D (KD)) from the license;

applying (PR-D) to (PU-D (KD)) to produce (KD).

51. The method of claim 50 wherein the license has a license rights description specifying terms and conditions that must be satisfied before the digital content may be rendered, the license rights description being encrypted with the decryption key (KD) (i.e., (KD (DRL))), the method further comprising applying (KD) to (KD(DRL)) to obtain the license terms and conditions.

52. The method of claim 50 wherein the license has a license rights description specifying terms and conditions that must be satisfied before the digital content may be rendered, the method further comprising:

evaluating the license terms and conditions to determine whether the digital content is permitted to be rendered in the manner sought;

if so, applying (KD) to the encrypted digital content to decrypt such encrypted digital content; and

rendering the decrypted digital content.

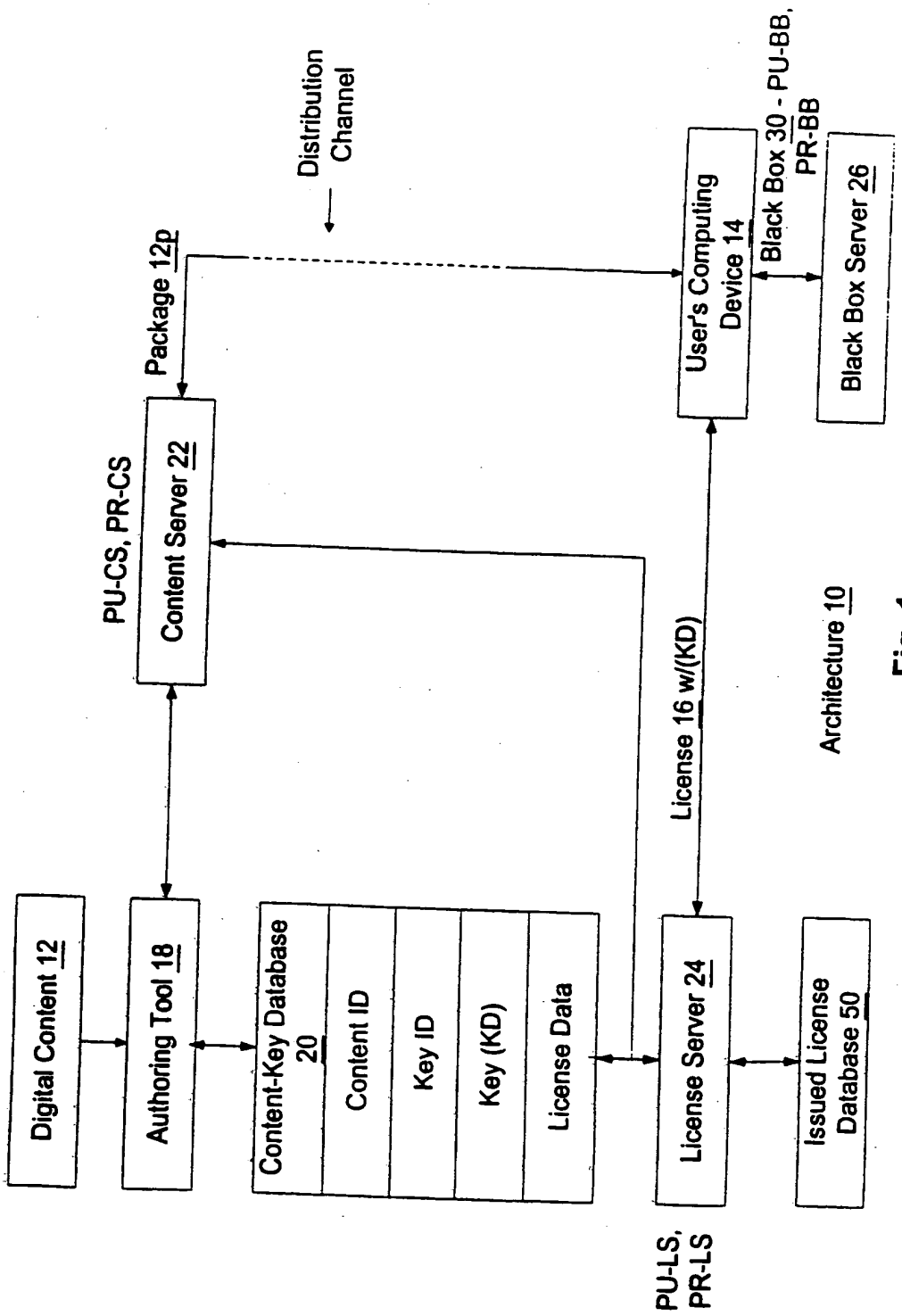

Distribution Channel
Package 12p
User's Computing Device 14
Black Box 30 - PU-BB, PR-BB
Black Box Server 26
PU-CS, PR-CS
Content Server 22
License 16 w/(KD)
Architecture 10
Digital Content 12
Authoring Tool 18
Content-Key Database 20
Content ID
Key ID
Key (KD)
License Data
License Server 24
Issued License Database 50
PU-LS, PR-LS

Fig. 1

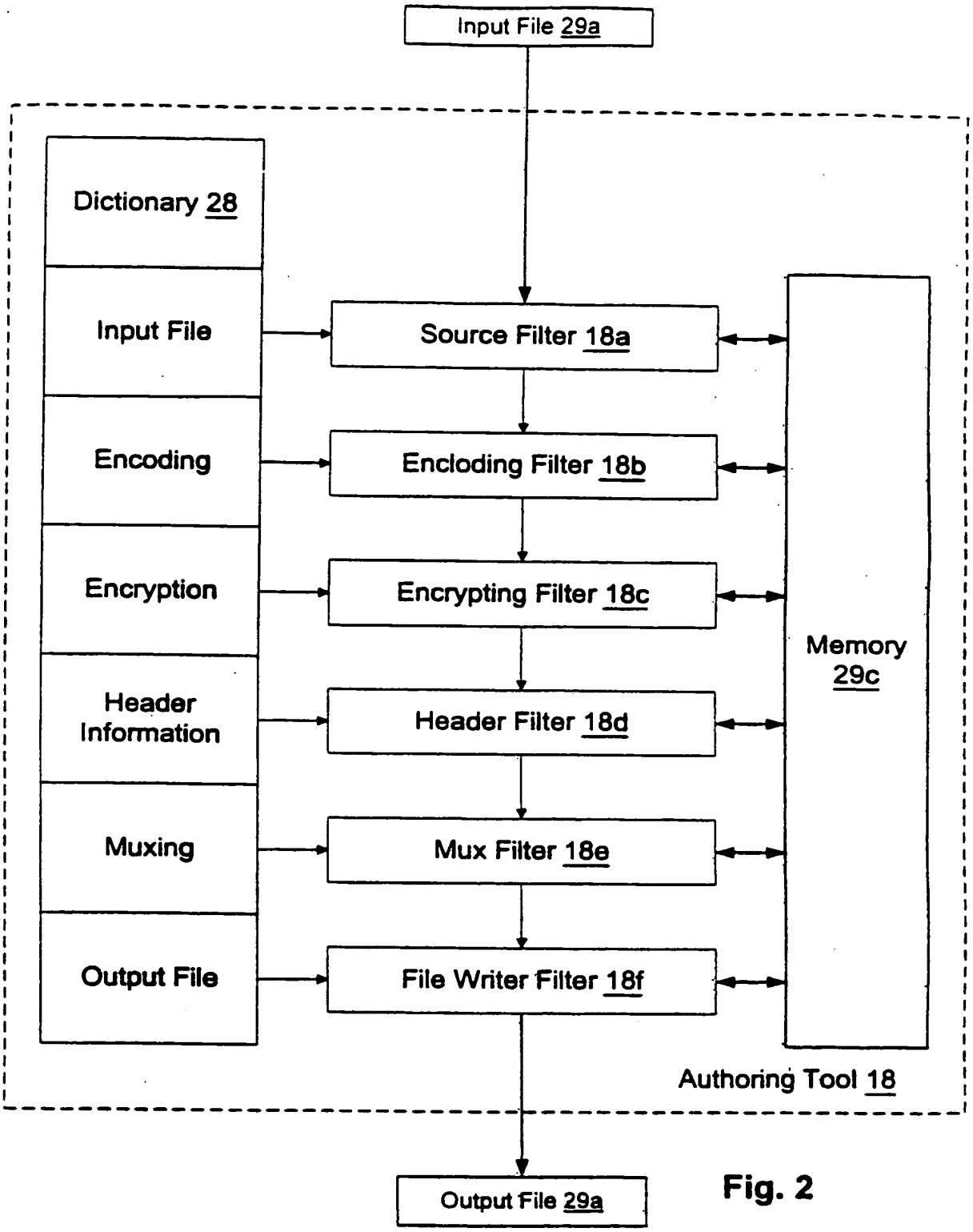

Input File 29a
Dictionary 28
Input File
Encoding
Encryption
Header Information
Muxing
Output File
Source Filter 18a
Encloding Filter 18b
Encrypting Filter 18c
Header Filter 18d
Mux Filter 18e
File Writer Filter 18f
Memory 29c
Authoring Tool 18
Output File 29a

Fig. 2

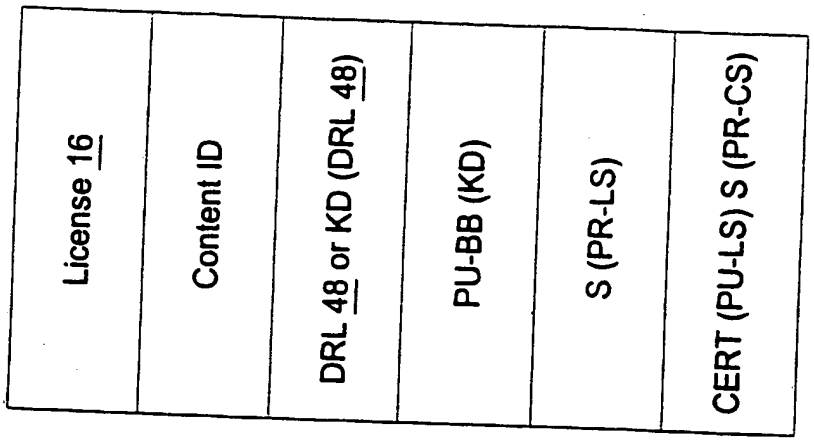

License 16
Content ID
DRL 48 or KD (DRL 48)
PU-BB (KD)
S (PR-LS)
CERT (PU-LS) S (PR-CS)

Fig. 8

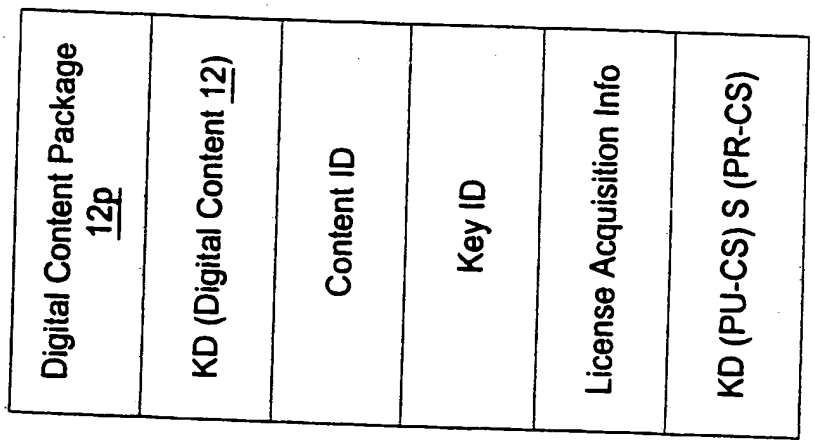

Digital Content Package 12p
KD (Digital Content 12)
Content ID
Key ID
License Acquisition Info
KD (PU-CS) S (PR-CS)

Fig. 3

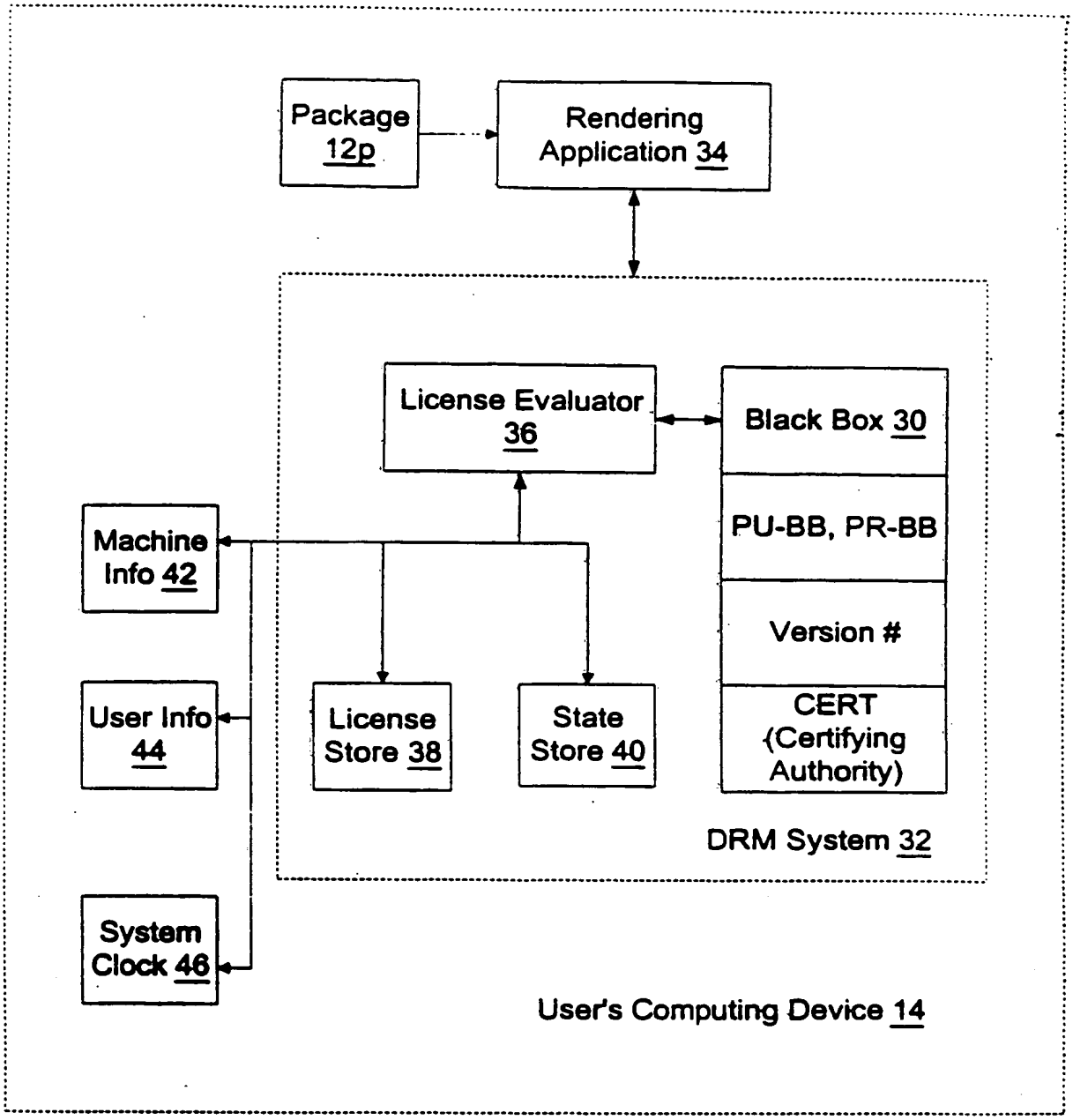

Package 12p
Rendering Application 34
License Evaluator 36
Black Box 30
PU-BB, PR-BB
Version #
CERT (Certifying Authority)
Machine Info 42
User Info 44
License Store 38
State Store 40
DRM System 32
System Clock 46
User's Computing Device 14

Fig. 4

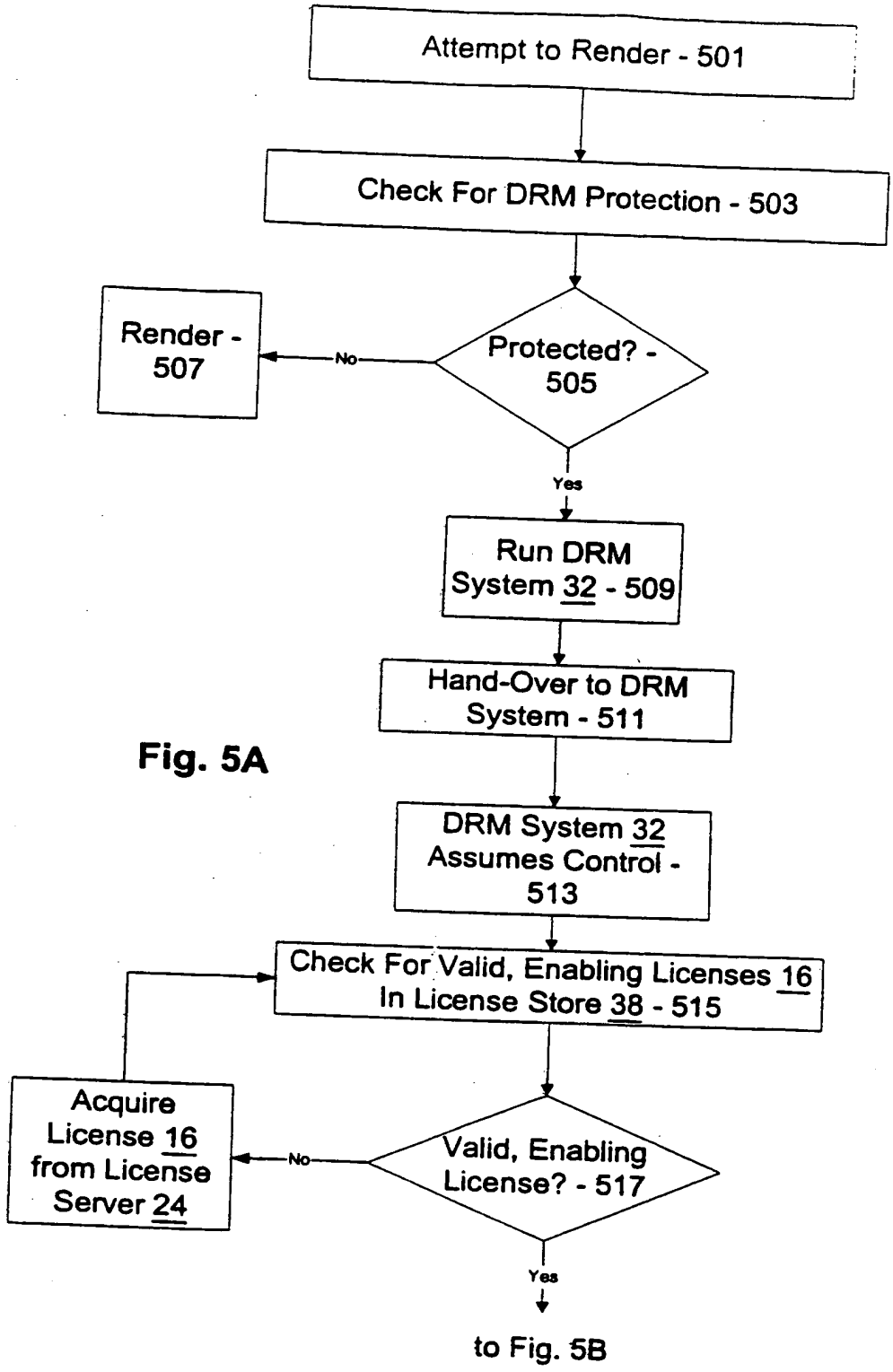

Attempt to Render - 501
Check For DRM Protection - 503
Protected? - 505
No
Render - 507
Yes
Run DRM System 32 - 509
Hand-Over to DRM System - 511
DRM System 32 Assumes Control - 513
Check For Valid, Enabling Licenses 16 In License Store 38 - 515
Valid, Enabling License? - 517
No
Acquire License 16 from License Server 24
Yes
to Fig. 5B

Fig. 5A

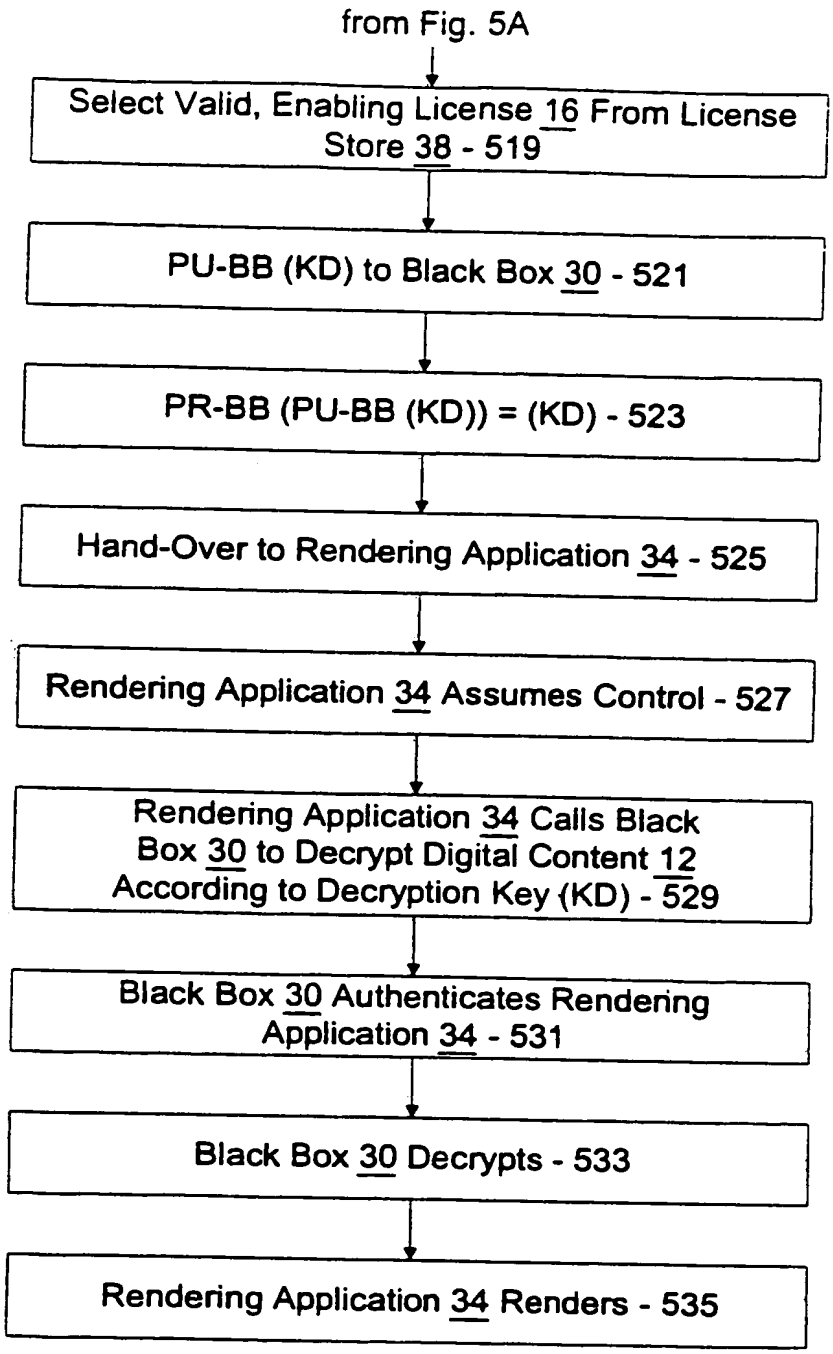

from Fig. 5A
Select Valid, Enabling License 16 From License Store 38 - 519
PU-BB (KD) to Black Box 30 - 521
PR-BB (PU-BB (KD)) = (KD) - 523
Hand-Over to Rendering Application 34 - 525
Rendering Application 34 Assumes Control - 527
Rendering Application 34 Calls Black Box 30 to Decrypt Digital Content 12 According to Decryption Key (KD) - 529
Black Box 30 Authenticates Rendering Application 34 - 531
Black Box 30 Decrypts - 533
Rendering Application 34 Renders - 535

Fig. 5B

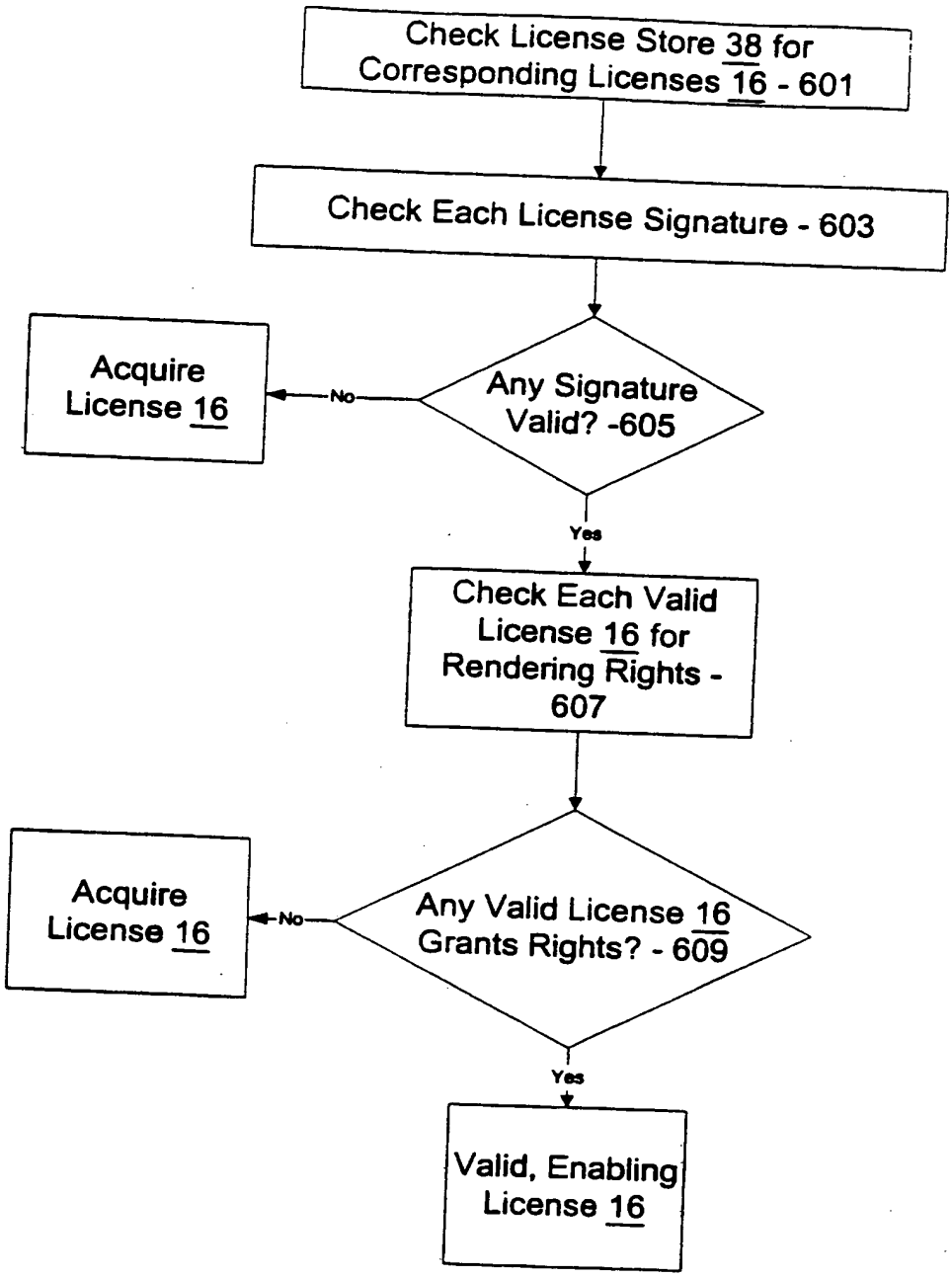

Check License Store 38 for Corresponding Licenses 16 - 601
Check Each License Signature - 603
Any Signature Valid? -605
No
Acquire License 16
Yes
Check Each Valid License 16 for Rendering Rights - 607
Any Valid License 16 Grants Rights? - 609
No
Acquire License 16
Yes
Valid, Enabling License 16

Fig. 6

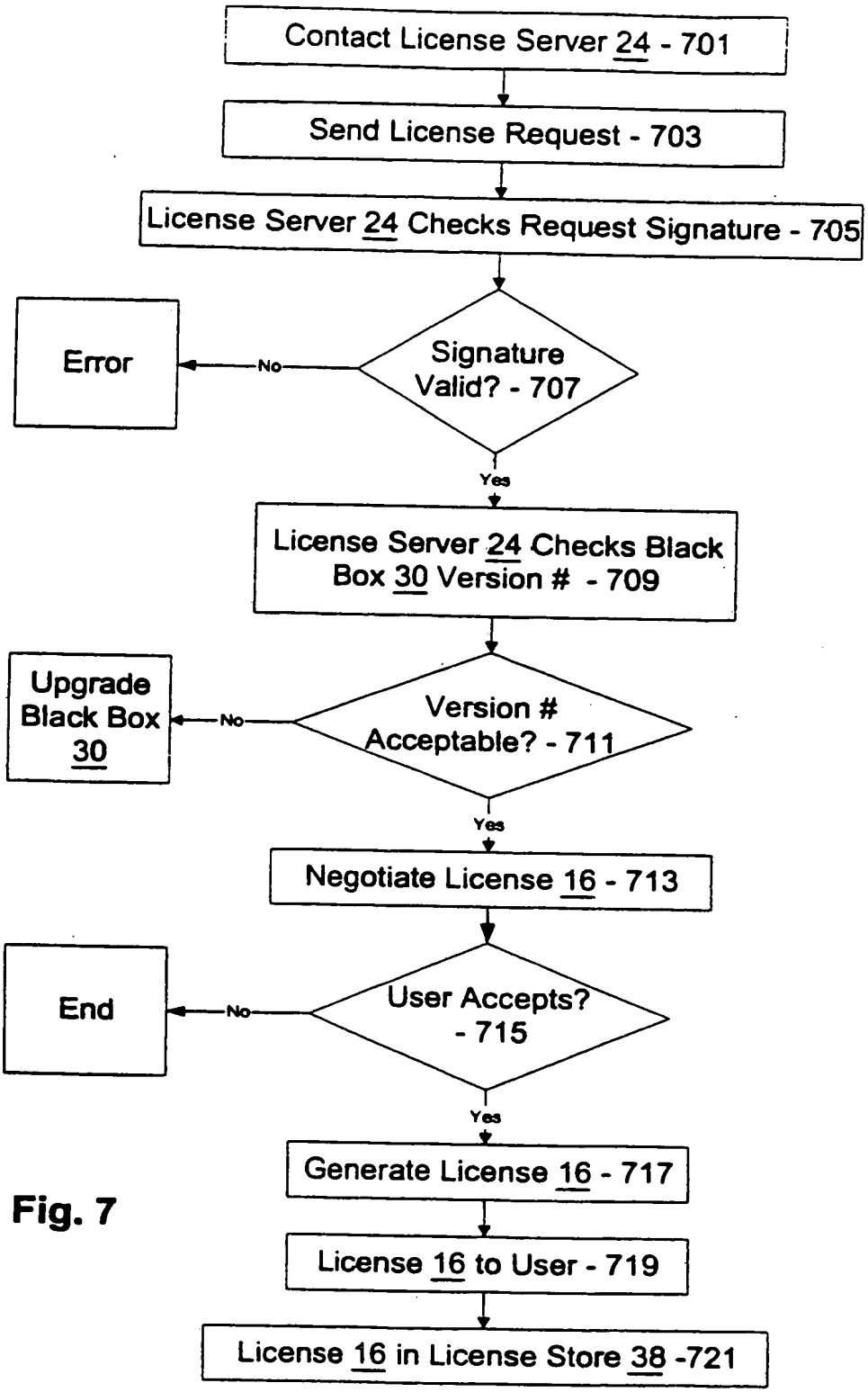

Contact License Server 24 - 701
Send License Request - 703
License Server 24 Checks Request Signature - 705
Signature Valid? - 707
No
Error
Yes
License Server 24 Checks Black Box 30 Version # - 709
Version # Acceptable? - 711
No
Upgrade Black Box 30
Yes
Negotiate License 16 - 713
User Accepts? - 715
No
End
Yes
Generate License 16 - 717
License 16 to User - 719
License 16 in License Store 38 -721

Fig. 7

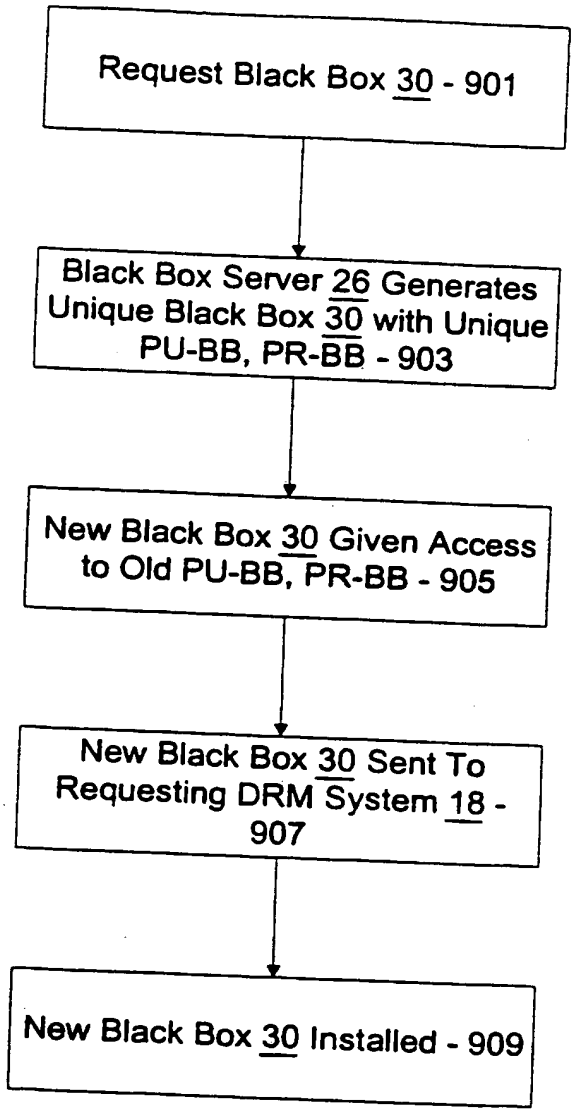

Request Black Box 30 - 901
Black Box Server 26 Generates Unique Black Box 30 with Unique PU-BB, PR-BB - 903
New Black Box 30 Given Access to Old PU-BB, PR-BB - 905
New Black Box 30 Sent To Requesting DRM System 18 - 907
New Black Box 30 Installed - 909

Fig. 9

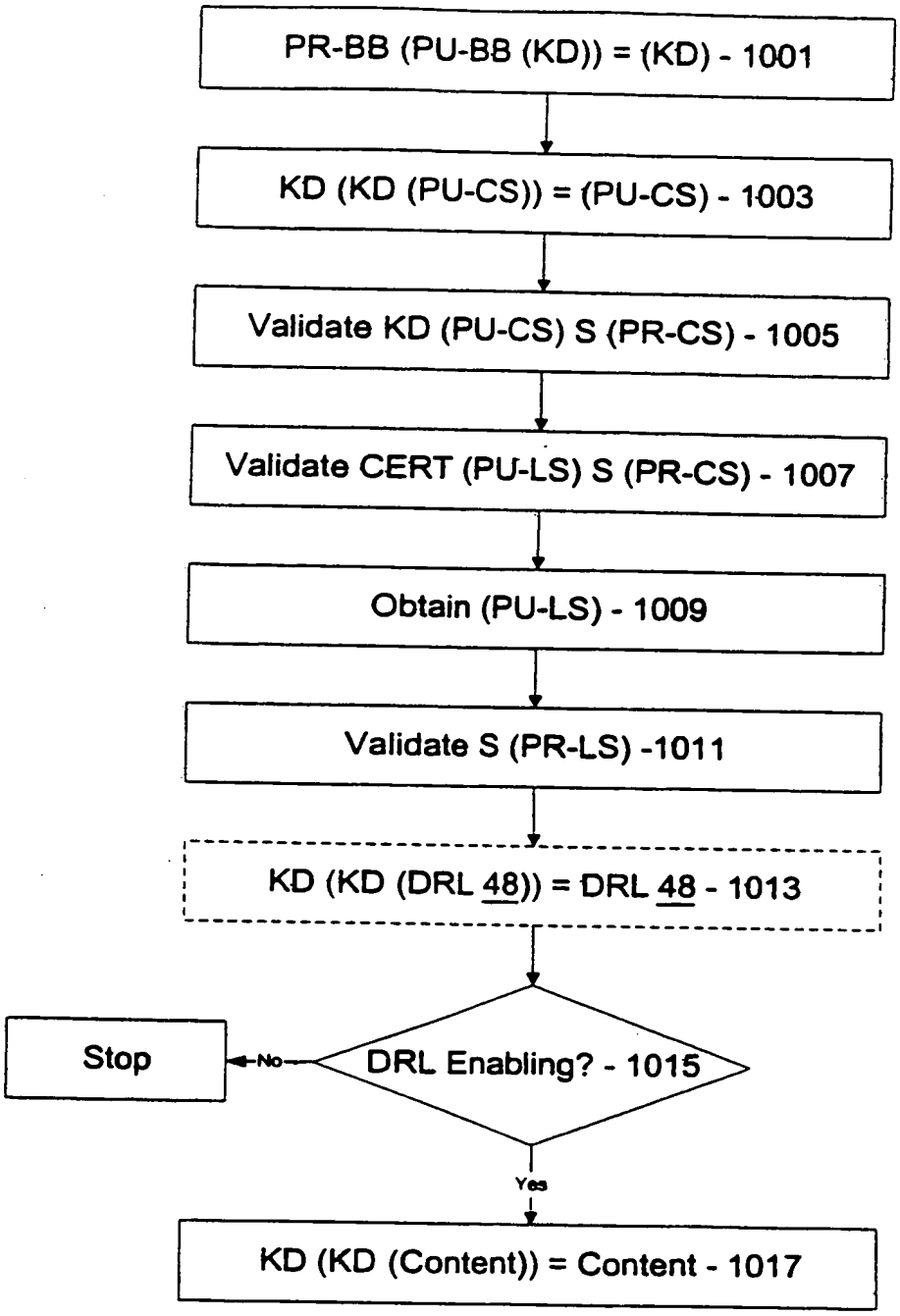

PR-BB (PU-BB (KD)) = (KD) - 1001
KD (KD (PU-CS)) = (PU-CS) - 1003
Validate KD (PU-CS) S (PR-CS) - 1005
Validate CERT (PU-LS) S (PR-CS) - 1007
Obtain (PU-LS) - 1009
Validate S (PR-LS) -1011
KD (KD (DRL 48)) = DRL 48 - 1013
Stop
No
DRL Enabling? - 1015
Yes
KD (KD (Content)) = Content - 1017

Fig. 10

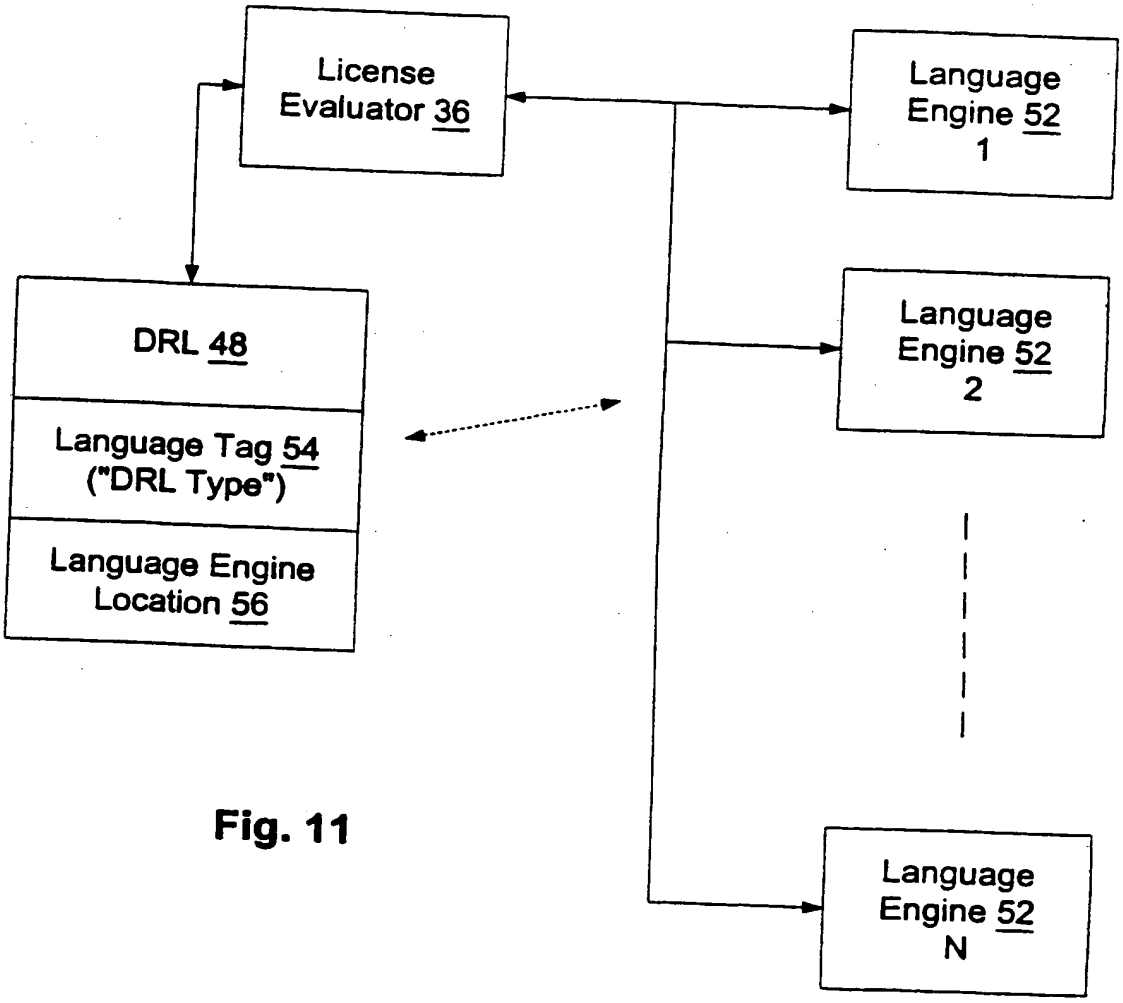

License Evaluator 36
Language Engine 52 1
DRL 48
Language Tag 54 ("DRL Type")
Language Engine Location 56
Language Engine 52 2
Language Engine 52 N

Fig. 11

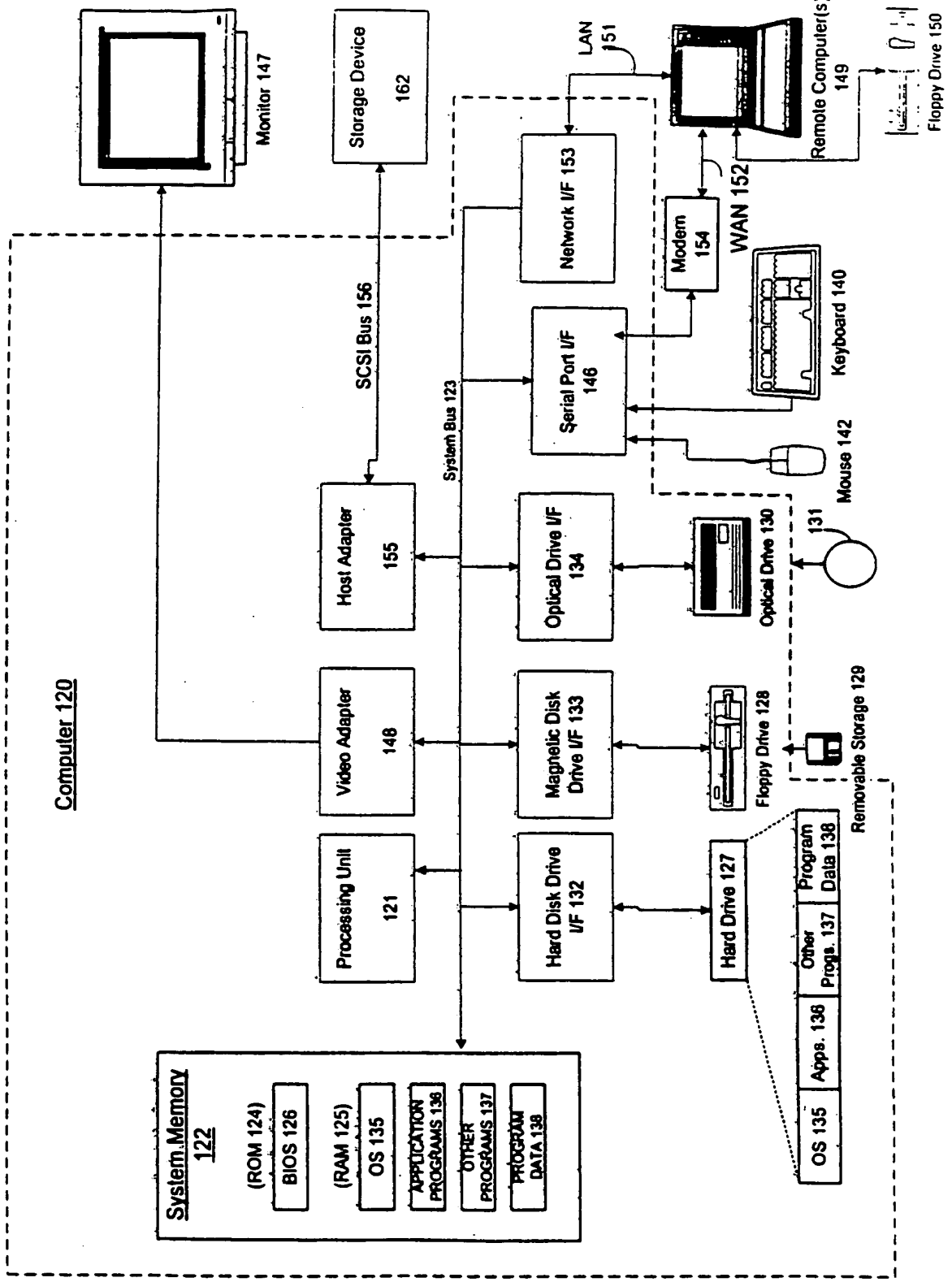

Computer 120
System Memory 122
(ROM 124)
BIOS 126
(RAM 125)
OS 135
APPLICATION PROGRAMS 136
OTHER PROGRAMS 137
PROGRAM DATA 138
Processing Unit 121
Video Adapter 148
Host Adapter 155
Hard Disk Drive I/F 132
Magnetic Disk Drive I/F 133
Optical Drive I/F 134
Serial Port I/F 146
Network I/F 153
Modem 154
WAN 152
LAN 151
Remote Computer(s) 149
Floppy Drive 150
Monitor 147
Storage Device 162
SCSI Bus 156
System Bus 123
Keyboard 140
Mouse 142
Optical Drive 130
131
Floppy Drive 128
Removable Storage 129
Hard Drive 127
OS 135
Appa. 138
Other Progs. 137
Program Data 138

Fig. 12

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 October 2000 (05.10.2000)

PCT

(10) International Publication Number
WO 00/059152 A3

(51) International Patent Classification[7]: G06F 17/60

(21) International Application Number: PCT/US00/04983

(22) International Filing Date: 25 February 2000 (25.02.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/126,614	27 March 1999 (27.03.1999)	US
09/290,363	12 April 1999 (12.04.1999)	US
09/482,928	13 January 2000 (13.01.2000)	US

(71) Applicant: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, WA 98052 (US).

(72) Inventors: **BLINN, Arnold, N.**; 9401 NE 27th Street, Bellevue, WA 98004 (US). **JONES, Thomas, C.**; 23617 NE 6th Street, Redmond, WA 98053-3618 (US).

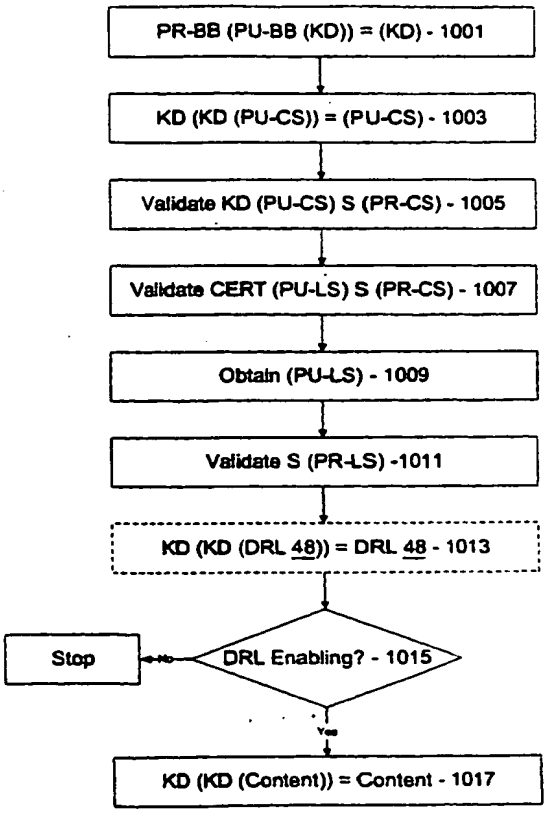
(74) Agents: **ROCCI, Steven, J.** et al.; Woodcock Washburn Kurtz Mackiewicz & Norris LLP, 46th floor, One Liberty Place, Philadelphia, PA 19103 (US).

(81) Designated States *(national)*: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, 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, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, 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), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHOD FOR INTERDEPENDENTLY VALIDATING A DIGITAL CONTENT PACKAGE AND A CORRESPONDING DIGITAL LICENSE



(57) Abstract: A method is disclosed for a device to interdependently validate a digital content package having a piece of digital content in an encrypted form, and a corresponding digital license for rendering the digital content. A first key is derived from a source available to the device, and a first digital signature is obtained from the digital content package. The first key is applied to the first digital signature to validate the first digital signature and the digital content package. A second key is derived based on the first digital signature, and a second digital signature is obtained from the license. The second key is applied to the second digital signature to validate the second digital signature and the license.

WO 00/059152 A3

Published:
— *with international search report*

(88) Date of publication of the international search report:
6 February 2003

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.

INTERNATIONAL SEARCH REPORT

Int ıal Application No
PCT/US 00/04983

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/60

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 7 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)
EPO-Internal, WPI Data, INSPEC, IBM-TDB, PAJ, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 98 09209 A (INTERTRUST TECHNOLOGIES CORP) 5 March 1998 (1998-03-05) page 51, line 5 -page 52, line 9 page 147, line 6 -page 149, line 21 page 174, line 10 -page 177, line 24 page 180, line 20 -page 195, line 10 page 334, line 10 -page 337, line 9 page 375, line 7 -page 390, line 14 page 393, line 18 -page 420, line 17 page 456, line 23 -page 460, line 19 page 466, line 13 -page 468, line 17 page 497, line 9 -page 505, line 7 page 577, line 1 -page 581, line 17 page 598, line 9 -page 648, line 15 --- -/--	1-52

[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

Date of the actual completion of the international search	Date of mailing of the international search report
18 July 2002	30/07/2002

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	Marcu, A

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Int al Application No
PCT/US 00/04983

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 715 245 A (XEROX CORP) 5 June 1996 (1996-06-05) abstract page 6, line 45 - line 57 page 7, line 31 -page 8, line 30 page 15, line 43 -page 16, line 4 page 25, line 35 -page 26, line 7	1-52
A	EP 0 665 486 A (AT & T CORP) 2 August 1995 (1995-08-02) abstract column 6, line 8 -column 8, line 28	1-52

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

formation on patent family members

In nal Application No
PCT/US 00/04983

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
WO 9809209	A	05-03-1998	US	5892900 A	06-04-1999
			AU	4170397 A	19-03-1998
			EP	0922248 A1	16-06-1999
			WO	9809209 A1	05-03-1998
EP 0715245	A	05-06-1996	US	5629980 A	13-05-1997
			EP	0715245 A1	05-06-1996
			JP	8263441 A	11-10-1996
EP 0665486	A	02-08-1995	US	5509074 A	16-04-1996
			CA	2137065 A1	28-07-1995
			EP	0665486 A2	02-08-1995
			JP	3121738 B2	09-01-2001
			JP	7239828 A	12-09-1995

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

(11) EP 0 731 404 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
11.09.1996 Bulletin 1996/37

(51) Int. Cl.[6]: **G06F 1/00**, G06F 19/00

(21) Application number: **96100832.3**

(22) Date of filing: **22.01.1996**

(84) Designated Contracting States:
DE FR GB

(30) Priority: **07.03.1995 US 401484**

(71) Applicant: **International Business Machines Corporation**
Armonk, N.Y. 10504 (US)

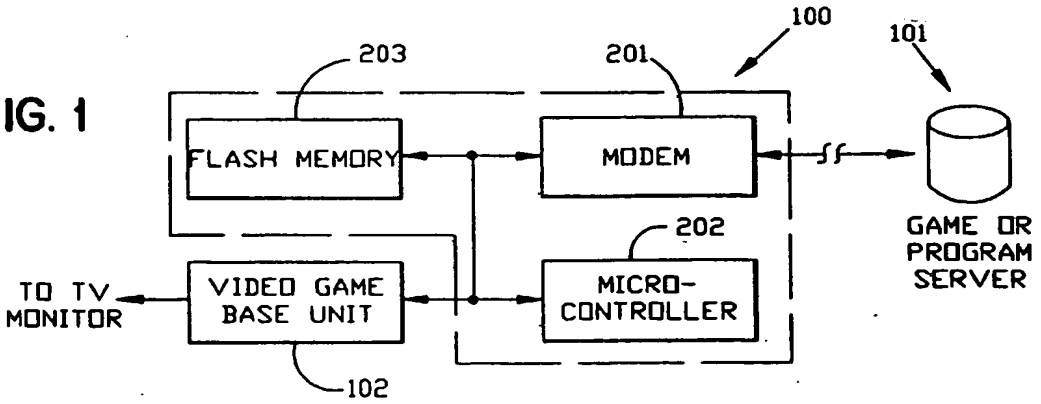
(72) Inventors:
- **Bakoglu, Halil Burhan**
  Ossining, New York 10562 (US)
- **Chen, Inching**
  Wappingers Falls, New York 12590 (US)
- **Lean, Andy Geng-Chyun**
  Merrick, New York 11566 (US)
- **Maruyama, Kiyoshi**
  Chappaqua, New York 10514 (US)
- **Yue, Chung-wai**
  Yorktown Heights, New York 10598 (US)

(74) Representative: **Rach, Werner, Dr.**
IBM Deutschland
Informationssysteme GmbH,
Patentwesen und Urheberrecht
70548 Stuttgart (DE)

(54) A universal electronic video game renting/distributing system

(57) A video game cartridge that can be plugged into a video game machine to enable a user to request and play a video game for a predetermined number of video frames. The cartridge has a receiver for receiving the video game program and the predetermined frame count in response to a request from the user. The program and frame count is then stored in a memory of the cartridge. Finally, the cartridge has a counter which changes its value when the user is actively playing the video game program. The counter ceases to change its value when the user is not playing the video game program. When the counter reaches a predetermined limit, the user is no longer authorized to play the video game program.

FIG. 1



EP 0 731 404 A1

Description

Technical Field

This invention relates to a video game cartridge for receiving video game programs from a remote server.

Description of the Prior Art

Today, there are many video games available for purchase or for rental at stores. Generally, there is no trial or test playing of the games in the stores, and there is no return on purchased games once the game package has been opened. Therefore, a person who is interested in any game has to buy it before playing it and thus may face the risk of not liking the game later. There is no return or refund of the game since the package has been opened. A person who rents a game from a store has to go through the usual VCR tape rental trouble of driving to the store, picking up the game and then later returning the game to the store.

To make video game rental easier for the consumer, Sega has created the Sega Channel. In this service, via cable and using a cable adapter unit which is plugged into the Sega Genesis game machine, people can play games that are downloaded to the cable adapter. It requires the on-line Sega Channel connection as well as the special adapter while the game is being played.

Down loading a software program to a personal computer over the modem connection exists today. Such software can come with a limited life where the life can be specified by expiration date, or time, or the number of times of the software usage. These schemes in limiting the software usage is not applicable to down loading video games to cartridges which are plugged into existing video game base units because these game base units do not have timer device built in. Thus a new scheme for controlling the usage of the game is needed.

The US Patent 4,905,280 to J.D. Wiedemer, et al describes a method for real time down loading of broadcast programs for pay-per-view or for subscription. Descrambling of broadcast programs is done by codes on a replaceable memory module, which is delivered to a subscriber by the service provider. This patent is applicable to the "purchase" of software content or real-time service, but it is not applicable to limiting the life of rented software.

US patent 5,251,909 to Reed et al describes software renting or distributing schemes in which access is granted to a subscriber prior to the actual programs being transmitted. This patent describes an off-line process and is not applicable to delivering software for rental purposes.

Summary of the Invention

It is an object of this invention to provide a portable video game cartridge which can be plugged into a video game machine base unit, such as Nintendos, Sega Genesis&tm. video game machine or Atari's Jaguar&tm. video game machine. The cartridge will allow a video game program to be used by receiving the video program over a telephone network or cable system.

The current invention describes a way of distributing and controlling the usage of a video game program (or any software program) by using a "watchdog mechanism" and by limiting the "life" of a game by limiting the total number of graphic frames that a video machine can generate. It offers a simple and effective way of software renting and distribution where game machines have no timer.

It is also an object of this invention to prevent piracy of video programs and programs in general by storing the frame count in a random location of the memory that is unknown to a potential pirate, especially if the count itself is encrypted. Since the count is part of the video game program or program execution path, the video game or program cannot be used without knowledge of the count.

This invention is generally an apparatus and method for enabling a user to request and use a program where the user receives the program and a frame count indicating the number of frames of the program that the user is authorized to execute or use. This program and the frame count is then stored in a memory. When the user is actively providing input to the program, the frame count changes. The frame count will cease to change when the user is not providing input to the program. When the count reaches a predetermined limit, the user is prevented from continuing use of the program.

This invention is a video game cartridge which can be plugged into a video game machine for enabling a user to receive and play a video game for a predetermined number of frames. The cartridge has a receiver for receiving the video program and for receiving a frame count indicating the number of video frames of the video game program that the user is authorized to play. The video program and frame count is then stored in a memory of the cartridge. The cartridge also has a counter which changes the frame count when the user is actively playing the video game program. When the user is not playing the video game program, the counter ceases to change its count. Finally when the counter reaches a predetermined limit, the user is prevented from further playing the video game program.

Brief Description of the Drawings

FIG. 1 schematically illustrates the major components of the video game cartridge along with a video game machine and a remote server.

FIG. 2 is a functional diagram showing the functions of each of the major components of the video game cartridge.

FIG. 3 schematically illustrates the flow chart for the watch "dog mechanism".

Description of the Preferred Embodiment

FIG. 1 illustrates a sample diagram of a electronic game or program renting system setup. The dotted line encloses the portable and programmable game cartridge unit 100 that can be plugged into a video game machine base unit 102, such as Sega Genesis&tm. video game machine, and remotely be connected to a video game server 101 via a modem connection. The connection to the remote video server can be through cable TV, or other telecommunication facilities.

When a video game base unit 102 is powered on, a user could either play a game (or games) stored in the programmable game cartridge 100 or place an order of a new game (either for rental or for purchase) to the game or program server 101. The cartridge 100 contains screen assistance (and voice assistance) to help place an order for a video game program to the server 101.

FIG. 2 illustrates the components of the video game cartridge unit 100. It consists of modem 201, microcontroller 202, flash memory 203 and an interface 204 to the video game base unit 102. The modem 201 performs the interface to the telephone or cable network. It can optionally perform decompression of received game or software if necessary. The received game is stored in flash memory 203. The game comes with its "life" which is indicated by the total number of graphic frames the video game machine 102 is authorized to generate when the game is actively played. For example, the game machine could render game graphics frame by frame at the rate of thirty framers per second.

After the number of graphic frames is exhausted, further playing of the game is prevented by the following mechanism. The flash memory 203 also stores a "watchdog mechanism" which keeps track of the remaining life of the game. An hourglass routine is embedded in the watchdog mechanism which is executed by microcontroller 202. This watchdog mechanism updates and tracks down a specified register in the flash memory 203 with its location randomly determined by the game server 101 in FIG. 1 during the down loading of the game.

The use of expiration date or time for voiding the game is an obvious approach if the video game base unit 102 comes with a timer. Since this patent application assumes a game base unit 102 which has no timer (which is the case of many existing game machines), the "life" of the rented game is determined by the total number of graphic frames that the base game unit can generate. This "life", or frame count, is what a renter gets when a game is down loaded. It is stored into a location in the flash memory 203. The location into which the frame count is stored in the flash memory is determined randomly by the video server at the time of the game down loading. The video game can resume at any time when it is being turned on, provided there is available frame count stored in the designated random location. The microcontroller 202 can pick up the frame count and allow the renting period, and thus the game or software, to be continued. As the rented game is being played, the frame count is decremented. When the user turns off the power, the hourglass routine in memory 203 will first store the remaining frame count to a random location in the non- volatile memory 203 and then shut down the game. The rental expires when there is no frame count remaining. The microcontroller 202 will not allow any portion of the game to be played by the game base unit 102 when the frame count reaches zero.

FIG. 3 illustrates the watchdog mechanism embedded with the video game program execution path that contains the hourglass routine which serves as part of the watchdog mechanism which can expire the game. When the user starts the game, the frame count is first fetched (305) and checked (306). If the frame count reaches zero, the game is over even though the game unit still has its power on (306N). If the frame count is still greater than zero (306Y), the scanner continues to monitor the game player's input in playing the video game (307). No active input (307) means the player is not playing the video game, and the scanner continues to monitor the player inputs from the key pad connected to the video game. When there is no active input, the video game will not render any game graphic frames. Therefore, the game program execution path will fall through decisions 308 and 309 and immediately return to continue scanning (307). When the game is not actively played and the player leaves the game machine's power on, the game will be sitting idle without rendering any new graphic frames. The frame count will not be consumed until the player becomes active again in playing the game as detected by the scanner (307 and 308).

If the player's input has been detected as active (307), a check is made to see if graphic rendering is required (309). Graphics rendering is required when the game program determines that the input signals from the key pad connected to the video game are valid signals. If rendering is required (309Y), the frame counter will be decremented (301). The hourglass routine (301 and 302) decrements the frame count and checks for any frame count left.

If the count is valid (302Y), then the program flows back to (310) which is the game program main collections, and then at the same time, 302 Y:sup.':esup. branches to check for power-off condition (303).

If the user decides to power-off the game, the watchdog mechanism will go through decision (303) and the shutdown routine (304) to store any remaining frame count in the flash memory. The shutdown routine stores the remaining frame count in the flash memory and exits the game. In summary flowchart components (301-306) and their associated flash memory form the "watchdog mechanism" that contains the hourglass rou-

tine (301 and 302) to keep track of the games "life" (remaining frame count). The watchdog mechanism also insures that the game can be resumed if there is still a valid frame count in the flash memory. Microcontroller (202) can also give advance warning when the rental is about to expire. Rental extension, if desired, can be downloaded again by the server (101) through a telephone or cable connection. Thus, server (101) in FIG. 1 has complete control over the game playing time, which should reflect the user's request for renting the game.

Although this embodiment was described in terms of a video game program in a cartridge, this invention can be extended to software programs in general. As long as the programs monitor user inputs, a scanner and watchdog mechanism can be implemented in similar fashion using a non-volatile memory.

The watchdog mechanism can even be made more secure by encrypting the frame count, which is stored at a random location in the memory. Even if the would-be pirate stumbles across the count in the memory, he/she wouldn't know what he/she found.

Claims

1. An apparatus for enabling a user to request and use a program, said apparatus comprising:

   a. a receiver for receiving the program and a frame count indicating a number of frames of the program that is authorized to be executed by the user;

   b. a memory for storing the program and the frame count received by the receiver; and

   c. a counter for changing the frame count when the user is actively providing input to the program, wherein the counter ceases to change its count when the user is not providing input to the program, and wherein the user is prevented from continuing use of the program when the counter reaches a predetermined limit.

2. An apparatus as recited in claim 1, further comprising:

   means for randomly determining an address in the memory in which the frame count is to be stored, and wherein the address is unknown to the user.

3. A method of enabling a user to request and use a program, said method comprising:

   a. receiving the game program and a frame count indicating a number of frames of the program that is authorized to be used by the user in response to a request;

   b. a memory for storing the program and the frame count; and

   c. changing the frame count when the user is actively using the program, wherein the frame count ceases to change when the user is not using the program and wherein the user is prevented from continuing use of the program when the counter reaches a predetermined limit.

4. A method as recited in claim 3, wherein the frame count is stored in a randomly determined location in the memory.

5. A video game cartridge which can be plugged into, for operation with, a video game machine to enable a user to request and play a video game program which is received from a remotely located server, said video game cartridge comprising:

   a. a receiver for receiving from the server the video game program and a frame count indicating a number of frames of the video game program that is authorized to be played by the user in response to a request;

   b. a memory for storing the video game program and the frame count received by the receiver; and

   c. a counter for changing the frame count when the user is actively playing the video game program, wherein the counter ceases to change its count when the user is not playing the video game program, and wherein the user is prevented from further playing the video game program when the counter reaches a predetermined limit, indicating that the user has played said video game for the number of frames.

6. A video game cartridge as recited in claim 5, further comprising:

   means for randomly determining an address in the memory in which the frame count is to be stored.

7. A video game cartridge as recited in claim 5, further comprising:

   a modem for transmitting to the server the request from the user to play a video game program.

8. A video game cartridge, as recited in claim 5, wherein said memory is a non-volatile memory.

9. A video game cartridge, as recited in claim 8, wherein the frame count indicated in the counter is stored in the memory when power for the video game machine is turned off.

10. A video game cartridge, as recited in claim 9, further comprising:

a means for fetching the frame count stored in the memory when power for said game machine is turned on.

11. A video game cartridge which can be plugged into, for operation with, a video game machine to enable a user to request and play a video game program which is received from a remotely located server, said video game cartridge comprising:

a. a modem for transmitting from the user over a telephone or cable network a request to receive the video game from the server, and for receiving the video game program and frame count from the server over the telephone or cable network, the frame count indicating a predetermined number of frames of the video game program that is authorized to be played by the user in response to the request;

b. a non-volatile memory for storing the video game program and the frame count;

c. a counter for changing the frame count when the player is actively playing the video game;

d. a means for storing the changed frame count of the counter in the memory when the power to the video game machine is turned off; and

e. a means for fetching the changed frame count stored in the memory in step (d) when the player resumes playing the video game, wherein the user is prevented from further playing of the video game program when the frame count of the counter reaches a predetermined limit, indicating that the user has played said video game for the predetermined number of frames.

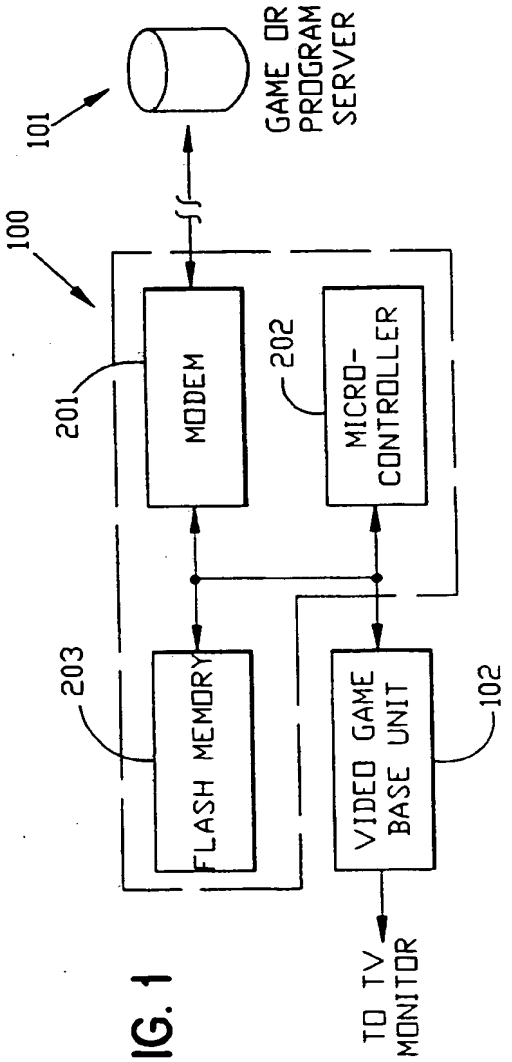

GAME OR PROGRAM SERVER
101
100
201
MODEM
202
MICRO-CONTROLLER
203
FLASH MEMORY
VIDEO GAME BASE UNIT
102
TO TV MONITOR

FIG. 1

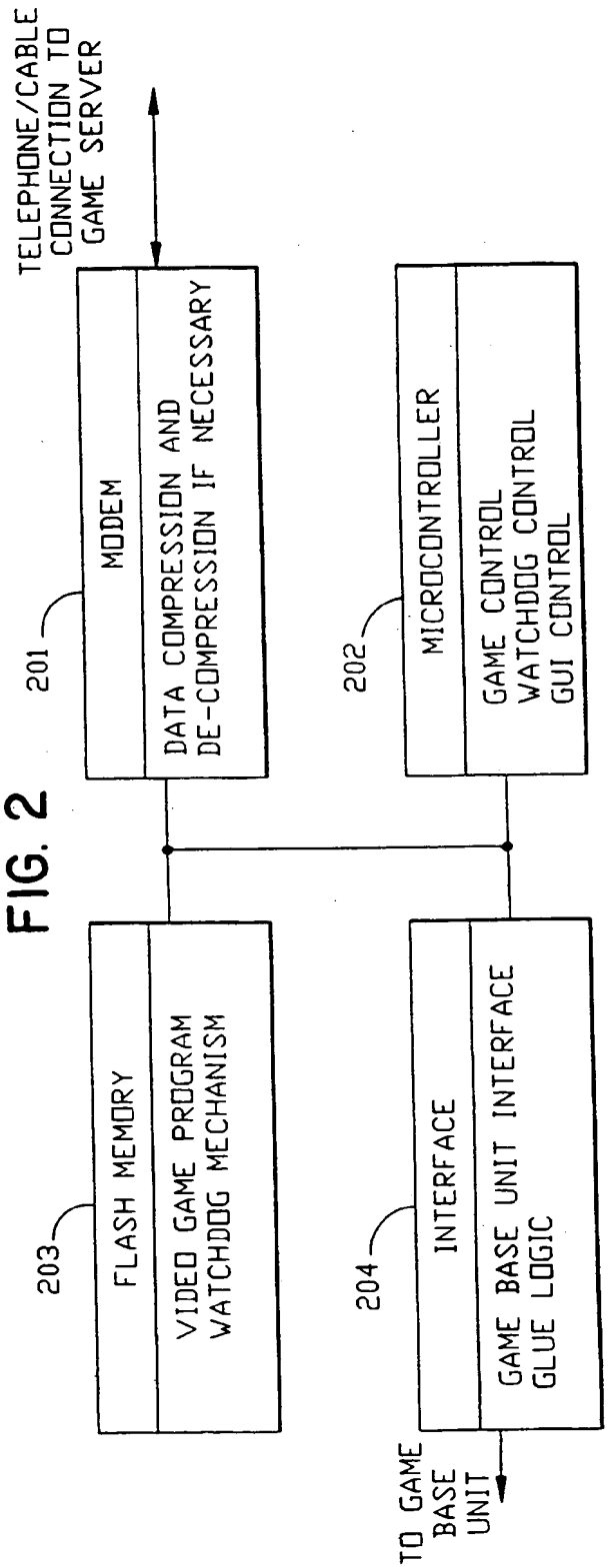
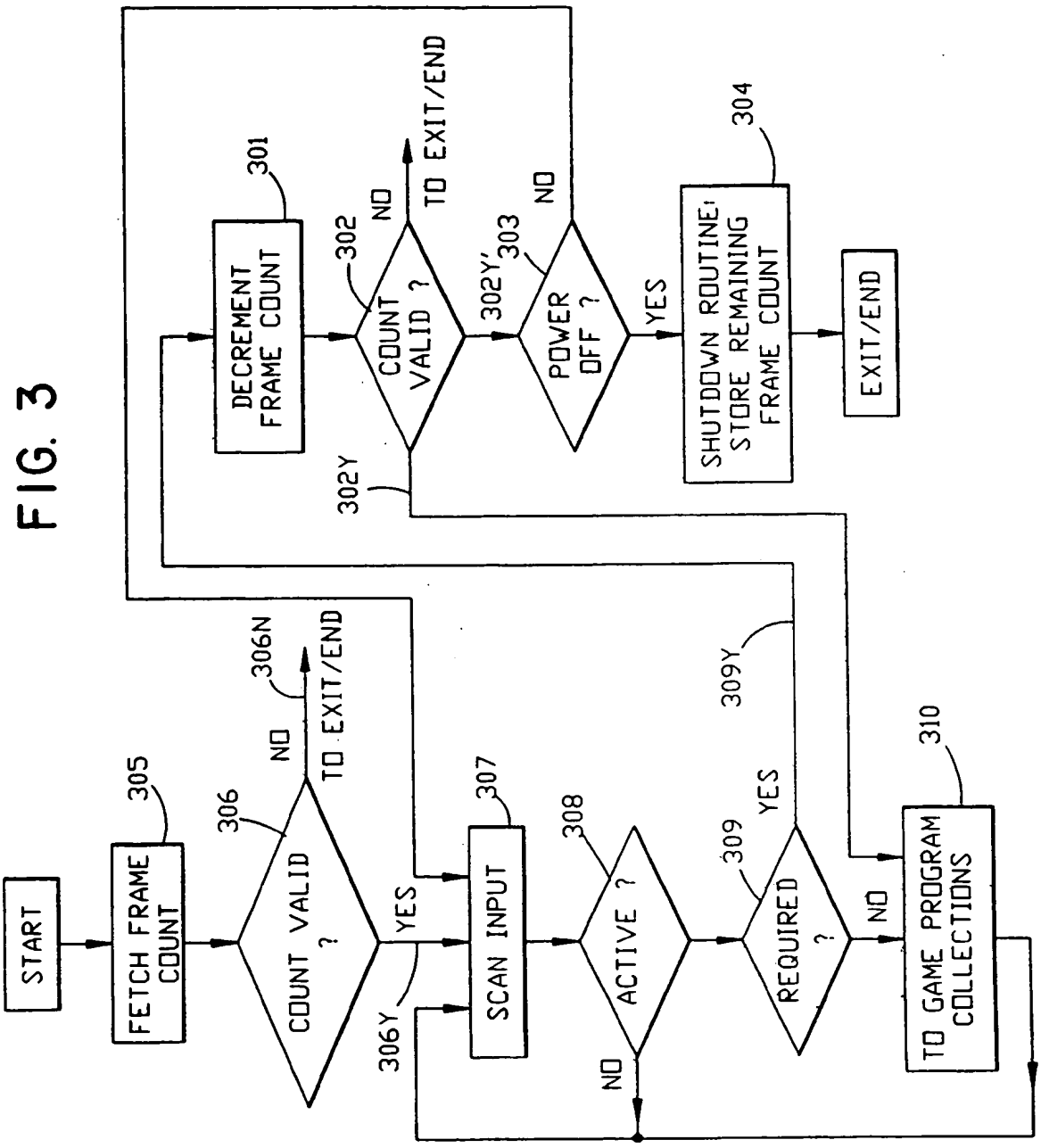

TELEPHONE/CABLE CONNECTION TO GAME SERVER
201
MODEM
DATA COMPRESSION AND DE-COMPRESSION IF NECESSARY
202
MICROCONTROLLER
GAME CONTROL
WATCHDOG CONTROL
GUI CONTROL
203
FLASH MEMORY
VIDEO GAME PROGRAM
WATCHDOG MECHANISM
204
INTERFACE
GAME BASE UNIT INTERFACE
GLUE LOGIC
TO GAME BASE UNIT

FIG. 2

FIG. 3


START
FETCH FRAME COUNT
305
COUNT VALID ?
306
NO
306N
TO EXIT/END
YES
306Y
SCAN INPUT
307
ACTIVE ?
308
NO
REQUIRED ?
309
YES
309Y
NO
TO GAME PROGRAM COLLECTIONS
310
DECREMENT FRAME COUNT
301
COUNT VALID ?
302
NO
TO EXIT/END
302Y
POWER OFF ?
303
302Y'
YES
SHUTDOWN ROUTINE: STORE REMAINING FRAME COUNT
304
EXIT/END

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 96 10 0832

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)
P,A	EP-A-0 671 711 (SEGA ENTERPRISES KK) 13 September 1995 * the whole document *	1-11	G06F1/00 G06F19/00
A	IBM TECHNICAL DISCLOSURE BULLETIN, vol. 37, no. 3, 1 March 1994, pages 413-417, XP000441522 "MULTIMEDIA MIXED OBJECT ENVELOPES SUPORTING A GRADUATED FEE SCHEME VIA ENCRYPTION" * page 413, line 1 - page 414, line 14 *	1-11	
A	WO-A-93 01550 (INFOLOGIC SOFTWARE INC) 21 January 1993 * page 1, line 1 - page 8, line 32 * * claims 1-3 *	1-11	

TECHNICAL FIELDS SEARCHED (Int.Cl.6)

G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	14 June 1996	Powell, D

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

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

(11) Publication number: **0 332 304 A3**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number: **89301510.7**

(51) Int. Cl.5: **G06F 1/00**

(22) Date of filing: **16.02.89**

(30) Priority: **07.03.88 US 164944**

(43) Date of publication of application:
13.09.89 Bulletin 89/37

(84) Designated Contracting States:
DE FR GB

(88) Date of deferred publication of the search report:
26.02.92 Bulletin 92/09

(71) Applicant: **DIGITAL EQUIPMENT CORPORATION**
111 Powdermill Road
Maynard Massachusetts 01754-1418(US)

(72) Inventor: **Robert, Gregory**
12 Carson Circle
Nashua New Hampshire 03062(US)
Inventor: **Chase, David**
28 Bay View Road
Wellesley Massachusetts 02181(US)
Inventor: **Schaefer, Ronald**
7 Gioconda Avenue
Acton Massachusetts 01720(US)

(74) Representative: **Goodman, Christopher et al**
Eric Potter & Clarkson St. Mary's Court St. Mary's Gate
Nottingham NG1 1LE(GB)

(54) Software licensing management system.

(57) A license management system which includes a license management facility that determines whether usage of a licensed program is within the scope of the license. The license management system maintains a license unit value for each licensed program and a pointer to a table identifying an allocation unit value associated with each use of the licensed program. In response to a request to use a licensed program, the license management system responds with an indication as to whether the license unit value exceeds the allocation unit value associated with the use. Upon receiving the response, the operation of the licensed program depends upon policies established by the licensor.

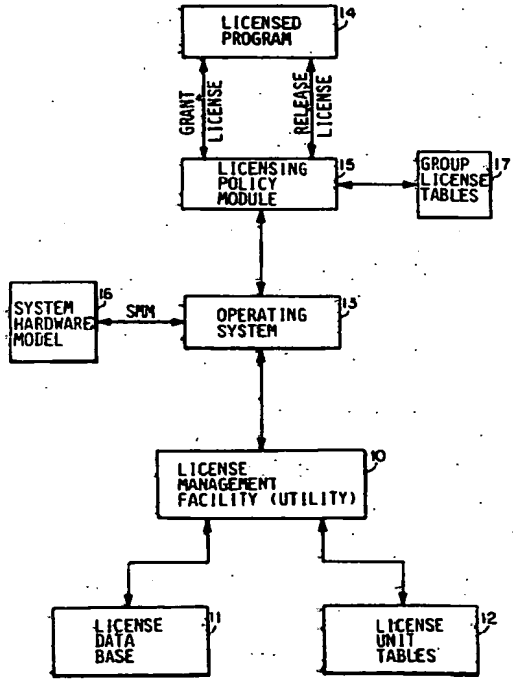


FIG. 1

EP 0 332 304 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number

EP 89 30 1510

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.4)
Y	US-A-4 471 163 (DONALD ET AL.) * figure 1 * * column 3, line 22 - line 33 * * column 4, line 4 - line 13 *	1-5,13	G06F1/00
Y	US-A-4 590 557 (LILLIE) * figures 1,3 * * column 2, line 47 - line 68 * * column 5, line 36 - line 43 *	1-5,13	
Y	IBM TECHNICAL DISCLOSURE BULLETIN. vol. 30, no. 9, February 1988, NEW YORK US pages 37 - 38; 'PROTECTION OF LICENSED SOFTWARE APPLICATIONS IN A NETWORK ENVIRONMENT'	13	
A	* the whole document *	1,3,8,9	

TECHNICAL FIELDS SEARCHED (Int. Cl.4)

G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	26 NOVEMBER 1991	WEISS P.

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 (P0401)

cense usage allocation unit value, the licensing policy module determines whether to allow the licensed program to be used in response to other licensing policy factors.

BRIEF DESCRIPTION OF THE DRAWINGS

This invention is pointed out with particularity in the appended claims. The above and further advantages of this invention may be better understood by referring to the following description taken in conjunction with the accompanying drawings, in which:

Fig. 1 is a general block diagram of a new system in accordance with the invention;

Figs. 2 and 3 are diagrams of data structures useful in understanding the detailed operation of the system depicted in Fig. 1; and

Figs. 4A-1 through 4B-2 are flow diagrams which are useful in understanding the detailed operations of the system depicted in Fig. 1.

DETAILED DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

Fig. 1 depicts a general block diagram of a system in accordance with the invention for use in connection with a digital data processing system which assists in managing software use in accordance with software licenses. With reference to Fig. 1, the new system includes a license management facility 10 which operates in conjunction with a license data base 11 and license unit tables 12, and under control of an operating system 13 and licensing policy module 15 to control use of licensed programs, such as licensed program 14, so that the use is in accordance with the terms of the software license which controls the use of the software program on a system 16 identified by a system marketing model (SMM) code in a digital data processing system.

As is conventional, the digital data processing system including the licensing management system may include one or more systems 16, each including one or more processors, memories and input/output units, interconnected in a number of ways. For example, the digital data processing system may comprise one processor, which may include a central processor unit which controls the system and one or more auxiliary processors which assist the central processor unit. Alternatively, the digital data processing system may comprise multiple processing systems, in which multiple central processor units are tightly coupled, or clustered or networked systems in which multiple central processor units are loosely coupled, generally operating relatively autonomously, interacting by means of messages transmitted over a cluster or network connection. In a tightly coupled multiple processing system, for example, it may be desirable to control the number of users which may use a particular software program at one time. A similar restriction may be obtained in a cluster or network environment by controlling the number of particular nodes, that is, connections to the communications link in the cluster or network over which messages are transferred. In addition, since the diverse processors which may be included in a digital data processing system may have diverse processing speeds and powers, represented by differing system marketing model (SMM) codes, it may be desirable to include a factor for speeds and power in determining the number of processors on which a program may be used concurrently.

As will be explained in greater detail below, the license data base 11 contains a plurality of entries 20 (described below in connection with Fig. 2) each containing information relating to the terms of the license for a particular licensed program 14. In one embodiment such information may include a termination date, if the license is for a particular time period or expires on a particular date, and a number of licensing units if the license is limited by usage of the license program. In that embodiment, the entry also includes identification of a license unit table 40 (described below in connection with Fig. 3) in the license unit tables 12 that identifies the number of allocation units for usage of the licensed program on the types of systems 16 which may be used in the digital data processing system as represented by the system marketing model (SMM) codes.

When a user wishes to use a licensed program 14, a GRANT LICENSE request message is generated which requests information as to the licensing status of the licensed program 14. The GRANT LICENSE request message is transmitted to the licensing policy module 15, which notifies the operating system of the request. The operating system 13, in turn, passes the request, along with the system marketing model of the specific system 16 being used by the user, to the license management facility 10 which determines whether use of the program is permitted under the license.

In response to the receipt of the GRANT LICENSE request from the user and the system marketing model (SMM) code of the system 16 being used by the user on which the licensed program will be processed, the license management facility 10 obtains from the license data base the entry 20 associated with the licensed program

14 and determines whether the use of the licensed program 14 is within the terms of the license as indicated by the information in the license data base 11 and the license unit tables 12.

In particular, the license management facility 10 retrieves the contents of the entry 20 associated with the licensed program. If the entry 20 indicates a termination data, the license management facility 10 compares the system data, which is maintained by the digital data processing system in a conventional manner, with the termination date identified in the entry. If the system date is after the termination date identified in the entry 20, the license has expired and the license management facility 10 generates a usage disapproved message, which it transmits to the operating system 13. On the other hand, if the termination date indicated in the entry 20 is after the system date, the license has not expired and the license management facility 10 proceeds to determine whether the usage of the licensed program 14 is permitted under other terms of the license which may be embodied in the entry 20.

In particular, the license management facility 10 then determines whether the usage of the licensed program is permitted under usage limitations. In that operation, the license management facility obtains the number of license units remaining, which indicates usage of the licensed program 14 not including the usage requested by the user, as well the identification of the table 40 in license unit tables 12 associated with the licensed program 14. The license management facility 10 then compares the number of license units which would be allocated for use of the licensed program 14, which it obtains from the table 40 identified by entry 20 in the license data base 11, and the number of remaining units to determine whether sufficient license units remain to permit usage of the licensed program 14.

If the number of remaining license units indicated by entry 20 in the license data base 11 exceeds the number, from license unit tables 12, of license units which would be allocated for use of the licensed progam 14, the usage of the licensed program is permitted under the license. Accordingly, the license management facility transmits a usage approved response to the operating system 13. In addition, the license management facility 10 adjusts the number of remaining license units in entry 20 by a function of the license units allocated to use of the licensed program to reflect the usage.

On the other hand, if the number of remaining license units indicated by entry 20 in the license data base is less than the number of license units which would be allocated for use of the licensed program 14, the usage of the licensed program 14 is not permitted by the license. In that case, the license management facility 10 transmits a usage disapproved response to the operating system 13. In addition, the license management facility 10 may also log the usage disapproved response; this information may be used by a system operator to determine whether usage of the licensed program 14 is such as to warrant obtaining an enlarged license.

Upon receipt of either a usage approved response or a usage disapproved response to the GRANT LICENSE request, the operating system 13 passes the response to the licensing policy module 15. If a usage approved response is received, the licensing policy module normally allows usage of the licensed program 14. If a usage disapproved response is received, the licensing policy module determines whether the usage of the licensed program may be permitted for other reasons. For example, usage of the licensed program 14 may be permitted under a group license, whose terms are embodied in entries in group license tables 17. Under a group license, usage may be permitted of any of a group of licensed programs. The operations to determine to whether usage is permitted may be performed in the same manner as described above in connection with license management facility 10. In addition, if the usage of the licensed program 14 is not permitted under a group license, usage may nonetheless be permitted under the licensor's licensing practices, which may be embodied in the licensing policy module 15. If the licensing policy module determines that usage of the program should be permitted, notwithstanding a usage disapproved response from the license management facility 10, because the usage is permitted under a group license or the licensor's licensing practices, the licensing policy module 15 permits usage of the licensed program. Otherwise, the licensing policy module does not permit usage of the licensed program in response to the GRANT LICENSE request.

When a user no longer requires use of a licensed program 14, it transmits a RELEASE LICENSE request to the licensing policy module 15. The operations performed by the licensing policy module depend on the basis for permitting usage of the licensed program. If usage was permitted as a result of a group license, if the group license is limited by usage, the licensing policy module 15, if necessary, adjusts the records in the group license tables 17 related to the group license to reflect the fact that the licensed program 14 related to the group license is not being used. If the usage was permitted as a result of a group license which is not limited by usage, but instead is limited in duration, or if the usage was permitted in response to the licensor's licensing policies, the licensing policy module 15 need do nothing. If the licensing

policy module 15 maintains a log of usage outside the scope of a group or program license, it may make an entry in the log of the RELEASE request.

Finally, if usage was permitted as a result of the license management facility 10 providing an approve usage response to the GRANT LICENSE request, the licensing policy module 15 transmits the RELEASE LICENSE request to the operating system 13. In response, the operating system 13 transfers the RELEASE LICENSE request to the license management facility 10, along with an identification of the system 16 using the licensed program 14. The license management facility 10 then obtains from the license data base the identification of the appropriate license usage allocation unit value table in license unit tables 12, and determines the number of allocation units associated with this use of the licensed program 14 based on the identified allocation table and the processor. The license management facility 10 then adjusts the number of license units for the licensed program 14 in the license data base 11 to reflect the release.

It will be appreciated by those skilled in the art that, the license management facility 10 may, in response to a GRANT LICENSE request, instead of deducting allocation units from the entries in the license data base 11 associated with the licensed programs 14, determine the number of allocation units which would be in use if usage of the licensed program 14 is permitted, and respond based on that determination. If the license mangement facility 10 operates in that manner, it may be advantageous for the entries in license data base 11 relating to each licensed program 14 to maintain a running record of the number of allocation units associated with its usage. The licensing policy module 15 may operate similarly in connection with group licenses that are limited by usage.

It will also be appreciated that the new license management system thus permits the digital data processing system to control use of a licensed program 14 based on licensing criteria in the license data base 11, the license unit tables 12, the group licensing tables 17 and the licensor's general licensing policies rather than requiring an operator to limit or restrict use of a licensed program or charging for the license based on some function of the capacity of all of the processors in the digital data processing system. The new license management system allows for very flexible pricing of licenses and licensing policies, since the digital data processing system itself enforces the licensing terms controlling use of the licensed programs 14 in the system.

Fig. 2 depicts the detailed structure of the license data base 12 (Fig. 1) used in the license management system depicted in Fig. 1. With reference to Fig. 2, the license data base includes a plurality of entries generally identified by reference numberal 20, with each entry being associated with one licensed program 14. Each entry 20 includes a number of fields, including an issuer name field 21 identifying the issuer of the license, an authorization number field 22 which contains an authorization number, a producer name field 23 which identifies the name of the vendor of the licensed program, and a product name field 24 which contains the name of the licensed program. The contents of these fields may be used, for example, in connection with other license management operations, such as determining the source of licensed programs in the event of detection of errors in programs, and in locating duplicate entries in the license data base or entries which may be combined as a result of licenses being obtained and entered by, perhaps different operators or at different times.

Each entry 20 in the licensing data base 11 also includes a license number field 25 whose contents identify the number of licensing units remaining. A license of a licensed program 14 identifies a number of licensing units, which may be a function of the price paid for the license. An availability table field 26 and an activity table field 27 identify license usage allocation unit value tables in the license unit tables 12 (described in connection with Fig. 3) to be used in connection with the GRANT LICENSE and RELEASE LICENSE requests.

By way of background, a license may be in accordance with a licensing paradigm which requires concurrent use of the licensed program 14 on several processors to be a function of the processor power and capacity, and the availability table field 26 identifies a license usage allocation unit table to be used in connection with that. In an alternative, a license may be in accordance with a licensing paradigm which requires concurrent use of the licensed program to be a function of the number of users using the program, and the activity table field 27 identifies a license usage allocation unit valve table in the license unit tables 12 to be used in connection with that. If either licensing paradigm is used to the exclusion of the other, one field contains a non-zero value and the other field contains a zero value. In addition, a license may be in accordance with both licensing paradigms, that is, concurrent use of a program may be limited by both processor power and capacity and by the number of concurrent users, and in that case both fields 26 and 27 have non-zero values.

In one embodiment of the licensing management system, fields 21 through 27 of an entry 20 in the licensing data base 11 are required. In that embodiment, an entry 20 in the licensing data may

also have several optional fields. In particular, an entry 20 may include a date/version number field 30 whose contents comprise either a date or version number to identify the licensed program. If a license is to terminate on a specific date, the entry 20 may include a licensor termination date field 31 or a licensee termination date field 32 whose contents specify the termination date assigned by the licensor or licensee. This may be particularly useful, for example, as a mechanism for permitting licensees to demonstrate or try a program before committing to a long or open term license.

Finally, an entry 20 in the license data base includes a checksum field 33, which includes a checksum of the contents of the other fields 21 through 27 and 30 through 32 in the entry 20, which may be established by means of a mathematical algorithm applied to the contents of the various fields. The general mechanism for establishing checksums is well known in the art, and will not be described further herein. The contents of all fields 21 through 27 and 30 through 33 of a new entry 20 are entered by an operator. Prior to establishment of an entry in the license data base 11, the license management facility 10 may verify correct entry of the information in the various fields by calculating a checksum and comparing it to the checksum provided by the operator. If the checksum provided by the operator and the checksum determined by the license management facility are the same, the entry 20 is established in the license data base 11. On the other hand, if the checksum provided by the operator and the checksum determinec by the license management facility differ, the license management facility 10 determines that the information is erroneous or the license is invalid and does not establish the entry 20 in the license data base 11. It will be appreciated that, if the checksum-generation algorithm is hidden from an operator, the checksum provides a mechanism for verifying, not only that the information has been properly loaded into the entry, but also that the license upon which the entry is based is authorized by the licensor.

The structure of group license tables 17 may be similar to the structure of the license data base 11, with the addition that the entries for each license reflected in the group license tables 17 will need to identify all of the licensed programs covered thereby.

As described above, the licensing unit tables 12 (Fig. 1) contain information as to the allocation units for use in determining the number of licensing units associated with use of a licensed program. The structure of a licensing unit table 40 is depicted in Fig. 3. With reference to Fig. 3, the licensing unit table includes a plurality of entries 41(1) through 41(N) (generally identified by reference numeral 41) each identified by a particular type of processor. One entry 41 in the table 40 is provided for each type of processor which can be included in the digital data processing system which can use the licensed programs 14 which reference the license unit table 40. The processor associated with each entry is identified by a processor identification field 42. The successive fields in the entries 41 (which form the various columns in the table 40 depicted in Fig. 3) form license usage allocation unit value tables 43(1) through 43-(M) (generally identified by reference numeral 43). The contents of the availability table field 26 and the activity table field 27 identify a license usage allocation unit value table 43. If there are non-zero contents in both availability field 26 and activity field 27, the contents which identify be the same license usage allocation unit value table 43 or different license usage allocation unit value tables 43. As described above, the contents of the license usage allocation unit value table identify the number of licensing units associated with use of the licensed programs which identify the particular license usage allocation unit value table, for each of the identified processors.

The operation of the licensing management system is depicted in detail in Figs. 4A-1 through 4-B. Figs. 4A-1 through 4A-4 depict, in a number of steps the details of operation of the licensing management system in connection with the GRANT LICENSE request from a licensed program 14. Figs. 4B-1 and 4B-2 depict, in a number of steps, the details of operation in connection with the RELEASE LICENSE request from a licensed program 14. In the Figs., the particular steps performed by the licensing policy module 15, the license management facility 10 and the operating system 13 are indicated in the respective steps. Since the operations depicted in Figs. 4A-1 through 4B-2 are substantially as described above in connection with Fig. 1, they will not be described further herein.

The foregoing description has been limited to a specific embodiment of this invention. It will be apparent, however, that variations and modifications may be made to the invention, with the attainment of some or all of the advantages of the invention. Therefore, it is the object of the appended claims to cover all such variations and modifications as come within the true spirit and scope of the invention.

Claims

1. A license management system for managing usage of a licensed software program comprising:
licensing storage means for storing a licensing unit value identifying a number of licensing units asso-

ciated with the licensed software program;
usage allocation value storage means for storing a usage allocation value identifying a number of licensing units associated with a use of the licensed software program; and
licensing verification means responsive to a usage request to use said licensed software program for determining, based on the contents of said licensing storage means and said usage allocation value storage means, whether usage of said licensed software program is permitted and, if usage is permitted, for adjusting the contents of said licensing storage means by a value to the contents of said usage allocation value storage means.

2. A license management system as defined in claim 1 for use in a digital data processing system which generates a system date value, said licensing storage means includes a plurality of fields including a licensing unit storage field for storing said licensing unit number identifying value and a field identifying a termination date, said licensing verification means further determining whether usage of said licensed software program is permitted in response to a comparison of said system date and said termination date.

3. A license management system as defined in claim 1 for managing usage of plurality of licensed software programs, wherein said licensing storage means includes a plurality of entries each containing a program identification field identifying a licensed software program and a licensing unit storage field for storing said licensing unit value, said licensing verification means including:
request receiving means for receiving a usage request identifying a licensed software program;
licensing unit retrieval means responsive to said request receiving means receipt of a usage request for retrieving the contents of said licensing unit storage field from the entry of said licensing storage means whose program identification field identifies the licensed software program identified in said usage request; and
licensing unit processing means for determining, based on the contents of retrieved licensing unit storage field and said usage allocation value storage means, whether usage of said licensed software program is permitted and, is usage is permitted, for adjusting the contents of said licensing storage means by a value related to the contents of said usage allocation value storage means.

4. A license management system as defined in claim 3 for use in a digital data processing system which generates a system date value, each entry in said licensing storage means further including a termination date field identifying a termination date, said licensing unit processing means further determining whether usage of said licensed software program is permitted in response to a comparison of said system date and said termination date.

5. A license management system as defined in claim 3 wherein said usage allocation value storage means includes a plurality of usage allocation tables each storing a value identifying a number of licensing units, each entry in said licensing storage means further including a usage allocation table identification field identifying a usage allocation table, said licensing verification means further including usage allocation table retrieval means responsive to said request receiving means receipt of a usage request for retrieving the contents of the usage allocation table identified by the contents of said usage allocation table identification field of said retrieved entry, said licensing unit processing means using said retrieved usage allocation table in its determination.

6. A license management system as defined in claim 5 wherein a request message further includes licensing usage allocation value selection criteria and each usage allocation table includes a plurality of entries each identifying a usage allocation value associated with a licensing usage allocation value selection criterion, said licensing verification means including means for retrieving, from the usage allocation table identified by said entry in said licensing storage means, the usage allocation value associated with the licensing usage allocation value selection criterion in said request message and using said retrieved usage allocation value in its determination.

7. A license management system as defined in claim 3 wherein a request message further includes licensing usage allocation value selection criteria and said usage allocation table includes a plurality of entries each identifying a usage allocation value associated with a licensing usage allocation selection criterion, said licensing verification means including means for retrieving the usage allocation value associated with the licensing usage allocation selection criterion in said request message and using said retrieved usage allocation value in its determination.

8. A license management system as defined in claim 1 wherein said licensing verification means further operates in response to a release request message for adjusting the contents of said licensing storage means by a value related to the contents of said usage allocation value storage means.

9. A license management system as defined in claim 8 for managing usage of a plurality of licensed software programs, wherein said licensing storage means includes a plurality of entries each containing a program identification field identifying a licensed software program and a licensing unit storage field for storing said licensing unit value,

said licensing verification means including:
request receiving means for receiving a release request identifying a licensed software program;
licensing unit processing means for adjusting the contents of said licensing storage means by a value related to the contents of said usage allocation value storage means.

10. A license management system as defined in claim 9 wherein said usage allocation value storage means includes a plurality of usage allocation tables each storing a value identifying a number of licensing units, each entry in said licensing storage means further including a usage allocation table identification field identifying a usage allocation table, said licensing verification means further including usage allocation table retrieval means responsive to said request receiving means receipt of a usage request for retrieving the contents of said usage allocation table identification field of said retrieved entry, said licensing unit processing means using retrieved usage allocation table in its adjusting.

11. A license management system as defined in claim 10 wherein a release message further includes licensing usage allocation value selection criteria and each usage allocation table includes a plurality of entries each identifying a usage allocation value associated with a licensing usage allocation value selection criterion, said licensing verification means including means for retrieving, from the usage allocation table identified by said entry in said licensing storage means, the usage allocation value associated with the licensing usage allocation value selection criterion in said request message and using said retrieved usage allocation value in its adjusting.

12. A license management system as defined in claim 8 wherein a release message further includes licensing usage allocation value selection criteria and each usage allocation table includes a plurality of entries each identifying a usage allocation value associated with a licensing usage allocation value selection criterion, said licensing verification means including means for retrieving, from the usage allocation value table identified by said entry in said licensing storage means, the usage allocation value associated with the licensing usage allocation value selection criterion in said request message and using said retrieved usage allocation value in its adjusting.

13. A license management system for use in a digital data processing system including a system date generating means for generating a system date value, said license management system comprising:
licensing storage means including a plurality of entries each associated with a licensed software program, each entry containing a licensing units field for storing a licensing unit value identifying a number of licensing units associated with the license software program, a usage allocation table, and a termination date;
usage allocation table storage means for storing a plurality of usage allocation tables, each usage allocation table having a plurality of usage allocation entries each usage allocation entry being associated with a licensing usage allocation value selection criterion and storing a usage allocation value identifying a number of licensing units; and
licensing verification means including:
usage grant means including:
usage request message receiving means for receiving a usage request message from a licensed software program, said usage request message identifying said licensed software program and usage grant criteria;
entry retrieval means responsive to the receipt of a usage request message for retrieving from said licensing storage means the licensing table entry associated with said licensed software program;
usage allocation table retrieval means for retrieving from said usage allocation table storage means a usage allocation entry identified by said retrieved licensing table entry and the licensing usage allocation value selection criterion identified by the received usage request message;
licensing request processing means including:
usage determination means including licensing unit comparing means for comparing the contents of said licensing units field and said usage allocation units field and date comparison means for comparing the system date value with the contents of said termination date field to determine whether usage of said licensed software program is permitted.
response generation means for generating a message in response to the determination by said usage determination means; and
licensing unit adjusting means for adjusting the contents of said licensing units field in response to a positive determination by said usage determination means;
usage release means including:
usage release message receiving means for receiving a usage request message from a licensed software program; said usage request message identifying said licensed software program and usage grant criteria;
entry retrieval means responsive to the receipt of a usage request message for retrieving from said licensing storage means the licensing table entry associated with said licensed software program;
usage allocation table retrieval means for retrieving from said usage allocation table storage means a usage allocation entry identified by said retrieved licensing table entry and the licensing usage allocation value selection criterion identified by the

received usage request message;
licensing release processing means for adjusting the contents of said licensing units field in relation to the value of said usage allocation entry.

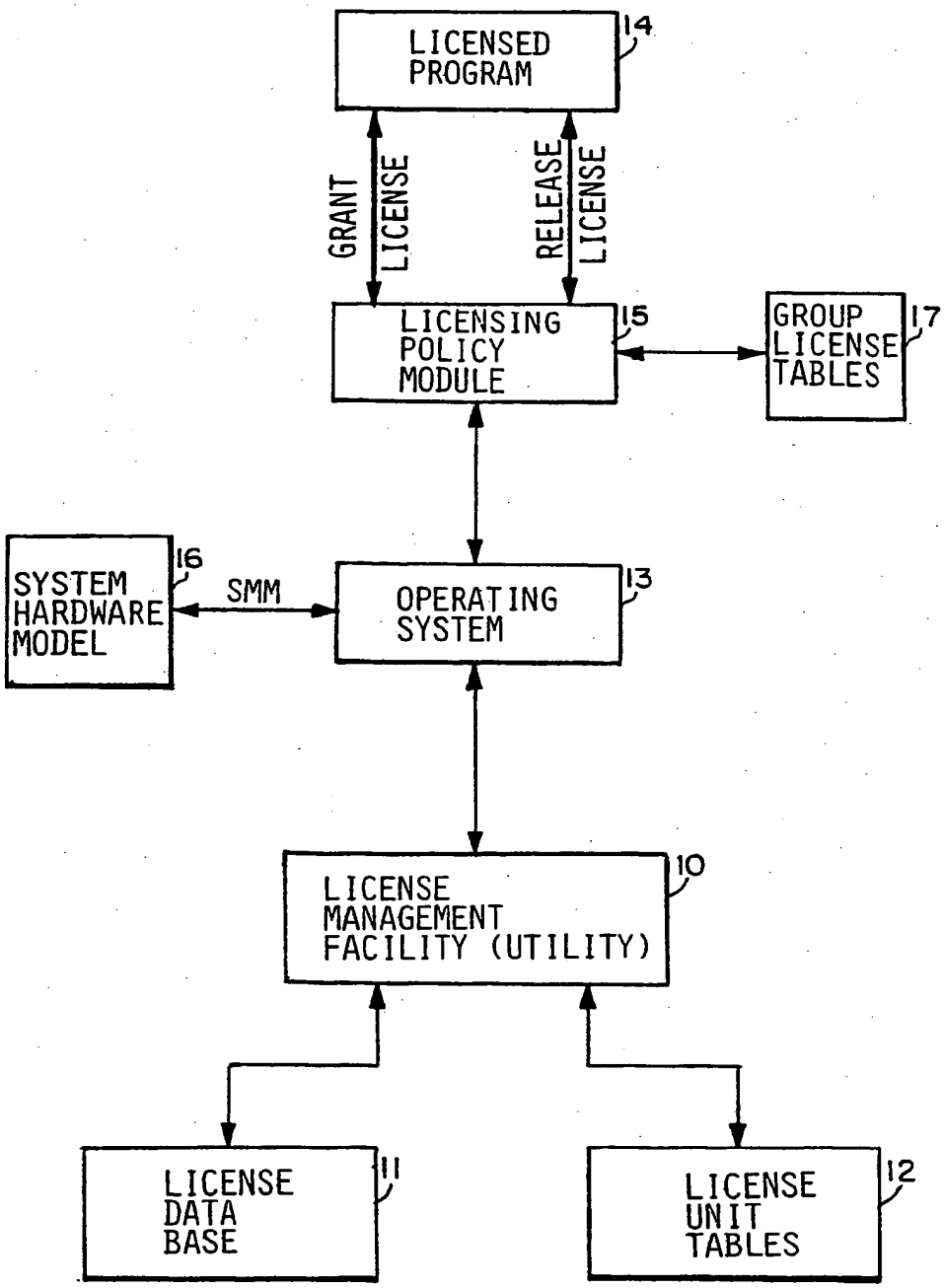


FIG. 1

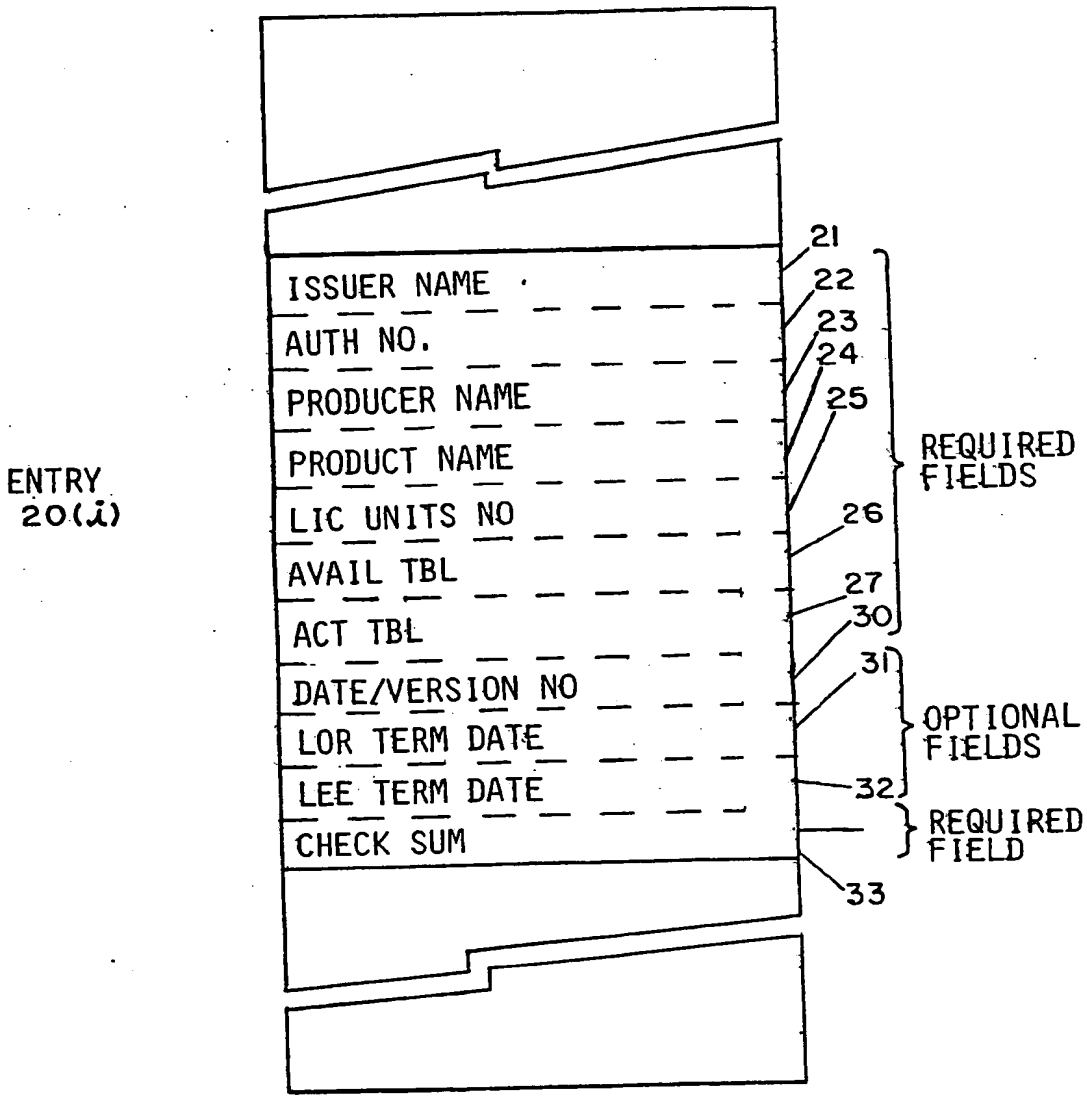
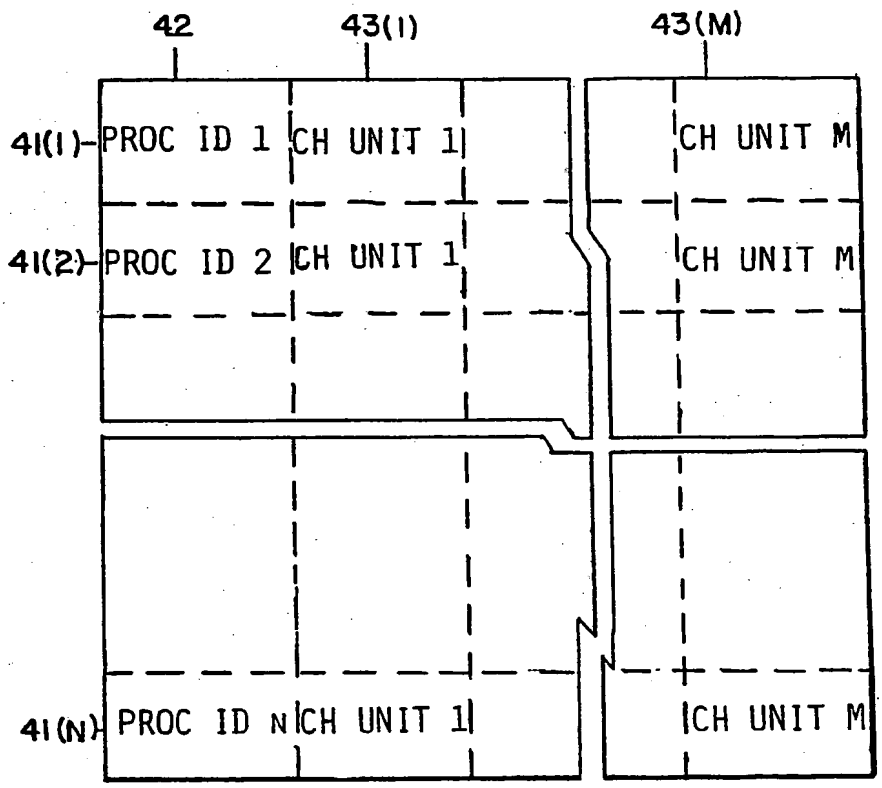


FIG. 2



LICENSE UNIT TABLE 40

FIG. 3

FIG. 4A-1

GRANT LICENSE

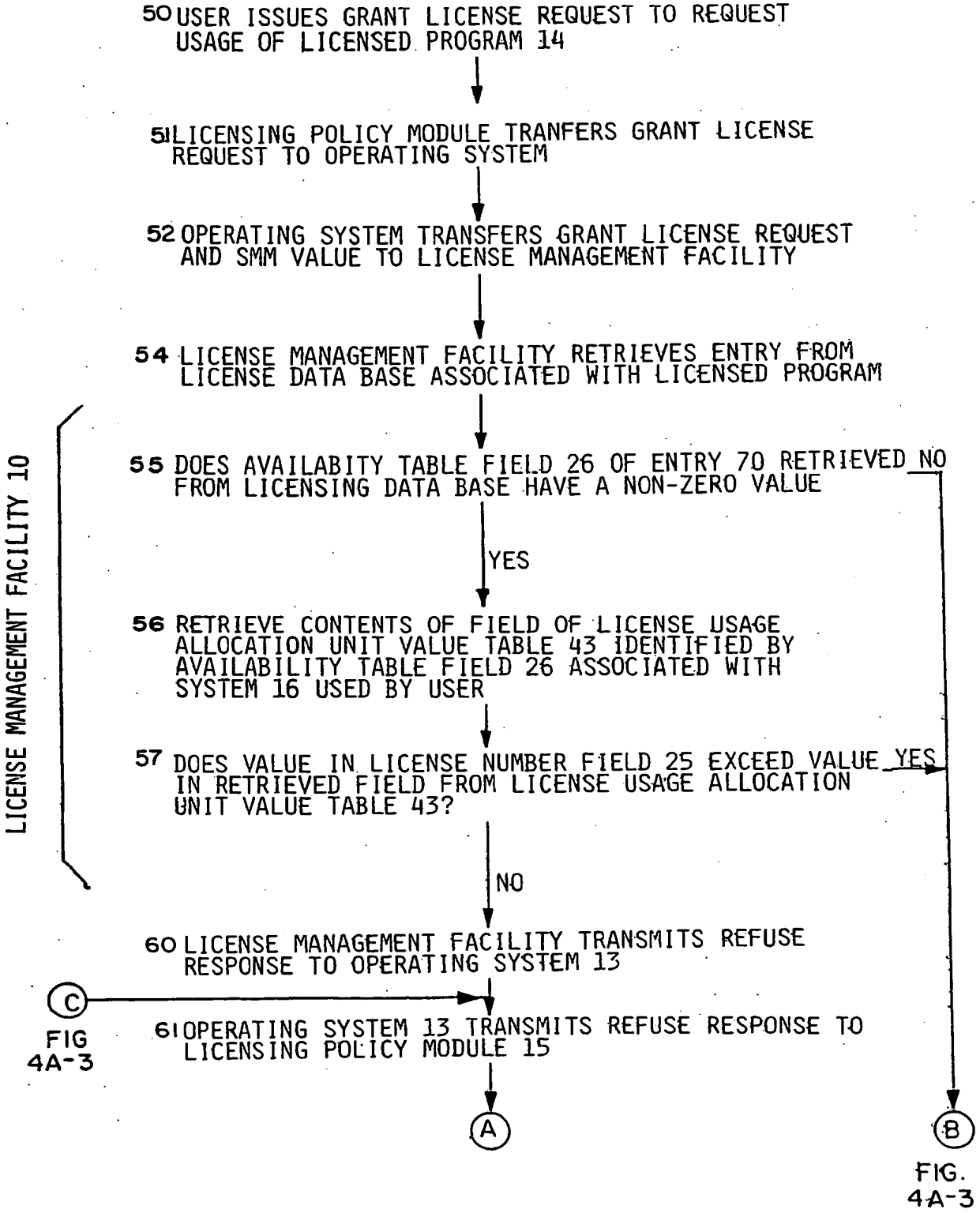
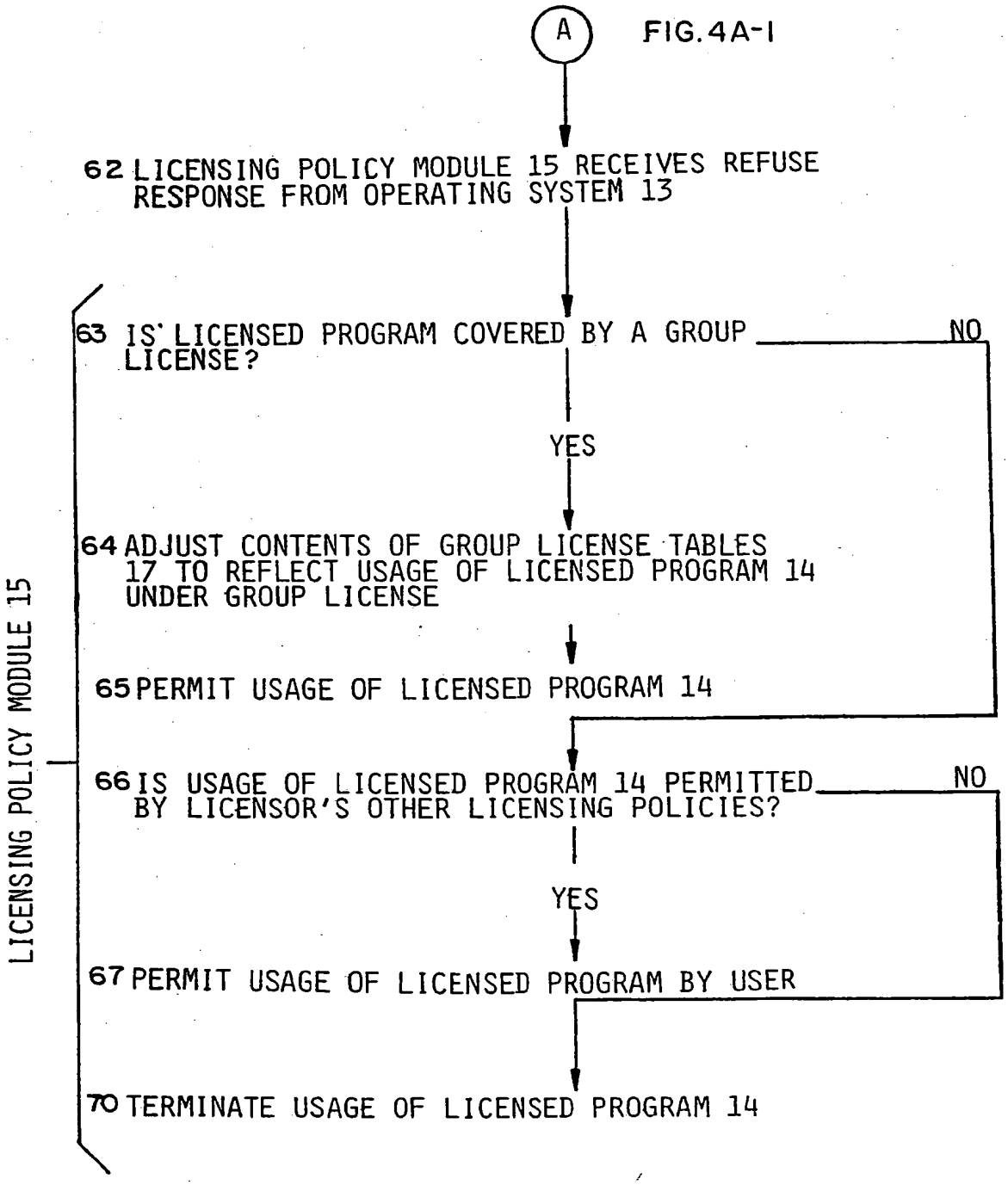


FIG. 4A-2



EP 25655 (5)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 April 2001 (05.04.2001)

PCT

(10) International Publication Number
WO 01/24530 A2

(51) International Patent Classification[7]: H04N 7/24

(21) International Application Number: PCT/US00/26832

(22) International Filing Date:
29 September 2000 (29.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/156,817 29 September 1999 (29.09.1999) US

(71) Applicant: LOUDEYE TECHNOLOGIES, INC. [US/US]; 414 Olive Way, Suite 300, Seattle, WA 98101 (US).

(72) Inventors: TOBIAS, Martin; 3601 East Union, Seattle, WA 98122 (US). KITE, Beverly; 420 N.W. 73rd, Seattle, WA 98122 (US). MATHEWS, Mat; 1118 E. John Street, Seattle, WA 98102 (US).

(74) Agents: BRANDT, Carl, L. et al.; Hickman Palermo Truong & Becker, 1600 Willow Street, San Jose, CA 95125 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, 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, 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), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *Without international search report and to be republished upon receipt of that report.*

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.

WO 01/24530 A2

(54) Title: STREAMING MEDIA ENCODING AGENT FOR TEMPORAL MODIFICATIONS

(57) Abstract: A method and apparatus for playing digital content at a client is disclosed. In one aspect, a plurality of versions of the digital content is generated. Each version of the plurality of versions is generated with the same amplitude but a different wavelength relative to the other plurality of versions. During playback of thedigital content at said client, a selected version of the plurality of versions is used for playing back the content. In response to user input received at said client, a change is made as to which of the plurality of versions to be used as the selected version.

STREAMING MEDIA ENCODING AGENT FOR TEMPORAL MODIFICATIONS

CLAIM OF PRIORITY

This patent application claims priority from, U.S. Provisional Patent Application No. 60/156,817, filed on September 29, 1999, entitled STREAMING MEDIA ENCODING AGENT FOR TEMPORAL MODIFICATIONS, the content of which is hereby incorporated by reference in its entirety.

FIELD OF THE INVENTION

The present invention relates to the encoding of streaming media files and more specifically to a mechanism that provides for the speeding up and slowing down of streaming media files.

BACKGROUND OF THE INVENTION

In recent years, the media industry has expanded its horizons beyond traditional analog technologies. Numerous systems have been developed for transmitting video information digitally from one location to be viewed in real time at another location.

As would be expected, the viewers of digital video desire the same functionality from the providers of digital video as they now enjoy while watching analog video tapes on video cassette recorders. For example, viewers want to be able to make the video jump ahead, jump back, fast forward, fast rewind, slow forward, slow rewind and freeze frame.

Conventionally, digital video delivered to a particular destination (the "client") is encoded to allow for playback at a specific rate. Thus, the user is typically required to play the entire video at one static rate as they generally have no mechanism for altering the playback speed of the video. In addition, with today's technology, even if a mechanism is provided for changing the playback speed of the digital video, the

amplitude of the video signal is altered such that the video sound will be undesirably distorted.

Based upon the foregoing, there is a clear need for an improved method for providing media content that allows for multiple playback speed control at a client.

SUMMARY OF THE INVENTION

According to one aspect of the invention, a method is provided for playing digital content at a client is disclosed. In one aspect, a plurality of versions of the digital content is generated. Each version of the plurality of versions is generated with the same amplitude but a different wavelength relative to the other plurality of versions. During playback of the digital content at said client, a selected version of the plurality of versions is used for playing back the content. In response to user input received at said client, a change is made as to which of the plurality of versions to be used as the selected version.

According to another feature, a method for incorporating temporal modifications in streaming media content is performed by generating one or more temporal media files by applying a temporal encoding process to media content. Streaming media data based on the one or more temporal media files is then generated. The streaming media data is delivered to a client and can be played at the client at multiple play rates.

The invention also encompasses a computer-readable medium, a computer data signal embodied in a carrier wave, and an apparatus configured to carry out the foregoing steps. Other features and aspects will become apparent from the following description and the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments are illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings in which like reference numerals refer to similar elements and in which:

FIG. 1A is a block diagram of a temporal media encoding system in which certain embodiments of the invention may be used;

FIG. 1B is another block diagram of a temporal media encoding system in which certain embodiments of the invention may be used;

FIG. 2 illustrates example of how applying the temporal encoding process allows streaming media data to be sent from the server side at one rate and played on the client side at a dynamically changing second rate;

FIG. 3 is a flow diagram that illustrates a method for incorporating temporal modifications in streaming media content in accordance with certain embodiments of the invention;

FIG. 4 is a block diagram of a computer system on which embodiments may be implemented; and

FIG. 5 is a block diagram that illustrates an example of another temporal media encoding system configuration in which certain embodiments of the invention may be used.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

A method and apparatus for incorporating temporal modifications in streaming media content is described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

TERMS AND TERMINOLOGY

Various terms are used herein to describe embodiments of the invention. In following description:

The term "live feed information" refers to information that may be received from analog or digital cameras, satellite or cable feeds or any other mechanism that is capable of providing live feed information.

The term "media content" includes one or more of a variety of different types of pre-recorded information, and/or live feed information.

The "wavelength" of a media content signal refers to the horizontal length of one cycle of the wave and can be calculated by measuring the distance between any two successive equivalent points on the wave.

The "amplitude" of a media content signal refers to the distance between a crest or a trough on the wave and its undisturbed position.

The "period" of a media content signal refers to the time required to complete one full cycle of motion.

The "frequency" of a wave is merely the reciprocal of the period, or the number of cycles completed in one second. Both the period and the frequency are dependent on the wavelength of the wave. If the media content signal has a long wavelength then its frequency would be lower than if had a short wavelength.

The "speed" of a media content signal refers to the rate of presentation of audio or video in the media file, based on changes in frequency of the wave for audio, or number of frames presented for video.

A "temporal" modification refers to a change in the frequency of a wave without a corresponding change in the amplitude of a wave. A "temporal media file" refers to a media file which, using temporal modification, the speed of the audio and video is changed through modification of the frequency of the waveform and frame rate of the video.

A "variable speed media file" refers to a media file that contains data from multiple temporal media files, where the data from the multiple temporal media files within the variable speed media file can be played selectively based on the user's preferences for presentation of the file.

FUNCTIONAL OVERVIEW

Techniques are disclosed for enabling a user to increase or decrease the speed in viewing a streaming media file during playback. In one embodiment, the techniques involve encoding streaming media files with wave frequency ("temporal") modifications. Specifically, a streaming media encoding mechanism is configured as a modular collection of methods or processes to manipulate the temporal nature of time-based media, and to encode streaming media data for Internet and broadband playback.

The process of manipulating the temporal nature of time-based media involves modifying the frequency of the wave by reducing the wavelength of the wave without modifying the amplitude of the wave. If necessary, the process also involves dropping corresponding frames in the media signal to synchronize audio and video.

In certain embodiments, the streaming media encoding mechanism employs a temporal encoding process that alters the frequency of the media content signal independent of the amplitude, and removes frames, to generate one or more temporal media files. In one embodiment, the one or more temporal media files are then bound together and encoded into streaming media packets to emulate variable speed streaming media data. Each packet includes a portion of each of the one or more temporal media files. Header information is attached to each of the packets describing the packets contents. Using a streaming media player, a user may dynamically adjust the playback speed of the streaming media data to emulate different playback rates, such as the "fast-forward" or "slow" functions of a media player.

SYSTEM OVERVIEW

FIG. 1A is a block diagram of a temporal media encoding system 100 in which certain embodiments of the invention may be used. Generally, temporal media encoding system 100 includes a server 102, a client 110, and one or more network systems 118.

As is depicted in FIG. 1A, server 102 and client 110 are connected through one or more network systems 118. These one or more network systems may include, but are not limited to, Local Area Networks (LAN), and Wide Area Networks (WAN), including the

Internet and/or other wireless communication mechanisms and/or communication mediums. Thus, embodiments of the invention are not limited to any particular type of communication mechanism, medium or protocol.

Server 102 is a combination of hardware and/or software components that are configured for encoding media content with temporal modifications to create one or more temporal media files. The one or more temporal media files are combined and encoded in streaming media format to produce variable speed streaming media data.

In this example, server 102 includes a phase encoding unit 104, a streaming encoding unit 106 and a delivery unit 108. The phase encoding unit 104 is configured for generating temporal media files by applying a temporal encoding process to media content.

In one embodiment, phase encoding unit 104 is used to manipulate the temporal nature of time-based content media to generate a set of temporal media files.

The streaming encoding unit 106 is configured to encode the one or more temporal media files into streaming media format ("streaming media data"). Streaming encoding unit 106 may represent a variety of different types of encoders that are capable of encoding data in a particular encoding format. For example, streaming encoding unit 106 may be configured as a Real, Windows Media, Liquid Audio, MPEG, A2B, Audiobase, MP3, Blade, Xing, QuickTime or any other similar type of encoder. In addition, streaming encoding unit 106 may be either an "off-the-self" encoding unit or a proprietary encoding unit, as embodiments of the invention are not limited to any particular type of encoder.

In one embodiment, streaming encoding unit 106 stores the streaming media data as one or more media files in a storage unit 120. Storage unit 120 may represent a non-volatile storage device that is part of server 102 or instead may be a separate storage unit that is accessible to server 102, for example over a network with delivery unit 108 to provide the streaming media data, thus potentially reducing the overhead of storing and retrieving the media data from storage unit 120.

Delivery unit 108 is configured for delivering streaming media data to client 110 over network 118. In one embodiment, in response to a user request from client 110, delivery unit 108 retrieves streaming media data from storage unit 120 and delivers the streaming media data to client 110 via network system 118. Alternatively, delivery unit 108 may communicate directly with streaming encoding unit 106 to obtain streaming media data for delivery to client 110.

Client 110 represents a device, such as a personal computer, workstation, or other like device that is capable of communicating with server 104. Client 110 may include a browser application, such as Microsoft Internet Explorer® or Netscape Navigator®, that can request, receive and display electronic documents over a network connect.

In one embodiment, client 110 includes a display unit 112 and a media player 122, such as a Real, or Windows Media player. In certain embodiments, media player 122 includes a phase decoder unit 114 and a streaming decoding unit 116. In one embodiment, phase decoder unit 114 is configured as a plug-in component that can be dynamically linked into media player 122. Streaming decoding unit 116 is configured to receive streaming media from server 102 and to communicate the streaming media data to the phase decoder unit 114. The streaming media data is then decoded by the phase decoder unit 114 and played on client 110.

Because the media data includes the temporal media information, a user may dynamically select the desired playback speed for playing the streaming media data on client 110. For example, in one embodiment the streaming media data (which includes the variable speed media file information) contains a number of temporal media file portions that each contain the same section of the media content for playback at a different speed (i.e., playback at the original speed, 2x the original, playback at 3x the original etc.) FIG. 2 illustrates an example of a section of a streaming media file 200 that includes multiple packets 202, 204, 206 which each contain a particular portion of the media content (214, 216, 218). As depicted, packets 202, 204, 206 each respectively include a header 208, 210, 212 that describes the information contained in media content portions 214, 216, 218. In one embodiment, the playback speed of the media content can

be dynamically changed by switching between the different media content portions 214, 216, 218 that are present within each packet. For example, by switching from content portion 214 of packet 202 to content portion 216 of packet 202, the playback speed of the media content can be increased from 1X (original speed) to 2X (twice the speed of normal).

In one embodiment, phase decoder unit 114 maintains indexes into the content portions 214, 216, 218 of each packet as it is played to allow for dynamically switching between the content portions 214, 216, 218 of a particular packet. In another embodiment, phase decoder unit 114 is configured to switch between the content portions 214, 216, 218 between the playing of two packets. For example, if phase decoder unit 114 receives a request to increase the playback speed (for example to 2X) while in the middle of processing the content portion 214 in packet 202, phase decoder unit 114 may wait for the processing of content portion 214 complete and initiate the playback speed change by selecting content portion 216 in packet 204.

In certain embodiments, the media player 122 includes a player control component, for example a plug-in component, that provides a set of selectable VCR-like controls on display unit 112. By interacting with the controls, a user can select different options for dynamically controlling the speed at which the streaming media data is played. For example, the player control component may cause two speed control buttons to be displayed on display unit 112; a speed up button and a slow down button. In response to selecting the speed up button, phase decoder 114 automatically switches to a different ("faster playing") temporal media file in the variable speed media file. In switching to the different ("faster playing") temporal media file, the media content begins to play at a faster speed on display unit 112.

In one embodiment, phase decoder 122 is configured to automatically switch between the different temporal media files based on a set of playback conditions. For example, if it is determined that a commercial is currently being played on display unit 112, phase decoder 122 may automatically switch to a different ("faster playing") temporal media file to "fast-forward" through the commercial. In addition, server 112

may include additional information in the streaming media content that signals phase decoder unit 114 to switch to a different temporal media file to either "fast-forward" or "slow-down" the playing of the media content.

OPERATIONAL OVERVIEW

As depicted, the temporal encoding process allows the playback speed of the encoded streaming media data to dynamically alter the audio and/or image frame rate independent of the streaming media data amplitude.

FIG. 3 is a flow diagram 300 that illustrates a method for incorporating temporal modifications in streaming media content in accordance with certain embodiments of the invention. For explanation purposes, the blocks of FIG. 3 are described in reference to the components of FIG. 1A and FIG. 2. However, embodiments of the methods disclosed herein are not limited to the example embodiments that are shown in FIG. 1A and FIG. 2.

At block 302, the temporal encoding process is applied to media content to generate one or more temporal media files. The media content may take a variety of forms, including but not limited to movies, music, and television shows. For example, the temporal encoding may be applied to a movie to generate one or more temporal media files based on the particular movie.

At block 304, the one or more temporal media files are encoded to generate streaming media data. In certain embodiments, the one or more temporal media files may be encoded in multiple media formats using a variety of different encoders. For example, the one or more temporal media files may be encoded to generate streaming media data in both Real and Windows Media format.

At block 306, the streaming media data is delivered to a client. For example, using delivery unit 108, the streaming media data may be transmitted from server 102 to client 110 over network 118.

At block 308, the streaming media data is received by client 110 and played at a rate that may be dynamically altered, either by the client automatically or in response to the user interacting with the client interface controls.

CREATING TEMPORAL MEDIA FILES

As previously indicated, a mechanism for incorporating temporal modifications in streaming media content is provided. The temporal encoding process can be applied to a raw media content waveform, before the streaming encoding process, thus allowing multiple output streaming media formats to be supported.

Various techniques may be used to generate temporal media files. For example, to generate one or more temporal media files, a "Phase Vocoding" algorithm, generally referred to as a phase vocoder process, may be used to alter the frequency of an audio waveform independent of the amplitude. This "stretching" and "squashing" of the audio signal can also be applied in conjunction with image frame rate modifications, for example the dropping or duplicating frames, to keep the audio and video information synchronized to change the playback speed of encoded streaming media video files.

In one embodiment, temporal encoding process uses a Fast Fourier Transform (FFT) to represent a signal as a set of sinusoids. These sinusoids can be manipulated independently to produce different results. In certain embodiments, the Discrete Time Fourier Transform (DTFT) is used to transform a function of the independent variable n (a function of time in this case) to a function of the independent variable omega (digital frequency). The Discrete Fourier Transform (DFT) is then the DTFT evaluated at a number of equally spaced digital frequency values from 0 to pi. The FFT is a tool used to evaluate numerous DFTs with fewer steps than are required by the defining equation.

DECODING VARIABLE SPEED STREAMING MEDIA

As previously indicated, each "variable speed" media file packet is wrapped with a header that describes the contents of the file. In one embodiment, the header of each file contains metadata that includes information about the temporal media data that is contained in each packet. In response to the user interacting with the speed control buttons at client 110, phase decoder 114 uses the metadata to switch between the different media content portions to dynamically change the playback speed of media content.

In one embodiment, server 102 is configured to switch between the temporal media files that are contained within a variable speed media file. For example FIG. 1B is a block diagram of a temporal media encoding system 150 in which certain embodiments of the invention may be used. FIG. 1B includes many of the same components as shown in FIG. 1A, and as such like components have been numbered alike. As depicted in FIG. 1B, delivery unit 152 includes a selection unit 154 that receives user playback requests from client 110. In one embodiment, delivery unit 152 maintains index information into each of the temporal media files that allows the delivery unit to dynamically switch between the temporal media files in a consistent manner. For example, in response to receiving a user playback request to increase the playback speed from 2X to 5X, selection unit 154 identifies a location in the 5X temporal media file that corresponds to the portion currently being played in the 2X temporal media file. Delivery unit 152 then begins streaming the media data from the identified location in the 5X temporal media file to client 116. In response to receiving the streaming media data, streaming decoding unit 116 continues to play the media data, generally unaware that the playback speed of the media data has changed.

Still, in certain other embodiments, media player 122 depicted in FIG 1A is itself configured to modify the frequency of the waveform of the streaming media content and to drop image frames as necessary to keep the images synchronized with the audio portion. For example, in one embodiment, media player 122 includes a phase encoding unit 104, possibly as a plug-in component, which dynamically generates multiple temporal media files and corresponding metadata based on conventional streaming media content. These files are then used by phase decoder unit 114 to dynamically provide variable speed media content for display on display unit 112.

HARDWARE OVERVIEW

Figure 4 is a block diagram that illustrates a computer system 400 upon which an embodiment of the invention may be implemented. Computer system 400 includes a bus 402 or other communication mechanism for communicating information, and a processor

404 coupled with bus 402 for processing information. Computer system 400 also includes a main memory 406, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 402 for storing information and instructions to be executed by processor 404. Main memory 406 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 404. Computer system 400 further includes a read only memory (ROM) 408 or other static storage device coupled to bus 402 for storing static information and instructions for processor 404. A storage device 410, such as a magnetic disk or optical disk, is provided and coupled to bus 402 for storing information and instructions.

Computer system 400 may be coupled via bus 402 to a display 412, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 414, including alphanumeric and other keys, is coupled to bus 402 for communicating information and command selections to processor 404. Another type of user input device is cursor control 416, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 404 and for controlling cursor movement on display 412. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

The invention is related to the use of computer system 400 for incorporating temporal modifications in streaming media. According to one embodiment of the invention, the insertion of temporal modifications in streaming media is provided by computer system 400 in response to processor 404 executing one or more sequences of one or more instructions contained in main memory 406. Such instructions may be read into main memory 406 from another computer-readable medium, such as storage device 410. Execution of the sequences of instructions contained in main memory 406 causes processor 404 to perform the process steps described herein. One or more processors in a multi-processing arrangement may also be employed to execute the sequences of instructions contained in main memory 406. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to

implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

The term "computer-readable medium" as used herein refers to any medium that participates in providing instructions to processor 404 for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 410. Volatile media includes dynamic memory, such as main memory 406. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 402. Transmission media can also take the form of acoustic or light waves, such as those generated during radio wave and infrared data communications.

Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

Various forms of computer readable media may be involved in carrying one or more sequences of one or more instructions to processor 404 for execution. For example, the instructions may initially be carried on a magnetic disk of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 400 can receive the data on the telephone line and use an infrared transmitter to convert the data to an infrared signal. An infrared detector coupled to bus 402 can receive the data carried in the infrared signal and place the data on bus 402. Bus 402 carries the data to main memory 406, from which processor 404 retrieves and executes the instructions. The instructions received by main memory 406 may optionally be stored on storage device 410 either before or after execution by processor 404.

Computer system 400 also includes a communication interface 418 coupled to bus 402. Communication interface 418 provides a two-way data communication coupling to a network link 420 that is connected to a local network 422. For example, communication interface 418 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 418 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 418 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

Network link 420 typically provides data communication through one or more networks to other data devices. For example, network link 420 may provide a connection through local network 422 to a host computer 424 or to data equipment operated by an Internet Service Provider (ISP) 426. ISP 426 in turn provides data communication services through the worldwide packet data communication network now commonly referred to as the "Internet" 428. Local network 422 and Internet 428 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 420 and through communication interface 418, which carry the digital data to and from computer system 400, are exemplary forms of carrier waves transporting the information.

Computer system 400 can send messages and receive data, including program code, through the network(s), network link 420 and communication interface 418. In the Internet example, a server 430 might transmit a requested code for an application program through Internet 428, ISP 426, local network 422 and communication interface 418. In accordance with the invention, one such downloaded application provides for the insertion of temporal modifications in streaming media as described herein.

The received code may be executed by processor 404 as it is received, and/or stored in storage device 410, or other non-volatile storage for later execution. In this manner, computer system 400 may obtain application code in the form of a carrier wave.

ALTERNATIVES, EXTENSIONS

The mechanism described herein provides several advantages over prior approaches for providing streaming media content. In particular, the described techniques provide an improved method for delivering streaming media data that allows the playback speed of the media data to be dynamically changed as it is played at a client without affecting the amplitude of the data. By allowing the speed of the media data to be dynamically altered, adjustments to the playback rate may be made based on the current bandwidth that is available between a server and a client. Thus, if a user is having trouble understanding the voice-over at a particular video speed, they can slow down the playing of the media in an attempt to better understand the content. In addition, if a user wishes to speed through a speech yet still understand the contents of the speech, they can increase the playing speed of the media content.

In describing certain embodiments of the invention, several drawing figures have been used for explanation purposes. However, the invention is not limited to any particular context as shown in drawing figures, and the spirit and scope of the invention include other contexts and applications. For example, although embodiments of the invention have illustrated a server delivering streaming media data to a single client, in certain embodiments, as depicted in FIG. 5, a server 502 may be configured with a plurality of phase encoding units 510,512, streaming encoding unit 516, 518 and delivery units 520, 522, 524 and which may be configured to communicate streaming media data to a plurality of clients 504, 506, 508. Additional configurations for encoding media data are described in co-pending U.S. Patent Application No. 09/499,961, filed on February 8, 2000, entitled DISTRIBUTED PRODUCTION SYSTEM FOR DIGITALLY ENCODING INFORMATION, the content of which is hereby incorporated by reference in its entirety. Thus, the specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

In addition, in this disclosure, including in the claims, certain process steps are set forth in a particular order, and alphabetic and alphanumeric labels are used to identify

certain steps. Unless specifically stated in the disclosure, embodiments of the invention are not limited to any particular order of carrying out such steps. In particular, the labels are used merely for convenient identification of steps, and are not intended to imply, specify or require a particular order of carrying out such steps.

CLAIMS

What is claimed is:

1. A method for incorporating temporal modifications in streaming media content, the method comprising the computer-implemented steps of:
generating one or more temporal media files by applying a temporal encoding process to media content;
generate streaming media data based on the one or more temporal media files; and
delivering the streaming media data to a client, wherein the streaming media data can be played at the client at multiple play rates.

2. The method as recited in Claim 1, further comprising the steps of:
combining the one or more temporal media files to generate a variable speed media file;
generating media content packets based on the variable speed media file, wherein each packet includes media content portions for playing the media content at multiple playback speeds; and
wherein the step of delivering the streaming media data comprises the step of delivering media content packets to said client to provide for variable speed playback rates of the media content.

3. The method as recited in Claim 1, wherein the step of generating one or more temporal media files includes the step of applying a phase vocoder process to the media content to generate the one or more temporal media files.

4. The method as recited in Claim 1, further comprising the steps of:
playing the streaming media data at a first playback speed at the client;
receiving user input at the client that requests that the streaming media data be played at a second playback speed at the client; and
in response to receiving the user input at the client, playing the streaming media data at said second playback speed at said client.

5. A computer-readable medium carrying one or more sequences of instructions for incorporating temporal modifications in streaming media content, wherein

execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the steps of:

generating one or more temporal media files by applying a temporal encoding process to media content;

encoding the one or more temporal media files to generate streaming media data; and

delivering the streaming media data to a client, wherein the streaming media data can be played at the client at multiple play rates.

6. The computer-readable medium as recited in Claim 5, further comprising instructions for performing the steps of:

combining the one or more temporal media files to generate a variable speed media file;

generating media content packets based on the variable speed media file, wherein each packet includes media content portions for playing the media content at multiple playback speeds; and

wherein the step of delivering the streaming media data comprises the step of delivering media content packets to said client to provide for variable speed playback rates of the media content.

7. The computer-readable medium as recited in Claim 5, wherein the step of generating one or more temporal media files includes the step of applying a phase vocoder process to the media content to generate the one or more temporal media files.

8. The computer-readable medium as recited in Claim 5, further comprising instructions for performing the steps of:

playing the streaming media data at a first playback speed at the client;

receiving user input at the client that requests that the streaming media data be played at a second playback speed at the client; and

in response to receiving the user input at the client, playing the streaming media data at said second playback speed at said client.

9. A server apparatus configured for incorporating temporal modifications in streaming media content, comprising:

means for generating one or more temporal media files by applying a temporal encoding process to media content;

means for encoding the one or more temporal media files to generate streaming media data; and

means for delivering the streaming media data to a client, wherein the streaming media data can be played at the client at multiple play rates.

10. The server apparatus as recited in Claim 9, further comprising:

means for combining the one or more temporal media files to generate a variable speed media file;

means for generating media content packets based on the variable speed media file, wherein each packet includes media content portions for playing the media content at multiple playback speeds; and

wherein the means for delivering the streaming media data comprises means for delivering media content packets to said client to provide for variable speed playback rates of the media content.

11. The server apparatus as recited in Claim 9, wherein the means for generating the one or more temporal media files includes means for applying a phase vocoder process to the media content to generate the one or more temporal media files.

12. A method playing digital content at a client, the method comprising the computer-implemented steps of:

generating a plurality of versions of said digital content, wherein each version of said plurality of versions has a same amplitude and a different wavelength relative to the other versions of said plurality of versions; and

during playback of said digital content at said client, performing the steps of

using a selected version of said plurality of versions for the playback of said content; and

changing which version of said plurality of versions to use as said selected version based on user input received at said client.

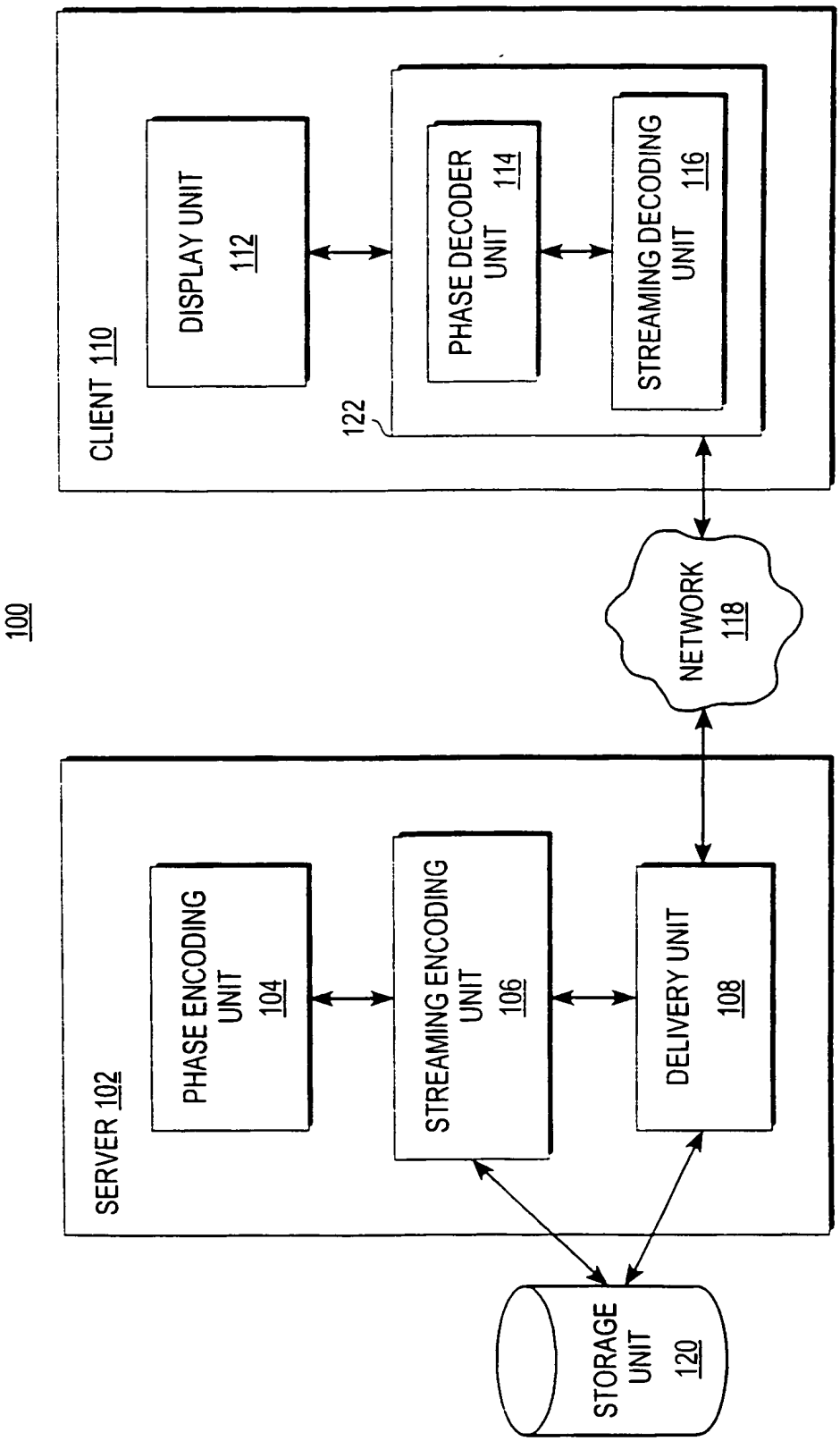


Fig. 1A

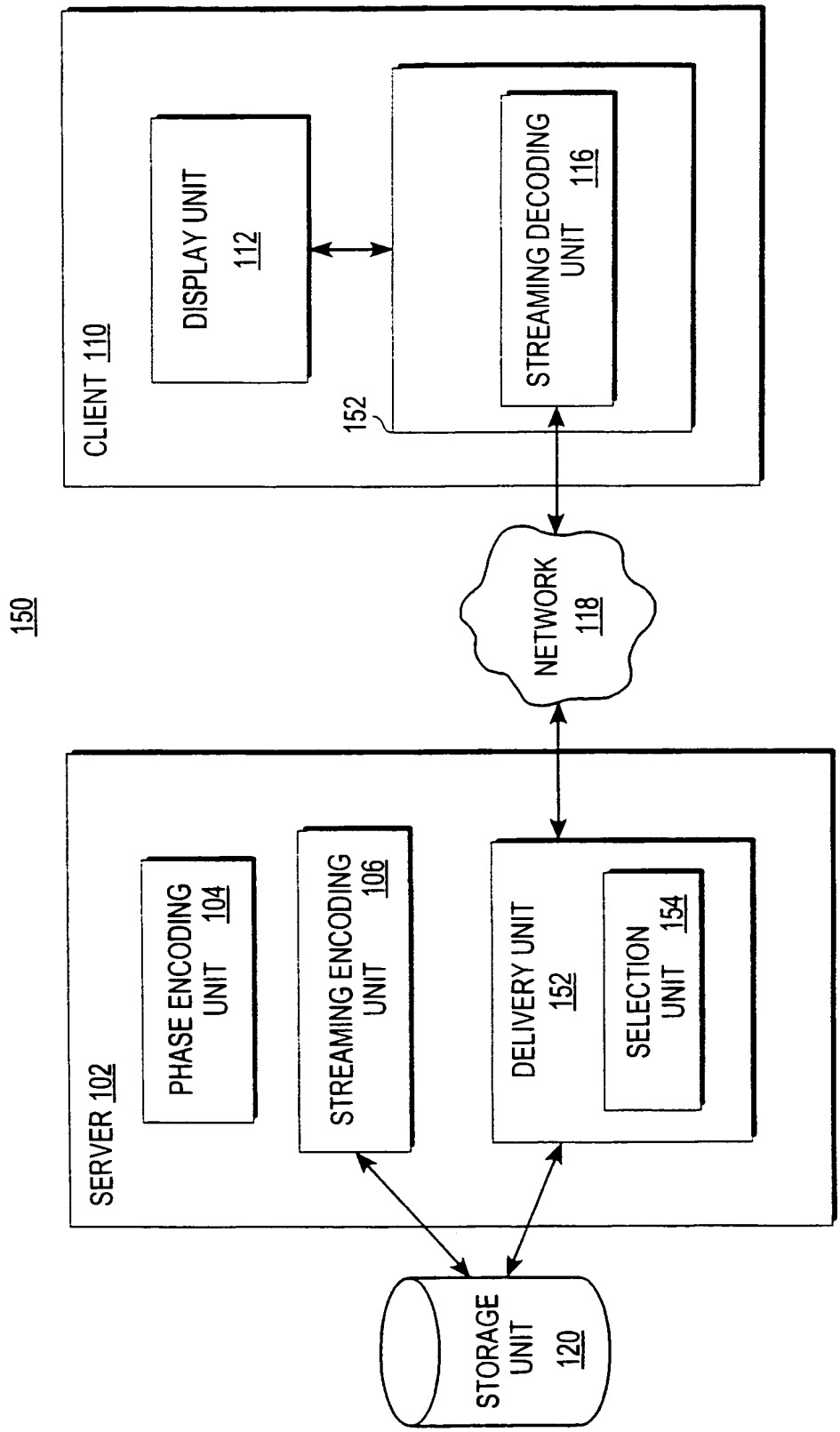


Fig. 1B

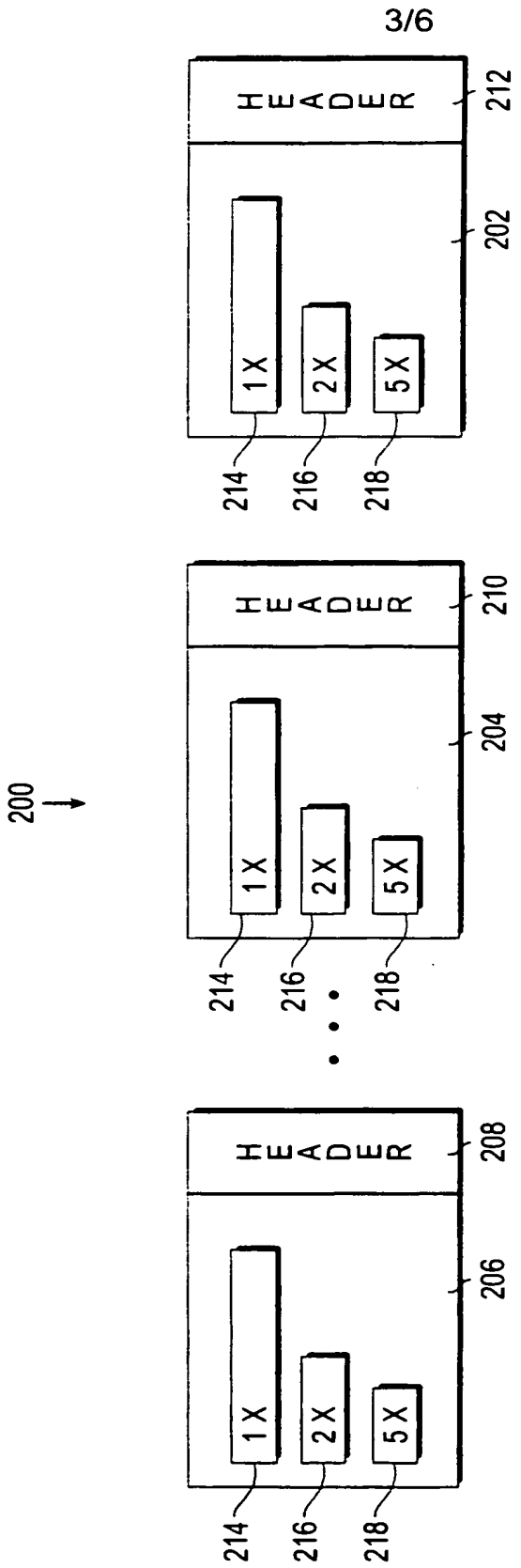


Fig. 2

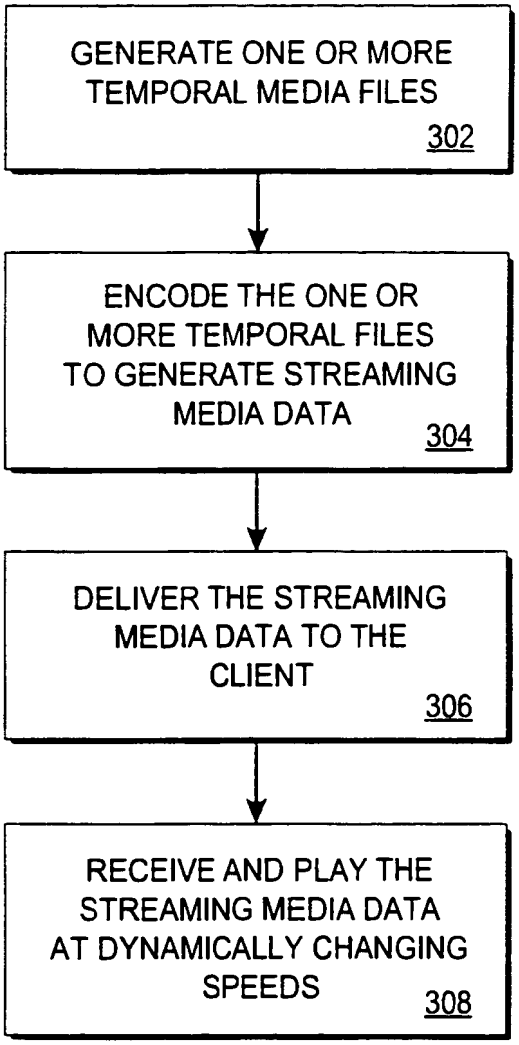


Fig. 3

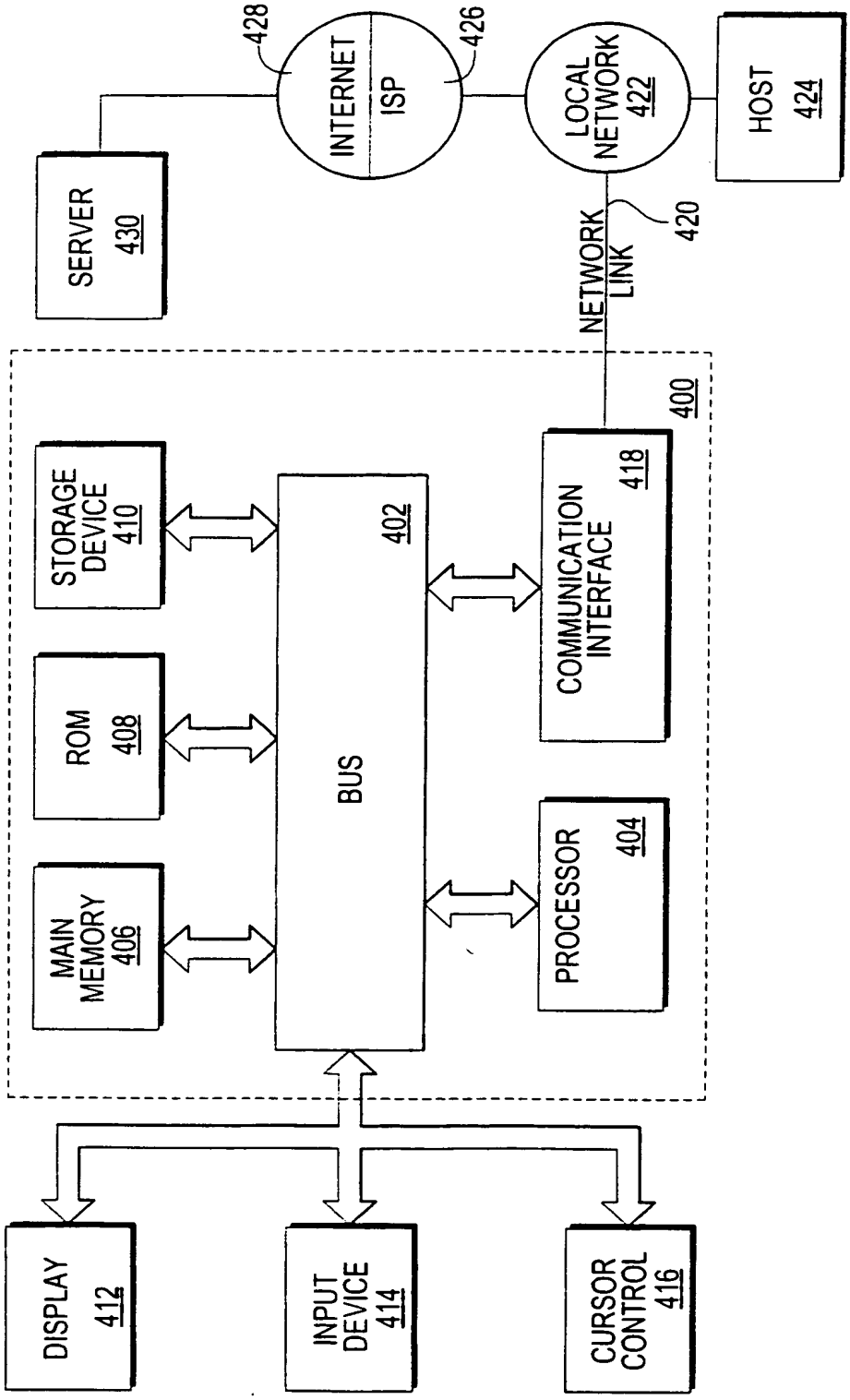


Fig. 4

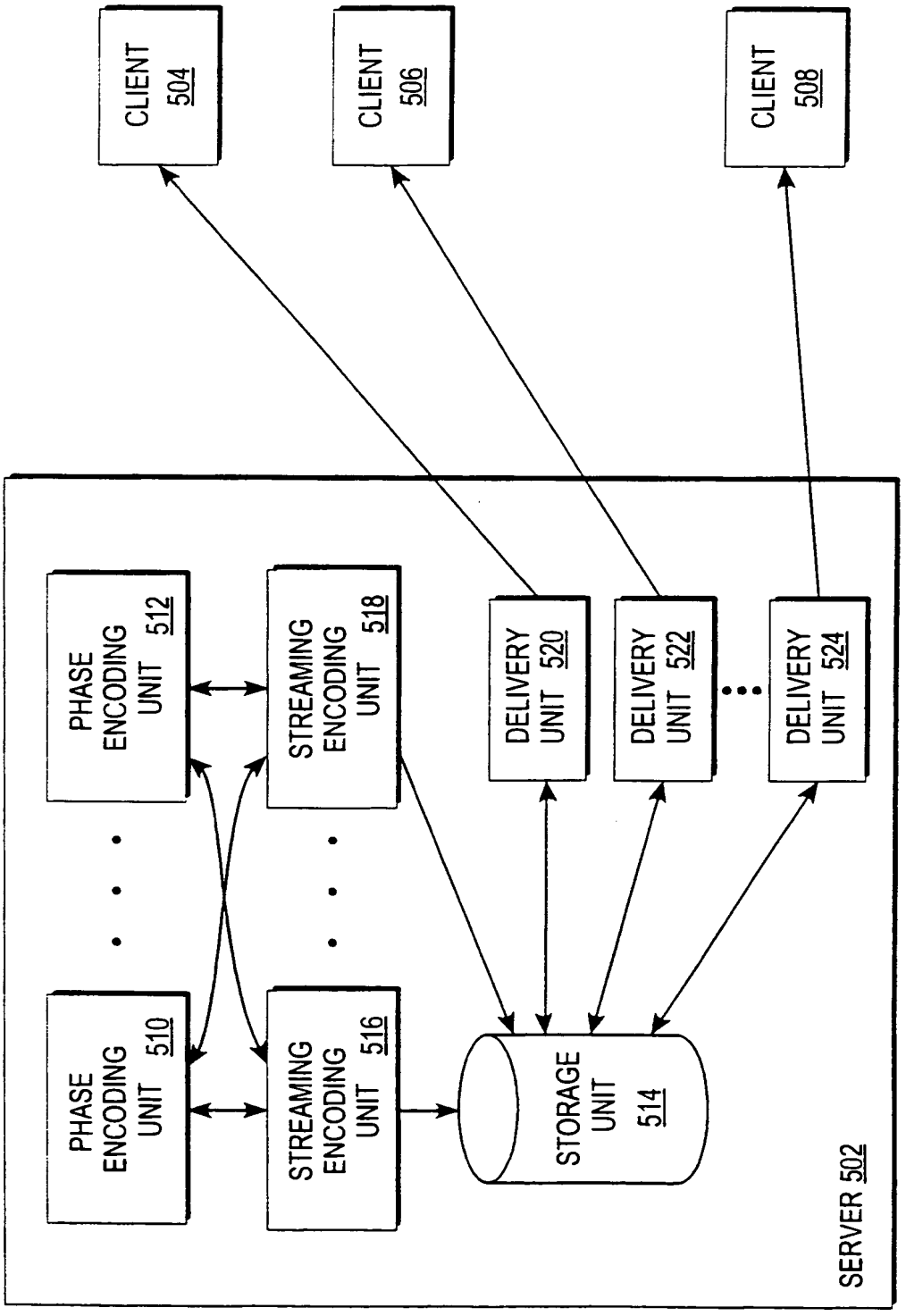


Fig. 5

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 April 2001 (05.04.2001)

PCT

(10) International Publication Number
WO 01/24530 A3

(51) International Patent Classification[7]: H04L 29/06

(21) International Application Number: PCT/US00/26832

(22) International Filing Date:
29 September 2000 (29.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/156,817 29 September 1999 (29.09.1999) US

(71) Applicant: LOUDEYE TECHNOLOGIES, INC. [US/US]; 414 Olive Way, Suite 300, Seattle, WA 98101 (US).

(72) Inventors: TOBIAS, Martin; 3601 East Union, Seattle, WA 98122 (US). KITE, Beverly; 420 N.W. 73rd, Seattle, WA 98122 (US). MATHEWS, Mat; 1118 E. John Street, Seattle, WA 98102 (US).

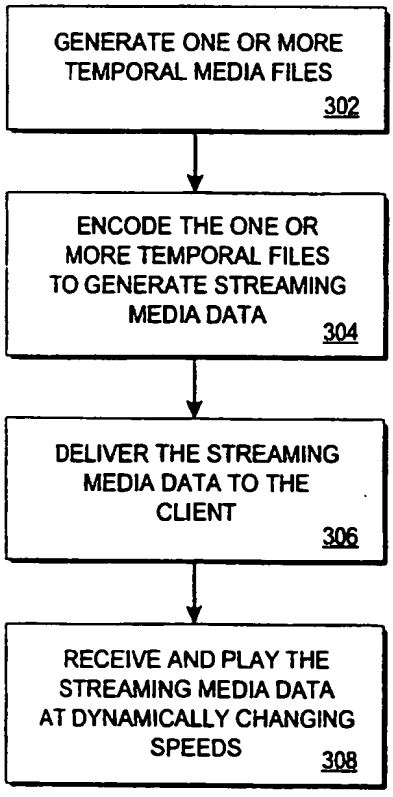
(74) Agents: BRANDT, Carl, L. et al.; Hickman Palermo Truong & Becker, 1600 Willow Street, San Jose, CA 95125 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, 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, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian

[Continued on next page]

(54) Title: STREAMING MEDIA ENCODING AGENT FOR TEMPORAL MODIFICATIONS



(57) Abstract: A method and apparatus for playing digital content at a client is disclosed. In one aspect, a plurality of versions of the digital content is generated. Each version of the plurality of versions is generated with the same amplitude but a different wavelength relative to the other plurality of versions. During playback of thedigital content at said client, a selected version of the plurality of versions is used for playing back the content. In response to user input received at said client, a change is made as to which of the plurality of versions to be used as the selected version.

WO 01/24530 A3

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*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

(88) Date of publication of the international search report:
20 December 2001

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.

INTERNATIONAL SEARCH REPORT

Int. ional Application No
PCT/US 00/26832

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06

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 7 H04L

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)
EPO-Internal, WPI Data, PAJ, INSPEC, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	DEY, J.K; SEN, S.; KUROSE, J.F.; TOWSLEY, D.; SALEHI, J.D.: "Playback restart in interactive streaming video applications" MULTIMEDIA COMPUTING AND SYSTEMS '97, 'Online! 3 - 6 June 1997, pages 458-465, XP002179183 Massachusetts Univ., Amherst, MA, USA ISBN: 0-8186-7819-4 Retrieved from the Internet: <URL:http.//ieeexplore.ieee.org> 'retrieved on 2001-10-03! abstract page 458, left-hand column, line 1 -right-hand column, line 8 page 459, left-hand column, line 7 -right-hand column, line 43 --- -/--	1-12

[X] Further documents are listed in the continuation of box C. [ ] 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

Date of the actual completion of the international search	Date of mailing of the international search report
3 October 2001	16/10/2001
Name and mailing address of the ISA 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	Authorized officer Adkhis, F

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inte onal Application No
PCT/US 00/26832

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	SUMARI, P.; MERABTI, M.; PEREIRA, R.: "Video-on-demand server: strategies for imrpoving performance" SOFTWARE, IEE PROCEEDINGS, 'Online! 9 - 10 July 1998, pages 33-37, XP002179184 Liverpoll John Moores Univ., UK ISSN: 1462-5970 Retrieved from the Internet: <URL:http://ieeexplore.ieee.org> 'retrieved on 2001-10-03! abstract page 33, left-hand column, line 1 -right-hand column, line 9 page 34, right-hand column, line 18 -page 35, right-hand column, line 44 -----	1-12

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

SP 25655

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 December 2000 (07.12.2000)

PCT

(10) International Publication Number
WO 00/73922 A2

(51) International Patent Classification[7]: G06F 17/00

(21) International Application Number: PCT/US00/11078

(22) International Filing Date: 25 April 2000 (25.04.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/323,635 1 June 1999 (01.06.1999) US

(71) Applicant: ENTERA, INC. [US/US]; 40971 Encyclopedia Circle, Fremont, CA 94538 (US).

(72) Inventor: SCHARBER, John, M.; 1616 Placer Circle, Livermore, CA 94550 (US).

(74) Agents: FAHMI, Tarek, N. et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, 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, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, 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), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *Without international search report and to be republished upon receipt of that report.*

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.

WO 00/73922 A2

(54) Title: CONTENT DELIVERY SYSTEM

(57) Abstract: Disclosed is a network content delivery system configured to: select a first content routing technique for processing a first set of network content; and select a second content routing technique for processing a second set of network content, wherein the first and second content routing techniques are selected based on one or more content routing variables. Also disclosed is a content delivery system comprising: a network node for storing network content; a first transmission medium communicatively coupled to the network node for transmitting a first set of network content to the network node; and a second transmission medium communicatively coupled to the network node for transmitting a second set of network content to the network node, wherein the first and second sets of network content are selected based on one or more routing variables.

CONTENT DELIVERY SYSTEM

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates to the transmission and storage of digital information across a network. More particularly, the invention relates to an improved system and method for caching and/or delivering various types of digital content using a plurality of network protocols.

Description of the Related Art

The World Wide Web (hereinafter "the Web") is a network paradigm which links documents known as "Web pages" locally or remotely across multiple network nodes (i.e., Web servers). A single Web page may have links (a.k.a., "hyperlinks") which point to numerous other Web pages. When a user points and clicks on a link using a cursor control device such as a mouse, the user can jump from the initial page to another page, regardless of where the Web pages are actually located. For example, the initial Web page might be stored on a Web server in New York and the second page (accessed via the hyperlink in the first page) might be stored on a Web server in California.

The underlying principles of the Web were developed 1989 at the European Center for Nuclear Research (CERN) in Geneva. By 1994 there were approximately 500 Web servers on the Internet. Today there are more than a million, with new sites starting up at an extraordinary rate. In sum, the Web has become the center of Internet activity and is the primary reason for the explosive growth of the Internet over the past several years.

In addition to providing a simple point-and-click interface to vast amounts of information on the Internet, the Web is quickly turning into a content delivery system. Well known Internet browsers such as Netscape Navigator™ and Microsoft Internet Explorer™ frequently provide plug-in software which allow additional features to be incorporated into the browser program. These include, for example, support for audio and video streaming, telephony, and videoconferencing.

The unparalleled increase in Web usage combined with the incorporation of high bandwidth applications (i.e., audio and video) into browser programs has created serious

performance/bandwidth problems for most Internet Service Providers (hereinafter "ISPs"). Moreover, the network traffic resulting from non-Web-based Internet services such as Internet News (commonly known as "Usenet" News) has increased on the same scale as the increase in Web traffic, thereby further adding to the bandwidth problems experienced by most ISPs.

These issues will be described in more detail with respect to **Figure 1** which illustrates an ISP 100 with a link 160 to a larger network 150 (e.g., the Internet) through which a plurality of clients 130, 120 can access a plurality of Web servers 140-144 and/or News servers 146-148. Maintaining a link 160 to the Internet 150 with enough bandwidth to handle the continually increasing traffic requirements of its clients 120, 130 represents a significant cost for ISP. At the same time, ISP 110 must absorb this cost in order to provide an adequate user experience for its clients 120, 130.

One system which is currently implemented to reduce network traffic across link 160 is a proxy server 210 with a Web cache 220, illustrated in **Figure 2**. When client 120 initially clicks on a hyperlink and requests a Web page (shown as address "www.isp.com/page.html") stored on Web server 144, client 120 will use proxy server 210 as a "proxy agent." This means that proxy server 210 will make the request for the Web page on behalf of client 120 as shown. Once the page has been retrieved and forwarded to client 120, proxy server will store a copy of the Web page locally in Web cache 220. Thus, when client 120 or another client – e.g., client 130 – makes a subsequent request for the same Web page, proxy server 210 will immediately transfer the Web page from its Web cache 210 to client 130. As a result, the speed with which client 130 receives the requested page is substantially increased, and at the same time, no additional bandwidth is consumed across Internet link 160.

While the foregoing proxy server configuration alleviates some of the network traffic across Internet link 160, several problems remain. One problem is that prior Web cache configurations do not have sufficient intelligence to deal with certain types of Web pages (or other Web-based information). For example, numerous Web pages and associated content can only be viewed by a client who pays a subscriber fee. As such, only those clients which provide proper authentication should be permitted to download the information. Today, proxy servers such as proxy server 210 will simply not cache a Web document which requires authentication.

In addition, Web caches do not address the increasing bandwidth problem associated with non-Web based Internet information. In particular, little has been done to alleviate the increasing bandwidth problems created by Usenet news streams. In fact, ISPs today must set aside a substantial amount of bandwidth to provide a continual Usenet news feed to its clients. Moreover, no mechanism is currently available for caching other data transmissions such as the streaming of digital audio and video. The term "streaming" implies a one-way transmission from a server to a client which provides for uninterrupted sound or video. When receiving a streaming transmission, the client will buffer a few seconds of audio or video information before it starts sending the information to a pair of speakers and/or a monitor, thus compensating for momentary delays in packet delivery across the network.

Accordingly, what is needed is a content delivery system which will reduce the bandwidth requirements for ISP 110 while still providing clients 120, 130 with an adequate user experience. What is also needed is a system which will work seamlessly with different types of Web-based and non-Web-based information and which can be implemented on currently available hardware and software platforms. What is also needed is an intelligent content delivery system which is capable of caching all types of Web-based information, including information which requires the authentication of a client before it can be accessed. What is also needed is a content delivery system which is easily adaptable so that it can be easily reconfigured to handle the caching of new Internet information and protocols. Finally, what is needed is a data replication system which runs on a distributed database engine, thereby incorporating well known distributed database procedures for maintaining cache coherency.

SUMMARY OF THE INVENTION

Disclosed is a network content delivery system configured to: select a first content routing technique for processing a first set of network content; and select a second content routing technique for processing a second set of network content, wherein the first and second content routing techniques are selected based on one or more content routing variables.

Also disclosed is a content delivery system comprising: a network node for storing network content; a first transmission medium communicatively coupled to the network node for transmitting a first set of network content to the network node; and a second transmission medium communicatively coupled to the network node for transmitting a second set of

network content to the network node, wherein the first and second sets of network content are selected based on one or more routing variables.

BRIEF DESCRIPTION OF THE DRAWINGS

A better understanding of the present invention can be obtained from the following detailed description in conjunction with the following drawings, in which:

FIG. 1 illustrates generally a network over which an ISP and a plurality of servers communicate.

FIG. 2 illustrates an ISP implementing a proxy server Web cache.

FIG. 3 illustrates one embodiment of the underlying architecture of an Internet content delivery system node.

FIG. 4 illustrates a plurality of Internet content delivery system nodes communicating across a network.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

One embodiment of the present system is a computer comprising a processor and a memory with which software implementing the functionality of the internet content delivery system described herein is executed. Such a computer system stores and communicates (internally or with other computer systems over a network) code and data using machine readable media, such as magnetic disks, random access memory, read only memory, carrier waves, signals, etc. In addition, while one embodiment is described in which the parts of the present invention are implemented in software, alternative embodiments can implement one or more of these parts using any combination of software, firmware and/or hardware.

The underlying architecture of one embodiment of the present internet content delivery system (hereinafter "ICDS") is illustrated in **Figure 3**. A single ICDS node 300 is shown including a cache 330, an ICDS application programming interface (hereinafter "API") 360 which includes a distributed database engine 361, and a plurality of software modules 310-326

which interface with the ICDS API 360. ICDS node 300 may communicate over a network 340 (e.g., the Internet) over communication link 370 and may also interface with a plurality of clients 350-351 and/or other ICDS nodes (e.g., through link 380).

As is known in the art, an API such as ICDS API 360 is comprised of a plurality of subroutines which can be invoked by application software (i.e., software written to operate in conjunction with the particular API). Thus, in **Figure 3** each of the plurality of software modules 310-326 may be uniquely tailored to meet the specific needs of a particular ISP. The modules interface with API 360 by making calls to the API's set of predefined subroutines. Another significant feature of ICDS API 360 is that it is platform-independent. Accordingly, it can be implemented on numerous hardware platforms including those that are Intel-based, Macintosh-based and Sun Microsystems-based.

In one embodiment, a portion of API 360's subroutines and a set of prefabricated modules can be marketed together as a Software Development Kit (hereinafter "SDK"). This will allow ISPs, corporations and/or end-users to customize the type of internet content delivery/caching which they require. In addition, because modules 310-326 may be dynamically linked, they may be loaded and unloaded without having to reboot the hardware platform on which cache 330 is executed.

I. Distributed Content Processing

As illustrated, ICDS node 300 includes a plurality of network protocol modules 310-319 which interface with API 360. These modules provide caching support on ICDS node 300 for numerous different Internet protocols including, but not limited to, Web protocols such as the Hypertext Transfer Protocol (hereinafter "HTTP") 310, Usenet news protocols such as the Network News Transport Protocol (hereinafter "NNRP") 312, directory access protocols such as the Lightweight Directory Access Protocol (hereinafter "LDAP") 314, data streaming protocols such as the Real Time Streaming Protocol (hereinafter "RTSP") 316, and protocols used to perform Wide Area Load Balancing (hereinafter "WALB") 318. Because the underlying architecture of the present ICDS system includes an open API, new protocol modules (e.g., module 319) can be seamlessly added to the system as needed.

One embodiment of the ICDS system includes a plurality of standardized service definitions through which individual service modules 320-326 may be configured to interface

with the ICDS API 360. These service modules provide the underlying functionality of ICDS node 300 and may include a data services module 320, an access services module 322, a transaction services module 323, a commercial services module 324, a directory services module 325, and a resource services module 326. The functionality of each of these modules will be described in more detail below.

In one embodiment of the ICDS system, the ICDS API includes a distributed relational database engine 361. As a result, a plurality of ICDS nodes 410-440 can be distributed across ISP 400's internal network and still maintain a coherent, up-to-date storage of Internet content. For example, if a particular data object is updated at two nodes simultaneously, the underlying distributed database system may be configured to resolve any conflicts between the two modifications using a predefined set of distributed database algorithms. Accordingly, the present system provides built in caching support for dynamically changing Internet content (e.g., Web pages which are modified on a regular basis). Such a result was not attainable with the same level of efficiency in prior art caching systems such as proxy server 210 of **Figure 2** (which are executed on, e.g., standard flat file systems such as UNIX or NFS file servers).

Data Services

Data services modules such as module 320 running on each ICDS node 410-440 provide support for data replication and distribution across ISP 400's internal network 480. This includes caching support for any data protocol included in the set of protocol modules 310-319 shown in **Figure 3** as well as for any future protocol which may be added as a module to the ICDS API 360. Because the ICDS API 360 provides a set of standardized service definitions for data services module 320, an ISP using a plurality of ICDS nodes 300 as illustrated in **Figure 4** can replicate data across its network without an extensive knowledge of distributed database technology. In other words, the ISP can configure its plurality of nodes by invoking the standardized service definitions associated with data services module 320 and leave the distributed database functionality to the distributed database engine 361.

Generally, three different types of data replication may be implemented by the present system: dynamic replication, database replication (or "actual" replication), and index replication. Using dynamic replication, if client 472 requests content from internal ICDS server 460 or from a server across network 490, the content will be delivered to client 472 and replicated in ICDS node 430. If client 473 (or any other client) subsequently requests the

same content, it will be transmitted directly from ICDS node 430 rather than from its original source (i.e., a second request to server 460 or a server across network 490 will not be required). Accordingly, bandwidth across ISP 400's internal network and across Internet link 405 is conserved.

The dynamic replication mechanism just described works well for replicating static content but not for replicating dynamically changing content. For example, if the replicated content is a magazine article then caching a copy locally works well because it is static information – i.e., there is no chance that the local copy will become stale (out of date). However, if the replicated content is a Web page which contains continually changing information such as a page containing stock market quotes, then dynamic replication may not be appropriate. No built in mechanism is available for proxy cache server 210 to store an up-to-date copy of the information locally.

The present ICDS system, however, may use database replication to maintain up-to-date content at each ICDS node 410-440. Because the present system includes a distributed database engine 361, when a particular piece of content is changed at one node (e.g., ICDS server 460) a store procedure may be defined to update all copies of the information across the network. This may be in the form of a relational database query. Thus, the present system may be configured to use dynamic replication for static content but to use database replication for time-sensitive, dynamically changing content.

The third type of database replication is known as index replication. Using index replication a master index of content is replicated at one or more ICDS nodes 410-440 across the network 480. Once again, this implementation is simplified by the fact that the underlying ICDS node engine is a distributed database engine. Certain types of information distributions are particularly suitable for using index replication. For example, news overview information (i.e., the list of news articles in a particular newsgroup) is particularly suited to index replication. Instead of replicating each individual article, only the news overview information needs to be replicated at various nodes 410-440 across the network 480. When a client 473 wants to view a particular article, only then will the article be retrieved and cached locally (e.g., on ICDS node 430).

ICDS node 430 is capable of caching and delivering various types of Internet data using any of the foregoing replication techniques. While prior art proxy servers such as proxy server 210 may only be used for caching Web pages, ICDS node 430 is capable of caching various other types of internet content (e.g., news content) as a result of the protocol modules 310-319 interfacing with ICDS API 360. Moreover, as stated above, ICDS node 430 (in conjunction with nodes 410, 420 and 440) may be configured to cache dynamic as well as static Web-based content using various distributed database algorithms.

One specific example of a data service provided by one embodiment of the present system is Wide Area Load Balancing (hereinafter "WALB") using layer 7 switching as described in the co-pending U.S. Patent Application entitled "WIDE AREA LOAD BALANCING" (Serial No. _____), which is assigned to the assignee of the present application and which is incorporated herein by reference. The present system may also perform dynamic protocol selection, dynamic query resolution, and/or heuristic adaptation for replicating content across a network as set forth in the co-pending U.S. Patent Application entitled Dynamic Protocol Selection and "QUERY RESOLUTION FOR CACHE SERVERS" (Serial No. ____), which is assigned to the assignee of the present application and which is incorporated herein by reference. Finally, the present system also may include network news (e.g., Usenet news) services set forth in the co-pending U.S. Patent Applications entitled "HYBRID NEWS SERVER" (Serial No. ___), and "SELF-MODERATED VIRTUAL COMMUNITIES" (Serial No. ___), each assigned to the assignee of the present application and each incorporated herein by reference.

Access Services

As stated above, prior art proxy server cache systems such as proxy server 210 are only capable of caching static, publicly available Web pages. A substantial amount of Web-based and non-Web-based content, however, requires some level of authentication before a user will be permitted to download it. Thus, client 472 (in Figure 4) may pay a service fee to obtain access to content on a particular web site (e.g., from server 460 or from another server over network 490). As a result, when he attempts to access content on the site he will initially be prompted to enter a user name and password. Once the user transmits this information to the Web server, he will then be permitted to download Web server content as per his service agreement.

A problem that arises, however, is that prior art cache systems such as proxy server 210 are not permitted to cache the requested content. This is because proxy server 210 has no way of authenticating subsequent users who may attempt to download the content. Thus, documents which require authentication are simply uncacheable using current network cache systems.

The present ICDS system, however, includes user authentication support embedded in access services module 322. Thus, when client 473, for example, attempts to access a Web page or other information which requires authentication, ICDS node will determine whether the requested content is stored locally. If it is, then ICDS node 430 may communicate with the authentication server (e.g., server 460 or any server that is capable of authenticating client 473's request) to determine whether client 473 should be granted access to the content. This may be accomplished using standardized authentication service definitions embedded in access services module 322. Using these definitions, ICDS node 430 will not only know what authentication server to use, it will also know what authentication *protocol* to use when it communicates to the authentication server. As a result of providing local access services module 322 for authentication, network information which requires authentication can now be cached locally in ICDS node 430, thereby conserving additional bandwidth across network link 405 and/or ISP network 480.

One particular embodiment of the present system replicates Remote Authentication Dial In User Service (hereinafter "RADIUS") information across network 480. RADIUS is an application-level protocol used by numerous ISP's to provide user authentication and profile services. This is achieved by setting up a central RADIUS server with a database of users, which provides both authentication services (i.e., verification of user name and password) and profile services detailing the type of service provided to the user (for example, SLIP, PPP, telnet, rlogin).

Users connect to one or more Network Access Servers (hereinafter "NASs") which operate as a RADIUS clients and communicate with the central RADIUS server. The NAS client passes the necessary user information to the central RADIUS server, and then acts on the response which is returned. RADIUS servers receive user connection requests, authenticate users, and then return all configuration information necessary for the client to deliver service to the user.

One problem associated with the RADIUS protocol is that it does not provide any built in facilities for replication of RADIUS information. Accordingly, on large ISP's such as America Online ("AOL"), which may have tens of millions of users, RADIUS servers are hard hit, potentially handling thousands of logon requests a minute. This may create severe performance/bandwidth problems during high traffic periods. In response, some ISP's have taken a brute-force approach to distributing RADIUS information by simply copying the information to additional servers across the network without any built in mechanism to keep the RADIUS data coherent and up-to-date.

One embodiment of the present ICDS system provides an efficient, dynamic mechanism for distributing RADIUS information. Specifically, a RADIUS module is configured to interface with ICDS API 360 in this embodiment (similar to the way in which protocol modules 310-319 interface with the ICDS API 360). RADIUS information can them be seamlessly distributed across the system using distributed database engine 361. For example, the RADIUS module in conjunction with access services module 322 on ICDS node 430 may maintain radius information for local users. [Exactly how will this work? I assume that access services module will be used but there will be a separate RADIUS protocol module to support the protocol??] Thus, when client 472 first logs in to the system, ICDS node 430 may communicate with a second ICDS node (e.g., central ICDS server 460) which contains the necessary RADIUS authentication and user profile information. Client 472 will input a user name and password and will then be permitted access to the network as per his service agreement with ISP 400.

Unlike previous RADIUS systems, however, ICDS node 430 in the present embodiment may locally cache client 472's RADIUS information so that the next time client 472 attempts to login to the network, the information will be readily available (i.e., no access to a second ICDS node will be necessary). ICDS node 430 may be configured to save client 472's RADIUS information locally for a predetermined period of time. For example, the information may be deleted if client 472 has not logged in to local ICDS node 430 for over a month.

Thus, if client 472 represents a user who frequently travels across the country and logs in to ISP 400's network 480 from various different ICDS nodes, the present system provides a quick, effective mechanism for dynamically replicating client 472's user information into

those geographical locations from which he most commonly accesses ISP 400. This reduces the load which would otherwise be borne by a central RADIUS server and also improves client 472's user experience significantly (i.e., by providing him with a quick login).

Database replication can also be used to update RADIUS information distributed across multiple ICDS nodes 410-440. This may be done using known store procedures defined in relational database 361. For example, if client 472 cancels his service agreement with ISP 400, he should not be able to continually log in to local ICDS node 430 using the RADIUS information which has been cached locally. Thus, under the present ICDS system, ISP 400 may simply issue a relational database query such as [let's add another update query here using database terminology as an example] to immediately update ICDS node 430's radius information.

One of ordinary skill in the art will readily recognize from the preceding discussion that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the invention. Throughout the foregoing description, specific embodiments of the ICDS system were described using the RADIUS protocol in order to provide a thorough understanding of the operation of the ICDS system. It will be appreciated by one having ordinary skill in the art, however, that the present invention may be practiced without such specific details. For example, the ICDS system may also distribute authentication and user profile information in the form of the Lightweight Directory Access Protocol ("LDAP"). In other instances, well known software and hardware configurations/techniques have not been described in detail in order to avoid obscuring the subject matter of the present invention.

Access services module 322 may also provide local encryption/decryption and watermarking of internet content. Audio or video content delivery systems, for example, commonly use encryption of content to protect the rights of the underlying copyright holder. When a user requests a particular piece of content some delivery systems encrypt the content using a unique client encryption key. Only a client who possesses the encryption key (presumably the client who paid for the content) will be permitted to play the content back. Other systems provide for the "watermarking" of content (rather than encrypting) so that the rightful owner of the content may be identified. This simply entails embedding a unique "tag"

which identifies the source of the content and/or the owner of the content (i.e., the one who paid for it).

Prior art caching systems such as proxy server 210 are not capable of dealing with encrypted or watermarked content because the encryption/watermarking functionality was not provided locally (i.e., proxy server 210 was not "smart" enough). In one embodiment of the present ICDS system, however, access services module 322 of ICDS nodes 410-440 includes a local encryption module and/or a local watermarking module. For example, if client 473 requests specific content such as a copyrighted music content stored on a music server (e.g., ICDS server 460), the initial request for the content will be made from ICDS node 430 on behalf of client 473. ICDS node 430 will retrieve the requested content and cache it locally. If the requested content requires encryption, ICDS node 430 will use its local encryption module to encrypt the requested content using a unique user encryption key for client 473.

If a second client – e.g., client 472 – requests the same content, the copy stored locally on ICDS module 430 can be used. ICDS module 430 will simply encrypt the content using a *different* encryption key for user 472. Thus, frequently requested multimedia content (which, as is known in the art, can occupy a substantial amount of storage space) may be cached locally at ICDS node 430 notwithstanding the fact that the content requires both user authentication and encryption.

The same functionality may be provided for watermarking of content. ICDS node will use a watermarking module (which may comprise a component of access services module 322) to individually watermark multimedia information requested by individual clients, thereby protecting the rights of the copyright holder of the underlying multimedia content. This information can then be regularly communicated back to a central database repository.

As is known in the art, multimedia files can be extremely large and, accordingly, take substantially more time to communicate across a network than do, for example, generic Web pages. As such, the ability to locally cache multimedia files significantly reduces traffic across network 480, and also significantly improves the user experience for local users when downloading multimedia information.

Transaction Services

In addition to replicating data services and access services information across a network, the present ICDS system also provides for the replication of transaction services. Transaction services includes maintaining information on client payments for use of ISP 400's services as well as information relating to the client's online access profile (i.e., recording of the times when the user is online).

When a client logs in to an ISP today, the client's online information is maintained on a single central server. The central server maintains records of when and for how long the client logged in to the network and may also include information about what the client did while he was online. As was the case with maintaining a central RADIUS server, maintaining a central transaction server for all users of a large ISP is inefficient and cumbersome. The present system solves the performance and bandwidth problems associated with such a configuration by storing transaction information locally via transaction module 323 and algorithms build around distributed database engine 361.

Thus, if client 472 only logs on to ISP 400's network 480 via ICDS node 430, all of his transaction and billing information will be stored locally. The information may then be communicated across network 480 to a central billing server at predetermined periods of time (e.g., once a month). [We didn't go into great detail on transaction services and the rest; please add information as you feel appropriate]

Commercial Services

Commercial services module 324 provides a significantly improved local caching capability for add rotation and accounting. An add rotation system operating in conjunction with a typical proxy cache server will now be described with respect to **Figure 2**. When client 120 downloads a web page from Web server 142 the Web page may contain an ad tag or an add tag may automatically be inserted. The add tag will identify add server 170 and will indicate that an add should be inserted into the requested Web page from add server 170. Add server will then identify a specific add to insert into the downloaded Web page from add content server. The Web page plus the inserted add will then be forwarded to proxy server 210 and on to client 120.

Add server 170 will keep an accounting of how many different users have downloaded Web pages with adds inserted as described above. However, one problem with accounting on this system is that proxy server 210 requests Web pages *on behalf of* its clients. Accordingly, once the requested Web pages has been cached locally in Web cache 220, add server will only receive requests from proxy server 210 for any subsequent requests for the Web page. This will result in an inaccurate accounting of how many unique clients actually requested the Web page (and how many adds were viewed by unique users).

Because one embodiment of the present system provides built in caching support for ad rotation services, an accurate accounting of the number of hits that a particular ad receives may be maintained. More specifically, one embodiment of the present ICDS system solves this problem by providing a commercial services module that monitors and records how often individual clients request Web pages containing particular adds from add content server 171. This information than then be communicated to a central server (e.g., ICDS server 460) at predetermined intervals for generating add rotation usage reports.

Directory Services

Directory services provide the ability to cache locally a directory of information across network 480 or 490. That is, the question here is not whether the particular information is available but where exactly over networks 480 or 490 it is located. It should be noted that there may be some overlap between the directory services concept and the index replication concept described above with respect to data services. [I'm still not 100% sure what this is – please elaborate with an example]

II. Content Routing

The term "content routing" refers to the ability to select among various techniques/protocols for maintaining a coherent set of content across a network. The selection of a particular technique may be based on several routing variables including, but not limited to, the type of content involved (i.e., FTP, HTTP . . . etc), the size of the content involved (i.e., small files such as HTTP vs. large files such as audio/video streaming), the location of the content on network 480 and/or network 490, the importance of a particular piece of content (i.e., how important it is that the content be kept up-to-date across the entire network), the particular user requesting the content and the terms of his subscription agreement (i.e., some users may be willing to pay more to be insured that they receive only the most up-to-date

content without having to wait), the frequency with which the content is accessed (e.g., 5%-10% of content on the Internet represents 90% of all the *requested* content), and the underlying costs and bandwidth constraints associated with maintaining up-to-date, coherent content across a particular network (e.g., network 480).

Three content routing techniques which may be selected (based on one or more of the foregoing variables) to maintain coherent content across the plurality of nodes illustrated in **Figure 4** are content revalidation, content notification, and content synchronization.

Content Revalidation

When content validation is selected, the original content source will be checked only when the content is requested locally. For example, client 473 may request an installation program for a new Web browser (e.g., the latest version of Microsoft's™ Internet Explorer™). The file may then be transmitted form ICDS server 460 to client 473 and a copy of the file cached locally on ICDS node 430. Consequently, if client 472 requests the same program, for example, two weeks later, ICDS node 430 may be configured to check ICDS server 460 to ensure that it contains the most recent copy of the file before passing it on to client 472 (i.e., ICDS node 430 "revalidates" the copy it has locally).

ICDS node 430 may also be configured to revalidate a piece of content only if has been stored locally for a predetermined amount of time (e.g., 1 week). The particular length of time selected may be based on one or more of the variables discussed above. Moreover, in one embodiment, the age/revision of a particular piece of content is determined based on tags (e.g., HTML metatags) inserted in the particular content/file.

Revalidation may work more efficiently with certain types of content than with others. For example, revalidation may be an appropriate mechanism for maintaining up-to-date copies of larger files which do not change very frequently (i.e., such as the program installation files described above). However, revalidation may not work as efficiently for caching smaller and/or continually changing files (e.g., small HTML files) because the step of revalidating may be just as time consuming as making a direct request to ICDS server 460 for the file itself. If the file in question is relatively small and/or is changing on a minute-by-minute basis (e.g., an HTML file containing stock quotes) then one or more other content routing techniques may be more appropriate.

Of course, other routing variables may influence the decision on which technique to use, including the issue of how strong the data transmission connection is between ICDS node 430 and ICDS server 460 (i.e., how reliable it is, how much bandwidth is available . . . etc) and the necessity that the underlying information cached locally (at ICDS node 430) be accurate. The important thing to remember is that ICDS node 430 – because of its underlying open API architecture – may be configured based on the unique preferences of a particular client.

Content Notification

Content notification is a mechanism wherein the central repository for a particular piece of content maintains a list of nodes, or "subscribers," which cache a copy of the content locally. For example, in **Figure 4**, a plurality of agents may run on ICDS server 460 which maintain a list of content subscribers (e.g., ICDS node 430, ICDS node 420 . . . etc) for specific types of content (e.g., HTML, data streaming files, FTP files . . . etc). In one embodiment of the system, a different agent may be executed for each protocol supported by ICDS server 460 and/or ICDS nodes 410-440.

When a particular piece of content is modified on ICDS server 460, a notification of the modification may be sent to all subscriber nodes (i.e., nodes which subscribe to that particular content). Upon receiving the notification, the subscriber node – e.g., ICDS node 430 – may then invalidate the copy of the content which it is storing locally. Accordingly, the next time the content is requested by a client (e.g., client 472), ICDS node 430 will retrieve the up-to-date copy of the content from ICDS server 460. The new copy will then be maintained locally on ICDS node 430 until ICDS node 430 receives a second notification from an agent running on ICDS server 460 indicating that a new copy exists.

Alternatively, each time content is modified on ICDS server 460 the modified content may be sent to all subscriber nodes along with the notification. In this manner a local, up-to-date copy of the content is always ensured. In one embodiment of the system, notification and/or transmittal of the updated content by the various system agents is done after a predetermined period of time has elapsed (e.g., update twice a day). The time period may be selected based on the importance of having an up-to-date copy across all nodes on the network 480, 490.

As was the case with content revalidation, the different varieties of content notification may work more efficiently in some situations than in others. Accordingly, content notification may be selected as a protocol (or not selected) based on one or more of the routing variables recited at the beginning of this section (i.e., the "content routing" section). For example, content notification may be an appropriate technique for content which is frequently requested at the various nodes across networks 480 and 490 (e.g., for the 5-10% of the content which is requested 90% of the time), but may be a less practical technique for larger amount of content which is requested infrequently. As another example, large files which change frequently may not be well suited for content notification (i.e., particularly the type of content notification where the actual file is sent to all subscribers along with the notification) due to bandwidth constraints across networks 480 and/or 490 (i.e., the continuous transmission of large, frequently changing files may create too much additional network traffic).

Content Synchronization

Content synchronization is a technique for maintaining an exact copy of a particular type of content on all nodes on which it is stored. Using content synchronization, as soon as a particular piece of content is modified at, for example, ICDS node 430, it will immediately be updated at all other nodes across networks 480 and/or 490. If the same piece of data was concurrently modified at one of its other storage locations (e.g., ICDS node 410) then the changes may be backed off in order to maintain data coherency. Alternatively, an attempt may be made to reconcile the two separate modifications if it is possible to do so (using, e.g., various data coherency techniques).

Once again, as with content notification and content revalidation, content synchronization is more suitable for some situations than it is for others. For example, content synchronization is particularly useful for information which can be modified from several different network nodes (by contrast, the typical content notification paradigm assumes that the content is modified at one central node). Moreover, content synchronization may be useful for maintaining content across a network which it is particularly important to keep current. For example, if network 480 is an automatic teller machine (hereinafter "ATM") network, then when a user withdraws cash from a first node (e.g., ICDS node 440), his account will be instantly updated on all nodes (e.g., ICDS node 410, 420, 460, and 430) to reflect the withdrawal. Accordingly, the user would not be able to go to a different node in a different part of the country and withdraw more than what he actually has in his account.

As another example, a user's account status on a network (i.e., whether he is a current subscriber and/or what his network privileges are) may be maintained using content synchronization. If, for example, a user of network 480 were arrested for breaking the law over network 480 (e.g., distributing child pornography), it would be important to disable his user account on all network nodes on which this information might be cached. Accordingly, using content synchronization, once his account was disabled at one node on network 480 this change would automatically be reflected across all nodes on the network.

As previously stated, the choice of which content routing technique to use for a particular type of content may be based on any of the variables set forth above. In one embodiment, the frequency with which content is accessed across the networks 480 and/or 490 may be an important factor in deciding which protocol to use. For example, the top 1% accessed content may be selected for content synchronization, the top 2%-10% accessed content may be selected for content notification, and the remaining content across networks 480, 490 may be selected for content revaildation.

III. Content Delivery Medium Selection

In addition to the content routing flexibility provided by the content delivery system as set forth above, one embodiment of the system allows content delivery nodes such as ICDS node 430 to select from a plurality of different transmission media. For example, ICDS node 430 may receive content from ICDS server 460 via a plurality of communication media, including, but not limited to, satellite transmission, wireless RF transmission, and terrestrial transmission (e.g., fiber).

Moreover, as with the selection of a particular content routing technique, the selection of a particular transmission medium may be based on any of the variables set forth above (see, e.g., routing variables listed under counter routing heading; page 24, line 18 through page 25, line 9). Moreover, the choice of a particular transmission medium may be dynamically adjustable based on performance of that medium. For example, ICDS node 430 may be configured to receive all of its content over terrestrial network 480 as long as network 480 is transmitting content at or above a threshold bandwidth. When transmissions over network 480 dip below the threshold bandwidth, ICDS node 480 may then begin receiving certain content via satellite broadcast or wireless communication.

In addition, a transmission medium may be selected for transmitting specific content based on how frequently that content is accessed. For example, the top 10% frequently accessed content may be continually pushed out to ICDS node 430 via satellite broadcast while the remaining content may be retrieved by (i.e., "pulled" to) ICDS node 430 over network 480 upon request by clients (e.g., client 473). Accordingly, those employing ICDS nodes such as node 430 can run a cost-benefit analysis to determine the most cost effective way to implement their system by taking in to consideration, for example, the needs of their users, the importance of the content involved and the expense of maintaining multiple transmission connections into ICDS node 430 (e.g., the cost associated with maintaining an ongoing satellite connection).

In one embodiment of the system, tags (e.g., HTML metatags) may be inserted into particular types of content to identify a specific transmission path/medium for delivering that content to ICDS node 480. The tags in this embodiment may identify to various nodes (and/or routers) across networks 480 and/or 490 how the particular content should be routed across the networks (e.g., from node 410 to node 420 via terrestrial network 480; from node 420 to node 430 via wireless transmission).

One of ordinary skill in the art will readily recognize from the preceding discussion that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the invention. Throughout this detailed description, numerous specific details are set forth such as specific network protocols (i.e., RADIUS) and networks (i.e., the Internet) in order to provide a thorough understanding of the present invention. It will be appreciated by one having ordinary skill in the art, however, that the present invention may be practiced without such specific details. In other instances, well known software and hardware configurations/techniques have not been described in detail in order to avoid obscuring the subject matter of the present invention. The invention should, therefore, be measured in terms of the claims which follow.

CLAIMS

What is claimed is:

1. A network content delivery system configured to:

select a first content routing technique for processing a first set of network content; and

select a second content routing technique for processing a second set of network content, wherein said first and second content routing techniques are selected based on one or more content routing variables.

2. The network content delivery system as claimed in Claim 1 wherein one of said selected content routing techniques is a content revalidation technique.

3. The network content delivery system as claimed in Claim 1 wherein one of said selected content routing techniques is a content notification technique.

4. The network content delivery system as claimed in Claim 1 wherein one of said selected content routing techniques is a content synchronization technique.

5. The network content delivery system as claimed in Claim 1 wherein one of said content routing variables is the frequency with which said network content is accessed by users.

6. The network content delivery system as claimed in Claim 1 wherein one of said content routing variables is the size of said network content.

7. The network content delivery system as claimed in Claim 1 wherein one of said content routing variables is the frequency with which said network content is modified.

8. The network content delivery system as claimed in Claim 1 wherein one of said content routing variables is the type of network content (e.g., HTML, Usenet News).

9. The network content delivery system as claimed in Claim 1 wherein one of said content routing variables is identity of the user requesting said network content.

10. The network content delivery system as claimed in Claim 2 wherein said content revalidation technique is selected based on the size of said network content.

11. The network content delivery system as claimed in Claim 2 wherein said content revalidation technique is selected based on the frequency with which said network content is accessed.

12. The network content delivery system as claimed in Claim 3 wherein said content notification technique is selected based on the size of said network content.

13. The network content delivery system as claimed in Claim 3 wherein said content notification technique is selected based on the frequency with which said network content is accessed.

14. The network content delivery system as claimed in Claim 4 wherein said content synchronization technique is selected based on the size of said network content.

15. The network content delivery system as claimed in Claim 4 wherein said content synchronization technique is selected based on the frequency with which said network content is accessed.

16. The network content delivery system as claimed in Claim 1 including the additional step of selecting a first transmission medium for a first group of network content based on one or more of said content routing variables.

17. The network content delivery system as claimed in Claim 16 including the additional step of selecting a second transmission medium for a second group of network content based on one or more of said content routing variables.

18. The network content delivery system as claimed in Claim 1 including an application programming interface for interfacing with a plurality of network protocol and service modules.

19. A content delivery system comprising:

a network node for storing network content;

a first transmission medium communicatively coupled to said network node for transmitting a first set of network content to said network node; and

a second transmission medium communicatively coupled to said network node for transmitting a second set of network content to said network node,

wherein said first and second sets of network content are selected based on one or more routing variables.

20. The content delivery system as claimed in Claim 19 wherein said first transmission medium is a satellite transmission.

21. The content delivery system as claimed in Claim 19 wherein said first transmission medium is a wireless radio frequency transmission.

22. The content delivery system as claimed in Claim 19 wherein said first transmission medium is terrestrial-based transmission.

23. The content delivery system as claimed in Claim 19 wherein said network node monitors transmission bandwidth of said first transmission medium and reallocates content

from said first set to said second set if said first transmission medium drops below a predetermined threshold value.

24. The content delivery system as claimed in Claim 23 wherein said first transmission medium is terrestrial and said second transmission medium is non-terrestrial.

25. The content delivery system as claimed in Claim 19 wherein content is included in said first set based on the frequency with which said content is accessed.

26. The network content delivery system as claimed in Claim 19 including an application programming interface for interfacing with a plurality of network protocol and service modules.

27. An article of manufacture including a sequence of instructions stored on a computer-readable media which, when executed by a network node, cause the network node to perform the acts of:

establishing a plurality of groups of network content to be cached on said network node based on one or more content routing variables;

selecting a first content routing technique for maintaining data coherency in a first group of said plurality; and

selecting a second content routing technique for maintaining data coherency in a second group of said plurality.

28. The article of manufacture as claimed in claim 27 wherein said first content routing technique is content revalidation.

29. The article of manufacture as claimed in Claim 28 wherein said second content routing technique is content notification.

30. The article of manufacture as claimed in Claim 28 wherein said second content routing technique is content synchronization.

31. The article of manufacture as claimed in Claim 28 wherein said content routing variable used to select said content for said first group is the frequency with which said content is accessed.

32. The article of manufacture as claimed in Claim 29 wherein said content routing variable used to select said content for said first group is the frequency with which said content is accessed.

33. The article of manufacture as claimed in Claim 30 wherein said content routing variable used to select said content for said first group is the frequency with which said content is accessed.

34. A network node comprising:

an application programming interface ("API"), said API including a distributed relational database engine;

a plurality of protocol modules for interfacing with said API, said protocol modules configured to allow said system to communicate over a network using a plurality of network protocols;

a cache memory for caching data communicated to said cache memory using said plurality of protocol modules; and

a data services module for maintaining coherency between said data stored in said cache memory and data stored at other nodes across said network.

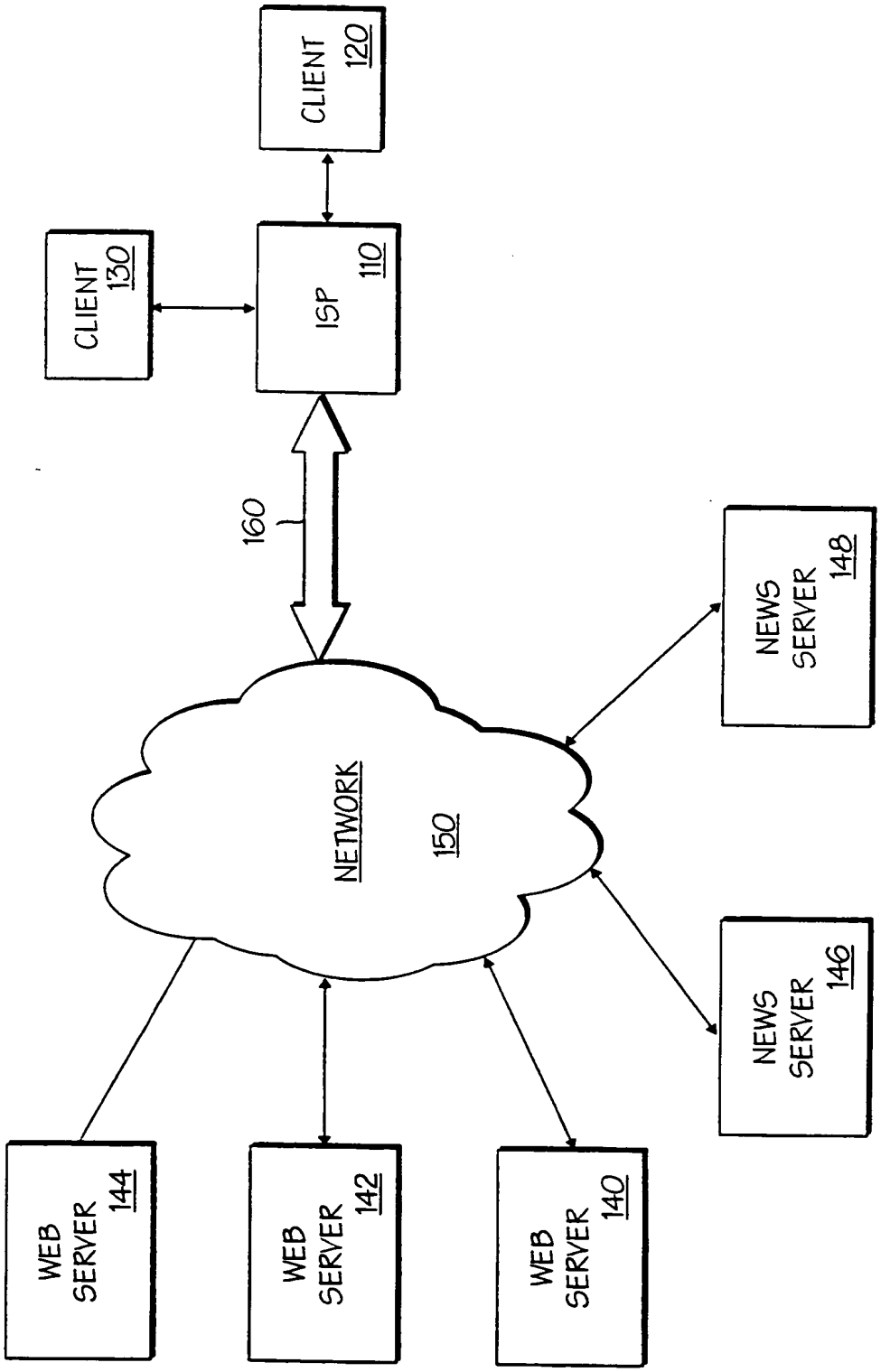

CLIENT 120
CLIENT 130
ISP 110
160
NETWORK 150
NEWS SERVER 148
NEWS SERVER 146
WEB SERVER 144
WEB SERVER 142
WEB SERVER 140

FIG. 1

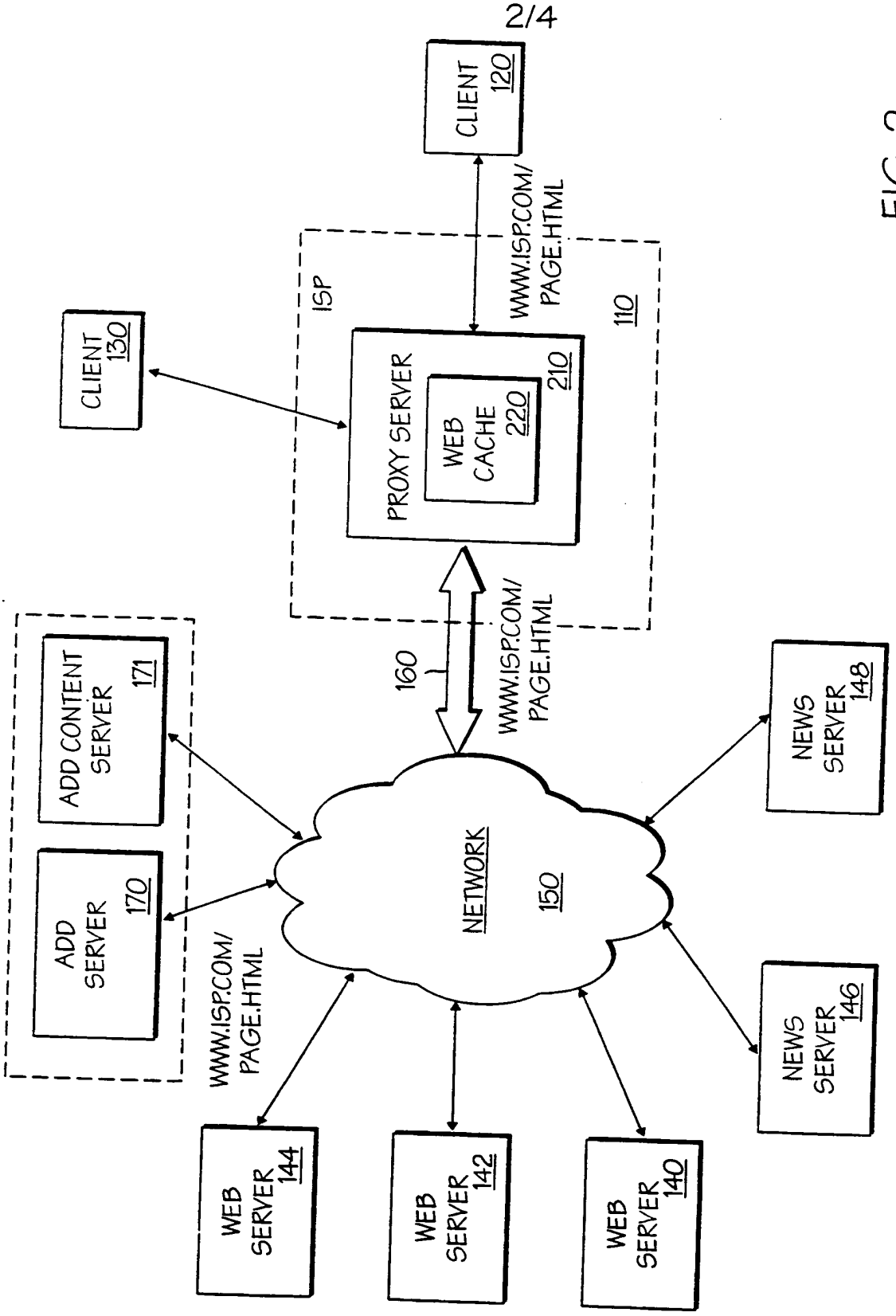

CLIENT 120
WWW.ISP.COM/ PAGE.HTML
ISP
110
CLIENT 130
PROXY SERVER 210
WEB CACHE 220
160
WWW.ISP.COM/ PAGE.HTML
ADD CONTENT SERVER 171
ADD SERVER 170
WWW.ISP.COM/ PAGE.HTML
NETWORK 150
NEWS SERVER 148
NEWS SERVER 146
WEB SERVER 144
WEB SERVER 142
WEB SERVER 140

FIG. 2

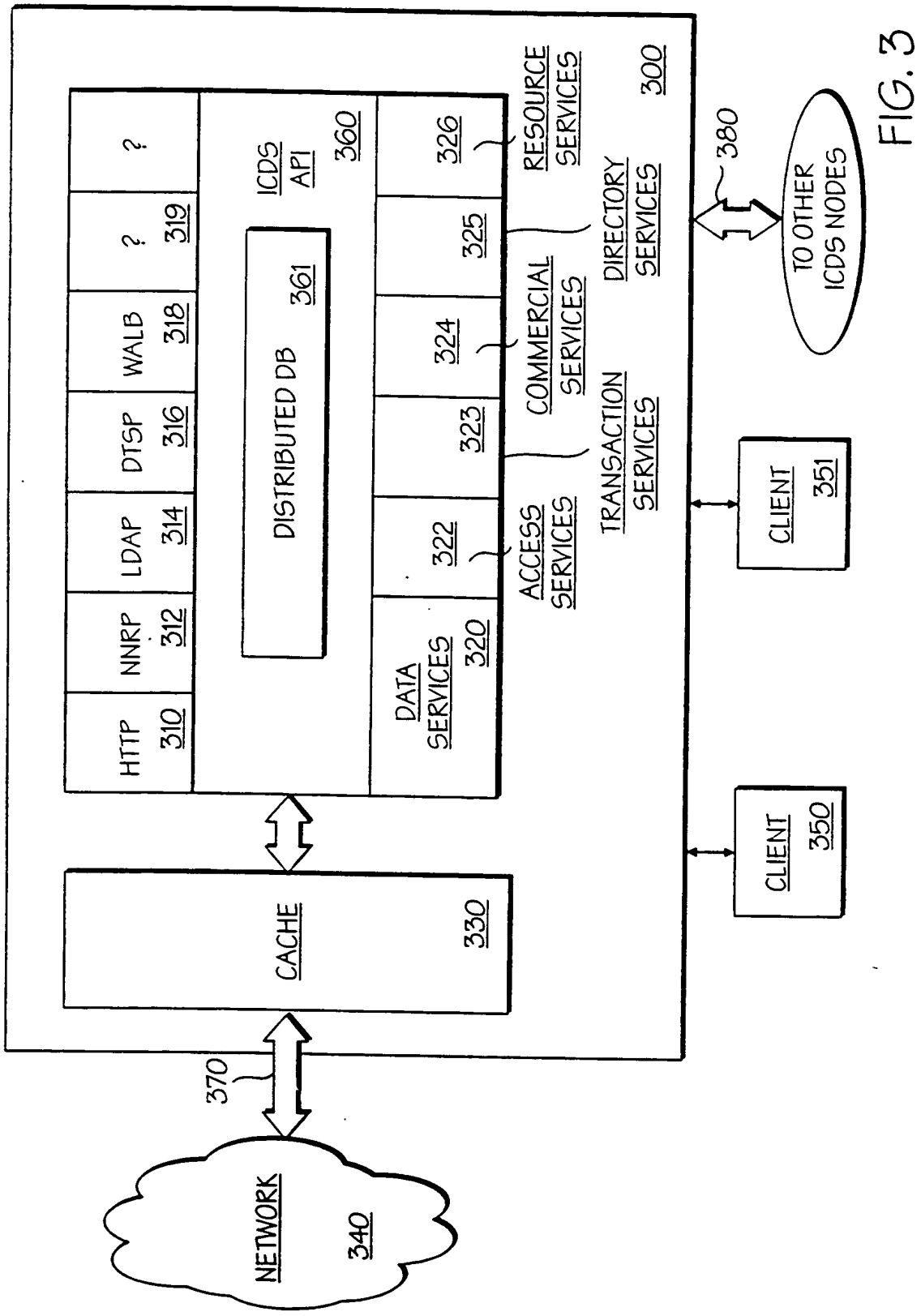

300
380
TO OTHER ICDS NODES
CLIENT 351
CLIENT 350
370
NETWORK 340
CACHE 330
HTTP 310
NNRP 312
LDAP 314
DTSP 316
WALB 318
? 319
?
DISTRIBUTED DB 361
ICDS API 360
DATA SERVICES 320
ACCESS SERVICES 322
TRANSACTION SERVICES 323
COMMERCIAL SERVICES 324
DIRECTORY SERVICES 325
RESOURCE SERVICES 326

FIG. 3

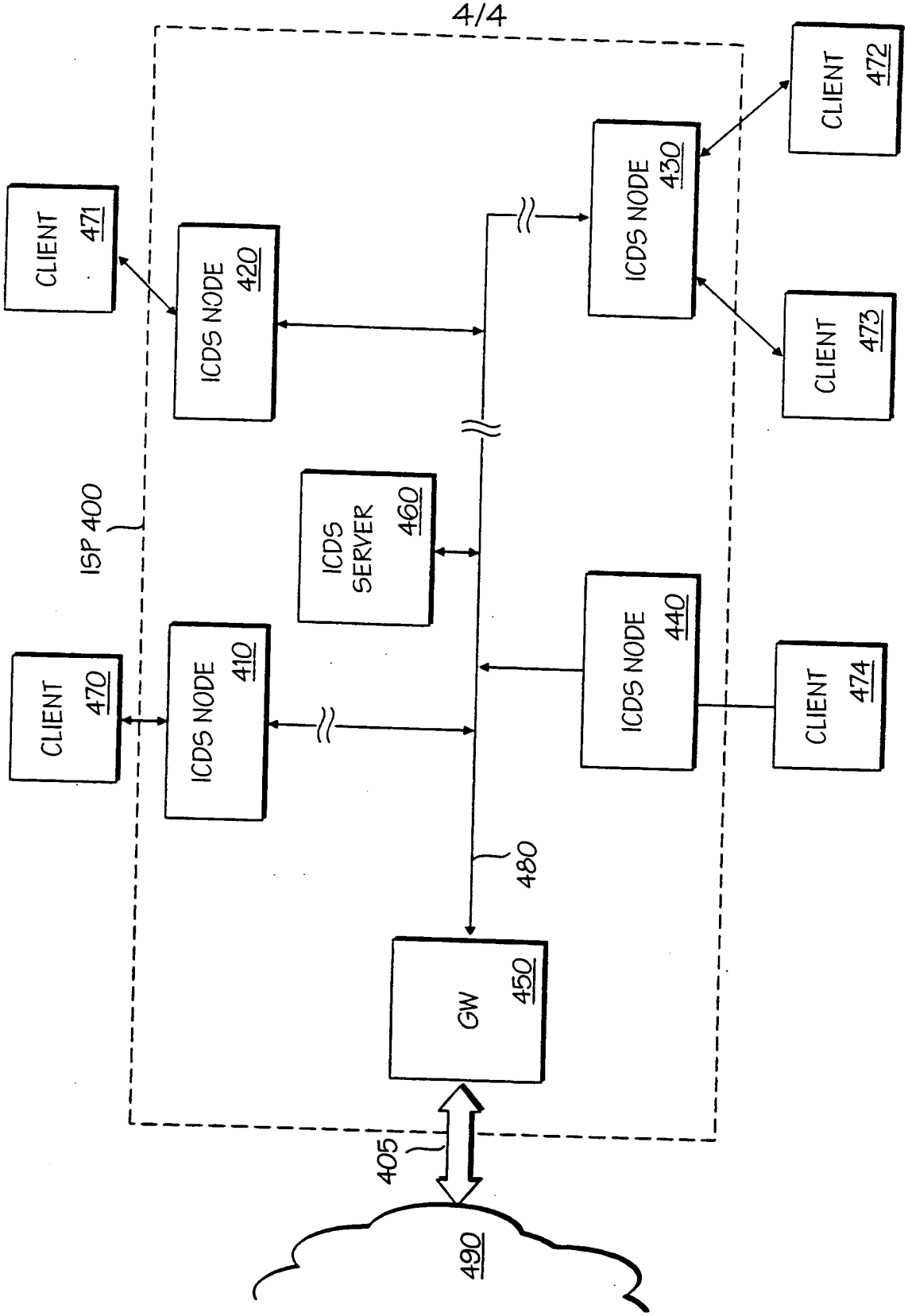

ISP 400
CLIENT 471
ICDS NODE 420
ICDS NODE 430
CLIENT 472
CLIENT 473
ICDS SERVER 460
CLIENT 470
ICDS NODE 410
ICDS NODE 440
CLIENT 474
480
GW 450
405
490

FIG. 4

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 December 2000 (07.12.2000)

PCT

(10) International Publication Number
WO 00/73922 A3

(51) International Patent Classification[7]: H04L 29/06, G06F 17/30

(21) International Application Number: PCT/US00/11078

(22) International Filing Date: 25 April 2000 (25.04.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/323,635 1 June 1999 (01.06.1999) US

(71) Applicant: ENTERA, INC. [US/US]; 40971 Encyclopedia Circle, Fremont, CA 94538 (US).

(72) Inventor: SCHARBER, John, M.; 1616 Placer Circle, Livermore, CA 94550 (US).

(74) Agents: FAHMI, Tarek, N. et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, 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, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

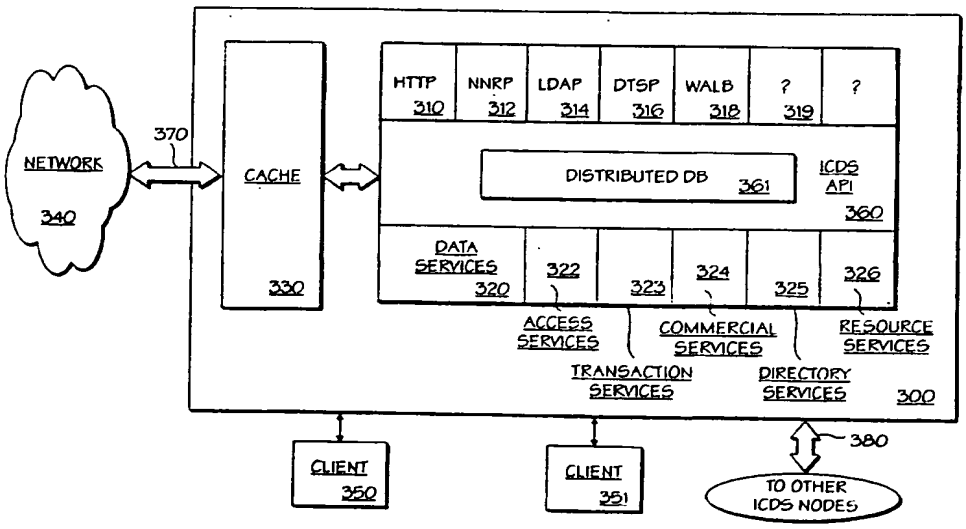
(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(88) Date of publication of the international search report:
16 August 2001

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.

(54) Title: CONTENT DELIVERY SYSTEM



(57) Abstract: Disclosed is a network content delivery system configured to: select a first content routing technique for processing a first set of network content; and select a second content routing technique for processing a second set of network content, wherein the first and second content routing techniques are selected based on one or more content routing variables. Also disclosed is a content delivery system comprising: a network node for storing network content; a first transmission medium communicatively coupled to the network node for transmitting a first set of network content to the network node; and a second transmission medium communicatively coupled to the network node for transmitting a second set of network content to the network node, wherein the first and second sets of network content are selected based on one or more routing variables.

WO 00/73922 A3

INTERNATIONAL SEARCH REPORT

Inter ›nal Application No
PCT/US 00/11078

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06 G06F17/30

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 7 H04L G06F H04B

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)
EPO-Internal, WPI Data, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CAUGHEY S J ET AL: "Flexible open caching for the Web" COMPUTER NETWORKS AND ISDN SYSTEMS,NL,NORTH HOLLAND PUBLISHING. AMSTERDAM, vol. 29, no. 8-13, 1 September 1997 (1997-09-01), pages 1007-1017, XP004095299 ISSN: 0169-7552 abstract page 1009, right-hand column, paragraph 1 -page 1013, right-hand column, paragraph 1	1-5,7,8, 11,13, 18, 27-32,34
A	page 1015, left-hand column, line 6 - line 12 page 1015, right-hand column, paragraph 3	6,9,10, 12,14, 15,33

-/--

[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

Date of the actual completion of the international search	Date of mailing of the international search report
2 March 2001	14 03. 2001

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	Blanco Cardona, P

Form PCT/ISA/210 (second sheet) (July 1992)

3

INTERNATIONAL SEARCH REPORT

Inte. onal Application No
PCT/US 00/11078

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	CAO P ET AL: "MAINTAINING STRONG CACHE CONSISTENCY IN THE WORLD WIDE WEB" IEEE TRANSACTIONS ON COMPUTERS,US,IEEE INC. NEW YORK, vol. 47, no. 4, 1 April 1998 (1998-04-01), pages 445-457, XP000740725 ISSN: 0018-9340 abstract	1-5,7, 11,13, 27-32
A	page 445, right-hand column, paragraph 2 - paragraph 3	15,33
A	DINGLE A ET AL: "Web cache coherence" COMPUTER NETWORKS AND ISDN SYSTEMS,NL,NORTH HOLLAND PUBLISHING. AMSTERDAM, vol. 28, no. 11, 1 May 1996 (1996-05-01), pages 907-920, XP004018195 ISSN: 0169-7552 page 908, left-hand column, paragraph 1 -right-hand column, paragraph 1	3,12,13, 29,32
A	GB 2 294 132 A (MARCONI GEC LTD) 17 April 1996 (1996-04-17) abstract	4,14,15, 30,33
A	WOOSTER R P ET AL: "Proxy caching that estimates page load delays" COMPUTER NETWORKS AND ISDN SYSTEMS,NL,NORTH HOLLAND PUBLISHING. AMSTERDAM, vol. 29, no. 8-13, 1 September 1997 (1997-09-01), pages 977-986, XP004095296 ISSN: 0169-7552 abstract page 978, left-hand column, paragraph 3 -right-hand column, paragraph 1 page 979, left-hand column, paragraph 3	6,10,12, 14
X	DE 43 08 161 A (PHILIPS PATENTVERWALTUNG) 22 September 1994 (1994-09-22) the whole document	19-24

-/--

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inte. .onal Application No
PCT/US 00/11078

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	ARANGO M ET AL: "GUARANTEED INTERNET BANDWIDTH" PHOENIX, MAY 4 - 7, 1997,NEW YORK, IEEE,US, vol. CONF. 47, 18 November 1996 (1996-11-18), pages 862-866, XP000741554 ISBN: 0-7803-3660-7 abstract page 862, right-hand column, paragraph 4 -page 863, left-hand column, paragraph 3 page 864, left-hand column, paragraph 3 -page 865, left-hand column, paragraph 2 page 866, right-hand column, line 9 - line 12	1,16,17, 19,23,26
X	EP 0 689 338 A (SONY CORP) 27 December 1995 (1995-12-27) abstract column 1, line 46 -column 2, line 3	1,16,17
A	column 11, line 51 -column 12, line 1	19-22, 24,25

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 00/11078

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☒ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☒ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. Claims: 1-15,18,27-34

   System where content routing techniques are selected based on the size of network content

2. Claims: 16,17,19-26

   content delivery system comprising two possible transmission mediums

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inter. ...nal Application No
PCT/US 00/11078

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
GB 2294132 A	17-04-1996	NONE	
DE 4308161 A	22-09-1994	NONE	
EP 0689338 A	27-12-1995	JP 8008860 A US 5801750 A	12-01-1996 01-09-1998

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

(11) EP 1 041 823 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
04.10.2000 Bulletin 2000/40

(51) Int Cl.[7]: **H04N 7/167**

(21) Application number: **00302721.6**

(22) Date of filing: **31.03.2000**

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

(30) Priority: **31.03.1999 JP 9391699**

(71) Applicant: **KABUSHIKI KAISHA TOSHIBA**
Kawasaki-shi, Kanagawa-ken 210-8572 (JP)

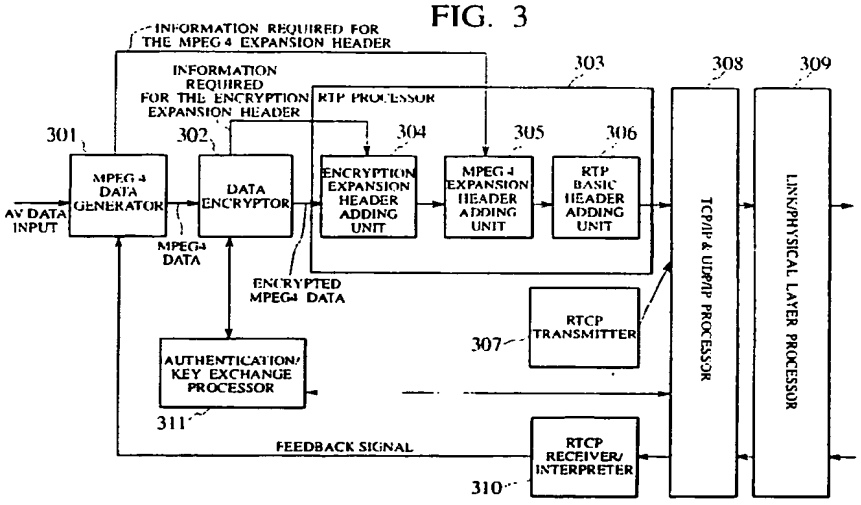
(72) Inventors:
- **Saito, Takeshi, Intellectual Property Division**
  Tokyo (JP)
- **Kato, Taku, Intellectual Property Division**
  Tokyo (JP)
- **Tomoda, Ichiro c/o Intellectual Property Division**
  Tokyo (JP)
- **Takabatake, Yoshiaki, Intell. Prop. Div.**
  Tokyo (JP)
- **Ami, Junko, Intellectual Property Division**
  Tokyo (JP)

(74) Representative: **Midgley, Jonathan Lee**
Marks & Clerk
57-60 Lincoln's Inn Fields
GB-London WC2A 3LS (GB)

(54) Content distribution apparatus, content receiving apparatus, and content distribution method

(57) A content distribution apparatus for implementing copy protection when distributing digital content as a real-time stream on the Internet is provided. This apparatus encrypts content and distributes them to a receiving apparatus via the Internet, and performs an authentication procedure and a key exchange procedure between with the receiving apparatus. The encoded content encoded by a prescribed encoding system is encrypted (S401), an encryption expansion header is generated that includes at least one attribute information of attribute information indicating whether or not the content is encrypted and attribute information indicating the encryption system used (S403), transport protocol processing required to transfer the content is performed and a basic transport header is generated (S407), a packet being sent which includes the basic transport header, the encryption expansion header, and the encrypted content (S409).

FIG. 3



EP 1 041 823 A2

Printed by Jouve, 75001 PARIS (FR)

Description

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0001] The present invention relates to a content distribution apparatus, a content receiving apparatus, and a content distribution method. More particularly, it relates to technology in real-time distribution of content via an open network such as the Internet, whereby protection is provided against content theft by third parties and unauthorized copying of content by a content recipient. thereby enabling transmission and reception of content data while considering copyright protection.

2. Description of the Related Art

[0002] In recent years, the digitization of the home sound and image (audio/visual; hereinafter abbreviated AV) environment, as exemplified by the start of digital broadcasts and the sales of digital AV equipment has gained much attention. Digital AV data enables diverse types of compression, is amenable to processing as multimedia data, tolerates unlimited playback without deterioration. and has other features which are expected to result in an expansion in its application in the future.

[0003] On the other hand, digital AV data technology is accompanied by the problem of easy unauthorized copying of content. More specifically, any type of digital content can in principle be used to create a copy identical to the original and indefinitely durable, by means of bit copying. in the process of unauthorized content copying.

[0004] A variety of technologies are being studied for the purpose of preventing unauthorized copying. One of these is the 1394CP Content Protection System Specification being studied by the CPTWG (Content Protection Technical Working Group). In this technology, an authentication procedure is performed beforehand between sending and receiving nodes for content (such as MPEG data) to be transferred between nodes connected by the IEEE 1394 bus, to enable shared use of an encryption key (content key). Thereafter, this encryption key is used to encrypt the content and then encrypted content is transferred, and it is not possible for nodes other than the nodes that performed the authentication procedure to decrypt the content. By doing this, because a node other than the authenticated nodes (that is, a third party node) does not know the encryption key, even the node were able to capture the transferred data (that is, the encrypted content data), it would not be able to decrypt it. Nodes that can participate in this authentication procedure are limited to nodes that receive permission to do so by an authentication organization beforehand, thereby preventing an unauthorized node from obtaining the encryption key, and thus preventing unauthorized copying of content.

[0005] The distribution of digital content is not, of course, limited to transfer via the IEEE 1394 bus, and general networks are expected to be used. The Internet is a strong candidate for building a technology infrastructure that is not wedded to public networks or physical/link networks.

[0006] In conventional content distribution as practiced, however, digital content on the Internet (and in particular digital AV streams) was mainly transferred in its raw form by the RTP (Real-time Transport Protocol), without copyright protection provided by prevention of third-party theft and unauthorized copying by a recipient.

SUMMARY OF THE INVENTION

[0007] Accordingly, it is an object of the present invention, in consideration of the above-noted situation, to provide content distribution apparatus, a content receiving apparatus, and a content distribution method, which provide protection from copying when digital content is transferred on the Internet by real-time streaming.

[0008] A feature of the present invention is that information with regard to encryption and encoding etc. required to provide copy protection for digital content is efficiently appended as various headers to the content, making use of the characteristics of the Internet.

[0009] One aspect of the present invention provides a content information distribution apparatus for distributing encrypted content information, via a network in accordance with a prescribed transport protocol, to other end apparatus in a communication authenticated by an authentication process including at least one procedure of an authentication procedure and a key exchange procedure, comprising: (a) a unit for encrypting content information encoded by a prescribed encoding system; (b) a unit for generating an encryption attribute header including attribute information with regard to the encryption of the content information; (c) a unit for performing transport protocol processing required to transfer the content information and for generating a basic transport header to be added to the content information to which the encryption attribute header has been added; and (d) a unit for sending to the other end apparatus that is authenticated a packet including the basic transport header, the encryption attribute header, and the encrypted content information, wherein the encryption attribute header is set into an expansion transport header within a packet header of the packet or into a payload header within a payload to be encrypted of the packet.

[0010] It is preferable that the encryption attribute header includes at least one of the existence or non-existence of encryption of the content information and the encryption system of the content information.

[0011] It is preferable that the encryption attribute header includes a copy attribute field having a plurality of bits with regard to the number of copying of the con-

tent information.

[0012] It is preferable that the encryption attribute header includes a counter field indicating a change in an encryption key.

[0013] It is preferable that the unit (b) sets the encoding information, which indicates the encoding system for the content information into the expansion transport header or into the payload header.

[0014] It is preferable that the unit (c) further codes into the basic transport header at least information to the effect that there is a possibility that the content information is encrypted, and wherein the unit (b) codes into the expansion header at least information as to whether or not the content information to be transferred is encrypted.

[0015] It is preferable that the unit (b) codes into the expansion header information as to whether or not the content information to be transferred is encrypted.

[0016] The above-noted content information distribution apparatus can further comprising: (e) a unit for generating a content attribute header that includes content attribute information with regard to content information, and for setting this content attribute header into the expansion transport header or into the payload header.

[0017] The content attribute header need not be encrypted.

[0018] The unit (a) generates the encryption key based on an identifier that uniquely identifies a storage medium sent from the other end apparatus in the communication.

[0019] Another aspect of the present invention provides a content information receiving apparatus authenticated by an authentication process including at least one procedure of an authentication procedure and a key exchange procedure and which receives encrypted content information via a network in accordance with a prescribed transport protocol, comprising: (aa) a unit for receiving from a sending apparatus a packet containing a basic transport header, an encryption attribute header including attribute information with regard to the encryption of the content information, and encrypted content information; (bb) a unit for referring to the basic transport header or encryption attribute header and judging whether or not the content information is encrypted or whether there is a possibility that the content information is encrypted; and (cc) a unit that, when a judgment is made by the unit (bb) that the content information is encrypted, decrypts the encrypted content information, based on the attribute information with regard to encryption included in the encryption attribute header.

[0020] It is preferable that the unit (bb), when there is a possibility that the content information is encrypted, refers to the encryption attribute header and judges whether or not the content information is encrypted.

[0021] It is preferable that the unit (bb) refers to the basic transport header or to the encryption attribute header to make a judgment as to the encoding system of the content information.

[0022] The above-noted apparatus can further comprising: (dd) a unit for referring to a received basic transport header and, when a prescribed delay time has elapsed or a prescribed number of packets have been discarded, requesting that the sending apparatus send a prescribed encryption parameter.

[0023] Another aspect of the present invention provides a method of distributing encrypted content information, via a network in accordance with a prescribed transport protocol, to other end apparatus in a communication authenticated by an authentication process including at least one procedure of an authentication procedure and a key exchange procedure, comprising the steps of: (a) encrypting content information encoded by a prescribed encoding system; (b) adding an encryption attribute header including attribute information with regard to the encryption of the content information to the encrypted content information; (c) adding a content attribute header indicating attributes of the content information to content information to which the encryption attribute header has been added; (d) performing transport protocol processing required to transfer the content information, and adding a basic transport header to content information to which the content attribute header has been added; and (e) sending a packet including the basic transport header, the encryption attribute header, the content attribute header, and the encrypted content information to the other end authenticated apparatus, wherein the encryption attribute header is set into either an expansion transport header within a packet header of the packet, or into a payload header within an encrypted payload of the packet.

[0024] Another aspect of the present invention provides a method of distributing encrypted content information, via a network in accordance with a prescribed transport protocol, to other end apparatus in the communication authenticated by an authentication process including at least one procedure of an authentication procedure and a key exchange procedure, comprising the steps of: (a') adding a content attribute header indicating attributes of the content information to the content information to be transferred; (b') encrypting content information that are encoded by a prescribed encoding system and to which the content attribute header has been added; (c') adding to the encrypted content information an encryption attribute header including attribution information with regard to the encryption of the content information; (d') performing transport protocol processing required to transfer the content information, and adding a basic transport header to content information to which the encryption attribute header has been added; and (e') sending a packet including the basic transport header, the encryption attribute header, the content attribute header, and the encrypted content information to the other end authenticated apparatus, wherein the encryption attribute header is set into either an expansion transport header within a packet header of the packet, or into a payload header within a payload

to be encrypted of the packet.

[0025] Another aspect of the present invention provides a method of receiving encrypted content information, via a network in accordance with a prescribed transport protocol, by an authentication process including at least one procedure of an authentication procedure and a key exchange procedure, comprising the steps of: (aa) receiving a packet including a basic transport header, an encryption attribute header including encryption attribute information with regard to the encryption of the content information, and encrypted content information; (bb) referring to the basic transport header and judging whether or not the content information is encrypted or whether or not there is a possibility that the content information is encrypted; (cc) referring to the encryption attribute header and extracting encryption attribute information with regard to encryption of the content information; (dd) referring to an expansion transport header within a packet header of the packet and extracting content attribute information with regard to the content information; and (ee) in the case in which a judgment is made at (bb) that the content information is encrypted, decrypting the encrypted content information, based on the extracted encryption attribute information.

[0026] Another aspect of the present invention provides a method of receiving encrypted content information, via a network in accordance with a prescribed transport protocol, by a authentication process including at least one procedure of an authentication procedure and a key exchange procedure, comprising the steps of: (aa') receiving a packet including a basic transport header, an encryption attribute header including encryption attribute information with regard to the encryption of the content information, and encrypted content information; (bb') referring to the basic transport header and judging whether or not the content information is encrypted or whether or not there is a possibility that the content information is encrypted; (cc') in the case in which a judgment is made at (bb') that the content information is encrypted, referring to the encryption attribute header and extracting encryption attribute information with regard to the encryption of the content information; (dd') in the case in which a judgment is made at (bb') that the content information is encrypted, decrypting the encrypted content information based on the extracted encryption attribute information; and (ee') referring to an expansion transport header within a packet header of the packet and extracting content attribute information with regard to the content information.

[0027] Another aspect of the present invention provides a computer-readable recording medium for recording a program to be executed by a computer, the program performing distribution of encrypted content information, via a network in accordance with a prescribed transport protocol, to other end apparatus in a communication authenticated by an authentication process including at least one procedure of an authentication procedure and a key exchange procedure, the program comprising: (a) a module for generating an encryption attribute header including attribute information with regard to encryption of the content information; (b) a module for performing transport protocol processing required to transfer the content information and for generating a basic transport header to be added to the content information to which the encryption attribute header has been added; and (c) a module for sending a packet including the basic transport header, the encryption attribute header, and the encrypted content information to the other end authenticated apparatus, wherein the encryption attribute header is set either into an expansion transport header within a packet header of the packet or into a payload header within a payload to be encrypted of the packet.

[0028] Another aspect of the present invention provides a computer-readable recording medium for recording a program to be executed by a computer, the program performing receiving of encrypted content information, via a network in accordance with a prescribed transport protocol, by an authentication process including at least one procedure of an authentication procedure and a key exchange procedure, the program comprising: (aa) a module for receiving from a sending apparatus a packet including a basic transport header, an encryption attribute header including attribute information with regard to encryption of the content information, and encrypted content information; (bb) referring to the basic transport header or the encryption attribute header and judging whether or not the content information is encrypted or whether there is a possibility that the content information is encrypted; and (cc) in the case in which a judgment is made by module (bb) that the content information is encrypted, decrypting the encrypted content information based on attribute information with regard to encryption included in the encryption attribute header.

[0029] Other features and advantages of the present invention will become apparent from the following description taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0030] The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate presently preferred embodiments of the invention, and together with the general description given above and the detailed description of the preferred embodiments given below, serve to explain the principles of the invention, wherein:

Fig. 1 is a drawing showing an example of a configuration of a network associated with the first embodiment to the sixth embodiment of the present invention;

Fig. 2 is a diagram showing an example of a sequence of content distribution in the first embodi-

ment to the sixth embodiment of the present invention;

Fig. 3 is a block diagram showing an example of the configuration of an MPEG distribution server according to the first embodiment of the present invention;

Fig. 4 is a flowchart showing the procedure for content distribution processing in an MPEG distribution server according to the first embodiment of the present invention;

Fig. 5 is a drawing showing a first example of a format of a code expansion header;

Fig. 6 is a drawing showing a first example of the format of a header given by an RTP processor;

Fig. 7 is a drawing showing a first example of a format of a transferred IP packet;

Fig. 8 is a block diagram showing an example of the configuration of a receiving apparatus according to the first embodiment of the present invention;

Fig. 9 is a flowchart showing the procedure for content receiving processing in a receiving apparatus according to the first embodiment of the present invention;

Fig. 10 is a block diagram showing an example of the configuration of an MPEG distribution server according to the second, third, or fifth embodiment of the present invention;

Fig. 11 is a drawing showing a second example of a format of a header given by an RTP processor;

Fig. 12 is a drawing showing the second example of the format of a transferred IP packet;

Fig. 13 is a flowchart showing the procedure for content distribution processing in an MPEG distribution server according to the second embodiment of the present invention;

Fig. 14 is a block diagram showing an example of the configuration of a receiving apparatus according to the second, third, or fifth embodiment of the present invention;

Fig. 15 is a flowchart showing the procedure for content receiving processing in a receiving apparatus according to the second embodiment of the present invention;

Fig. 16 is a drawing showing a third example of a format of a header given by an RTP processor;

Fig. 17 is a drawing showing a third example of a format of a transferred IP packet;

Fig. 18 is a block diagram showing an example of the configuration of an MPEG distribution server according to the fourth or sixth embodiment of the present invention;

Fig. 19 is a drawing showing a fourth example of a format of a header given by an RTP processor;

Fig. 20 is a drawing showing a fourth example of a format of a transferred IP packet;

Fig. 21 is a flowchart showing a procedure for content distribution processing in an MPEG distribution server according to the fourth embodiment of the present invention;

Fig. 22 is a block diagram showing an example of the configuration of a receiving apparatus according to the fourth or sixth embodiment of the present invention;

Fig. 23 is a flowchart showing a procedure for content receiving processing in a receiving apparatus according to the fourth embodiment of the present invention;

Fig. 24 is a drawing showing a fifth example of a format of a header given by an RTP processor;

Fig. 25 is a drawing showing a second example of a format of a code expansion header;

Fig. 26 is a drawing showing a fifth example of a format of a transferred IP packet;

Fig. 27 is a drawing showing a sixth example of a format of a header given by an RTP processor;

Fig. 28 is a drawing showing a sixth example of a format of a transferred IP packet;

Fig. 29 is a drawing showing an example of the configuration of a network according to the seventh embodiment of the present invention;

Fig. 30 is a drawing showing an example of the sequence of content distribution according to the seventh embodiment of the present invention;

Fig. 31 is a block diagram showing an example of the configuration of an MPEG distribution server according to the seventh embodiment of the present invention;

Fig. 32 is a drawing showing a seventh example of a format of a transferred IP packet;

Fig. 33 is a drawing showing a seventh example of a format of a header given by an RPT processor;

Fig. 34 is a block diagram showing an example of the configuration of a receiving apparatus according to the seventh embodiment of the present invention; and

Fig. 35 is a drawing showing an example of the sequence of content distribution according to the eighth embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0031] Embodiments of a content distribution apparatus, a content receiving apparatus, and a content distribution method according to the present invention are described in detail below, with reference to Fig. 1 through Fig. 35.

First Embodiment

[0032] The first embodiment of a content distribution apparatus, a content receiving apparatus, and a content distribution method according to the present invention is described in detail below, with reference to Fig. 1 through Fig. 9.

[0033] Fig. 1 shows an example of the configuration

of a content distribution system according to this embodiment of the present invention. In Fig. 1, an MPEG4 distribution server 101 and a receiving apparatus 102 according to this embodiment are connected to the Internet 103, MPEG4 AV stream data being securely communicated between the MPEG4 distribution server 101 and the receiving apparatus 102, via the Internet 103. Of course, other MPEG4 distribution servers and receiving apparatuses and other types of equipment can additionally be connected to the Internet 103.

[0034] In the description of embodiments to follow, while the type of data is MPEG (Motion Picture Experts Group) 4, it will be understood that the present invention is not restricted to application to this type of data, and can be applied to other data types as well.

[0035] The MPEG4 distribution server 101 performs distribution of MPEG4 data to the receiving apparatus 102. MPEG4 data is distributed not in the form of file transfer, but rather as data stream. The MPEG4 data that is to be copyright protected is distributed in encrypted form. When doing this, an authentication procedure or key exchange procedure is performed between the MPEG4 distribution server 101 and the receiving apparatus 102.

[0036] The sequence of this procedure is illustrated by example in Fig. 2.

[0037] Fig. 2 shows the sequence of content layer encryption and authentication, and it should be noted that security in layers such as the IP layer and transport layer and authentication procedures in those layers have been omitted from this drawing, as has the procedure for assessing charges at the content layer, which is performed earlier (although there are cases in which charge assessment and authentication/encryption at other layers are not performed).

[0038] Consider the case in which the receiving apparatus 102 makes a request to the MPEG4 distribution server 101 for distribution. In this case, the first authentication request is sent from the receiving apparatus 102 (S201). In this authentication request, there can also be a simultaneous exchange of a certificate (equipment certification) that is received by the apparatus (receiving apparatus 102) from a certification organization, the certificate certifying that the equipment is capable of performing transfer of copyright protected content.

[0039] The "equipment ID" that is used in the equipment certification can be an IP address, and in the case in which the IP address is given by a DHCP (Dynamic Host Configuration Protocol) server, there is a possibility that this value will differ each time the apparatus is booted. Because of this situation, it is possible to use as the "equipment ID" for equipment certification the MAC address of the equipment, or the EUI64 address, or an address created by adding to these address a partial module number. It is further possible to use as the "equipment ID" the CPU ID number of the apparatus, or the MPEG4 decoder ID number or the like, which (ideally) is a value that is unique worldwide (or that can be expected to be unique or almost unique within the region).

[0040] An MPEG4 distribution server 101 that receives a message from the receiving apparatus 101, performs a response to the authentication request and performs exchange of a certificate (equipment certification) (S202).

[0041] Next, the MPEG4 distribution server 101 and the receiving apparatus 102 perform a process that generates an authentication key, for the purpose of generating a common authentication key (S203). Details of this procedure can be the same as, for example, the IEEE 1394 copy protection key generation process. When this process is completed, the MPEG4 distribution server 101 and the receiving apparatus 102 can possess a common authentication key Kauth, which is not knowable to a third party.

[0042] Next, the MPEG4 distribution server 101 sends G(Kx, Kauth) which is generated by certain function G with the use of an exchange key Kx, the authentication key Kauth as arguments, and a random number Nc to the receiving apparatus 102 (S204, S205). At the receiving apparatus 102, reverse function of G(Kx, Kauth) is calculated so as to extract the exchange key Kx.

[0043] At this point in time, the MPEG4 distribution server 101 and the receiving apparatus 102 share the three values of authentication key Kauth, exchange key Kx, and the random number Nc.

[0044] At this point, the encryption key (content key) Kc, which is the encryption key for encrypting the MPEG4 data to be sent by the MPEG4 distribution server 101 and for the receiving apparatus 102 to decrypt the received encrypted MPEG4 data (that is the shared key), is calculated in the MPEG4 distribution server 101 and the receiving apparatus 102, respectively, by using one and the same pre-established function J, as a function of part of the above-noted value. For example, the calculation is made as Kc=J[Kx, f(EMI), Nc], in which EMI indicates the copy attribute for the data (content), which expresses such attributes as the data being copiable without limit, the data being copiable only 1 time, the data being copiable only 2 times, the data being uncopiable under any conditions, or the data being already copied and therefore not further copiable. f(EMI) is obtained by transforming the attribute value of EMI with the use of certain specific function f. These functions J and f can also be kept maintained as secret with respect to the outside.

[0045] After the encryption key (content key) Kc is generated, the MPEG4 distribution server 101 encrypts the content (MPEG4 data) using the encryption key Kc, and sends the encrypted content to the Internet (S206, S207, ...).

[0046] As will be described below, because the encrypted content is in the form of AV stream data transferred in real time over the Internet, in the first embodiment of the present invention RTP (Real-time Transport Protocol) is used as the transport protocol.

[0047] It is also possible to make the encryption key Kc vary with time (that is, have its value change with the passage of time).

[0048] For example, if the elapse of a prescribed amount of time (which can be a fixed amount of time, or a variable amount of time) from the previous change is recognized, the value of the variable Nc is incremented, and the above-noted function J is used to calculate the encryption key Kc. When this is done, the timing of updating of the encryption key Kc value (or the data at the point at which the encryption key Kc is to be updated) must be recognized synchronously at the sending and receiving sides. For this reason, regions such as Even/Odd field is provided in the transferred MPEG4 data (AV data), and the point at which there is a change in the field is established as the point at which the value of the variable Nc, that is, the value of the encryption key Kc is changed, so that data after the field change point is encrypted with the updated encryption key Kc.

[0049] The MPEG4 distribution server 101 monitors the above-noted elapse of time and, when the timing for the updating of the encryption key Kc is detected, the value of the variable Nc is incremented and the encryption key Kc is calculated again, the recalculated value of the encryption key Kc being used to encrypt the MPEG4 data that is to be sent, the Even/Odd field value being incremented, and transmission being performed. Thereafter, until the timing for the next updating, this updated encryption key Kc is used to perform encryption.

[0050] At the receiving apparatus 102, the received Even/Odd field value is monitored, comparing the value with the immediately previously received value and, if the value is detected as being different from the immediately previously received value, the value of the variable Nc is incremented, and the value of encryption key Kc is recalculated, the encryption key Kc after this recalculation being used in decrypting received encrypted data. Thereafter, until the next change in the Even/Odd field value is detected, this value of encryption key Kc is used to perform decryption.

[0051] In this manner, encrypted MPEG4 data is transferred between the MPEG4 distribution server 101 and the receiving apparatus 102.

[0052] Fig. 3 shows an example of the internal configuration of the MPEG4 distribution server 101.

[0053] As shown in Fig. 3, the MPEG4 distribution server 101 of this embodiment comprises a MPEG data generator 301, a data encryptor 302, an RTP processor 303, an RTCP transmitter 307, a TCP/IP and UDP/IP processor 308, a link/physical layer processor 309, an RCTP receiver/interpreter 310, and an authentication/key exchange processor 311. The RTP processor 303 includes an encryption expansion header adding unit 304, an MPEG4 expansion header adding unit 305, and an RTP basic header adding unit 306, and performs processing related to the RTP.

[0054] Next, the procedure for processing the content distribution in the MPEG4 distribution server 101 according to the first embodiment is described below, with reference to Fig. 4.

[0055] Processing related to authentication and encryption in the sequence of Fig. 2 (processing from S201 to S205) and processing related to the above-described updating of the encryption key is performed by the authentication/key exchange processor 311. This processing can be performed before or after the sending of content to the receiving apparatus 102.

[0056] The inputted AV information (for example, an analog signal) is compressed to MPEG4 data by the MPEG4 data generator 301.

[0057] When this is done, if MPEG attribute information such as the location of I pictures (intra-coded pictures) and the encoding rate are sent simultaneously with the MPEG4 data to notify the receiving side, playback (decoding) at the receiving side is facilitated. In particular on the Internet, where such things as discarded and delayed packets and a change in the sequence of arrival of packets can occur, this attribute information is essential to achieve high-quality playback at the receiving side. For example, in the case of MPEG4 in the first embodiment, information about, for example, the VOP header corresponds to these attributes. Cases can be envisioned in which information with regard to the MPEG4 system, for example transmission of synchronization information at a sync layer, information for multiplexing when sending a plurality of MPEG4 streams in multiplexed format, or information with regard to initial and latest values of object descriptors is required. For this reason, when sending AV data by RTP, the above-noted information is coded into the RTP expansion header or into the payload header of the RTP payload (that is, user area), so that the MPEG attribute information is sent along with the AV data.

[0058] In the first embodiment, this MPEG attribute information is sent in the form of an RTP expansion header. That is, it is sent as an RTP expansion header of the ID type of MPEG4 expansion header. For this reason, required information with regard to encoding is sent from the MPEG4 data generator 301 as notification to the MPEG4 expansion header adding unit 305.

[0059] Next, the MPEG4 data outputted from the MPEG4 data generator 301 is encrypted by the data encryptor 302 (step S401 in Fig. 4). When this is done, the encryption key may be the above-noted time-variant encryption key Kc. With regard to the encryption processing as well, a variety of attribute information can be envisioned. In the first embodiment, an RTP expansion header whose ID type indicates the encryption expansion header is added by the encryption expansion header adding unit 304 (step S403). For this reason, information required for the generation of an encryption expansion header is sent as notification from the data encrypter 302 to the encryption expansion header adding unit 304.

[0060] In order to perform the above-noted encryption processing, at the authentication/key exchange proces-

sor 311 when the timing for updating the encryption key Kc is reached, Nc is incremented and the above-described function J is used to generate a new encryption key Kc which is passed to the data encrypter 302. Along with this, the value of Even/Odd field is incremented and passed to the data encrypter 302. The value Even/Odd field is passed, as noted above, from the data encrypter 302 to the encryption expansion header adding unit 304.

[0061] Fig. 5 shows an example of an encryption expansion header. The encryption expansion header has a expansion header type field, a encryption on/off field, encryption type indication field, an encryption mode indicator (EMI) field, and an Even/Odd field.

[0062] The expansion header type field is for coding information that indicates the type of corresponding expansion header. In this case, information that indicates the encryption expansion header is coded into the expansion header type field.

[0063] The encryption on/off field is a field for coding information indicating whether or not the data transferred in the corresponding RTP packet is encrypted.

[0064] The encryption type indicator field is a field for coding the type of encryption used with respect to data transferred in an RTP packet. For example, in Fig. 5, information is described that indicates "an M6 encryption type".

[0065] The encryption mode indicator (EMI) field is a field for coding the above-noted copy attribute value EMI.

[0066] The Even/Odd field, as noted above, is a field for notifying the receiving side from the sending side of the timing of updating of the encryption key.

[0067] Although in the example shown each field has 8 bits, there is no restriction to the number of bits, and the number of bits can be appropriately established as desired.

[0068] When AV data is encrypted and sent to the receiving side, if it is desired to perform such trick play as fast-forwarding, or sending of partial static images, there are cases in which processing at the receiving side becomes difficult. This is because it is difficult to perform a task such as sending just a part of an encrypted AV data stream (for example, because the Nc value would not be incremented but would rather skip over a number of values). For this reason, in certain cases it is desirable to send AV data to the receiving side without encrypting it. In such cases, it is necessary to notify the receiving side by information as to whether or not the AV data has been encrypted. The above-noted encryption on/off field is provided for such purposes.

[0069] Additionally, on the Internet there is the possibility that one stream will use one encryption type and another stream use a different encryption type. In such cases, if there is a field that indicates to what type of encryption the AV data has been subjected, the receiving side can examine this field so as to select a proper decryption engine for describing the data. The above-noted encryption type indicator field is provided for this type of purpose.

[0070] As shown in Fig. 5, the encryption mode indicator (EMI) field is 8 bits rather than the 2 bits that are used in the case of IEEE 1394. This is in order to establish the freedom to select a value of N used to give notification of the specification of the number of copies N of the AV data that are to be permitted, and for cases in which a special type of copying (for example, permitting copying only when some condition is satisfied), in which case this field must be able to take a larger number of values than would be possible with 2 bits.

[0071] As shown in Fig. 5, in the Even/Odd field, in contrast to the 1 bit used in IEEE 1394, there are 8 bits provided. This is because 1 bit would not allow enough information for the Internet, in which as noted above it is possible to have discarded or delayed packets and an altered sequence of packet arrivals. Thus, for example, on the Internet a case can be envisioned in which, with the Even/Odd bit 1 as shown at step S207, all the packets are discarded. In this case, if the Even/Odd field were to be just 1 bit, the Even/Odd field would return to 0 at the next packet, so that as seen from the receiving side the condition in which the Even/Odd bit is 0 is continuing (that is, not changing). Thus, although Nc should actually be incremented by 2, if the Even/Odd bit value is not changed, the Nc value is not incremented, thereby preventing generation of the correct encryption key. Because of this situation, more than 2 bits, for example 8 bits, are provided in the Even/Odd field, so that even if packet discarding and delay, or resequencing of packet arrival occurs, as it can on the Internet, it is possible to perform appropriate processing on the receiving side.

[0072] In the first embodiment of the present invention, data encryption is performed only with respect to MPEG4 data itself, and not with respect to the MPEG4 expansion header. Because the MPEG4 expansion header is not content that need to be copyright protected, but is rather used on the receiving side before the MPEG4 data itself is used, enabling it to omitted from the data that is encrypted.

[0073] In the RTP processor 303, an encryption expansion header is added to the MPEG4 data by the encryption expansion header adding unit 304, an MPEG4 expansion header is added to the MPEG4 data by the MPEG4 expansion header adding unit 305 (step S405 in Fig. 4), and an RTP basic header is added to the MPEG4 data by the RTP basic header adding unit 306 (step S407), the ultimate result being the addition of the RTP header such as shown in Fig. 6. The encryption expansion header is generated based on information from the data encrypter 302, and the MPEG4 expansion header is generated based on information from the MPEG4 data generator 301. The RTP header has elements that are basic parameters required for AV data transfer via the Internet, such as a time stamp and a sequence number (for details, refer to RFC 1889).

[0074] Encrypted MPEG4 data to which the RTP header has been added is sent to the Internet 103 by

the TCP/IP and-UDP/IP processor 308 as an IP packet as shown in Fig. 7, via the link/physical layer processor 309.

[0075] Next, the configuration of the receiving apparatus 102 and the procedure for processing therein will be described.

[0076] Fig. 8 shows the internal configuration of the receiving apparatus 102.

[0077] As shown in Fig. 8, the receiving apparatus 102 according to the first embodiment comprises a link/physical layer processor 701, a TCP/IP and UDP/IP processor 702, an RTP processor 703, a data encryptor/decryptor 707, anMPEG4 data decoder 708, a receiving condition interpreter 709, an RTCP transmitter 710, and an authentication/key exchange processor 711. The RTP processor 703 includes an RTP basic header receiver/interpreter 704, an MPEG4 expansion header receiver/interpreter 705, and an encryption expansion header receiver/interpreter 706, and performs processing related to the RTP.

[0078] Fig. 9 shows a procedure for encrypted content receiving processing performed by the receiving apparatus 102 of the first embodiment of the present invention.

[0079] Processing related to authentication and encryption of the sequence of Fig. 2 (processing from S201 to S205) and processing related to encryption key updating is performed by the authentication/key exchange processor 711.

[0080] The receiving apparatus 102 basically performs processing in a sequence that is the reverse of the processing performed by the MPEG4 distribution server 101.

[0081] Specifically, MPEG4 data (encrypted data with an added RTP header) transferred via the Internet 103 passes through the link/physical layer processor 701 to the TCP/IP and UDP/IP processor 702 and is then inputted to the RTP processor 703 (step S901 in Fig. 9).

[0082] At the RTP processor 703, the RTP basic header is interpreted by the RTP basic header receiver/interpreter 704 (step S903 in Fig. 9), the MPEG4 expansion header is interpreted by the MPEG4 expansion header receiver/interpreter 705 (Step S905), and the encryption expansion header is interpreted by the encryption expansion header receiver/interpreter 706 (Step S907). Information required for decoding is sent as notification from the MPEG4 expansion header receiver/interpreter 705 to the MPEG4 data decoder 708, and information required for decryption is sent from the encryption expansion header receiver/interpreter 706 to the data encyptor/decryptor 707.

[0083] The encrypted data stored in the RTP payload is passed from the RTP processor 703 to the data encryptor/decryptor 707. The data encryptor/decryptor 707 performs decryption of data, based on information from the encryption expansion header receiver/interpreter 706.

[0084] The encryption key used in decryption is the above-described time-variant encryption key Kc. That is, the data encryptor/decryptor 707 refers to a value of the encryption on/off field of the encryption expansion header sent as notification from the encryption expansion receiver/interpreter 706, so as to learn that the received data is encrypted (as a result of which the decryption thereof is determined), after which the Even/Odd field is referenced and a comparison is made between the value thereof and a previously received value. If the value had been an increment, it is known that the encryption key is to be updated (but if the values are the same, the encryption key will not be updated). Then, the fact that the encryption key is to be updated is notified to the authentication/key exchange processor 711 from the data encryptor/decryptor 707, and in the authentication/key exchange processor 711 since the timing for updating the encryption key Kc has been reached, Nc is incremented and the above-noted function J is used to generate a new encryption key Kc, which is passed to the data encryptor/decryptor 707.

[0085] Decrypted MPEG4 data is passed from the data encryptor/decryptor 707 to the MPEG4 data decoder 708. The MPEG4 data decoder 708 decodes this MPEG4 data, based on information from the MPEG4 expansion header receiver/interpreter 705, and outputs the results as an AV output data (for example, an analog signal).

[0086] In the above, the RTP has associated with it RTCP (Real-time Transport Control Protocol). RTCP monitors the RTP sequence number and time stamp and has the function of notifying the sending side (in the first embodiment, the MPEG4 distribution server 101) from the receiving side (in the first embodiment, the receiving apparatus 102) with regard to the receiving condition (packet discarding rate, packet transmission delay time, and the like). This is performed by the receiving condition interpreter 709 and the RTCP transmitter 710.

[0087] The MPEG4 distribution server 101 receives this RTCP packet at the RTCP receiver/interpreter 310 and, if necessary, can apply feedback to the MPEG4 data generator 301, to attempt to achieve optimization. For example, in the case in which there is a great amount of packet discarding, network crowding can be envisioned, in response to which the bit rate of the MPEG4 data generation can be lowered by feedback.

[0088] The RTCP transmitter 307 of the MPEG4 distribution server 101 transmits information required for RTCP.

[0089] On the other hand, as described above, in the receiving apparatus 102 based on the results of monitoring the Even/Odd field included in the encryption expansion header of the received packet, the value of the variable Nc used in the calculation of the encrypt ion key Kc is changed. Therefore, if it is not possible for the sending side to reliably inform the receiving side that the value of Nc has been updated, the receiving side cannot calculate the encrypt ion key Kc, thereby making it impossible to decrypt the arriving encrypted data.

[0090] Because the Internet is intrinsically a network on which discarding of packets can occur, it is not guaranteed that the meaning of the Even/Odd field value (timing of incrementing) will be accurately informed to the other apparatus in a communication link (particularly in the case in which there are few bits in the Even/Odd field). Because of this situation, in the case in which the receiving apparatus 102 wishes to know the precise value of Nc, it can be provided with the option to issue a request to the sending apparatus (in this embodiment, the MPEG4 distribution server 101) for the Nc value.

[0091] As an example of a case in which the receiving apparatus 102 wishes to know the precise Nc value, consider the case in which there is more skipping than expected in the time stamp or the sequence number of the RTP basic header, in which case one solution envisionable is that of sending a packet to the sending side to request the value of Nc. This is because a skip greater than a pre-established limit could mean the possibility that the Even/Odd bit value has changed. This processing is performed by the authentication/key exchange processor 311 of the MPEG4 distribution server 101 or the authentication/key exchange processor 711 of the receiving apparatus 102. By doing this, even in the event that synchronization of the Even/Odd bit is lost between the MPEG4 distribution server 101 and the receiving apparatus 102, it is possible to perform appropriate recovery processing. Furthermore, in the case in which there is notification of Nc value from the MPEG4 distribution server 101 to the receiving apparatus 102. it is possible to simultaneously send the time stamps and sequence numbers of the corresponding RTP, expansion header, or payload header as notification.

[0092] A case can be envisioned in which distributed data is accumulated in the receiving apparatus (or in some form of storage medium, such as DVD-RAM or the like, installed in the receiving apparatus), in which case the distributed data can be stored as is in the form of encrypted data, and the corresponding encryption key Kc stored together therewith.

Second Embodiment

[0093] Next, the second embodiment, in which there is a variation of the packet format in the first embodiment, is described below, with reference to Fig. 10 through Fig. 15. Because the basic configuration and operation of the second embodiment is the same as that of the first embodiment, the description that follows focuses on the differences introduced with the second embodiment compared to the first embodiment.

[0094] The difference in the second embodiment with respect to the first is that, whereas in the first embodiment an encryption expansion header and MPEG4 expansion header were added as expansion header in an RTP header (Fig. 6 and Fig. 7), in the second embodiment the encryption expansion header is added as the expansion header in the RTP header and the MPEG4 expansion header is added as a payload header to the RTP payload (Fig. 11 and Fig. 12).

[0095] The overall network configuration according to the second embodiment is similar to that of the first embodiment (Fig. 1). The processing sequence is also similar to that of the first embodiment (Fig. 2). The encryption expansion header format is also similar to that of the first embodiment (Fig. 5).

[0096] Fig. 10 shows an example of the configuration of an MPEG4 distribution server 101 according to the second embodiment. In this embodiment, because the MPEG4 expansion header is provided as a payload header for the RTP payload, the processing for applying the MPEG4 header is removed from the RTP processing, so that the MPEG4 expansion header adding unit 305 of Fig. 3 is removed from within the RTP processor 305 and placed outside, this becoming the MPEG4 payload header adding unit 315, which is different than in the first embodiment.

[0097] Fig. 11 shows the format of the RTP header used when sending encrypted AV data in the second embodiment. In the second embodiment, a payload type field that indicates an attribute of data that is transferred by an RTP packet (for example, encoding system) is provided in the RTP basic header. In the second embodiment, for example, if the transferred data is encrypted MPEG4 data, this field will be coded with information indicating "the data is encrypted MPEG4 data". The receiving apparatus 102 can know that transferred data is encrypted MPEG4 data with reference to this field.

[0098] Additionally in the second embodiment, the RTP basic header is provided with an X bit field that indicates whether or not there is an expansion header added to the RTP header. In the second embodiment, the arrangement is that a bit indicating "the existence of an expansion header" is set.

[0099] Fig. 12 shows the overall format of an IP packet transferred over the Internet by the second embodiment.

[0100] Fig. 13 is a flowchart shows the procedure for processing of content distribution in the second embodiment of the present invention. In this second embodiment, first the MPEG4 payload header adding unit 315 adds an MPEG4 payload header to the content (step S400), the step S405 shown in Fig. 4 not being carried out. The processing of the other steps S401, S403, S407, and S409 is similar to the processing as described with regard to the first embodiment. However, the header has the above-noted information set into it.

[0101] Next, the receiving apparatus 102 according to the second embodiment is described below.

[0102] Fig. 14 shows an example of the configuration of the receiving apparatus 102 of the second embodiment. Similar to the above-noted MPEG4 distribution server 101, the processing of the MPEG4 expansion header is moved to outside of the RTP processor, the MPEG4 expansion header receiver/interpreter 705 of Fig. 8 being moved from inside the RTP processor 703

to outside, this becoming the MPEG4 payload header receiver/interpreter 715, which is different than in the first embodiment.

[0103] Fig. 15 is a flowchart showing the procedure for receiving processing in a receiving apparatus 102 according to the second embodiment.

[0104] At the receiving apparatus 102, first a packet is received (step S901), and at the RTP basic header receiver/interpreter 704 it is learned that the received data is encrypted MPEG4 data and that an expansion header has been added to the RTP header (step S903). Then, at the expansion header receiver/interpreter 706 it is learned that the expansion header is an encryption expansion header, and possible to learn from the encryption expansion header the encryption system and whether or not there is updating of the encryption key (step S907). Then, similar to the case of the first embodiment, at the data encryptor/decryptor 707 the encrypted MPEG4 data is decrypted (step S909), and at the MPEG payload header receiver/interpreter 715 the MPEG4 payload header is interpreted (step S910), and further, similar to the first embodiment, at the MPEG4 data generator 708, MPEG4 data is decoded, based on the results of the above interpreting, the results being outputted as an AV output data (for example, an analog signal).

[0105] In this second embodiment, in the case in which the payload type field is coded with information that includes notification of encryption, the encryption on/off field of the encryption expansion header need not be referenced, and in the case in which the payload type field is coded with information that includes notification of the encryption, this can be taken as a notification of the possibility of encryption, so that encryption on/off field in the encryption expansion header can be used for the final determination of whether or not there is encryption.

Third Embodiment

[0106] Next, the third embodiment of the present invention is described in detail below, with reference to Fig. 16 and Fig. 17. In this embodiment, the description will focus on differences with respect to the second embodiment.

[0107] The configuration and processing in the third embodiment are similar to those of the third embodiment.

[0108] Fig. 16 shows the format of the RTP header format used in transmitting encrypted AV data in the third embodiment, and Fig. 17 shows the overall IP packet format transferred via the Internet in the third embodiment.

[0109] Specifically, whereas in the second embodiment, in the payload type field within the RTP basic header information was coded that provides notification of the existence of encryption or the possibility of encryption, such as "encrypted MPEG4 data", in the third embodiment, only "MPEG4" is coded, and information including notification of encrypting or the possibility thereof is not coded in the payload type field.

[0110] In the third embodiment, therefore, while the receiving apparatus 102 can refer to the payload type field to know that the received data is MPEG4 data, recognition with regard to whether or not there is encryption is done by referencing the RTP expansion header (the encryption expansion header) of the RTP header.

[0111] In the receiving apparatus 102, at the RTP basic header receiver/interpreter 704 it is learned that the received data is MPEG4 data, and that an expansion header is added to the RTP header. Then, at the encryption expansion header receiver/interpreter 706, it is learned that the expansion header is an encryption expansion header, and it is possible to learn from the encryption expansion header whether or not there is encryption and whether or not the encryption key is updated. Thereafter, the processing is the same as in the second embodiment.

Fourth Embodiment

[0112] Next, the fourth embodiment of the present invention is described below with reference to Fig. 18 to Fig. 23, focusing on the difference between it and the second embodiment.

[0113] The difference between the fourth embodiment and the second embodiment is that, whereas in the second embodiment the encryption expansion header is added to the expansion header of the RTP header and the MPEG4 expansion header is provided as a payload header of the RTP payload (Fig. 11 and Fig. 12), in the fourth embodiment, both the encryption expansion header and the MPEG4 expansion header are provided as a payload header in the RTP payload (Fig. 19 and Fig. 20).

[0114] The overall configuration of a network according to the fourth embodiment is similar to that of an above-noted embodiment (Fig. 1), and the processing sequence is also similar to an above-noted embodiment (Fig. 2). The format of the encryption expansion header (encryption payload header in the fourth embodiment) is also the same as described above (Fig. 5).

[0115] Fig. 18 shows an example of the configuration of an MPEG4 distribution server 101 according to the fourth embodiment. In this embodiment, because the encryption expansion header added to the MPEG4 expansion header is also provided as a payload header in the RTP payload, the processing for the encryption expansion header is placed outside the RTP processor, and the encryption expansion header adding unit 304 of Fig. 10 is also moved from within the RTP processor 304 to outside, this serving as the encryption payload header adding unit 314, which is a difference with respect to the second embodiment.

[0116] Fig. 21 is a flowchart showing the procedure for content distribution processing according to the fourth embodiment. In the fourth embodiment, in place

of step S403 of Fig. 13, an encryption payload header adding unit 314 adds an encryption payload header to the encrypted MPEG4 data (step 403b). The processing at other steps S400, S401, S407, and S409 are similar to that described with regard to the above embodiment, with the exception that the above-noted information is set into the header.

[0117] Fig. 19 shows the format of the RTP header used in transmission of encrypted AV data in the fourth embodiment. With regard to the payload type field, the fourth embodiment is similar to the second embodiment. The function of the X bit field is similar to that in the second embodiment, and in this embodiment the arrangement is that a bit indicating "no expansion header (no RTP expansion header)" is set.

[0118] Fig. 20 shows the overall format of an IP packet transferred via the Internet in the fourth embodiment.

[0119] Fig. 22 shows an example of the configuration of a receiving apparatus 102 according to the fourth embodiment. Similar to the case of the above-noted MPEG4 distribution server 101, the processing of the encryption expansion header is moved to outside the RTP processor, and the encryption expansion header receiver /interpreter 706 of Fig. 11 is moved from within the RTP processor 703 to outside, this serving as the encryption payload header receiver/interpreter 716, which is a difference with respect to the second embodiment.

[0120] Fig. 23 shows the procedure for receiving processing in the receiving apparatus 102 in the fourth embodiment. In the receiving apparatus 102, first a packet is received (step S901), and at the RTP basic header receiver/interpreter 704 it is learned that the received data is encrypted MPEG4 data, and learned that no expansion header is added to the RTP header (step S903). In this embodiment, subsequent processing is processing for the payload. First at the encryption payload receiver/interpreter 716 it is learned that the payload is an encryption expansion header, and it is also possible to learn from the encryption payload header such information as the encryption system and whether the encryption key is updated (step S907b). Subsequent processing is similar to the case of the second embodiment, the encrypted MPEG4 data being decrypted at the data encryptor/decryptor 707 (step S909), the MPEG4 payload header being interpreted at the MPEG4 payload header receiver/interpreter 715 (step S910), the MPEG4 data being decoded at the MPEG4 data generator 708, based on the above interpretation results, and the results of this being output as an AV output data (for example, an analog signal).

[0121] Similar to the case of the second embodiment, in the fourth embodiment in a case in which information including notification of encryption is coded into the payload header, it is not necessary to refer to the encryption on/off field of the encryption payload header, and if information including notification of encryption is coded into the payload type field, this can be taken as a notification of the possibility of encryption so that the encryption on/off field of the encryption payload header can be used for the final determination of whether or not there is encryption.

Fifth Embodiment

[0122] Next, the fifth embodiment of the present invention is described below, with reference to Fig. 24 through Fig. 26, the description thereof focusing on the difference with respect to the second embodiment.

[0123] Fig. 24 shows the format of the RTP header used when transmitting encrypted AV data in the fifth embodiment. Fig. 25 shows the format of the encryption expansion header in the fifth embodiment, and Fig. 26 shows the overall IP packet transferred via the Internet in the fifth embodiment.

[0124] Whereas in the second embodiment information including notification of encrypted data attributes (for example, encoding system) such as "encrypted MPEG4" was coded into the payload type field within the RTP basic header (Fig. 11 and Fig. 12), in the fifth embodiment only information giving notification of the fact that the data is encrypted (for example, "encrypted data") is coded into the payload type header (Fig. 24 and Fig. 26). While the addition of an encryption expansion header as an expansion header to the RTP header, and the provision of an MPEG4 expansion header as a payload header to the RTP payload are the same as with the second embodiment, in the fifth embodiment the above-noted encrypted data attributes (encoding system or the like) are coded into the encryption expansion header (Fig. 5 and Fig. 25).

[0125] The overall configuration of the network according to the fifth embodiment is similar to that of an above-noted embodiment (Fig. 1), and the sequence of processing is also similar to an above-noted embodiment (Fig. 2). The internal configurations of the MPEG4 distribution server 101 and the receiving apparatus 102 are also similar to those of the second embodiment (Fig. 11 and Fig. 13).

[0126] As shown in Fig. 24, in the fifth embodiment a value indicating "encrypted data" is coded into the payload type field of the RTP basic header. The receiving apparatus 102 can refer this field to learn that the transferred data is encrypted. In the fifth embodiment, the X bit field has a bit that indicates "there is an expansion header".

[0127] As shown in Fig. 25, in the fifth embodiment, a payload type field is provided in the encryption expansion header. Information indicating the type of data (MPEG4 in this embodiment) in the payload is coded into the payload type field. The receiving apparatus 102 can refer to this field to learn the type of data transferred.

[0128] In the receiving apparatus 102, at the RTP basic header receiver/interpreter 704 it is learned that the received data is encrypted data, and that there is an expansion header added to the RTP header. Then, at the

encryption expansion header receiver/interpreter 706 is it learned that this expansion header is an encryption expansion header, and from the encryption expansion header it is possible to learn such information as the encryption system, whether or not the encryption key is updated, and the type of data in the payload. Similar to the case of the second embodiment, the encrypted MPEG4 data is decrypted at the data encryptor/decryptor 707, the MPEG4 payload header is interpreted at the MPEG4 payload header receiver/interpreter 715, at the MPEG4 data generator the MPEG4 data is decoded based on the results of the above interpretation, and the results are output as an AV output data (for example, an analog signal).

[0129] In the fifth embodiment, similar to the case of the second embodiment, in the case in which information including notification of encryption is coded into the payload type field, it is not necessary to refer to the encryption on/off field of the encryption expansion header, and if information including notification of encryption is coded into the payload type field in the RTP basic header, this can be taken as a notification of the possibility of encryption, so that the encryption on/off field of the encryption expansion header can be used for the final determination of whether or not there is encryption.

Sixth Embodiment

[0130] Next, the sixth embodiment of the present invention is described below with reference to Fig. 27 and Fig. 28, the description focusing on the difference with respect to the fourth embodiment.

[0131] Fig. 27 shows the format of the RTP header used when transmitting encrypted AV data in the sixth embodiment. The encryption expansion header of this embodiment is similar to that shown in Fig. 25. Fig. 28 shows the overall format of an IP packet transferred via the Internet in the sixth embodiment.

[0132] That both the encryption expansion header and the MPEG4 expansion header are provided as an RTP payload header is similar to the fourth embodiment (Fig. 19 and Fig. 20). However, in contrast to the fourth embodiment, wherein information including notification of attributes of encrypted data, such as the encoding system, for example, "encrypted MPEG4" are coded into the payload type field within the RTP basic header, in the sixth embodiment, only information giving notification about the existence of encryption (such as "encrypted data") is coded into the payload type field (Fig. 27 and Fig. 28). Additionally, while the fact that an encryption expansion header is added as an expansion header of the RTP header, and an MPEG4 expansion header is added as a payload header to the RTP payload are the same as in the second embodiment, in the sixth embodiment the above-noted encrypted data attributes (for example, encoding system) are coded within the encryption expansion header (Fig. 5 and Fig. 25)

[0133] The overall configuration of the network according to the sixth embodiment is similar to an above-noted embodiment (Fig. 1), and the sequence of processing is also similar to an above-noted embodiment (Fig. 2). The internal configuration of the MPEG4 distribution server 101 and the receiving apparatus 102 are also similar to those of the fourth embodiment (Fig. 18 and Fig. 22).

[0134] As shown in Fig. 27, in the sixth embodiment a value indicating "encrypted data" is entered into the payload type field of the RTP basic header. By referring to this field, the receiving apparatus 102 can know that the transferred data is encrypted data. The X bit field has a bit that indicates "there is no expansion header". Similar to the case of the fourth embodiment, information indicating the type of data in the payload (MPEG4 in this embodiment) is coded into the payload type field of the encryption expansion header.

[0135] In the receiving apparatus 102, at the RTP basic header receiver/interpreter 704 it is learned that the received data is encrypted, and it is learned that no expansion header is added to the RTP header. In the sixth embodiment, subsequent processing is with respect to the payload. First, at the encryption payload receiver/interpreter 716 it is learned that the payload header is an encryption expansion header, and it is possible to learn from the encryption payload head such information as the encryption system, whether the encryption key is updated, and type of data in the payload. In the same manner as in the fourth embodiment, at the data encryptor/decryptor 707 the encrypted MPEG4 data is decrypted, at the MPEG payload header receiver/interpreter 715 the MPEG4 payload header is interpreted, at the MPEG4 data generator 708 the MPEG4 data is decoded based on results of the above interpretation, the results being output as an AV output data (for example, an analog signal).

[0136] In the sixth embodiment, similar to the case of the fourth embodiment, in the case in which information including notification of encryption is coded into the payload type field of the RTP basic header, it is not necessary to refer to the encryption on/off field in the encryption expansion header, and if information including notification of encryption is coded into the payload type field, this can be taken as a notification of the possibility of encryption, so that the encryption on/off field of the encryption expansion header can be used for the final determination of whether or not there is encryption.

Seventh Embodiment

[0137] Whereas in the first embodiment to the sixth embodiment, the present invention was applied to a system in which RTP was used as a transport protocol, the present invention can also be applied to systems using other protocols.

[0138] In the seventh embodiment, instead of using RTP as the transport protocol, the distribution of MPEG4 data is performed using HTTP (Hyper-Text Transfer Pro-

tocol), that is, a protocol for use between WWW servers and Web browsers.

[0139] Fig. 29 shows an example of the configuration of an information distribution system in the seventh embodiment. In system shown in Fig. 29, an MPEG4 distribution server 6101 according to the seventh embodiment is connected to the Internet 103, and a receiving apparatus 6102 according to the seventh embodiment is connected to a LAN 6105, the LAN 6105 being connected to the Internet 103 via a proxy server 6104. The receiving apparatus 6102 performs AV stream communication secretly with the MPEG4 distribution server 6101, via the LAN 6105, the proxy server 6104 and the Internet 103. Of course, other MPEG4 distribution servers and other types of equipment can also be connected to the Internet 103, and other receiving apparatuses and other types of equipment can also be connected to the LAN 6105.

[0140] Although in the seventh embodiment, the description is that of the case in which the data type is MPEG4 it will be understood, of course, that the present invention is not restricted to this type of data.

[0141] In Fig. 29, the various equipment supports IP. However, because of the HTTP proxy server 6104 between the LAN 6105 and the Internet 103, the IP address on the LAN 6105 can be either a global IP address or a private (local) IP address. The term proxy server as used herein refers to a server that at one point terminates HTTP (or some other protocol) between the Internet and an intranet, and functions so as to join HTTP sessions at both ends of the proxy server, and is provided to enable the distribution of HTTP content data requested by substantial receiving apparatus (Web browser) to a distribution server (Web server), and functions in the reverse direction as well. Details about proxy servers can be found at, for example, the URL http://squid.nlanr.net/Squid. In the seventh embodiment, the MPEG4 distribution server 6101 can be a WWW server, and the receiving apparatus 6102 can also be a browser.

[0142] Fig. 30 shows an example of the sequence of the authentication process, key exchange process, and encrypted data transmission. Because the proxy server 6104 is disposed between the receiving apparatus 6102 and the MPEG4 distribution server 6101, the actual messages (messages transferred as HTTP messages) are in reality relayed via the proxy server 6104, this being the only difference with respect to previously described embodiments (Fig. 2), with other elements of the procedure being the same as previously described procedures.

[0143] With regard to the MPEG4 distribution server 6101, the receiving apparatus 6102, the packet format, if parts of the units in the first through the sixth embodiments dependent upon the transport protocol are modified to accommodate the HTTP protocol, it is possible to configure an MPEG4 distribution server 6101, a receiving apparatus 6102, and a packet format conforming to the HTTP protocol. In the following, the example is that in which an encryption expansion header is added as an expansion header, and in which an MPEG4 expansion header is provided as a payload header (as in the second embodiment).

[0144] Fig. 31 shows the internal configuration of the MPEG4 distribution server 6101.

[0145] As shown in Fig. 31, the MPEG4 distribution server 6101 according to the seventh embodiment comprises an MPEG4 data generator 6301, a data encrypter 6302, an MPEG4 payload header adding unit 6305, an HTTP processor 6303 that includes an encryption header adding unit 6304 and a MIME header adding unit 6306, a TCP/IP and UDP/IP processor 6308, a link/physical layer processor 6309, and an authentication/key exchange processor 6311.

[0146] Processing related to authentication and encryption of the sequence shown in Fig. 30 (from S6201 to S6205) and processing related to encryption key updating is performed by the authentication/key exchange processor 6311.

[0147] The HTTP processor 6303 corresponds to the RTP processor in previously described embodiments, and the MIME (Multipurpose Internet Mail Extensions) header adding unit 6306 corresponds to the RTP basic header adding unit in previously described embodiments.

[0148] Fig. 32 shows an IP packet that is transferred on the Internet (and LAN), and Fig. 33 shows details of the MIME basic header and encryption expansion header.

[0149] In the seventh embodiment, the encryption expansion header is transferred as part of the MIME. For this reason, with regard to the encryption expansion header, information indicating that "this is an encryption expansion header" is coded into the Content-Type of the MIME. The MPEG4 expansion header is transferred as part of the MIME, along with the encrypted MPEG4 data as payload header. For the encrypted MPEG4 data to which MPEG4 expansion header added, information indicating that "this is MPEG4 data" is coded into the MIME Content-Type. For details with regard to MIME, refer to RFC 2045, for example.

[0150] The format of the encryption expansion header is the same as was described for the first embodiment.

[0151] Fig. 34 shows an example of the internal configuration of the receiving apparatus 6102.

[0152] As shown in Fig. 34, the receiving apparatus 6102 according to the seventh embodiment comprises a link/physical layer processor 6701, a TCP/IP and UDP/IP processor 6702, an HTTP processor 6703 that includes a MIME header interpreter 6704 and an encryption header interpreter 6706, an MPEG4 payload header interpreter 6705, a data encryptor/decryptor 6707, an MPEG4 data decoder 6708, and an authentication/key exchange processor 6711.

[0153] Processing related to authentication and encryption in the sequence of Fig. 30 (processing from S6201 to S6205) and processing related to encryption

key updating is performed by the authentication/key exchange processor 6711.

[0154] The HTTP processor 6703 corresponds to the RTP processor shown in the above-described embodiments, and the MIME header interpreter 6704 corresponds to the RTP basic header receiver/interpreter in the above-described embodiments.

[0155] In the MPEG4 distribution server 6101, the inputted AV content (for example, an analog signal) is compressed to MPEG4 data by the MPEG4 data generator 6301. Required information is sent as notification to the MPEG4 payload header adding unit 6305 from the MPEG4 data generator 6301.

[0156] Next, the MPEG4 data outputted from the MPEG4 data generator 6301 is encrypted by the data encrypter 6302. The encryption key used when doing this is the above-described time-variant encryption key Kc. Required information is sent as notification to the encryption header adding unit 6304 from the data encrypter 6302.

[0157] Next, in the HTTP processor 6303, at the encryption header adding unit 6304, the encryption expansion header is added, and at the MIME header adding unit 6306, a MIME header is added.

[0158] Encrypted MPEG4 data to which has been added a MIME header is sent as a packet shown in Fig. 32 to the Internet 6103 by the TCP/IP and UDP/IP processor 6308, via the link/physical layer processor 6309.

[0159] In the receiving apparatus 6102, at the MIME header interpreter 6704, it is learned that there is a possibility that the received data is encrypted, and that an encryption expansion header is added as part of the MIME. At the encryption header interpreter 6706 it can be learned from the encryption expansion header whether or not encryption is present, the encryption system and whether the encryption key is updated. In the same manner as in the case of the second embodiment, the data encryptor/decryptor 6707 decrypts the encrypted MPEG4 data, at the MPEG4 payload header interpreting section 6715 the MPEG4 payload header (similar to the MPEG4 payload header in previously described embodiments) is interpreted, and at the MPEG4 data generator 6708 the MPEG4 data is decoded based on the results of the above interpretation, the result being output as an AV output data (for example, an analog signal).

[0160] In the above, the encryption expansion header is added as an expansion header and the MPEG4 expansion header was provided as a payload header on the payload. However, it is also possible to use a different configuration, for example one in which the encryption expansion header and MPEG4 expansion header are added as an expansion header, or in which the encryption expansion header and MPEG4 expansion header are provided as a payload header on the payload.

[0161] While in the first to seventh embodiments an Even/Odd field in the encryption expansion header (or encryption payload header) was used to notify the receiving side from the sending side of updating of the variable value Nc used for generating the encryption key Kc, instead of using the Even/Odd field, it is possible to send the value of Nc, in which case the Nc value can be randomly generated, as opposed to simply being incremented. The value of Nc can also be changed for each individual packet.

[0162] While in the first to seventh embodiments RTP or HTTP was used as the transfer protocol, it will be understood that other protocols can also be used, and further that the network to which the present invention is applied is not limited to the Internet. For example, any kinds of local area network, such as Bluetooth, can use the methods of this invention. Further, the present invention has no restriction in application to the case in which the transferred data is MPEG4 data.

[0163] Although in the second, third, and fifth embodiments, the data encryptor 302 and data encryptor/decryptor-707 were provided outside the RTP processors 303 and 307, these can alternately be provided within the RTP processors 303 and 307.

Eighth Embodiment

[0164] Next, the eighth embodiment of the present invention is described below with reference to Fig. 35.

[0165] Whereas in the first to the seventh embodiments, the description was for the sequence shown in Fig. 2, it is understood that the present invention can be applied to other sequences as well.

[0166] The description that follows is for other sequences, for the case in which MPEG4 data distributed from an MPEG4 distribution server is stored by the receiving apparatus.

[0167] The configuration of the information distribution system according to the eighth embodiment is similar to that shown in Fig. 1. In Fig. 1, an MPEG4 distribution server 101 and a receiving apparatus 102 according to the eighth embodiment are connected to the Internet 103, MPEG4 AV stream data being secretly communicated between the MPEG4 distribution server 101 and the receiving apparatus 102, via the Internet 103. Of course, other MPEG4 distribution servers and receiving apparatuses and other types of equipment can additionally be connected to the Internet 103.

[0168] In the description of the eighth embodiment, while the type of data is MPEG4, it will be understood that the eighth embodiment is not restricted to application to this type of data, and can be applied to other data types as well.

[0169] The MPEG4 distribution server 101 performs distribution of MPEG4 data to the receiving apparatus 102. MPEG4 data is distributed not in the form of file transfer, but rather as a stream. When this is done, the MPEG4 data that is to be copyright protected is distributed in encrypted form. Before distribution, an authentication procedure or key exchange procedure is per-

formed between the MPEG4 distribution server 101 and the receiving apparatus 102.

[0170] An example of the sequence used is shown in Fig. 35.

[0171] In Fig. 35 shows the sequence of content layer encryption and authentication, and it should be noted that security in layers such as the IP layer and transport layer and authentication procedures in those layers have been omitted from this drawing, as has the procedure for assessing charges at the content layer, which is performed earlier (although there are cases in which charge assessment and authentication/encryption at other layers are not performed).

[0172] Similar to the case of the first embodiment, the MPEG4 distribution server 101 and the receiving apparatus 102 perform authentication and exchange of a certificate (equipment certificate) (S7201, S7202).

[0173] The MPEG4 distribution server 101 must notify the receiving apparatus 102 of the encryption key Kc for decrypting the content (AV data that is sent), and the following measure is taken to prevent the unlimited unauthorized copying of the content at the receiving apparatus 102. Specifically, when performing storage onto a storage medium (for example, a DVD-RAM) of the receiving apparatus 102, AV data is stored in encrypted form. When the data stored on the storage medium is to be played back, a check is made as to whether the data was properly stored on the storage medium and, if not, playback is not possible. That is, if digital copying is done from this storage medium onto another storage medium (for example, onto another DV-RAM), playback is prevented from the copying destination medium.

[0174] For this reason, notification of the MID, which is the ID (serial number) of the storage medium used at the receiving apparatus 102 is given from the receiving apparatus 102 to the MPEG4 distribution server 101 (step S7203), and at the MPEG4 distribution server 101 this MID value is used to encrypt the encryption key Kc and notify the receiving apparatus 102 (step S7204). More specifically, using a pre-established function g an encryption key W is generated of the form W=g(MID), and this encryption key W is used to encrypt the encryption key Kc (the encryption key Kc encrypted by the encryption key W being represented as [Kc]w), [Kc]w being then transmitted. In this arrangement, the value of MID is unique to each individual storage medium, and is located in a region of ROM, for example, that cannot be overwritten.

[0175] Having received the above-noted [Kc]w, the receiving apparatus 102 generates the encryption key W in the form W=g(MID), using the same function g that was used at the MPEG4 distribution server 101, this encryption key W being used to decrypt [Kc]w so as to recover the encryption key Kc.

[0176] Thereafter, at the MPEG4 distribution server 101 MPEG4 data is generated from the AV data, and MPEG4 data is encrypted using the encryption key Kc shared as described above, the encrypted MPEG4 data being then transmitted to the receiving apparatus 102 (step S7206).

[0177] At the receiving apparatus 102, the received encrypted MPEG4 data is decrypted using the encryption key Kc determined as noted above and decrypted MPEG4 data is decoded, resulting in output of an AV output data.

[0178] In the eighth embodiment, the receiving apparatus 102 has a function which, simultaneously with receiving the AV data (MPEG4 data encrypted with the encryption key Kc), or after entering it into a buffer or the like, stores the received AV data in the form of MPEG4 data encrypted using the encryption key Kc, along with the value of [Kc]w, onto a storage medium having the above-noted MID.

[0179] When the above is done, an apparatus for playing back the AV data (MPEG4 data encrypted with the encryption key Kc) stored on the proper storage medium (which can be the receiving apparatus 102, or another equipment) first reads the values of [Kc]w and MID from the storage medium, and then generates the encryption key W in the form W=g(MID), this encryption key W being used to decrypt [Kc]w so as to recover the encryption key Kc. Then the AV data (MPEG4 data encrypted by the encryption key Kc) stored on the storage medium is read out and, after decrypting with the encryption key Kc, the decrypted MPEG4 data is decoded.

[0180] If AV data (MPEG4 data encrypted by the encryption key Kc) recorded on a storage medium having a given MID of MID1 is copied onto a storage medium having a MID of MID2, at the equipment that plays back the data recorded on the copy destination storage medium, because it is not possible to obtain the original MID value, it is not possible to generate W, and therefore not possible to determine the encryption key Kc from the recorded [Kc]w. As a result, it is not possible to decrypt the recorded encrypted data.

[0181] That is, if proper values of Kc, W, and [Kc]w are Kcl, W1=g(MID1), and [Kc]wl, because the MID read out from the copy destination storage medium is MID2, the encryption key W generated therefrom is W2=g(MID2) and if the [Kc]wl read from the storage medium is decrypted using W2, a value that is different from Kc results (this being called Kc'). Therefore, an attempt to decrypt data [Data]Kc encrypted with Kc using Kc' will result in generation of Data', which is different from the original data Data, making it impossible to recover the original data Data.

[0182] Thus, even if the received AV data (MPEG4 data encrypted by the encryption key Kc) is copied onto a different storage medium, because the value of MID of that storage medium is different, it is possible to prevent playback of the AV data, thereby enabling prevention of unauthorized copying.

[0183] In the eighth embodiment, the RTP header, encryption expansion (payload) header, and MPEG4 expansion (payload) header and the like can have the same format as described with regard to the first through

the seventh embodiments.

[0184] Although the above-noted description is for the case in which MPEG4 was used as the encoding system, it will be understood that the present invention can be applied to other encoding systems as well, in which case it is merely necessary to modify the constituent elements of each of the embodiments (for example, the MPEG4 data generator, the MPEG4 expansion header adding unit, the MPEG4 data decoder, the MPEG4 expansion header receiver/interpreter, and the like), the expansion header (MPEG4 expansion header and MPEG4 payload header), and the payload type coding to suit the selected type of encoding system.

[0185] The distribution server of the described embodiments, if necessary, can also be made to send content in unencrypted form. That is, the coding of the encryption on/off field and payload type field and the like can be established as appropriate to the existence or non-existence of encryption. The receiving side as well can check for the presence of encryption from the header of a received packet, and control decryption processing accordingly.

[0186] The transport protocol used in the foregoing embodiments includes at least RTP or HTTP.

[0187] The present invention can also be embodied as a method and a method according to the present invention can be embodied as an apparatus. Additionally, the functions of the present invention can be embodied as software as well.

[0188] The present invention embodied as either an apparatus or a method can be further embodied as a computer-readable storage medium for storage of a program to be executed by a computer, following a procedure corresponding to the present invention (or a program for causing a computer to function as ameans corresponding to the present invention or a program for implementing the functions of the present invention with a computer). That is, a program for the purpose of implementing the processing for content distribution and receiving according to the present invention can be stored onto various types of storage media. The storage medium is then read by the CPU of a computer implemented in hardware, and the stored program is then executed to embody the present invention. The term storage medium used herein can be taken as referring to semiconductor memory, magnetic disk (floppy disk or hard disk), optical disk (CD-ROM or DVD or the like), and any other medium that can be used to store a program. Additionally, the program can be distributed by various communications means, such as a network.

[0189] In summary, according to the present invention an encryption expansion header is provided in the form of a expansion header or payload header in a transport protocol such as RTP (Real-time Transport Protocol) or HTTP (Hyper-text Transfer Protocol), and encryption attribute information with regard to encryption is coded into this encryption expansion header (for example, presence of encryption, encryption system, information with regard to copying (encryption mode indicator: EMI), information (Even/Odd field) on which to base generation of a content key (common key) and the like), and by doing so content data is sent securely from the sending side to the receiving side, and it is possible at the receiving side to decrypt the encrypted content data transferred as a payload.

[0190] In RTP in the past, only the type of encoding system used with data of the payload was coded in the payload type field, so that in the case in which the data stored in the payload was encrypted not in at the network or transporter layer, but rather at the content layer, there was no method of giving notification of this to the other side.

[0191] On the other hand, with the present invention, because coding is provided in the RTP payload type field to the effect that the content is "encrypted data" or "encrypted data encoded by a specific encoding system", it is possible to give notification of this to the other side, thereby enabling sufficient copy protection in sending and receiving encrypted content as described above.

[0192] According to the present invention as described above, it is possible to expand the distribution of digital content, providing copy protection for AV streaming that covers not only IEEE 1394, but also networks such as the Internet and LAN.

[0193] It is to be noted that, besides those already mentioned above, many modifications and variations of the above embodiments may be made without departing from the noel and advantageous features of the present invention. Accordingly, all such modifications and variations are intended to be included within the scope of the appended claims.

Claims

1. A content information distribution apparatus for distributing encrypted content information, via a network in accordance with a prescribed transport protocol, to other end apparatus in a communication authenticated by an authentication process including at least one procedure of an authentication procedure and a key exchange procedure, comprising:

   (a) a unit (302) for encrypting content information encoded by a prescribed encoding system;
   (b) aunit (304) for generating an encryption attribute header including attribute information with regard to the encryption of the content information;
   (c) aunit (306) for performing transport protocol processing required to transfer the content information and for generating a basic transport header to be added to the content information to which the encryption attribute header has been added; and

(d) a unit (309) for sending to the other end apparatus that is authenticated a packet including the basic transport header. the encryption attribute header, and the encrypted content information, wherein the encryption attribute header is set into an expansion transport header within a packet header of the packet or into a payload header within a payload to be encrypted of the packet.

2. The apparatus according to claim 1, wherein the encryption attribute header includes at least one of the existence or non-existence of encryption of the content information and the encryption system of the content information.

3. The apparatus according to claim 1, wherein the encryption attribute header includes a copy attribute field having a plurality of bits with regard to the number of copying of the content information.

4. The apparatus according to claim 1, wherein the encryption attribute header includes a counter field indicating a change in an encryption key.

5. The apparatus according to claim 1, wherein the unit (b) sets the encoding information, which indicates the encoding system for the content information into the expansion transport header or into the payload header.

6. The apparatus according to claim 1, wherein the unit (c) further codes into the basic transport header at least information indicating that there is a possibility that the content information is encrypted, and wherein the unit (b) codes into the expansion header at least information as to whether or not the content information to be transferred is encrypted.

7. The apparatus according to claim 1, wherein the unit (b) codes into the expansion header information as to whether or not the content information to be transferred is encrypted.

8. The apparatus according to claim 1, further comprising:

(e) a (305) unit for generating a content attribute header that includes content attribute information with regard to content information, and for setting this content attribute header into the expansion transport header or into the payload header.

9. The apparatus according to claim 8, wherein the content attribute header is not encrypted.

10. The apparatus according to claim 1, wherein the unit (a) generates the encryption key based on an identifier that uniquely identifies a storage medium sent from the other end apparatus in a communication.

11. A content information receiving apparatus authenticated by an authentication process including at least one procedure of an authentication procedure and a key exchange procedure and which receives encrypted content information via a network in accordance with a prescribed transport protocol, comprising:

(aa) a unit (701) for receiving from a sending apparatus a packet containing a basic transport header, an encryption attribute header including attribute information with regard to the encryption of the content information, and encrypted content information;
(bb) a unit (703) for referring to the basic transport header or encryption attribute header and judging whether or not the content information is encrypted or whether there is a possibility that the content information is encrypted; and
(cc) a unit (707) that, when a judgment is made by the unit (bb) that the content information is encrypted, decrypts the encrypted content information, based on the attribute information with regard to encryption included in the encryption attribute header.

12. The apparatus according to claim 11, wherein the unit (bb), when there is a possibility that the content information is encrypted, refers to the encryption attribute header and judges whether or not the content information is encrypted.

13. The apparatus according to claim 11, wherein the unit (bb) refers to the basic transport header or to the encryption attribute header to make a judgment as to the encoding system of the content information.

14. The apparatus according to claim 11, further comprising:

(dd) a unit (709,710),for referring to a received basic transport header and, when a prescribed delay time has elapsed or a prescribed number of packets have been discarded, requesting that the sending apparatus send a prescribed encryption parameter.

15. A method of distributing encrypted content information, via a network in accordance with a prescribed transport protocol, to other end apparatus in a communication authenticated by an authentication process including at least one procedure of an authentication procedure and a key exchange procedure, comprising the steps of:

(a) encrypting content information encoded by a prescribed encoding system (S401);
(b) adding an encryption attribute header including attribute information with regard to the encryption of the content information to the encrypted content information (S403);
(c) adding a content attribute header indicating attributes of the content information to content information to which the encryption attribute header has been added (S405);
(d) performing transport protocol processing required to transfer the content information, and adding a basic transport header to content information to which the content attribute header has been added (S407); and
(e) sending a packet including the basic transport header, the encryption attribute header, the content attribute header, and the encrypted content information to the other end authenticated apparatus (S409),

wherein the encryption attribute header is set into either an expansion transport header within a packet header of the packet, or into a payload header within an encrypted payload of the packet.

16. A method of distributing encrypted content information, via a network in accordance with a prescribed transport protocol, to other end apparatus in a communication authenticated by an authentication process including at least one procedure of an authentication procedure and a key exchange procedure, comprising the steps of:

(a') adding a content attribute header indicating attributes of the content information to the content information to be transferred (S400);
(b') encrypting content information that are encoded by a prescribed encoding system and to which the content attribute header has been added (S401);
(c') adding to the encrypted content information an encryption attribute header including attribution information with regard to the encryption of the content information (S403);
(d') performing transport protocol processing required to transfer the content information, and adding a basic transport header to content information to which the encryption attribute header has been added (S407); and
(e') sending a packet including the basic transport header, the encryption attribute header, the content attribute header, and the encrypted content information to the other end authenticated apparatus (S409),

wherein the encryption attribute header is set into either an expansion transport header within a packet header of the packet, or into a payload header within a payload to be encrypted of the packet.

17. A method of receiving encrypted content information, via a network in accordance with a prescribed transport protocol, by an authentication process including at least one procedure of an authentication procedure and a key exchange procedure, comprising the steps of:

(aa) receiving a packet including a basic transport header, an encryption attribute header including encryption attribute information with regard to the encryption of the content information, and encrypted content information (S901);
(bb) referring to the basic transport header and judging whether or not the content information is encrypted or whether or not there is a possibility that the content information is encrypted (S903);
(cc) referring to the encryption attribute header and extracting encryption attribute information with regard to encryption of the content information (S907);
(dd) referring to an expansion transport header within a packet header of the packet and extracting content attribute information with regard to the content information (S905); and
(ee) in the case in which a judgment is made at (bb) that the content information is encrypted, decrypting the encrypted content information, based on the extracted encryption attribute information (S909).

18. A method of receiving encrypted content information, via a network in accordance with a prescribed transport protocol, by an authentication process including at least one procedure of an authentication procedure and a key exchange procedure, comprising the steps of:

(aa') receiving a packet including a basic transport header, an encryption attribute header including encryption attribute information with regard to the encryption of the content information, and encrypted content information (S901);
(bb') referring to the basic transport header and judging whether or not the content information is encrypted or whether or not there is a possibility that the content information is encrypted (S903);
(cc') in the case in which a judgment is made at (bb') that the content information is encrypted, referring to the encryption attribute header and extracting encryption attribute information with regard to the encryption of the content information (S907);
(dd') in the case in which a judgment is made

at (bb') that the content information is encrypted, decrypting the encrypted content information based on the extracted encryption attribute information (S909): and
(ee') referring to an expansion transport header within a packet header of the packet and extracting content attribute information with regard to the content information (S910).

19. A computer-readable recording medium for recording a program to be executed by a computer, the program performing distribution of encrypted content information, via a network in accordance with a prescribed transport protocol, to other end apparatus in a communication authenticated by an authentication process including at least one procedure of an authentication procedure and a key exchange procedure, the program comprising:

(a) a module (304) for generating an encryption attribute header including attribute information with regard to encryption of the content information;
(b) a module (306) for performing transport protocol processing required to transfer the content information and for generating a basic transport header to be added to the content information to which the encryption attribute header has been added; and
(c) a module (309) for sending a packet including the basic transport header, the encryption attribute header, and the encrypted content information to the other end authenticated apparatus,

wherein the encryption attribute header is set either into an expansion transport header within a packet header of the packet or into a payload header within a payload to be encrypted of the packet.

20. A computer-readable recording medium for recording a program to be executed by a computer, the program performing receiving of encrypted content information, via a network in accordance with a prescribed transport protocol, by an authentication process including at least one procedure of an authentication procedure and a key exchange procedure, the program comprising:

(aa) a module (701) for receiving from a sending apparatus a packet including a basic transport header, an encrypt ion attribute header including attribute information with regard to encryption of the content information, and encrypted content information;
(bb) a module (703) for referring to the basic transport header or the encryption attribute header and judging whether or not the content information is encrypted or whether there is a possibility that the content information is encrypted: and
(cc) a module (707) for decrypting the encrypted content information based on attribute information with regard to encryption included in the encryption attribute header, in the case in which a judgment is made by module (bb) that the content information is encrypted.

FIG. 1

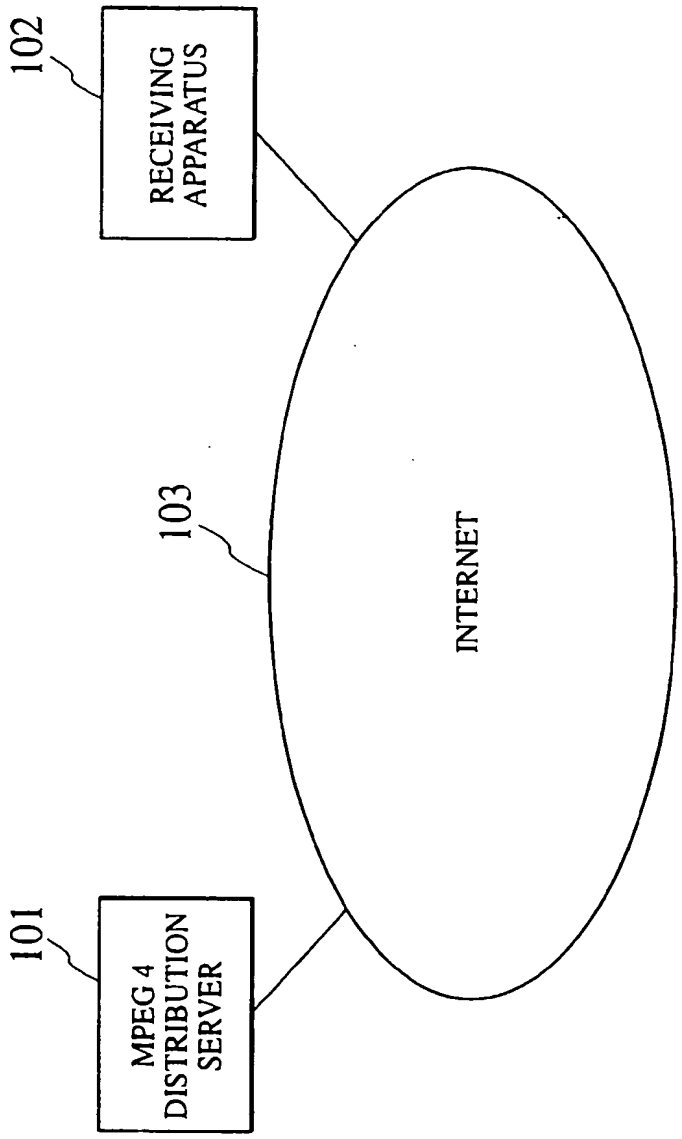


FIG. 2

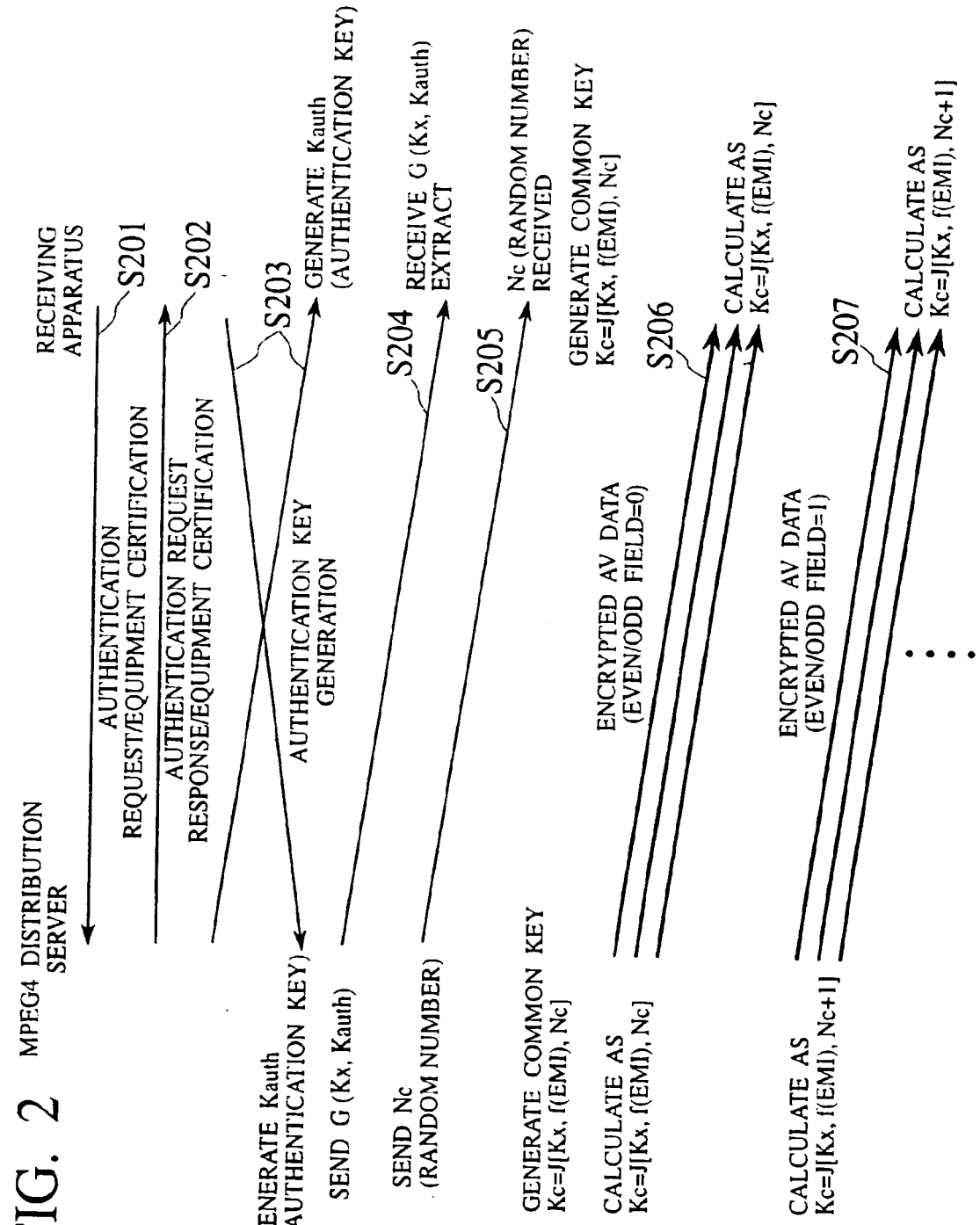


FIG. 3

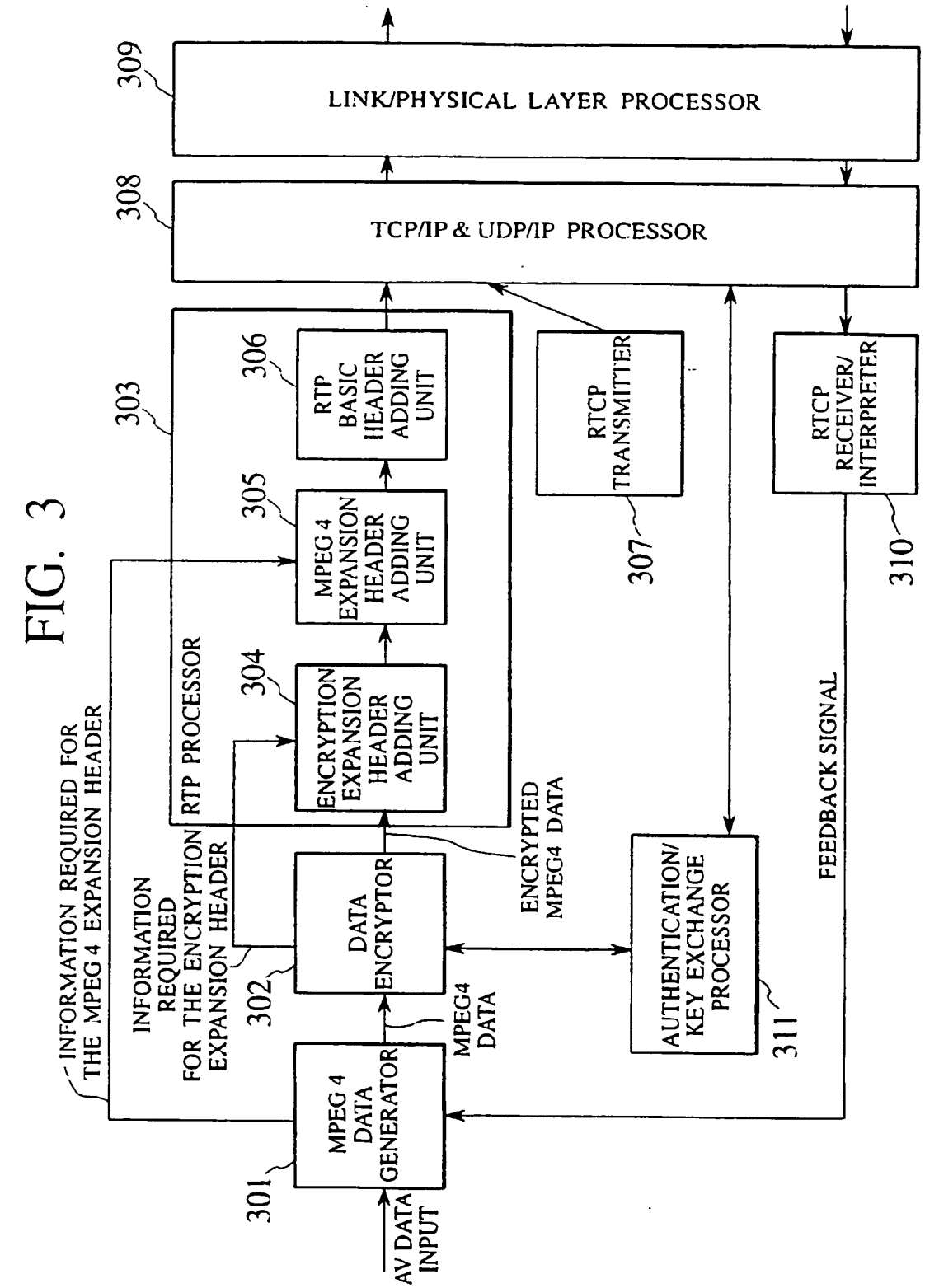


FIG. 4

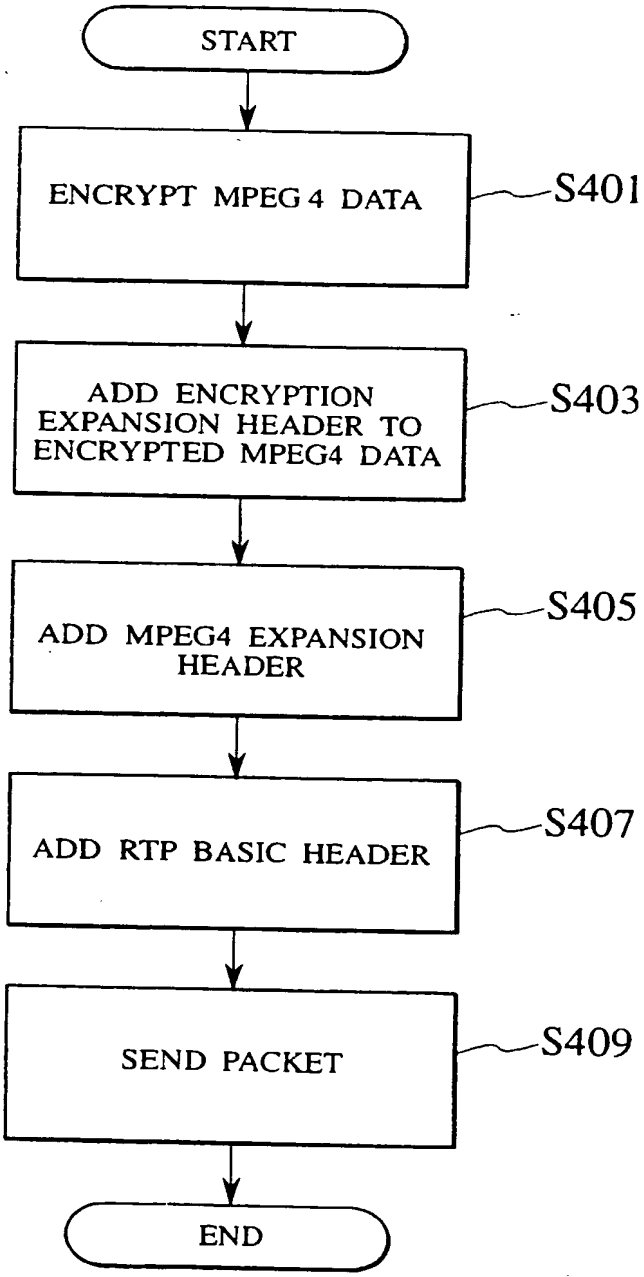


FIG. 5

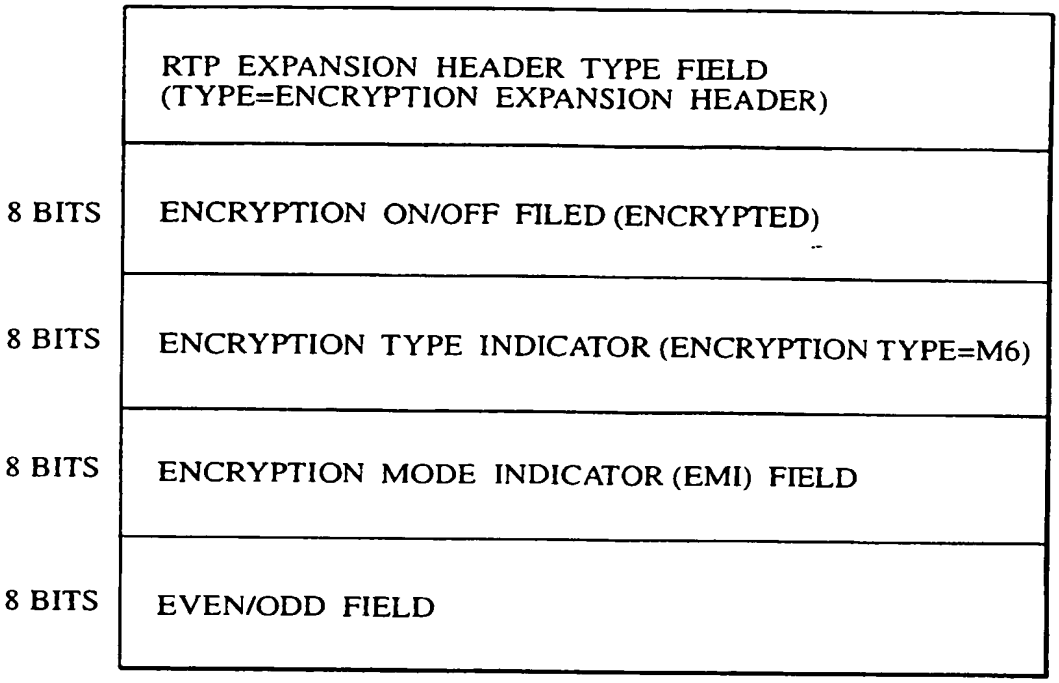


FIG. 6

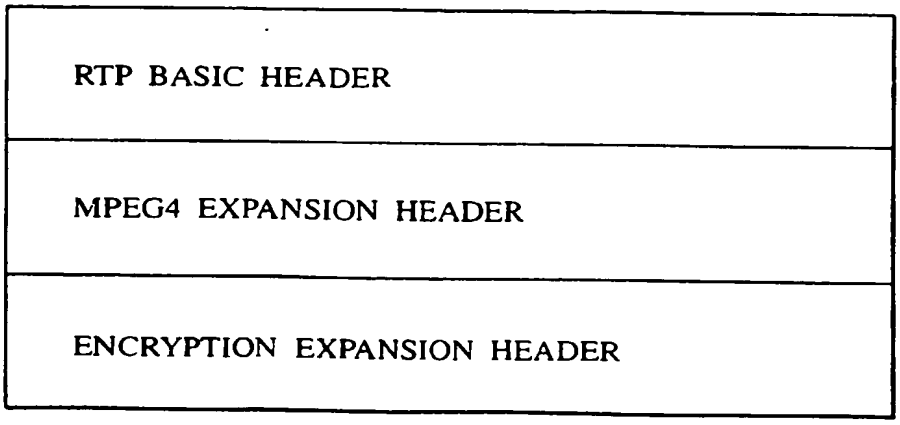


FIG. 7

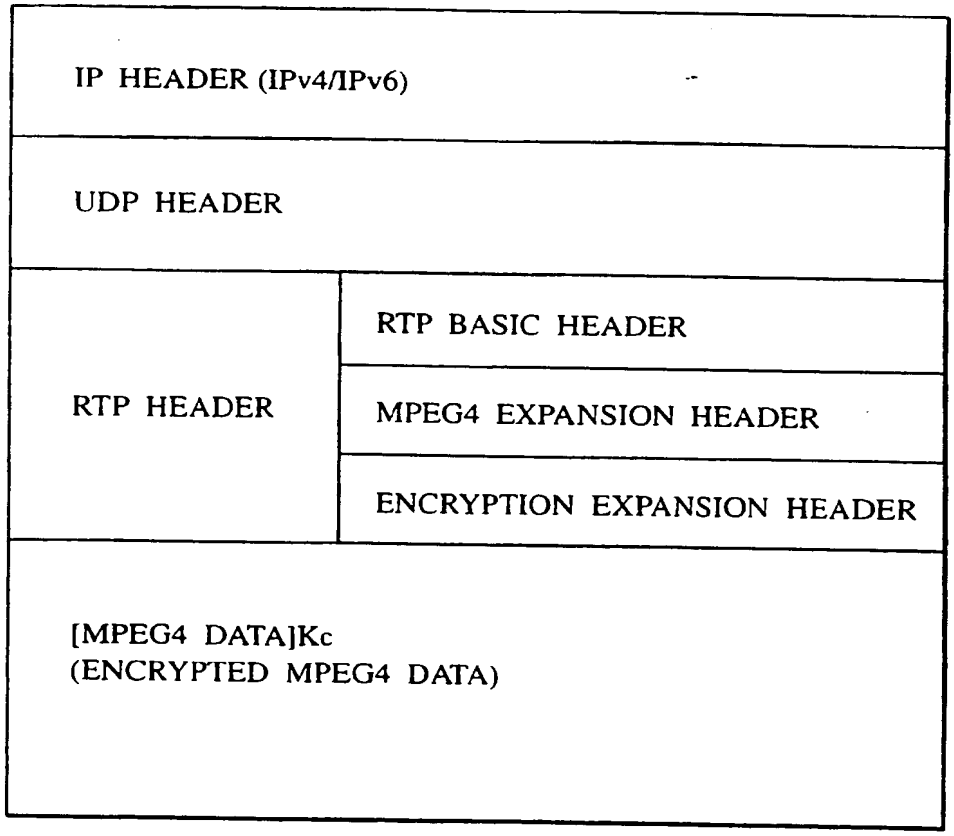


FIG. 8

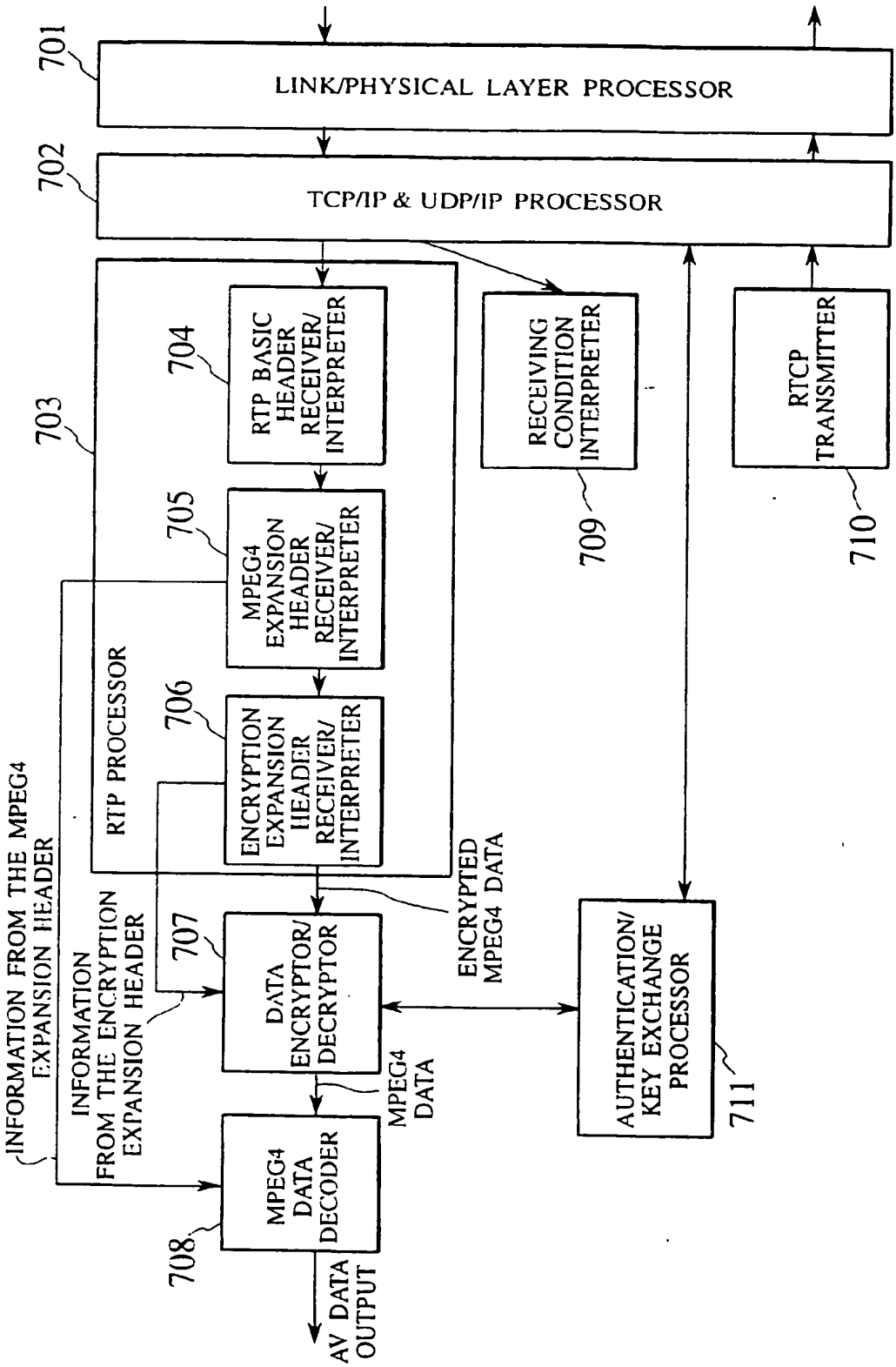


FIG. 9

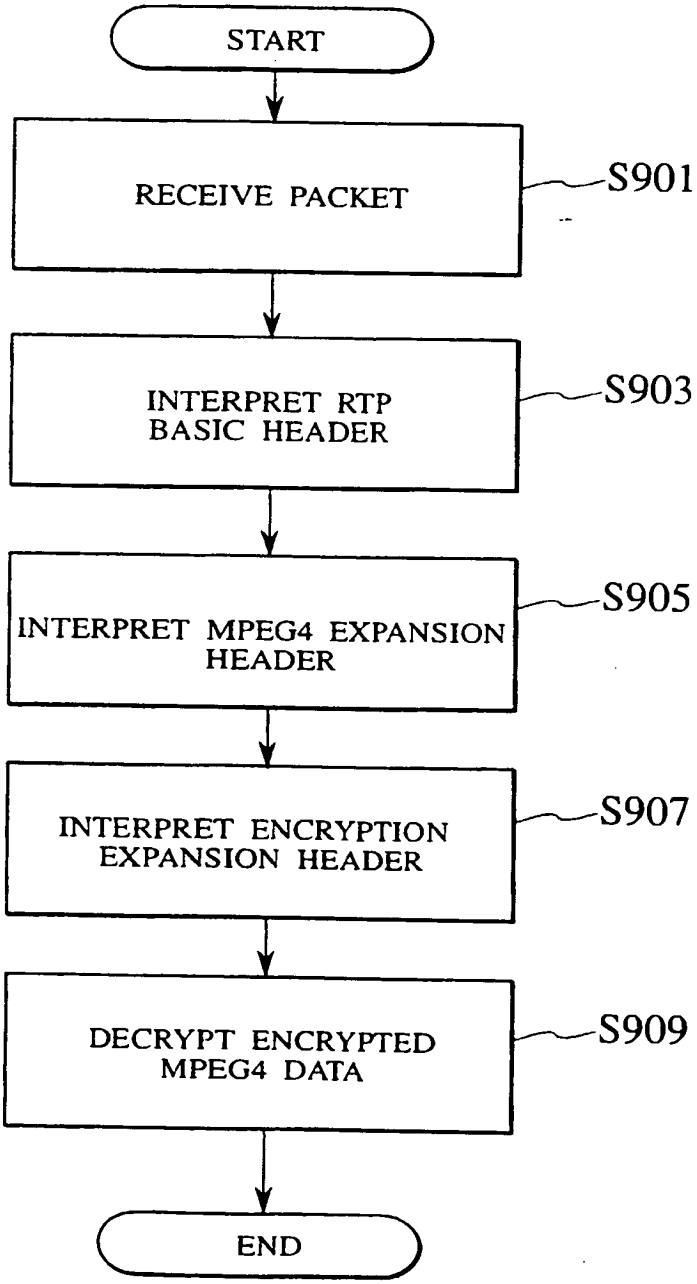


FIG. 10

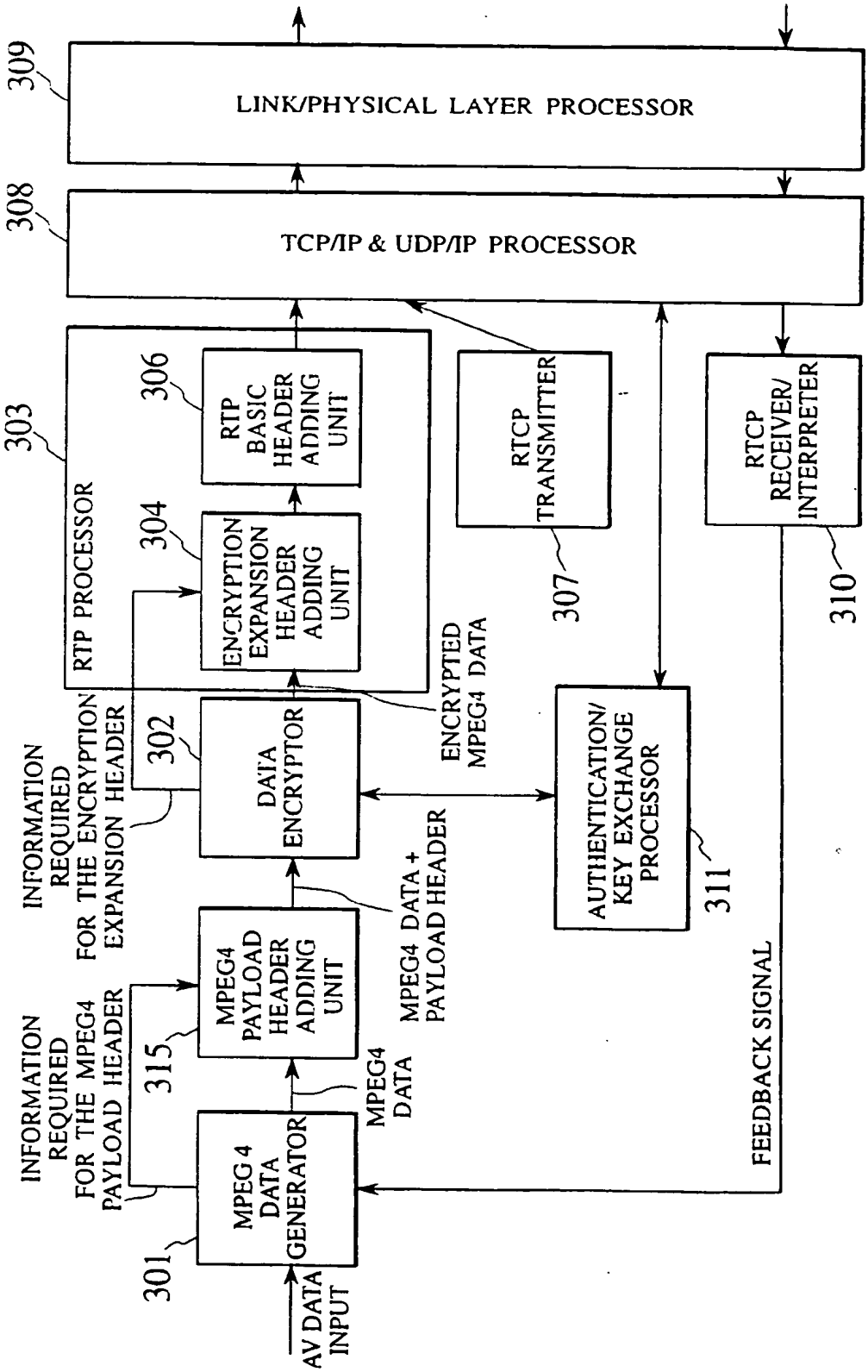


FIG. 11

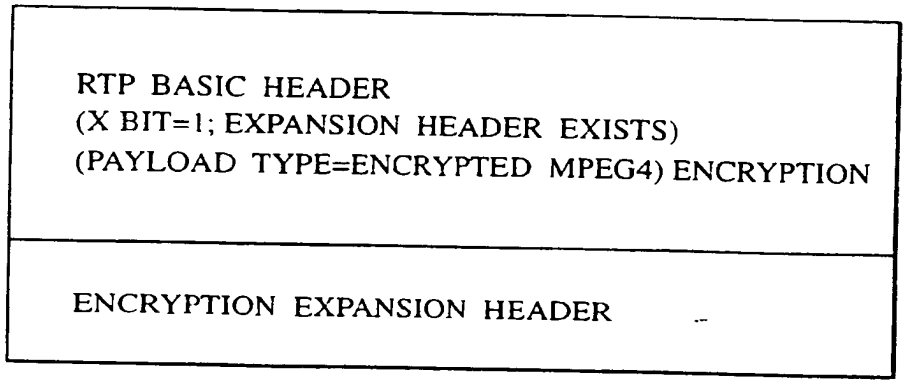


FIG. 12

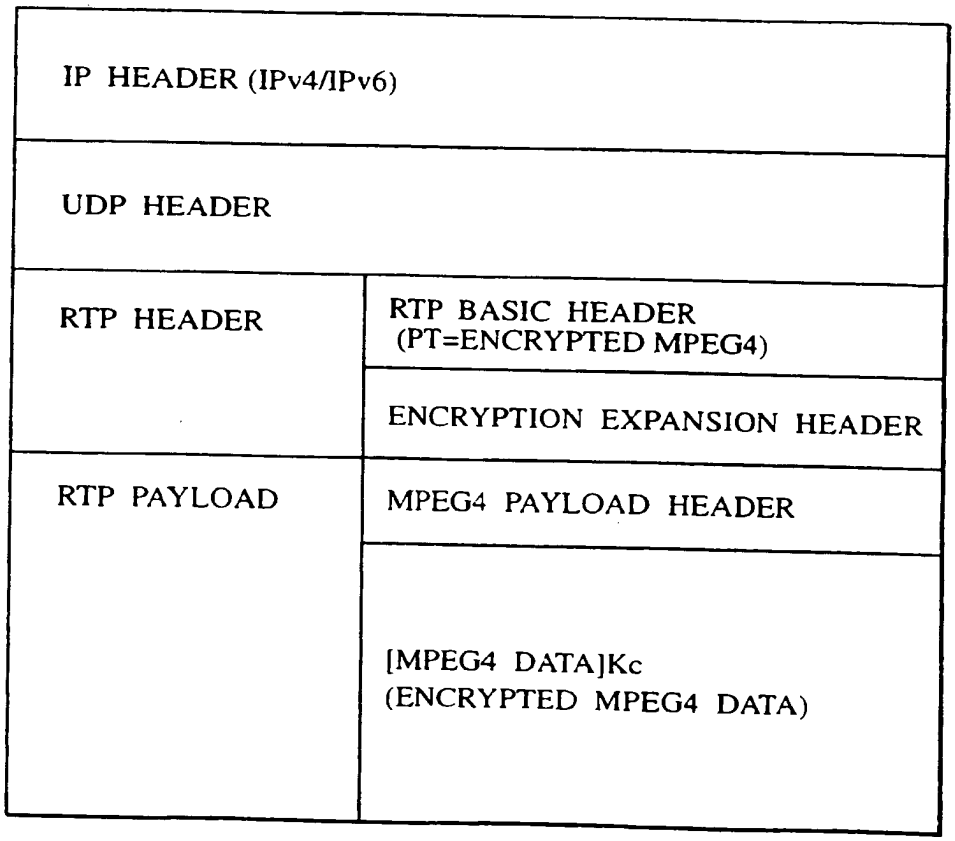


FIG. 13

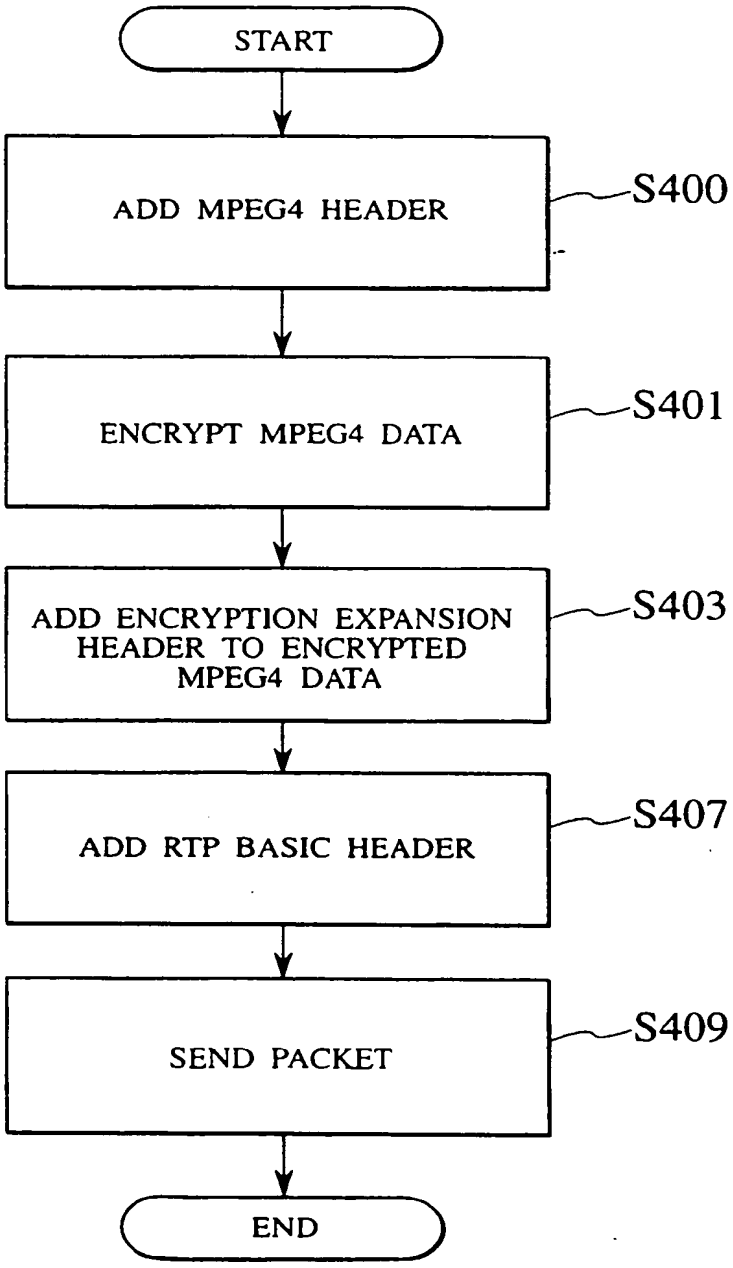


FIG. 14

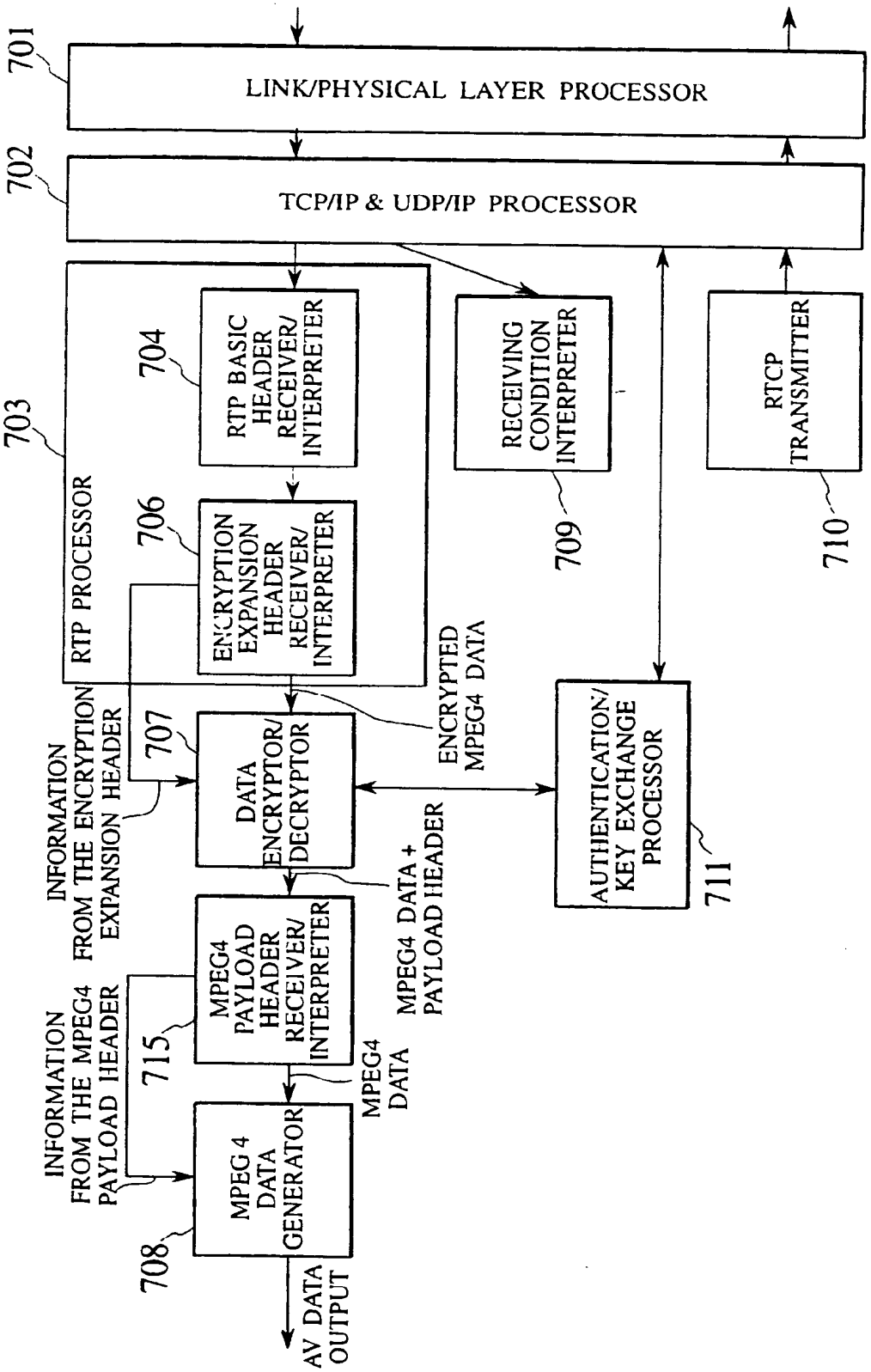


FIG. 15

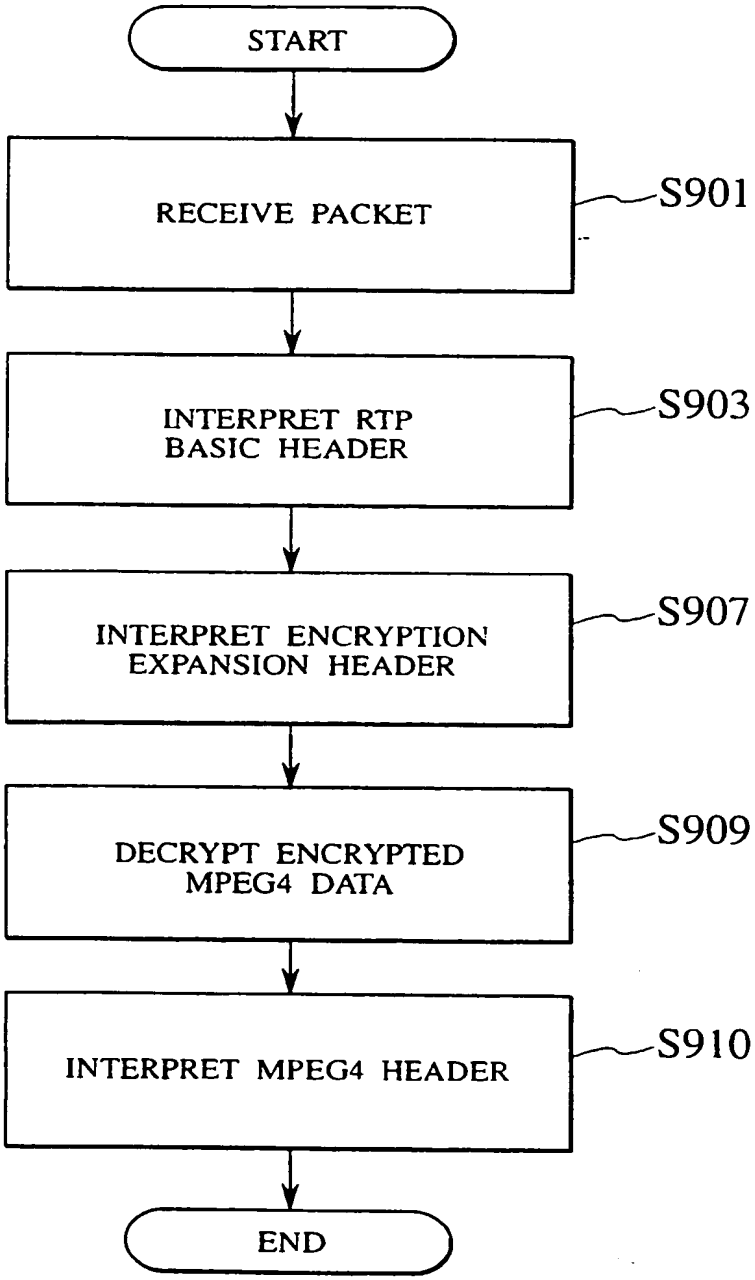


FIG. 16

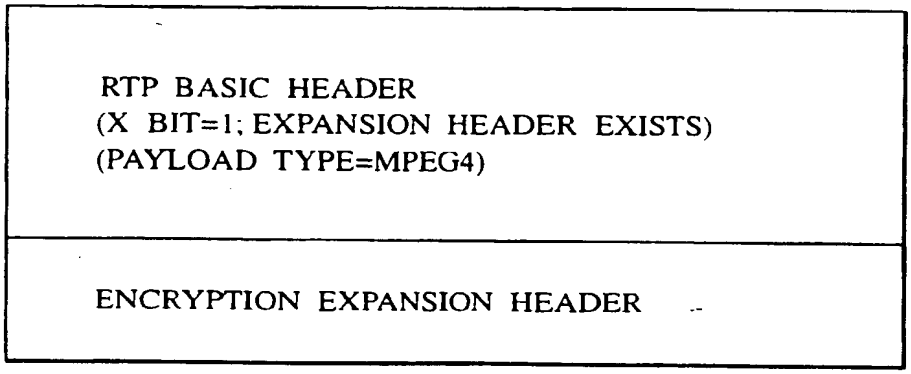


FIG. 17

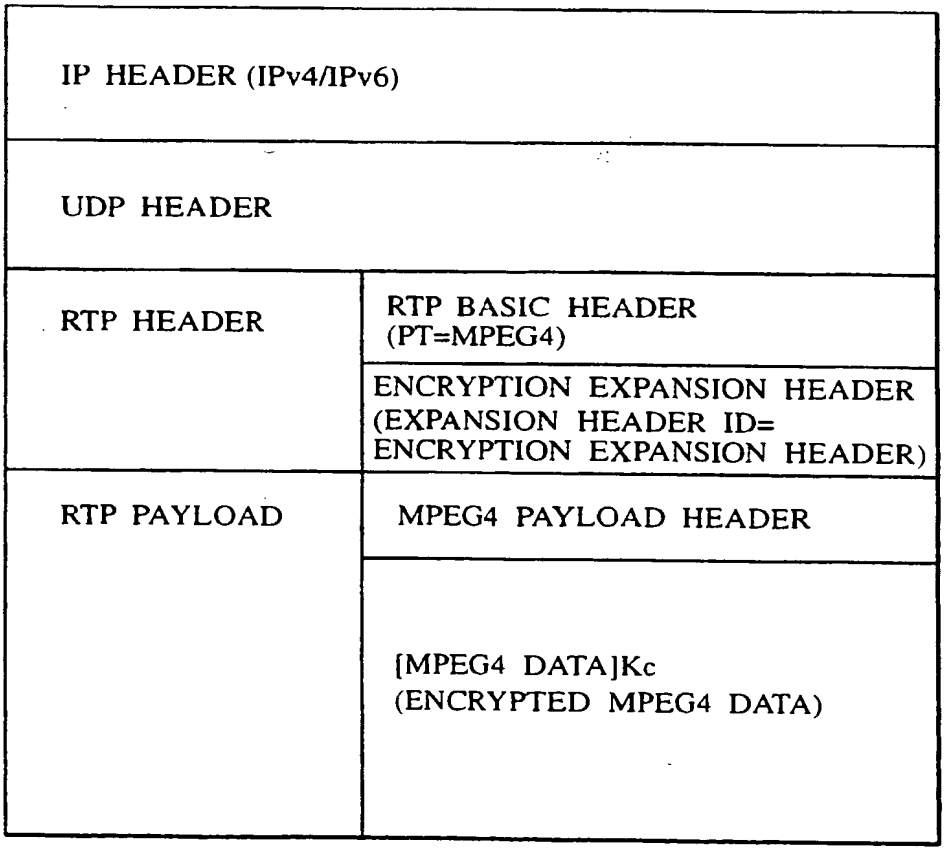


FIG. 18

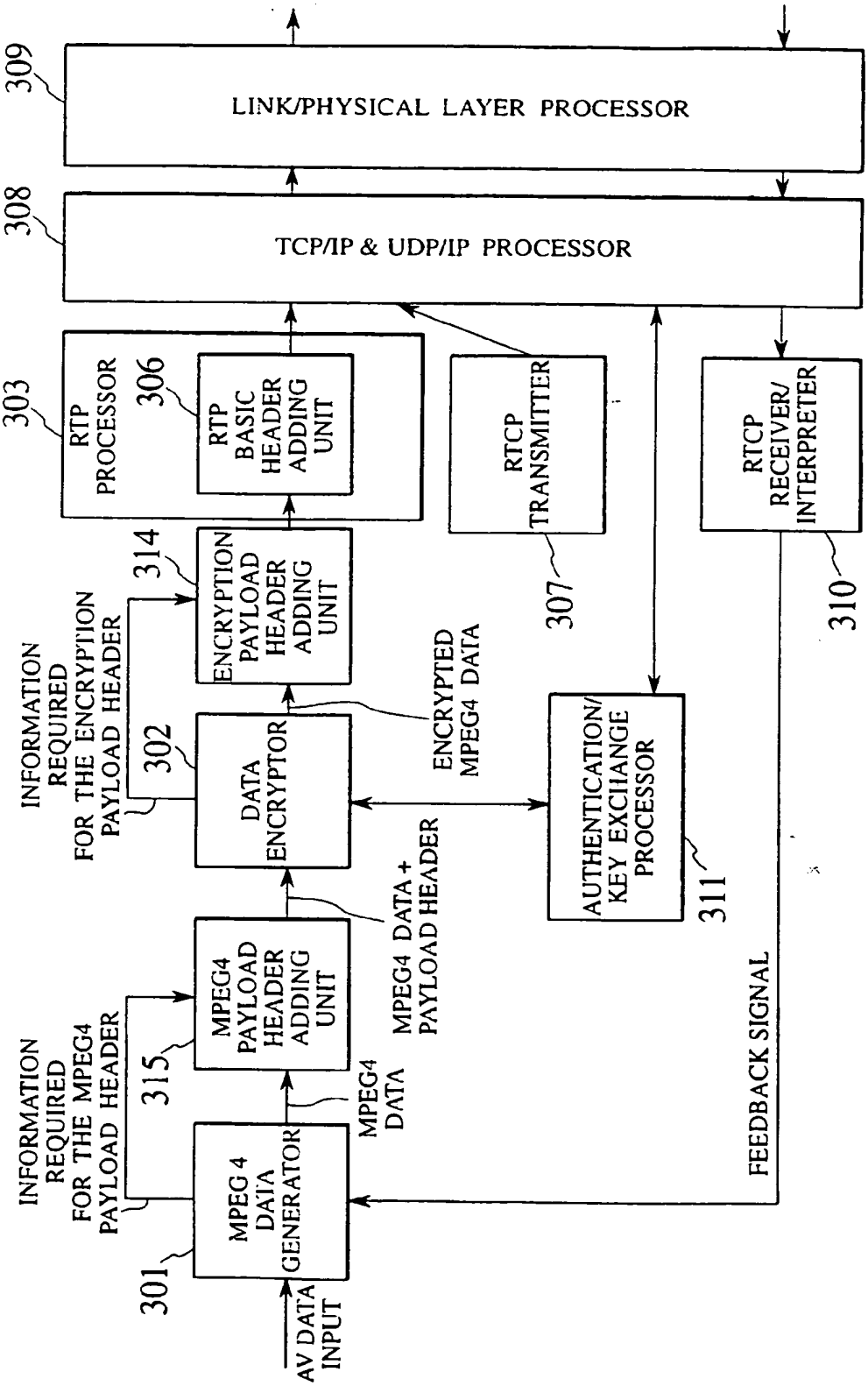


FIG. 19

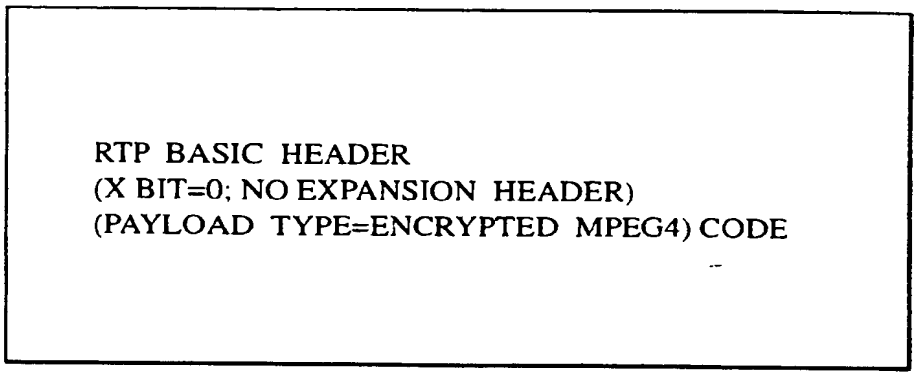


FIG. 20

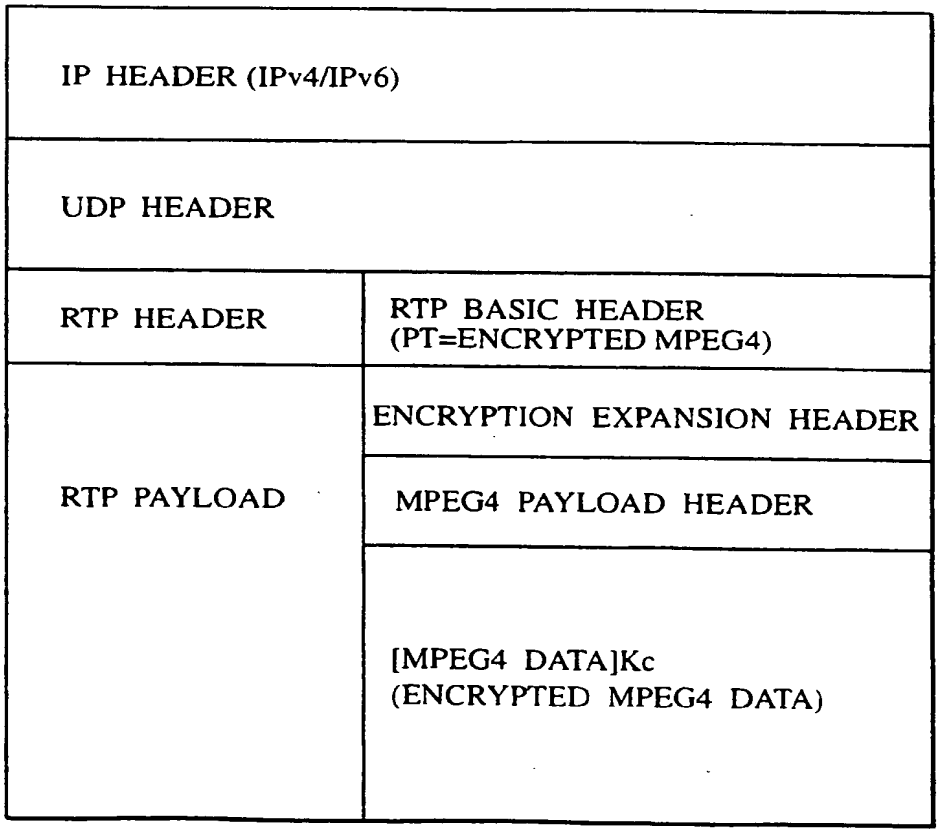


FIG. 21

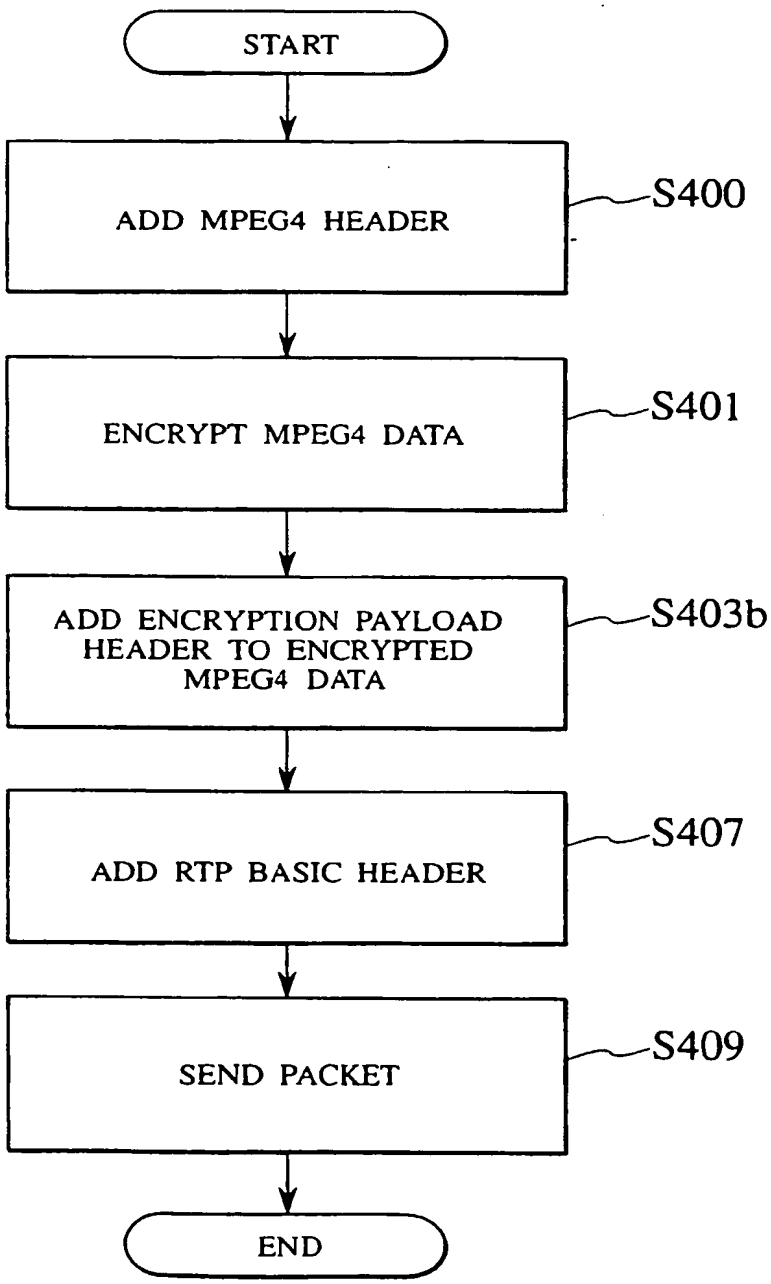


FIG. 22

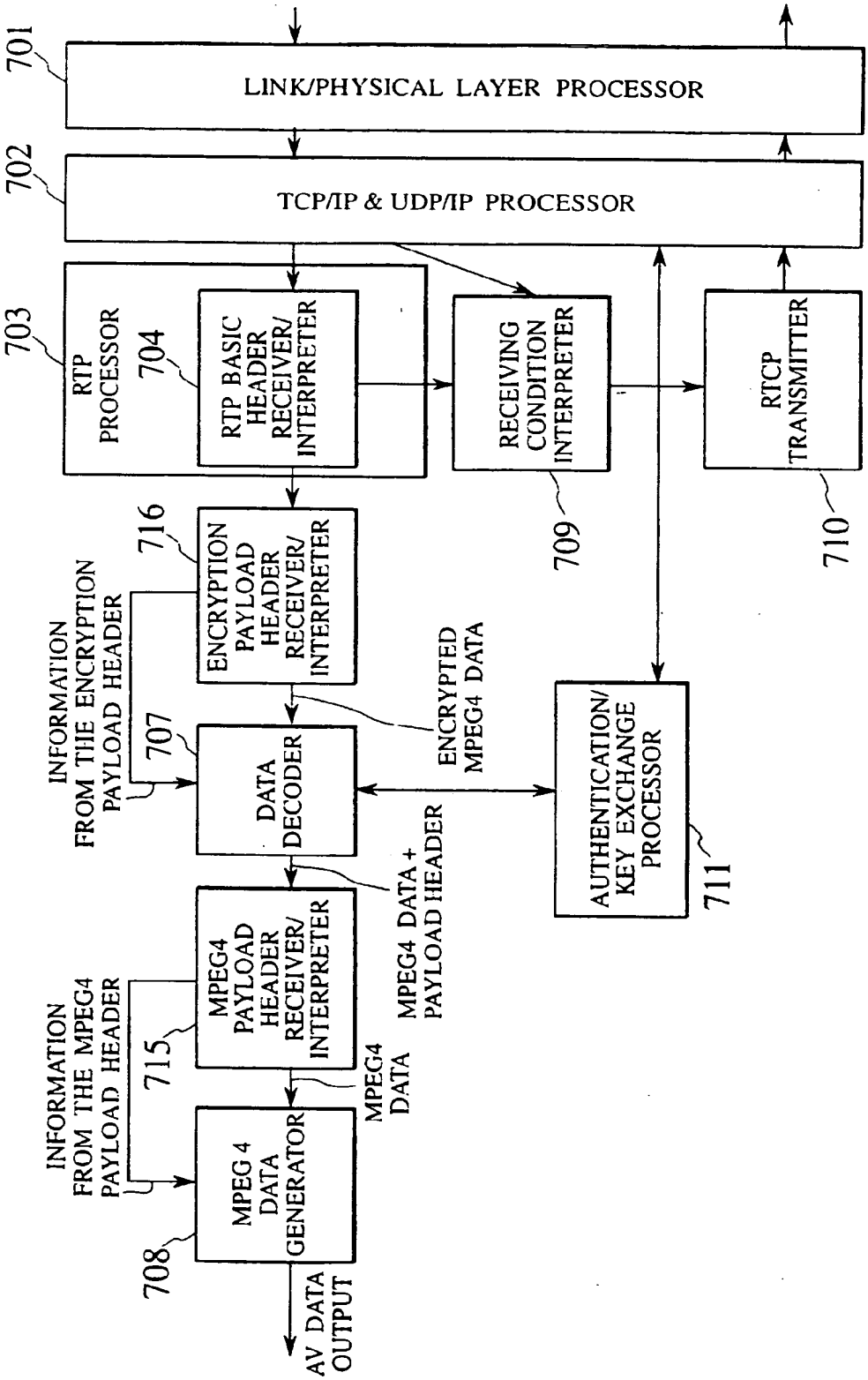


FIG. 29

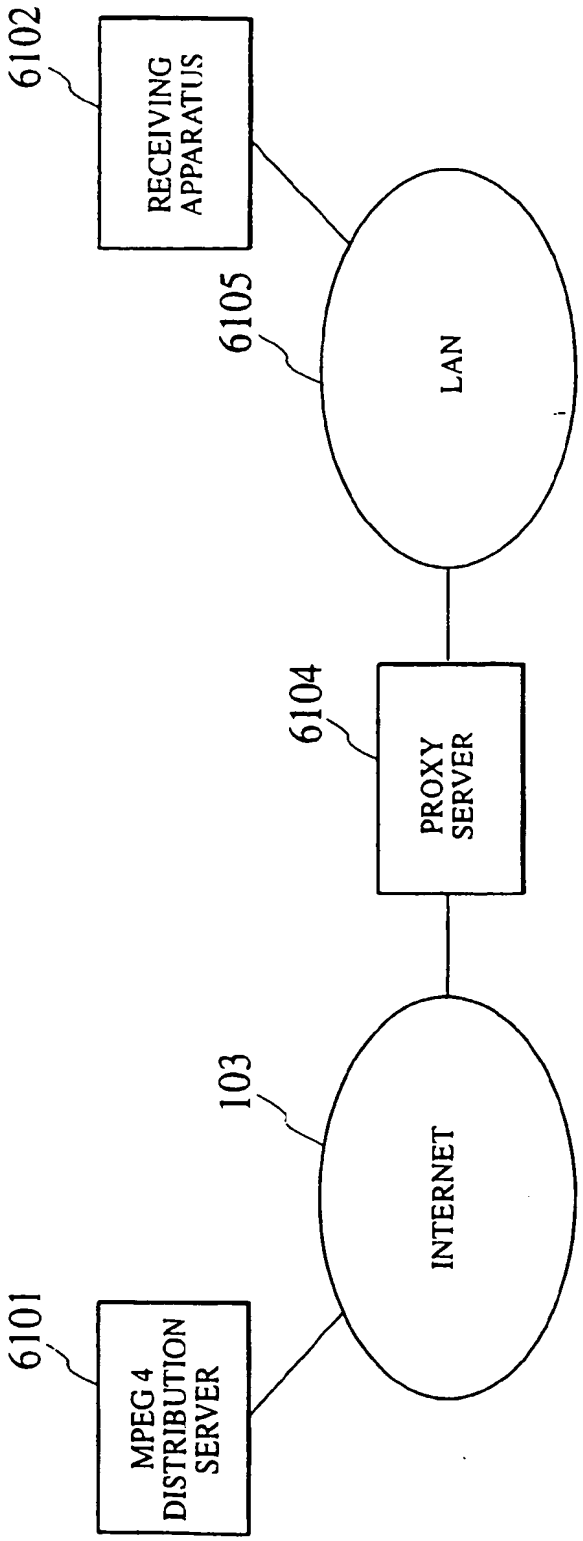


FIG. 30

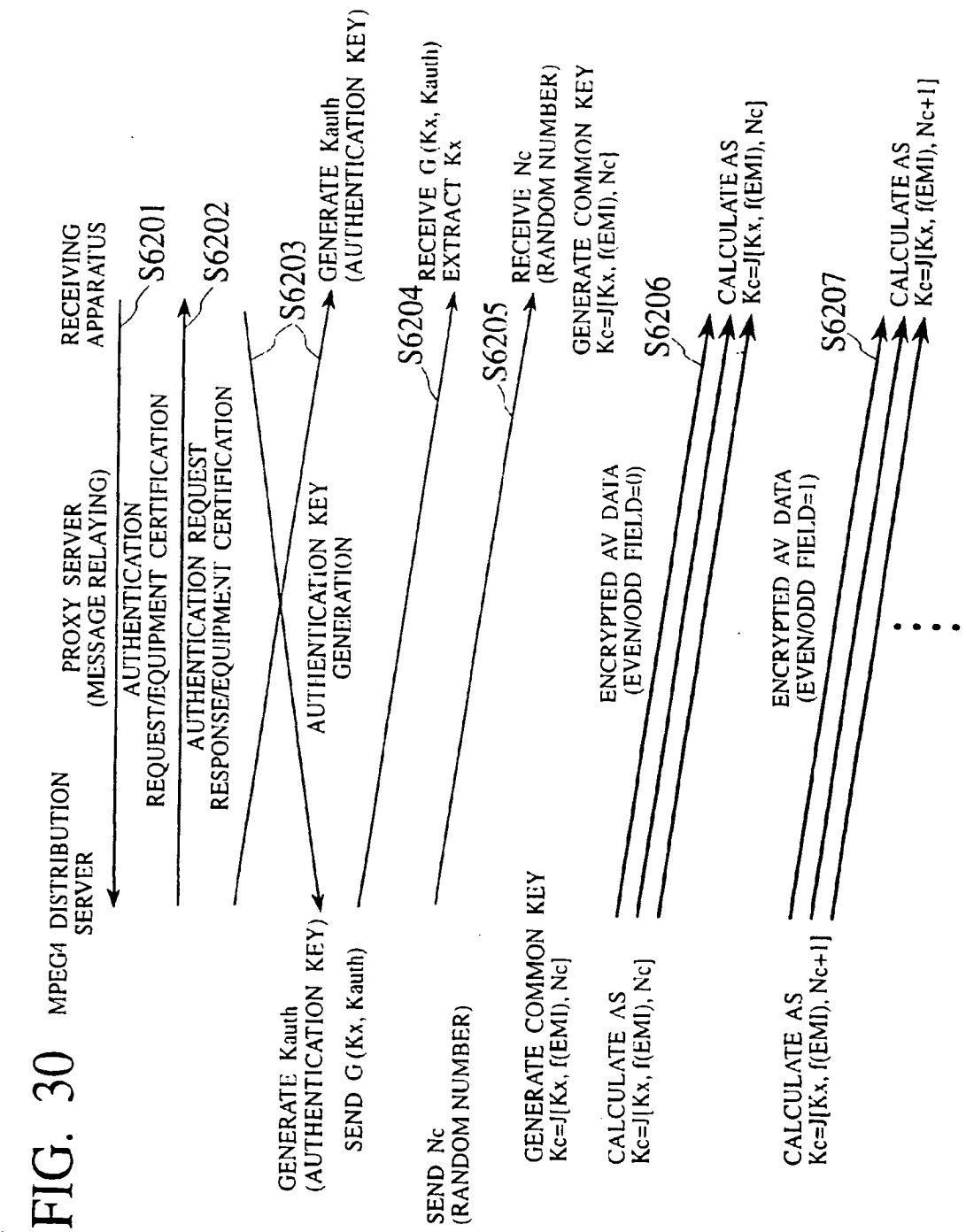


FIG. 23

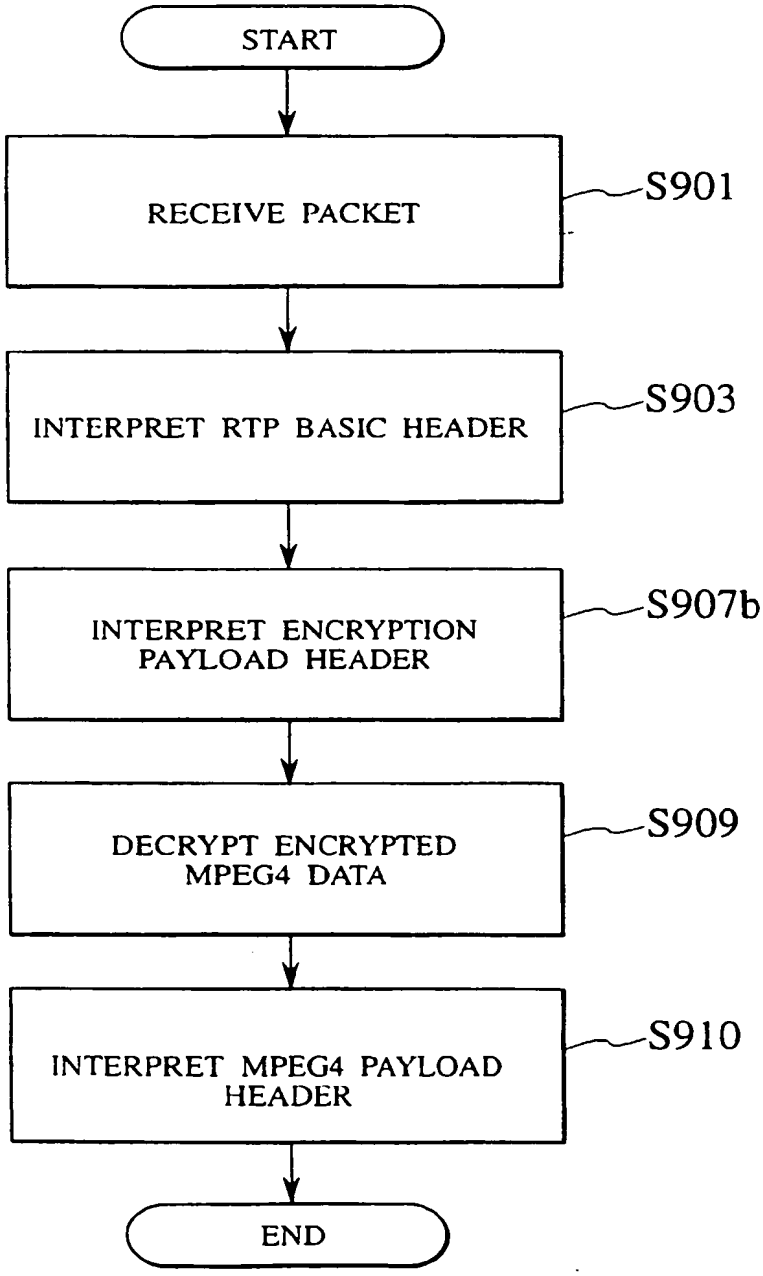


FIG. 24

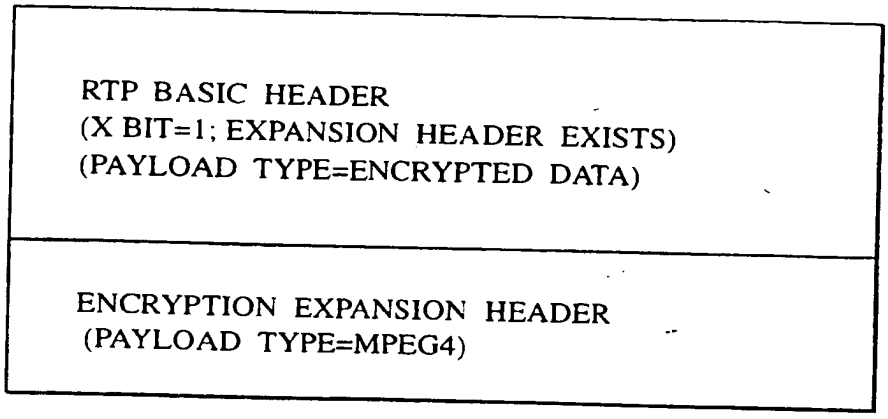


FIG. 25

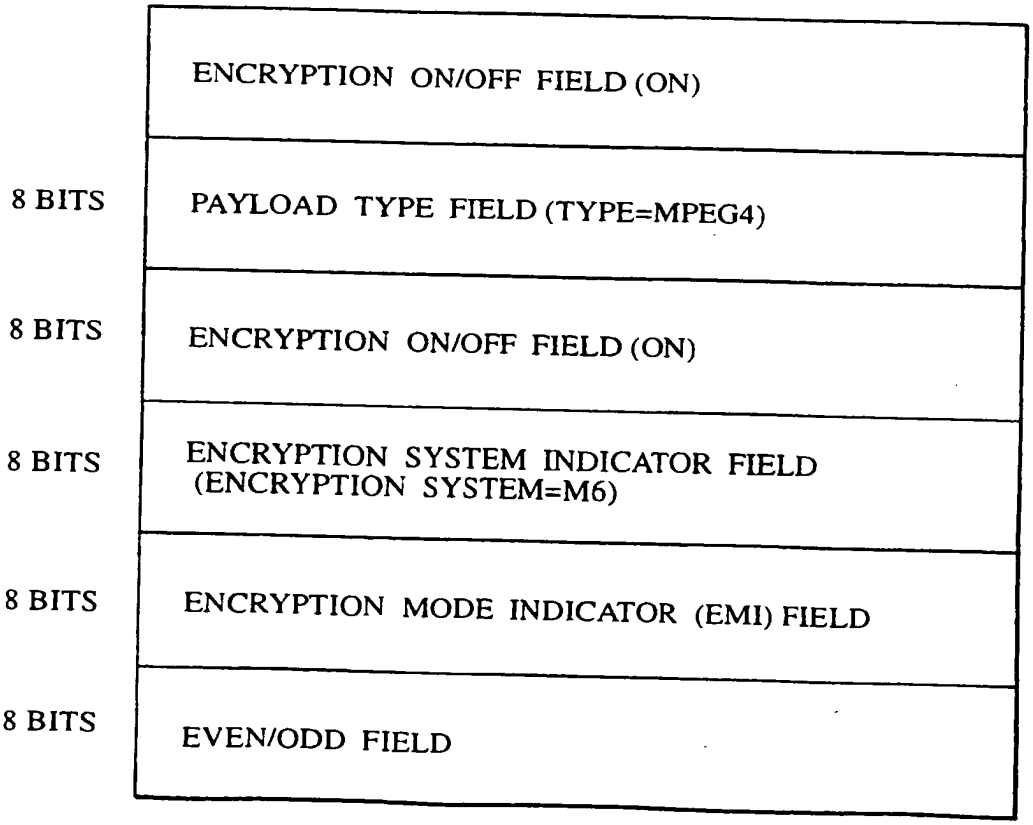


FIG. 31

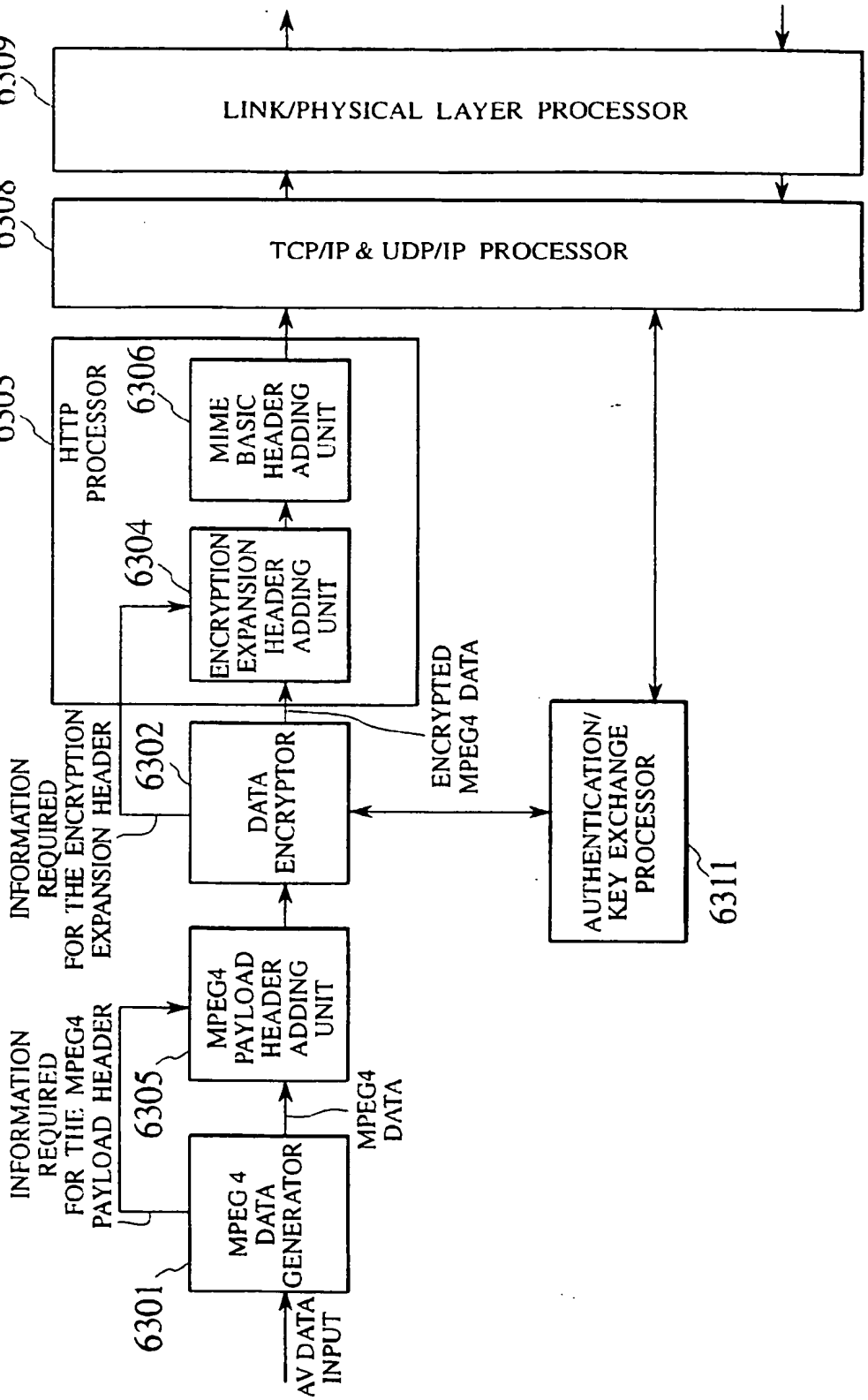


FIG. 32

<table>
<tr><td colspan="2">IP HEADER (IPv4/IPv6)</td></tr>
<tr><td colspan="2">TCP HEADER</td></tr>
<tr><td colspan="2">MIME BASIC HEADER</td></tr>
<tr><td>Content-Type=
application/X-encription</td><td>ENCRYPTION EXPANSION HEADER</td></tr>
<tr><td rowspan="2">Content-Type=
application/X-MPEG4</td><td>MPEG4 PAYLOAD HEADER</td></tr>
<tr><td>[MPEG4 DATA]Kc
(ENCRYPTED MPEG4 DATA)</td></tr>
</table>

FIG. 33

<table>
<tr><td>MIME BASIC HEADER
MIME-Version:1.0
Message-ID:User1-1
Content-Type:Multipasrt/mixed
boundary=''-boundary1-''</td></tr>
<tr><td>ENCRYPTION EXPANSION HEADER
Content-Type=application/x-encription
ENCRYPTION SYSTEM INDICATOR FIELD (ENCRYPTION SYSTEM=M6)
ENCRYPTION ON/OFF FIELD (ON)
ENCRYPTION MODE INDICATOR (EMI) FIELD
EVEN/ODD FIELD</td></tr>
</table>

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 00 30 2721

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

14-03-2003

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
EP 0910003	A	21-04-1999	JP	11122239 A	30-04-1999
			EP	0910003 A2	21-04-1999
			US	6519701 B1	11-02-2003
US 5870474	A	09-02-1999	AU	7009896 A	28-07-1997
			DE	872077 T1	06-05-1999
			EP	0872077 A1	21-10-1998
			ES	2123479 T1	16-01-1999
			JP	2000502857 T	07-03-2000
			WO	9724832 A1	10-07-1997
			US	6157719 A	05-12-2000
			US	2002094084 A1	18-07-2002
			US	6424717 B1	23-07-2002
			US	6292568 B1	18-09-2001
			US	6246767 B1	12-06-2001
			US	6252964 B1	26-06-2001
			US	2001001014 A1	10-05-2001
			US	2001046299 A1	29-11-2001
			US	2002044658 A1	18-04-2002
			US	2001053226 A1	20-12-2001
US 5841864	A	24-11-1998	NONE		
EP 0393806	A	24-10-1990	US	4956863 A	11-09-1990
			EP	0393806 A2	24-10-1990
			JP	2288746 A	28-11-1990

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

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

(11) EP 1 041 823 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
07.05.2003 Bulletin 2003/19

(43) Date of publication A2:
04.10.2000 Bulletin 2000/40

(21) Application number: **00302721.6**

(22) Date of filing: **31.03.2000**

(51) Int Cl.[7]: **H04N 7/167**, H04N 7/173

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

(30) Priority: **31.03.1999 JP 9391699**

(71) Applicant: **KABUSHIKI KAISHA TOSHIBA**
Kawasaki-shi, Kanagawa-ken 210-8572 (JP)

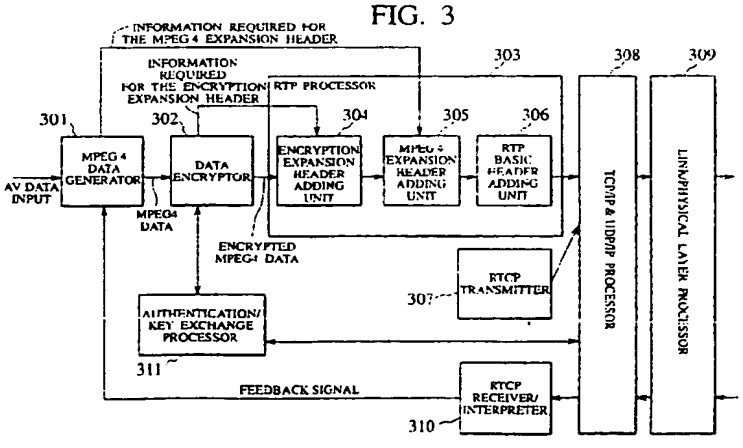
(72) Inventors:
- **Saito, Takeshi, Intellectual Property Division**
  Tokyo (JP)
- **Kato, Taku, Intellectual Property Division**
  Tokyo (JP)
- **Tomoda, Ichiro c/o Intellectual Property Division**
  Tokyo (JP)
- **Takabatake, Yoshiaki, Intell. Prop. Div.**
  Tokyo (JP)
- **Ami, Junko, Intellectual Property Division**
  Tokyo (JP)

(74) Representative: **Midgley, Jonathan Lee**
Marks & Clerk
57-60 Lincoln's Inn Fields
GB-London WC2A 3LS (GB)

(54) Content distribution apparatus, content receiving apparatus, and content distribution method

(57) A content distribution apparatus for implementing copy protection when distributing digital content as a real-time stream on the Internet is provided. This apparatus encrypts content and distributes them to a receiving apparatus via the Internet, and performs an authentication procedure and a key exchange procedure between with the receiving apparatus. The encoded content encoded by a prescribed encoding system is encrypted (S401), an encryption expansion header is generated that includes at least one attribute information of attribute information indicating whether or not the content is encrypted and attribute information indicating the encryption system used (S403), transport protocol processing required to transfer the content is performed and a basic transport header is generated (S407), a packet being sent which includes the basic transport header, the encryption expansion header, and the encrypted content (S409).

FIG. 3



Printed by Jouve, 75001 PARIS (FR)

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 00 30 2721

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.7)
P,Y	EP 0 910 003 A (SONY CORP) 21 April 1999 (1999-04-21) * column 6, line 16 - column 12, line 39; figures 4-11 *	1-20	H04N7/167 H04N7/173
Y	US 5 870 474 A (WASILEWSKI ANTHONY JOHN ET AL) 9 February 1999 (1999-02-09) * column 5, line 21 - column 10, line 12; figure 3A *	1-20	
Y	US 5 841 864 A (CLANTON CHRISTOPHER L ET AL) 24 November 1998 (1998-11-24) * column 4, line 54 - column 5, line 48; figure 4 *	1-20	
Y	EP 0 393 806 A (TRW INC) 24 October 1990 (1990-10-24) * column 4, line 19 - column 6, line 51 *	1-20	
			TECHNICAL FIELDS SEARCHED (Int.Cl.7)
			H04N

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	14 March 2003	Luckett, P

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)

FIG. 26

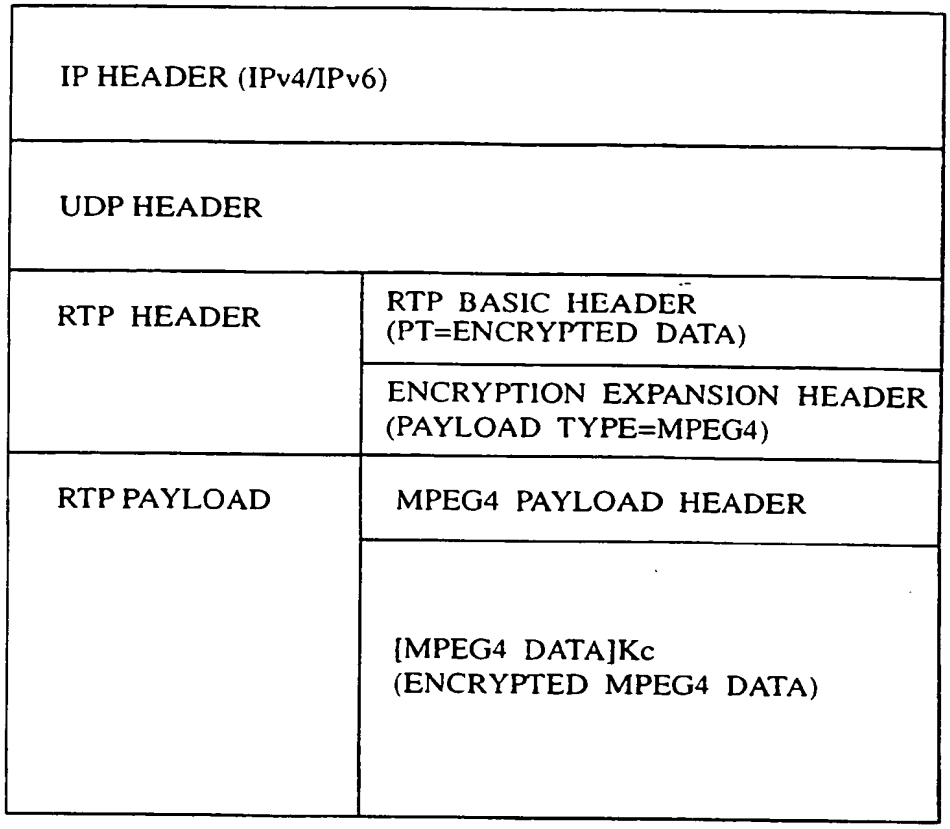


FIG. 27

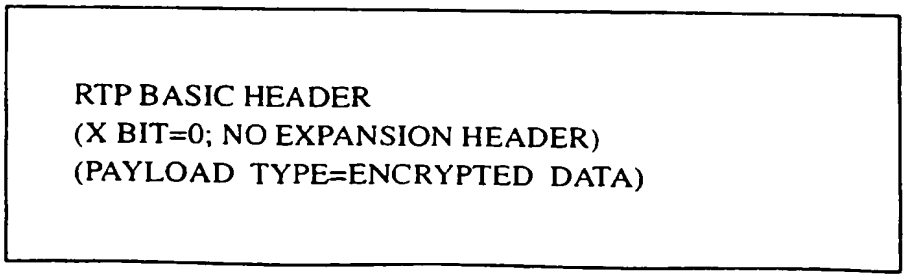


FIG. 28

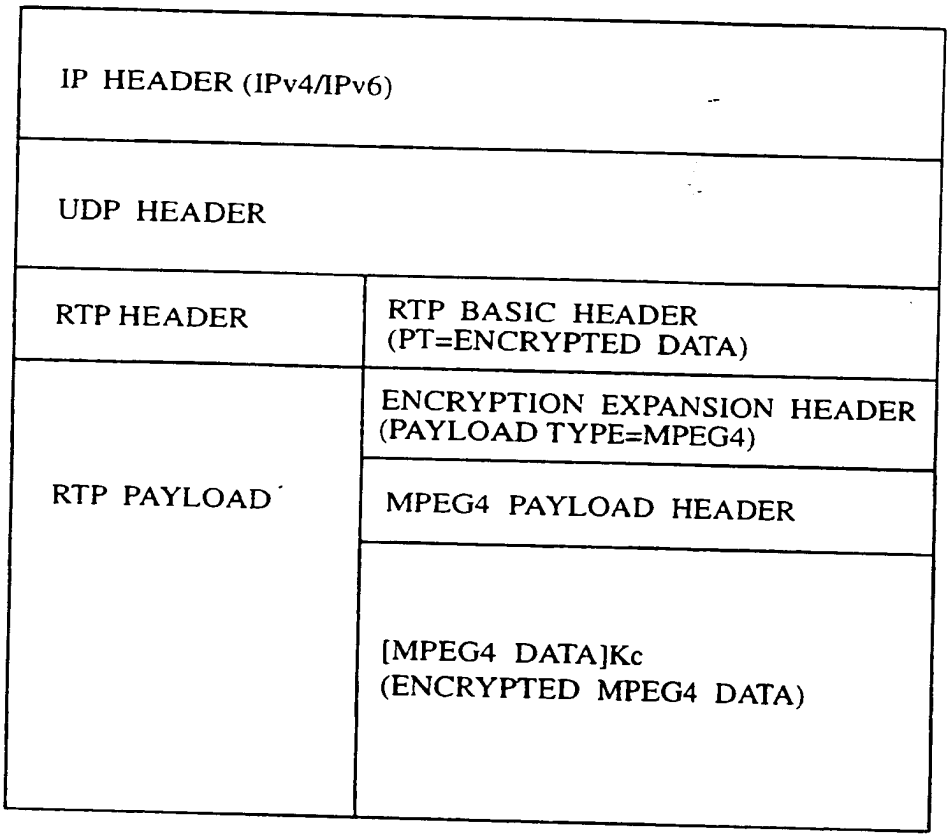


FIG. 34

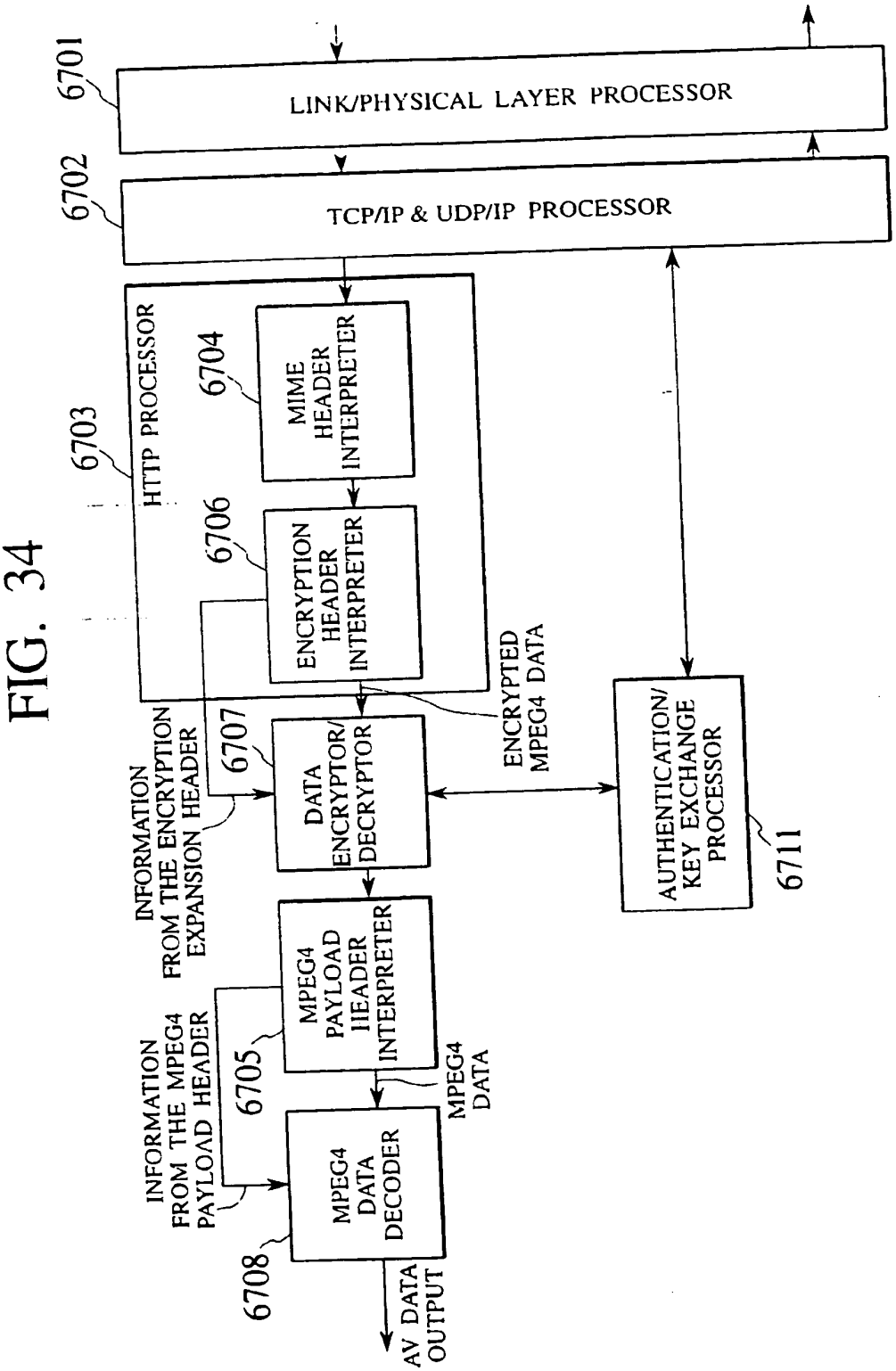
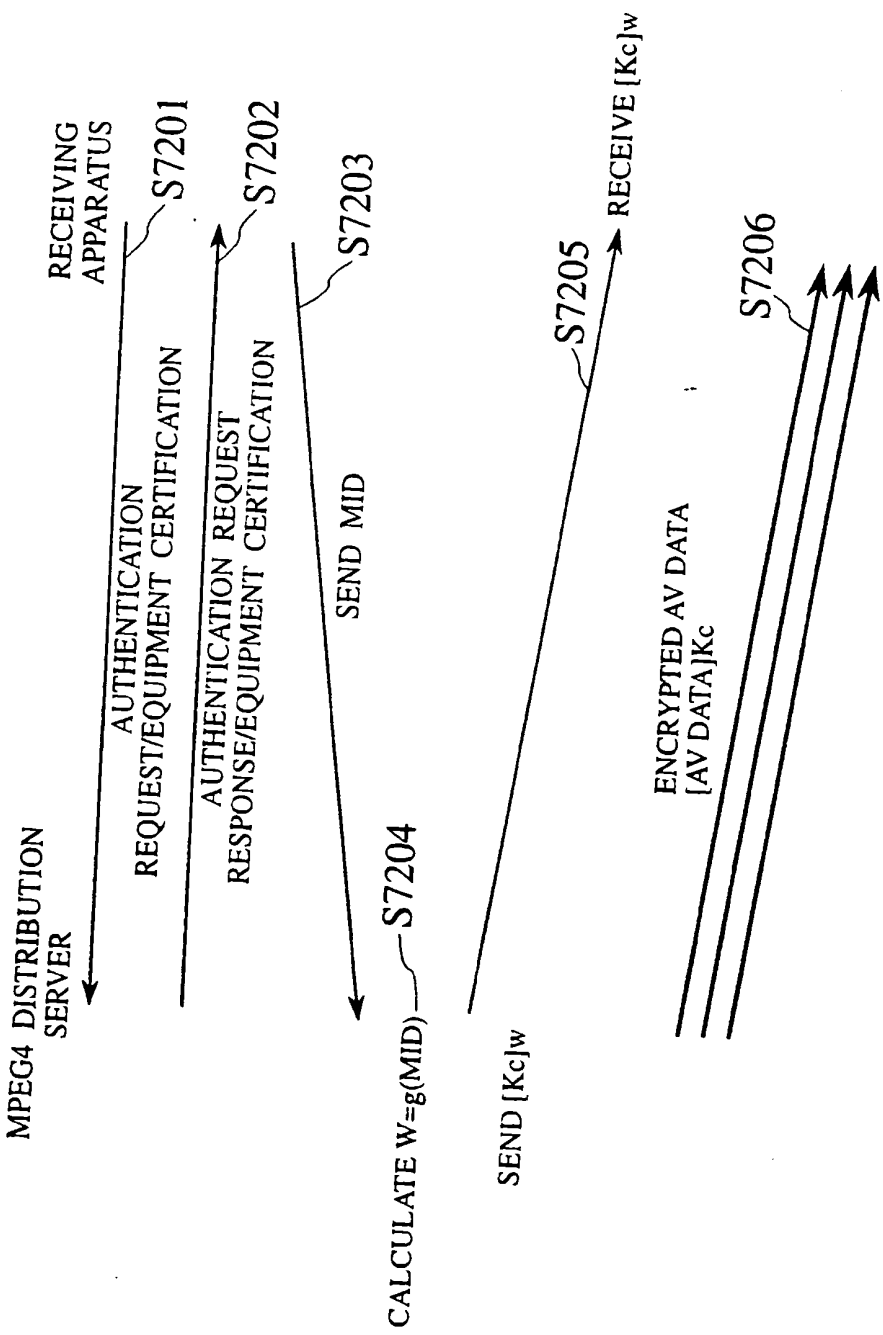


FIG. 35



NIXON PEABODY LLP
ATTORNEYS AT LAW

401 9th Street, N.W.
Suite 900
Washington, DC 20004
(202) 585-8000
Fax: (202) 585-8080

Carlos R. Villamar
Direct Dial: (202) 585-8204
E-Mail: cvillamar@nixonpeabody.com

July 15, 2004

VIA U.S. MAIL

Mr. Charles Gilliam
Vice President, Secretary and General Counsel
ContentGuard, Inc.
6500 RockSpring Drive, Suite 110
Bethesda, MD 20817-1105

RE: U.S. Continuation-in-Part Patent Application No.: 10/162,212
Based on Serial No. 09/867,745, Ref. No. 111325-66
Inventor(s): Xin **WANG**, *et al.*
Title: **RIGHTS OFFERING AND GRANTING**
Our Reference: 111325-104 (230300)
Action Required: None

Dear Charles:

Pursuant to my email dated June 22, 2004, we prepared and filed a Preliminary Amendment in the above-identified application with the U.S. Patent and Trademark Office **today, July 15, 2004**. Copies of the as-filed papers are attached for your records.

We will continue to keep you advised of any further developments in connection with this application as they occur.

Best regards.

Very truly yours,
NIXON PEABODY LLP

Carlos R. Villamar

CRV/kla
Enclosures

W316724.1

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin **WANG**, *et al.*
Group Art Unit	3621
Examiner Name	Mary Cheung
Attorney Docket Number	111325-104 (230300)
Confirmation Number	3700

Total Number of Pages in This Submission

ENCLOSURES *(check all that apply)*

- ☒ Fee Transmittal Form
  - ☐ 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)
- ☐ Response to Missing Parts/ Incomplete Application
  - ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Assignment Papers *(for an Application)*
- ☐ Drawing(s)
- ☐ 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
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to Group (Appeal Notice, Brief, Reply Brief)
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Application Data Sheet
- ☐ Request for Corrected Filing Receipt with Enclosures
- ☐ A self-addressed prepaid postcard for acknowledging receipt
- ☐ Other Enclosure(s) (*please identify below*):

Remarks

☒ 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	Carlos R. Villamar Registration No.: 43,224 **NIXON PEABODY LLP** 401 9th Street, N.W., Suite 900 Washington, D.C. 20004-2128
Signature	
Date	July 15, 2004

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(a)]

I hereby certify that this correspondence is being:

- ☐ deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop _____, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450
- ☐ transmitted by facsimile on the date shown below to the United States Patent and Trademark Office at (703) _____.

_____ Date

_____ Signature

_____ Typed or printed name

FEE TRANSMITTAL FOR FY 2004

Current fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT | **$126.00**

Complete if Known	
Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin WANG, *et al.*
Examiner Name	Mary Cheung
Art Unit	3621
Attorney Docket No.	111325-104 (230300)

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☐ Credit Card ☐ Money Order ☐ Other ☐ None

☒ Deposit Account:

Deposit Account Number: 19-2380

Deposit Account Name: Nixon Peabody LLP

The Commissioner is authorized to: *(check all that apply)*

☒ Charge fee(s) indicated below ☒ Credit any overpayments

☒ Charge any additional fee(s)

☐ Charge fee(s) indicated below, **except for the filing fee** to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	770	2001	385	Utility filing fee	
1002	340	2002	170	Design filing fee	
1003	530	2003	265	Plant filing fee	
1004	770	2004	385	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

SUBTOTAL (1)

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

			Extra Claims		Fee from below	=	Fee Paid
Total Claims	28	-21** =	7	X	18.00	=	126.00
Independent Claims	3	-3** =	0	X		=	0
Multiple Dependent				X		=	0

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	86	2201	43	Independent claims in excess of 3
1203	290	2203	145	Multiple dependent claim, if not paid
1204	86	2204	43	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) **$126.00**

**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	
1051	130	2051	65	Surcharge – late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	420	2252	210	Extension for reply within second month	
1253	950	2253	475	Extension for reply within third month	
1254	1,480	2254	740	Extension for reply within fourth month	
1255	2,010	2255	1,005	Extension for reply within fifth month	
1401	330	2401	165	Notice of Appeal	
1402	330	2402	165	Filing a brief in support of an appeal	
1403	290	2403	145	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive – unavoidable	
1453	1,330	2453	665	Petition to revive – unintentional	
1501	1,330	2501	665	Utility issue fee (or reissue)	
1502	480	2502	240	Design issue fee	
1503	640	2503	320	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	770	2809	385	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	770	2810	385	For each additional invention to be examined (37 CFR 1.129(b))	
1801	770	2801	385	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3)

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(a)]

I hereby certify that this correspondence is being:

☐ deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop _____, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450

☐ transmitted by facsimile on the date shown below to the United States Patent and Trademark Office at (703) _____.

Date _____ Signature _____

Typed or printed name _____

SUBMITTED BY

Name *(Print/Type)*	Carlos R. Villamar	Registration No. *(Attorney/Agent)*	43,224	Complete *(if applicable)* Telephone	(202) 585-8000
Signature				Date	July 15, 2004

SEND TO: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

W316716.1

TC/AU 3621

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:	)	
Xin **WANG,** *et al.*	)	Examiner: Mary Cheung
Serial No. 10/162,212	)	Group Art Unit: 3621
Filed: June 5, 2002	)	Confirmation No. 3700
For: RIGHTS OFFERING AND GRANTING	)	

Commissioner of Patents
U.S. Patent and Trademark Office
220 20th Street S.
Customer Window, Mail Stop Non-Fee Amendment
Crystal Plaza Two, Lobby, Room 1B03
Arlington, VA 22202

PRELIMINARY AMENDMENT

Sir:

Prior to examination on the merits, please amend the above-identified patent application as follows.

Amendments to the Claims:

1. (Currently amended) A method for transferring usage rights adapted to be associated with items, said method comprising:

generating, by a supplier, at least one first offer containing usage rights and meta-rights for the items ~~item~~, said usage rights defining a manner of use for the items, said meta-rights specifying rights to derive usage rights or other meta-rights;

presenting said offer to a first consumer;

receiving a selection from the first consumer indicating desired usage rights and meta-rights; and,

generating a first license granting the desired usage rights and meta-rights to the first consumer.

2. (Original) The method of claim 1, wherein said license specifies one or more conditions which must be satisfied in order for said usage right to be exercised and one or more conditions which must be satisfied in order for said meta-rights to be exercised.

3. (Original) The method of claim 1, further comprising the step of receiving a request for a license from the first consumer.

4. (Currently amended) The method of claim 1, further comprising:

receiving a request generated by a second consumer for a license containing at least one of usage rights and meta-rights for the items;

generating, by a second supplier, a second offer containing rights derived from said meta-rights contained in the second offer, wherein the second supplier is the ~~firs~~ first consumer; and

generating, by a second supplier, a second license containing rights derived from said meta-rights contained in the ~~second offer~~ first license, wherein the second supplier is the first consumer.

5. (Original) The method of claim 1, wherein the item comprises digital content.

6. (Original) The method of claim 1, further comprising the steps of:

providing said first license as a customized draft license to the first consumer;

accepting a confirmation of said customized draft license from the first consumer; and

authenticating said draft license to create an authenticated license.

7. (Original) The method of claim 1, wherein said first license comprises a license identification, a digital signature, and at least one grant, said at least one grant including usage rights, meta-rights, a named principal designating the first consumer to whom rights are granted, and a condition list.

8. (Currently amended) The method of claim 1, wherein the first supplier is at ~~lest~~ least one of a provider, distributor, retailer, consumer, and ~~or~~ a user.

9. (Currently amended) The method of claim 1, wherein the first consumer is at least one of a provider, distributor, retailer, consumer, and ~~or~~ a user.

10. (Original) The method of claim 1, wherein the step of generating at least one offer comprises the steps of:

collecting usage rights and meta-rights available to be offered;

determining if the supplier has a right to offer the available usage rights and meta-rights;

terminating the generating of a set of offers, if a right to offer other usage and meta rights does not exist;

composing an offer based on available rights if the supplier has the right to offer other usage and meta rights; and

authenticating said offer.

11. (Original) The method of claim 10, wherein said composing step comprises:

determining if a consumer has requested an offer including specific usage rights and meta-rights;

applying the specific usage rights and meta-rights to the offer as a filter; and

determining if an offer template corresponds to the filtered offer and if so applying said offer template as an offer,

12. (Currently amended) The method of claim ~~1~~ 6, wherein said step of generating a first license further comprises the steps of:

determining if the suppler has the right to grant the rights;

terminating the step of customizing a draft license, if the supplier does not have the right to grant the rights;

analyzing one or more choices received from the consumer;

determining if the choices are acceptable; and

creating a draft license based on the choices if the choices are acceptable.

13. (Currently amended) The method of claim 12, wherein said step of generating a first license further comprises: ;

presenting the draft license to the consumer;

re-negotiating a license if the first license ~~if~~ is not approved by the consumer; and

authenticating the draft license if the first consumer approves the draft license.

14. (Currently amended) The method of claim 1, wherein said usage rights specify rights to copy, transfer, loan, play, print, back-up, restore, delete, extract, embed, edit, authorize, install, or un-install the ~~item~~ items.

15. (Currently amended) A system for transferring usage rights adapted to be associated with an item, to be licensed in multi-tier channels of distribution with downstream rights and conditions assigned at at least one level, said system comprising:

a supplier component, comprising:

a supplier user interface module;

an offer generator module for generating an offer containing at least usage rights and ~~of~~ meta-rights;

a rights composer module for composing a draft license;

a repository for supplier's rights;

a supplier management database; and

a consumer component comprising:

a consumer user interface module;

an offer-consideration module configured to analyze the offers generated by the supplier component and select offers based on the analysis;

a repository for consumer's rights;

a consumer management database; and

a communication link coupling said supplier component and said consumer component.

16. (Original) A system as recited in claim 15, wherein the supplier component further comprises offer-templates and consumer profile information, wherein said offer-template contains one or more predetermined usage rights and meta-rights, and wherein said consumer profile information comprises at least one of consumer identity information, account information, purchase history information, consumer preferences information, and credit rating information.

17. (Original) A system as recited in claim 15, wherein said consumer component further comprises a supplier-preference module for providing supplier information.

18. (Original) The system of claim 15, wherein said offer-consideration module comprises:

means for determining if the consumer can accept an offer;

means for applying selection logic to the offer;

means for specifying contingencies; and

means for authenticating choices and providing the choices to said supplier component.

19. (Original) The system of claim 18, wherein said means for applying comprises:

means for parsing the offer and selecting preferred usage rights and meta-rights in the offer;

means for filtering offers based on supplier preferences;

means for applying consumer preferences; and

means for selecting options based on the output of said means for parsing, said means for filtering, and said means for applying consumer preferences.

20. (Original) A method for generating a license to digital content to be used within a system for at least one of managing use and distribution of the digital content, said method comprising:

presenting a consumer with an offer including meta-rights;

receiving a selection by the consumer of at least one meta-right in the offer;

generating a license based on the selection, wherein the license permits the consumer to exercise the at least one meta-right and permits the consumer to offer at least one derived right derived from the at least one meta-right and generate a license including the at least one derived right.

21. (Currently amended) A method as recited in claim ~~15~~ 20, wherein the at least one derived right in the generated license ~~for the second party~~ includes usage rights to be exercised by ~~the~~ a second party and meta-rights permitting derived rights to be ~~offere~~ offered to a third party.

22. (New) A method as recited in claim 4, comprising:

specifying in said second license one or more conditions which must be satisfied in order for said derived rights to be exercised; and

associating at least one of said conditions with at least one state variable.

23. (New) A method as recited in claim 22, wherein said state variable inherits its state for content usage from said usage rights in said first license.

24. (New) A method as recited in claim 22, wherein said state variable shares its state for content usage with said usage rights in said first license.

25. (New) A method as recited in claim 22, wherein said state variable inherits a remaining state for content usage from said usage rights in said first license.

26. (New) A method as recited in claim 22, comprising:

associating at least one of said conditions with a plurality state variables, wherein said plurality of state variables collectively track a state of a right.

27. (New) A method as recited in claim 2, wherein said license specifies at least one state variable related to said one or more conditions.

28. (New) A method as recited in claim 27, comprising:

updating said state variable while generating usage rights and meta-rights from an offer, wherein said state variable includes at least one of information of a user, a preference of a user, a number of times content has been used, coupons given, and information of a supplier.

REMARKS

The present amendment amends claims 1, 4, 8, 9, 12, 13, 14, 15, and 21 to correct discovered informalities and adds new claims 22-28 directed to further features of Applicant's invention. No new matter is introduced.

In view of the foregoing, examination on the merits of claims 1-28 is respectfully requested. The Examiner is invited to contact the undersigned attorney to expedite the prosecution of the present case.

Respectfully submitted,

NIXON PEABODY, LLP

Carlos R. Villamar
Registration No. 43,224

Date: July 15, 2004

NIXON PEABODY LLP
CUSTOMER NO.: 22204
401 9th Street, N.W., Suite 900
Washington, DC 20004
Tel: 202-585-8000
Fax: 202-585-8080

MSK/CRV:kla

IFW

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin WANG, *et al.*
Group Art Unit	3621
Examiner Name	Unassigned
Total Number of Pages in This Submission	
Attorney Docket Number	111325-104 (230300)

ENCLOSURES *(check all that apply)*

- ☐ Fee Transmittal Form
  - ☐ 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)
- ☐ Response to Missing Parts/ Incomplete Application
  - ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Assignment Papers *(for an Application)*
- ☐ Drawing(s)
- ☐ 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
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to Group (Appeal Notice, Brief, Reply Brief)
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Application Data Sheet
- ☐ Request for Corrected Filing Receipt with Enclosures
- ☐ A self-addressed prepaid postcard for acknowledging receipt
- ☐ Other Enclosure(s) (*please identify below*):

Remarks

☒ 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	Marc S. Kaufman Registration No. 35,212 **Nixon Peabody LLP** 401 9th Street, N.W., Suite 900 Washington, D.C. 20004-2128
Signature	
Date	August 5, 2004

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(a)]

I hereby certify that this correspondence is being:

- ☐ deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop _____, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450
- ☐ transmitted by facsimile on the date shown below to the United States Patent and Trademark Office at (703) _____.

_____ Date

_____ Signature

_____ Typed or printed name

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: Xin **WANG**, *et al.*	)	Examiner: Unassigned
Application No.: 10/162,212	)	Group Art Unit: 3621
Filed: June 5, 2002	)	
For: **Rights Offering and Granting**	)	

Commissioner of Patents
U.S. Patent and Trademark Office
220 20th Street S.
Customer Window
Crystal Plaza Two, Lobby, Room 1B03
Arlington, VA 22202

Sir:

INFORMATION DISCLOSURE STATEMENT UNDER 37 C.F.R. § 1.97 (b)

Pursuant to 37 C.F.R. §§ 1.56 and 1.97(b), Applicants bring to the attention of the Examiner the documents listed on the attached PTO-1449. This Information Disclosure Statement is being filed before the mailing date of the first Office Action on the merits for the above-referenced application. The listed documents were cited in a communication from the European Patent Office. The Search Report was mailed on April 26, 2004. Copies of the listed documents are attached.

It is requested that the accompanying PTO-1449 be considered and made of record in the above-identified 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. (111325-104/230300).

Respectfully submitted,

NIXON PEABODY, LLP

By: _____

Marc S. Kaufman
Registration No.: 35,212

Dated: **August 5, 2005**

NIXON PEABODY LLP
Customer No.: 22204
401 9th Street, N.W., Suite 900
Washington, DC 20004-2128
Telephone: (202) 585-8000
FAX: (202) 585-8080

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(use as many sheets as necessary)

Sheet 1 of 1

Complete if Known	
Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin WANG, *et al.*
Art Unit	3621
Examiner Name	Unassigned
Attorney Docket Number	111325-104 (230300)

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	U.S. Patent Document: Number - Kind Code[2] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US – 5,917,912	June 29, 1999	Ginter, et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document: Country Code[3] Number[4] Kind Code[5] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Application of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]
		EP 0 715 246 A	June 5, 1996			
		WO 01 13198 A	February 22, 2001			
		EP 0 818 748 A	January 14, 1998			

OTHER PRIOR ART – NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		European Search Report dated April 26, 2004 (European Patent Application No. 02 739 696.9)	

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference 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 form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

[1] Applicant's unique citation designation number (optional). [2] Applicant is to place a check mark here if English language Translation is attached.

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

(11) EP 0 715 246 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
05.06.1996 Bulletin 1996/23

(51) Int Cl.6: **G06F 1/00**

(21) Application number: **95308421.7**

(22) Date of filing: **23.11.1995**

(84) Designated Contracting States:
DE FR GB

(30) Priority: **23.11.1994 US 344776**

(71) Applicant: **XEROX CORPORATION**
Rochester New York 14644 (US)

(72) Inventors:
- **Stefik, Mark J.**
  Woodside, California 94062 (US)
- **Pirolli, Peter L. T.**
  El Cerrito, California 94530 (US)
- **Bobrow, Daniel G.**
  Palo Alto, Caliofrnia 94301 (US)

(74) Representative: **Goode, Ian Roy**
Rank Xerox Ltd
Patent Department
Parkway
Marlow Buckinghamshire SL7 1YL (GB)

(54) System for controlling the distribution and use of composite digital works

(57) A system for controlling use and distribution of composite digital works. A digital work is comprised of a description part (700) and a content part. The description part contains control information for the composite digital work. The content part stores the actual digital data comprising the composite digital work. The description part (700) is logically organized in an acyclic structure, e.g. a tree structure. For a composite digital work each node of the acyclic structure represents an individual digital work or some distribution interest in the composite digital work. A node in the acyclic structure is comprised of an identifier (701) of the individual work, usage rights (704) for the individual digital work and a pointer (705.706) to the digital work. Composite digital works are stored in repositories. A repository has two primary operating modes. a server mode and a requester mode. When operating in a server mode, the repository is responding to requests to access digital works. When operating in requester mode. the repository is requesting access to a digital work. A repository will process each request to access a composite digital work by examining the usage rights for each individual digital work found in the description part of the composite digital work.

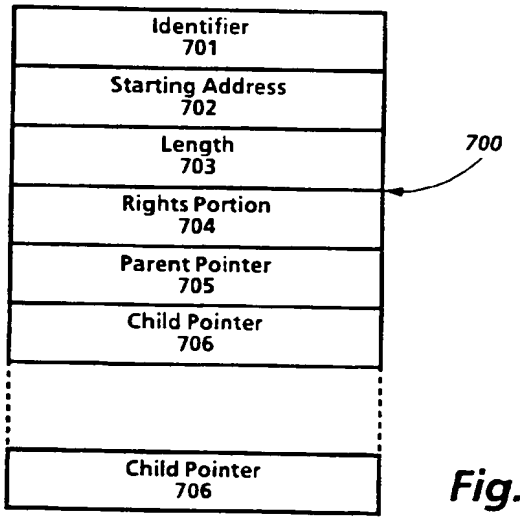


Fig. 7

EP 0 715 246 A1

Printed by Jouve 75001 PARIS (FR)

Description

The present invention relates to the field of distribution and usage rights enforcement for digitally encoded works

A fundamental issue facing the publishing and information industries as they consider electronic publishing is how to prevent the unauthorized and unaccounted distribution or usage of electronically published materials. Electronically published materials are typically distributed in a digital form and recreated on a computer based system having the capability to recreate the materials. Audio and video recordings, software, books and multimedia works are all being electronically published. Companies in these industries receive royalties for each accounted for delivery of the materials, e.g. the sale of an audio CD at a retail outlet. Any unaccounted distribution of a work results in an unpaid royalty (e.g. copying the audio recording CD to another digital medium.)

The ease in which electronically published works can be "perfectly" reproduced and distributed is a major concern. The transmission of digital works over networks is commonplace. One such widely used network is the Internet. The Internet is a widespread network facility by which computer users in many universities, corporations and government entities communicate and trade ideas and information. Computer bulletin boards found on the Internet and commercial networks such as CompuServ and Prodigy allow for the posting and retrieving of digital information. Information services such as Dialog and LEXIS/NEXIS provide databases of current information on a wide variety of topics. Another factor which will exacerbate the situation is the development and expansion of the National Information Infrastructure (the NII). It is anticipated that, as the NII grows, the transmission of digital works over networks will increase many times over. It would be desirable to utilize the NII for distribution of digital works without the fear of widespread unauthorized copying.

The most straightforward way to curb unaccounted distribution is to prevent unauthorized copying and transmission. For existing materials that are distributed in digital form, various safeguards are used. In the case of software, copy protection schemes which limit the number of copies that can be made or which corrupt the output when copying is detected have been employed. Another scheme causes software to become disabled after a predetermined period of time has lapsed. A technique used for workstation based software is to require that a special hardware device must be present on the workstation in order for the software to run, e.g., see US-A-4.932.054 entitled "Method and Apparatus for Protecting Computer Software Utilizing Coded Filter Network in Conjunction with an Active Coded Hardware Device." Such devices are provided with the software and are commonly referred to as dongles.

Yet another scheme is to distribute software, but which requires a "key" to enable its use. This is employed in distribution schemes where "demos" of the software are provided on a medium along with the entire product. The demos can be freely used, but in order to use the actual product, the key must be purchased. These schemes do not hinder copying of the software once the key is initially purchased.

It is an object of the present invention to provide an improved system and method for controlling the use and distribution of digital works.

The invention accordingly provides a system and method as claimed in the accompanying claims.

A system for controlling use and distribution of composite digital works is disclosed. A composite digital work is comprised of one or more individual digital works. An individual digital work is any written, aural, graphical or video based work that has been translated to or created in a digital form, and which can be recreated using suitable rendering means such as software programs. A folder containing one or more digital works may be treated as a composite digital work. The present invention allows the owner of an individual digital work to attach usage rights to their work which are honored when the individual digital work is incorporated into a composite digital work. The usage rights for the work define how the individual digital work may be used and distributed. The aggregation of the usage rights of the individual digital works of a composite digital work, as well as usage rights attached to the composite digital work as a whole define how the composite digital work may be used and distributed.

A digital work is comprised of a description part and a content part. The description part contains control information for the composite digital work. The content part stores the actual digital data comprising the composite digital work. The description part is logically organized in an acyclic structure (e.g. a tree structure.) For a composite digital work each node of the acyclic structure represents an individual digital work or some distribution interest in the digital work. A node in the acyclic structure is comprised of an identifier of the individual work, usage rights for the individual digital work and a pointer to the digital work. In this representation, the description part may naturally be stored separately on a separate medium from the content part.

Composite digital works are stored in repositories. A repository is comprised of a storage means for storing a digital work and its attached usage rights, an external interface for receiving and transmitting data, a processor and a clock. A repository has two primary operating modes, a server mode and a requester mode. When operating in a server mode, the repository is responding to requests to access digital works. When operating in requester mode, the repository is requesting access to a digital work. A repository will process each request to access a composite digital work by examining the usage rights for each individual digital work found in the description part. Access is granted if the composite digital work if access to each of the individual digital works can be granted. Alternatively, if access to all the

individual digital works cannot be granted. partial access can be granted only to those individual digital works which grant access

A system and method in accordance with the invention will now be described. by way of example with reference to the accompanying drawings. in which .-

Figure 1 is a flowchart illustrating a simple instantiation of the operation of the currently preferred embodiment of the present invention.

Figure 2 is a block diagram illustrating the various repository types and the repository transaction flow between them in the currently preferred embodiment of the present invention.

Figure 3 is a block diagram of a repository coupled with a credit server in the currently preferred embodiment of the present invention.

Figures 4a and 4b are examples of rendering systems as may be utilized in the currently preferred embodiment of the present invention.

Figure 5 illustrates a contents file layout for a digital work as may be utilized in the currently preferred embodiment of the present invention.

Figure 6 illustrates a contents file layout for an individual digital work of the digital work of Figure 5 as may be utilized in the currently preferred embodiment of the present invention.

Figure 7 illustrates the components of a description block of the currently preferred embodiment of the present invention.

Figure 8 illustrates a description tree for the contents file layout of the digital work illustrated in Figure 5.

Figure 9 illustrates a portion of a description tree corresponding to the individual digital work illustrated in Figure 6.

Figure 10 illustrates a layout for the rights portion of a description block as may be utilized in the currently preferred embodiment of the present invention.

Figure 11 is a description tree wherein certain d-blocks have PRINT usage rights and is used to illustrate "strict" and "lenient" rules for resolving usage rights conflicts.

Figure 12 is a block diagram of the hardware components of a repository as are utilized in the currently preferred embodiment of the present invention.

Figure 13 is a block diagram of the functional (logical) components of a repository as are utilized in the currently preferred embodiment of the present invention.

Figure 14 is diagram illustrating the basic components of a usage right in the currently preferred embodiment of the present invention.

Figure 15 lists the usage rights grammar of the currently preferred embodiment of the present invention.

Figure 16 is a flowchart illustrating the steps of certificate delivery. hotlist checking and performance testing as performed in a registration transaction as may be performed in the currently preferred embodiment of the present invention.

Figure 17 is a flowchart illustrating the steps of session information exchange and clock synchronization as may be performed in the currently preferred embodiment of the present invention. after each repository in the registration transaction has successfully completed the steps described in Figure 16.

Figure 18 is a flowchart illustrating the basic flow for a usage transaction. including the common opening and closing step, as may be performed in the currently preferred embodiment of the present invention.

Figure 19 is a state diagram of server and client repositories in accordance with a transport protocol followed when moving a digital work from the server to the client repositories. as may be performed in the currently preferred embodiment of the present invention.

OVERVIEW

A system for controlling use and distribution of digital works is disclosed. The present invention is directed to supporting commercial transactions involving digital works.

Herein the terms "digital work", "work" and "content" refer to any work that has been reduced to a digital representation. This would include any audio, video, text, or multimedia work and any accompanying interpreter (e.g. software) that may be required for recreating the work. The term composite work refers to a digital work comprised of a collection of other digital works. The term "usage rights" or "rights" is a term which refers to rights granted to a recipient of a digital work. Generally, these rights define how a digital work can be used and if it can be further distributed. Each usage right may have one or more specified conditions which must be satisfied before the right may be exercised.

Figure 1 is a high level flowchart omitting various details but which demonstrates the basic operation of the present invention. Referring to Figure 1, a creator creates a digital work step 101. The creator will then determine appropriate usage rights and fees. attach them to the digital work. and store them in Repository 1. step 102. The determination of appropriate usage rights and fees will depend on various economic factors The digital work remains securely in Repository 1 until a request for access is received. The request for access begins with a session initiation by another

repository. Here a Repository 2 initiates a session with Repository 1 step 103. As will be described in greater detail below, this session initiation includes steps which helps to insure that the respective repositories are trustworthy. Assuming that a session can be established, Repository 2 may then request access to the Digital Work for a stated purpose, step 104. The purpose may be, for example, to print the digital work or to obtain a copy of the digital work. The purpose will correspond to a specific usage right. In any event, Repository 1 checks the usage rights associated with the digital work to determine if the access to the digital work may be granted, step 105. The check of the usage rights essentially involves a determination of whether a right associated with the access request has been attached to the digital work and if all conditions associated with the right are satisfied. If the access is denied, repository 1 terminates the session with an error message, step 106. If access is granted, repository 1 transmits the digital work to repository 2, step 107. Once the digital work has been transmitted to repository 2, repository 1 and 2 each generate billing information for the access which is transmitted to a credit server, step 108. Such double billing reporting is done to insure against attempts to circumvent the billing process.

Figure 2 illustrates the basic interactions between repository types in the present invention. As will become apparent from Figure 2, the various repository types will serve different functions. It is fundamental that repositories will share a core set of functionality which will enable secure and trusted communications. Referring to Figure 2, a repository 201 represents the general instance of a repository. The repository 201 has two modes of operation: a server mode and a requester mode. When in the server mode, the repository will be receiving and processing access requests to digital works. When in the requester mode, the repository will be initiating requests to access digital works. Repository 201 is general in the sense that its primary purpose is as an exchange medium for digital works. During the course of operation, the repository 201 may communicate with a plurality of other repositories, namely authorization repository 202, rendering repository 203 and master repository 204. Communication between repositories occurs utilizing a repository transaction protocol 205.

Communication with an authorization repository 202 may occur when a digital work being accessed has a condition requiring an authorization. Conceptually, an authorization is a digital certificate such that possession of the certificate is required to gain access to the digital work. An authorization is itself a digital work that can be moved between repositories and subjected to fees and usage rights conditions. An authorization may be required by both repositories involved in an access to a digital work.

Communication with a rendering repository 203 occurs in connection with the rendering of a digital work. As will be described in greater detail below, a rendering repository is coupled with a rendering device (e.g. a printer device) to comprise a rendering system.

Communication with a master repository 205 occurs in connection with obtaining an identification certificate. Identification certificates are the means by which a repository is identified as "trustworthy". The use of identification certificates is described below with respect to the registration transaction.

Figure 3 illustrates the repository 201 coupled to a credit server 301. The credit server 301 is a device which accumulates billing information for the repository 201. The credit server 301 communicates with repository 201 via billing transactions 302 to record billing transactions. Billing transactions are reported to a billing clearinghouse 303 by the credit server 301 on a periodic basis. The credit server 301 communicates to the billing clearinghouse 303 via clearinghouse transactions 304. The clearinghouse transactions 304 enable a secure and encrypted transmission of information to the billing clearinghouse 303.

RENDERING SYSTEMS

A rendering system is generally defined as a system comprising a repository and a rendering device which can render a digital work into its desired form. Examples of a rendering system may be a computer system, a digital audio system, or a printer. A rendering system has the same security features as a repository. The coupling of a rendering repository with the rendering device may occur in a manner suitable for the type of rendering device.

Figure 4a illustrates a printer as an example of a rendering system. Referring to Figure 4, printer system 401 has contained therein a printer repository 402 and a print device 403. It should be noted that the the dashed line defining printer system 401 defines a secure system boundary. Communications within the boundary are assumed to be secure. Depending on the security level, the boundary also represents a barrier intended to provide physical integrity. The printer repository 402 is an instantiation of the rendering repository 205 of Figure 2. The printer repository 402 will in some instances contain an ephemeral copy of a digital work which remains until it is printed out by the print engine 403. In other instances, the printer repository 402 may contain digital works such as fonts, which will remain and can be billed based on use. This design assures that all communication lines between printers and printing devices are encrypted, unless they are within a physically secure boundary. This design feature eliminates a potential "fault" point through which the digital work could be improperly obtained. The printer device 403 represents the printer components used to create the printed output.

Also illustrated in Figure 4a is the repository 404. The repository 404 is coupled to the printer repository 402. The

repository 404 represents an external repository which contains digital works.

Figure 4b is an example of a computer system as a rendering system. A computer system may constitute a "multi-function" device since it may execute digital works (e.g. software programs) and display digital works (e.g. a digitized photograph). Logically, each rendering device can be viewed as having its own repository, although only one physical repository is needed. Referring to Figure 4b, a computer system 410 has contained therein a display/execution repository 411. The display/execution repository 411 is coupled to display device 412 and execution device 413. The dashed box surrounding the computer system 410 represents a security boundary within which communications are assumed to be secure. The display/execution repository 411 is further coupled to a credit server 414 to report any fees to be billed for access to a digital work and a repository 415 for accessing digital works stored therein.

STRUCTURE OF DIGITAL WORKS

Usage rights are attached directly to digital works. Thus, it is important to understand the structure of a digital work. The structure of a digital work, in particular composite digital works, may be naturally organized into an acyclic structure such as a hierarchy. For example, a magazine has various articles and photographs which may have been created and are owned by different persons. Each of the articles and photographs may represent a node in a hierarchical structure. Consequently, controls, i.e. usage rights, may be placed on each node by the creator. By enabling control and fee billing to be associated with each node, a creator of a work can be assured that the rights and fees are not circumvented.

In the currently preferred embodiment, the file information for a digital work is divided into two files, a "contents" file and a "description tree" file. From the perspective of a repository, the "contents" file is a stream of addressable bytes whose format depends completely on the interpreter used to play, display or print the digital work. The description tree file makes it possible to examine the rights and fees for a work without reference to the content of the digital work. It should be noted that the term description tree as used herein refers to any type of acyclic structure used to represent the relationship between the various components of a digital work.

Figure 5 illustrates the layout of a contents file. Referring to Figure 5, a digital work is comprised of story A 510, advertisement 511, story B 512 and story C 513. It is assumed that the digital work is stored starting at a relative address of 0. Each of the parts of the digital work are stored linearly so that story A 510 is stored at approximately addresses 0-30,000, advertisement 511 at addresses 30,001-40,000, story B 512 at addresses 40,001-60,000 and story C 513 at addresses 60,001-85K. The detail of story A 510 is illustrated in Figure 6. Referring to Figure 6, the story A 510 is further broken down to show text 614 stored at address 0-1500, soldier photo 615 at addresses 1501-10,000, graphics 616 stored at addresses 10,001-25,000 and sidebar 617 stored address 25,001-30,000. Note that the data in the contents file may be compressed (for saving storage) or encrypted (for security).

From Figures 5 and 6 it is readily observed that a digital work can be represented by its component parts as a hierarchy. The description tree for a digital work is comprised of a set of related descriptor blocks (d-blocks). The contents of each d-block is described with respect to Figure 7. Referring to Figure 7, a d-block 700 includes an identifier 701 which is a unique identifier for the work in the repository, a starting address 702 providing the start address of the first byte of the work, a length 703 giving the number of bytes in the work, a rights portion 704 wherein the granted usage rights and their status data are maintained, a parent pointer 705 for pointing to a parent d-block and child pointers 706 for pointing to the child d-blocks. In the currently preferred embodiment, the identifier 701 has two parts. The first part is a unique number assigned to the repository upon manufacture. The second part is a unique number assigned to the work upon creation. The rights portion 704 will contain a data structure, such as a look-up table, wherein the various information associated with a right is maintained. The information required by the respective usage rights is described in more detail below. D-blocks form a strict hierarchy. The top d-block of a work has no parent; all other d-blocks have one parent. The relationship of usage rights between parent and child d-blocks and how conflicts are resolved is described below.

A special type of d-block is a "shell" d-block. A shell d-block adds no new content beyond the content of its parts. A shell d-block is used to add rights and fee information, typically by distributors of digital works.

Figure 8 illustrates a description tree for the digital work of Figure 5. Referring to Figure 8, a top d-block 820 for the digital work points to the various stories and advertisements contained therein. Here, the top d-block 820 points to d-block 821 (representing story A 510), d-block 822 (representing the advertisement 511), d-block 823 (representing story B 512) and and d-block 824 (representing story C 513).

The portion of the description tree for Story A 510 is illustrated in Figure 9. D-block 925 represents text 614, d-block 926 represents photo 615, d-block 927 represents graphics 616 by and d-block 928 represents sidebar 617.

The rights portion 704 of a descriptor block is further illustrated in Figure 10. Figure 10 illustrates a structure which is repeated in the rights portion 704 for each right. Referring to Figure 10, each right will have a right code field 1050 and status information field 1052. The right code field 1050 will contain a unique code assigned to a right. The status information field 1052 will contain information relating to the state of a right and the digital work. Such information is

indicated below in Table 1. The rights as stored in the rights portion 704 may typically be in numerical order based on the right code

TABLE 1

DIGITAL WORK STATE INFORMATION		
Property	**Value**	**Use**
Copies-in-Use	Number	A counter of the number of copies of a work that are in use. Incremented when another copy is used; decremented when use is completed.
Loan-Period	Time-Units	Indicator of the maximum number of time-units that a document can be loaned out
Loaner-Copy	Boolean	Indicator that the current work is a loaned out copy of an authorized digital work.
Remaining-Time	Time-Units	Indicator of the remaining time of use on a metered document right.
Document-Descr	String	A string containing various identifying information about a document. The exact format of this is not specified, but it can include information such as a publisher name, author name, ISBN number, and so on.
Revenue-Owner	RO-Descr	A handle identifying a revenue owner for a digital work. This is used for reporting usage fees.
Publication-Date	Date-Descr	The date that the digital work was published.
History-list	History-Rec	A list of events recording the repositories and dates for operations that copy, transfer, backup, or restore a digital work.

The approach for representing digital works by separating description data from content assumes that parts of a file are contiguous but takes no position on the actual representation of content. In particular, it is neutral to the question of whether content representation may take an object oriented approach. It would be natural to represent content as objects. In principle, it may be convenient to have content objects that include the billing structure and rights information that is represented in the d-blocks. Such variations in the design of the representation are possible and are viable alternatives but may introduce processing overhead, e.g. the interpretation of the objects.

Digital works are stored in a repository as part of a hierarchical file system. Folders (also termed directories and sub-directories) contain the digital works as well as other folders. Digital works and folders in a folder are ordered in alphabetical order. The digital works are typed to reflect how the files are used. Usage rights can be attached to folders so that the folder itself is treated as a digital work. Access to the folder would then be handled in the same fashion as any other digital work As will be described in more detail below, the contents of the folder are subject to their own rights. Moreover, file management rights may be attached to the folder which define how folder contents can be managed.

ATTACHING USAGE RIGHTS TO A DIGITAL WORK

It is fundamental to the present invention that the usage rights are treated as part of the digital work. As the digital work is distributed, the scope of the granted usage rights will remain the same or may be narrowed. For example, when a digital work is transferred from a document server to a repository, the usage rights may include the right to loan a copy for a predetermined period of time (called the original rights). When the repository loans out a copy of the digital work, the usage rights in the loaner copy (called the next set of rights) could be set to prohibit any further rights to loan out the copy. The basic idea is that one cannot grant more rights than they have.

The attachment of usage rights into a digital work may occur in a variety of ways. If the usage rights will be the same for an entire digital work, they could be attached when the digital work is processed for deposit in the digital work server. In the case of a digital work having different usage rights for the various components, this can be done as the digital work is being created. An authoring tool or digital work assembling tool could be utilized which provides for an automated process of attaching the usage rights.

As will be described below, when a digital work is copied, transferred or loaned, a "next set of rights" can be specified. The"next set of rights" will be attached to the digital work as it is transported.

Resolving Conflicting Rights

Because each part of a digital work may have its own usage rights, there will be instances where the rights of a

"contained part" are different from its parent or container part. As a result conflict rules must be established to dictate when and how a right may be exercised. The hierarchical structure of a digital work facilitates the enforcement of such rules. A "strict" rule would be as follows: a right for a part in a digital work is sanctioned if and only if it is sanctioned for the part, for ancestor d-blocks containing the part and for all descendent d-blocks. By sanctioned it is meant that (1) each of the respective parts must have the right, and (2) any conditions for exercising the right are satisfied.

It also possible to implement the present invention using a more lenient rule. In the more lenient rule access to the part may be enabled to the descendent parts which have the right, but access is denied to the descendents which do not.

An example of applying both the strict rule and lenient is illustrated with reference to Figure 11. Referring to Figure 11, a root d-block 1101 has child d-blocks 1102-1105. In this case, root d-block represents a magazine, and each of the child d-blocks 1102-1105 represent articles in the magazine. Suppose that a request is made to PRINT the digital work represented by root d-block 1101 wherein the strict rule is followed. The rights for the root d-block 1101 and child d-blocks 1102-1105 are then examined. Root d-block 1101 and child d-blocks 1102 and 1105 have been granted PRINT rights. Child d-block 1103 has not been granted PRINT rights and child d-block 1104 has PRINT rights conditioned on payment of a usage fee.

Under the strict rule the PRINT right cannot be exercised because the child d-block does not have the PRINT right. Under the lenient rule, the result would be different. The digital works represented by child d-blocks 1102 and 1105 could be printed and the digital work represented by d-block 1104 could be printed so long as the usage fee is paid. Only the digital work represented by d-block 1103 could not be printed. This same result would be accomplished under the strict rule if the requests were directed to each of the individual digital works.

The present invention supports various combinations of allowing and disallowing access. Moreover, as will be described below, the usage rights grammar permits the owner of a digital work to specify if constraints may be imposed on the work by a container part. The manner in which digital works may be sanctioned because of usage rights conflicts would be implementation specific and would depend on the nature of the digital works.

REPOSITORIES

In the description of Figure 2, it was indicated that repositories come in various forms. All repositories provide a core set of services for the transmission of digital works. The manner in which digital works are exchanged is the basis for all transaction between repositories. The various repository types differ in the ultimate functions that they perform. Repositories may be devices themselves, or they may be incorporated into other systems. An example is the rendering repository 203 of Figure 2.

A repository will have associated with it a repository identifier. Typically, the repository identifier would be a unique number assigned to the repository at the time of manufacture. Each repository will also be classified as being in a particular security class. Certain communications and transactions may be conditioned on a repository being in a particular security class. The various security classes are described in greater detail below.

As a prerequisite to operation, a repository will require possession of an identification certificate. Identification certificates are encrypted to prevent forgery and are issued by a Master repository. A master repository plays the role of an authorization agent to enable repositories to receive digital works. Identification certificates must be updated on a periodic basis. Identification certificates are described in greater detail below with respect to the registration transaction.

A repository has both a hardware and functional embodiment. The functional embodiment is typically software executing on the hardware embodiment. Alternatively, the functional embodiment may be embedded in the hardware embodiment such as an Application Specific Integrated Circuit (ASIC) chip.

The hardware embodiment of a repository will be enclosed in a secure housing which if compromised, may cause the repository to be disabled. The basic components of the hardware embodiment of a repository are described with reference to Figure 12. Referring to Figure 12, a repository is comprised of a processing means 1200, storage system 1207, clock 1205 and external interface 1206. The processing means 1200 is comprised of a processor element 1201 and processor memory 1202. The processing means 1201 provides controller, repository transaction and usage rights transaction functions for the repository. Various functions in the operation of the repository such as decryption and/or decompression of digital works and transaction messages are also performed by the processing means 1200. The processor element 1201 may be a microprocessor or other suitable computing component. The processor memory 1202 would typically be further comprised of Read Only Memories (ROM) and Random Access Memories (RAM). Such memories would contain the software instructions utilized by the processor element 1201 in performing the functions of the repository.

The storage system 1207 is further comprised of descriptor storage 1203 and content storage 1204. The description tree storage 1203 will store the description tree for the digital work and the content storage will store the associated content. The description tree storage 1203 and content storage 1204 need not be of the same type of storage medium,

nor are they necessarily on the same physical device. So for example the descriptor storage 1203 may be stored on a solid state storage (for rapid retrieval of the description tree information), while the content storage 1204 may be on a high capacity storage such as an optical disk.

The clock 1205 is used to time-stamp various time based conditions for usage rights or for metering usage fees which may be associated with the digital works. The clock 1205 will have an uninterruptable power supply, e.g. a battery, in order to maintain the integrity of the time-stamps. The external interface means 1206 provides for the signal connection to other repositories and to a credit server. The external interface means 1206 provides for the exchange of signals via such standard interfaces such as RS-232 or Personal Computer Manufacturers Card Industry Association (PCMCIA) standards, or FDDI. The external interface means 1206 may also provide network connectivity

The functional embodiment of a repository is described with reference to Figure 13. Referring to Figure 13, the functional embodiment is comprised of an operating system 1301, core repository services 1302, usage transaction handlers 1303, repository specific functions, 1304 and a user interface 1305. The operating system 1301 is specific to the repository and would typically depend on the type of processor being used. The operating system 1301 would also provide the basic services for controlling and interfacing between the basic components of the repository.

The core repository services 1302 comprise a set of functions required by each and every repository. The core repository services 1302 include the session initiation transactions which are defined in greater detail below. This set of services also includes a generic ticket agent which is used to "punch" a digital ticket and a generic authorization server for processing authorization specifications. Digital tickets and authorizations are specific mechanisms for controlling the distribution and use of digital works and are described in more detail below. Note that coupled to the core repository services are a plurality of identification certificates 1306. The identification certificates 1306 are required to enable the use of the repository.

The usage transactions handlers 1303 comprise functionality for processing access requests to digital works and for billing fees based on access. The usage transactions supported will be different for each repository type. For example, it may not be necessary for some repositories to handle access requests for digital works.

The repository specific functionality 1304 comprises functionality that is unique to a repository. For example, the master repository has special functionality for issuing digital certificates and maintaining encryption keys. The repository specific functionality 1304 would include the user interface implementation for the repository.

Repository Security Classes

For some digital works the losses caused by any individual instance of unauthorized copying is insignificant and the chief economic concern lies in assuring the convenience of access and low-overhead billing. In such cases, simple and inexpensive handheld repositories and network-based workstations may be suitable repositories, even though the measures and guarantees of security are modest.

At the other extreme, some digital works such as a digital copy of a first run movie or a bearer bond or stock certificate would be of very high value so that it is prudent to employ caution and fairly elaborate security measures to ensure that they are not copied or forged. A repository suitable for holding such a digital work could have elaborate measures for ensuring physical integrity and for verifying authorization before use.

By arranging a universal protocol, all kinds of repositories can communicate with each other in principle. However, creators of some works will want to specify that their works will only be transferred to repositories whose level of security is high enough. For this reason, document repositories have a ranking system for classes and levels of security. The security classes in the currently preferred embodiment are described in Table 2.

TABLE 2

REPOSITORY SECURITY LEVELS	
Level	Description of Security
0	Open system. Document transmission is unencrypted. No digital certificate is required for identification. The security of the system depends mostly on user honesty, since only modest knowledge may be needed to circumvent the security measures. The repository has no provisions for preventing unauthorized programs from running and accessing or copying files. The system does not prevent the use of removable storage and does not encrypt stored files.
1	Minimal security. Like the previous class except that stored files are minimally encrypted, including ones on removable storage.

Continuation of the Table on the next page

TABLE 2 (continued)

Level	Description of Security
2	Basic security. Like the previous class except that special tools and knowledge are required to compromise the programming, the contents of the repository, or the state of the clock. All digital communications are encrypted. A digital certificate is provided as identification. Medium level encryption is used. Repository identification number is unforgeable.
3	General security. Like the previous class plus the requirement of special tools are needed to compromise the physical integrity of the repository and that modest encryption is used on all transmissions. Password protection is required to use the local user interface. The digital clock system cannot be reset without authorization. No works would be stored on removable storage. When executing works as programs, it runs them in their own address space and does not give them direct access to any file storage or other memory containing system code or works. Then can access works only through the transmission transaction protocol.
4	Like the previous class except that high level encryption is used on all communications. Sensors are used to record attempts at physical and electronic tampering. After such tampering, the repository will not perform other transactions until it has reported such tampering to a designated server.
5	Like the previous class except that if the physical or digital attempts at tampering exceed some preset thresholds that threaten the physical integrity of the repository or the integrity of digital and cryptographic barriers, then the repository will save only document description records of history but will erase or destroy any digital identifiers that could be misused if released to an unscrupulous party. It also modifies any certificates of authenticity to indicate that the physical system has been compromised. It also erases the contents of designated documents.
6	Like the previous class except that the repository will attempt wireless communication to report tampering and will employ noisy alarms.
10	This would correspond to a very high level of security. This server would maintain constant communications to remote security systems reporting transactions, sensor readings, and attempts to circumvent security.

The characterization of security levels described in Table 2 is not intended to be fixed. More important is the idea of having different security levels for different repositories. It is anticipated that new security classes and requirements will evolve according to social situations and changes in technology.

Repository User Interface

A user interface is broadly defined as the mechanism by which a user interacts with a repository in order to invoke transactions to gain access to a digital work, or exercise usage rights. As described above, a repository may be embodied in various forms. The user interface for a repository will differ depending on the particular embodiment. The user interface may be a graphical user interface having icons representing the digital works and the various transactions that may be performed. The user interface may be a generated dialog in which a user is prompted for information.

The user interface itself need not be part of the repository. As a repository may be embedded in some other device, the user interface may merely be a part of the device in which the repository is embedded. For example, the repository could be embedded in a "card" that is inserted into an available slot in a computer system. The user interface may be a combination of a display, keyboard, cursor control device and software executing on the computer system.

At a minimum, the user interface must permit a user to input information such as access requests and alpha numeric data and provide feedback as to transaction status. The user interface will then cause the repository to initiate the suitable transactions to service the request. Other facets of a particular user interface will depend on the functionality that a repository will provide.

CREDIT SERVERS

In the present invention, fees may be associated with the exercise of a right. The requirement for payment of fees is described with each version of a usage right in the usage rights language. The recording and reporting of such fees is performed by the credit server. One of the capabilities enabled by associating fees with rights is the possibility of supporting a wide range of charging models. The simplest model, used by conventional software, is that there is a

single fee at the time of purchase, after which the purchaser obtains unlimited rights to use the work as often and for as long as he or she wants. Alternative models include metered use and variable fees. A single work can have different fees for different uses. For example, viewing a photograph on a display could have different fees than making a hardcopy or including it in a newly created work. A key to these alternative charging models is to have a low overhead means of establishing fees and accounting for credit on these transactions.

A credit server is a computational system that reliably authorizes and records these transactions so that fees are billed and paid. The credit server reports fees to a billing clearinghouse. The billing clearinghouse manages the financial transactions as they occur. As a result, bills may be generated and accounts reconciled. Preferably the credit server would store the fee transactions and periodically communicate via a network with the billing clearinghouse for reconciliation. In such an embodiment, communications with the billing clearinghouse would be encrypted for integrity and security reasons. In another embodiment, the credit server acts as a "debit card" where transactions occur in "real-time" against a user account.

A credit server is comprised of memory, a processing means, a clock, and interface means for coupling to a repository and a financial institution (e.g. a modem). The credit server will also need to have security and authentication functionality. These elements are essentially the same elements as those of a repository. Thus, a single device can be both a repository and a credit server, provided that it has the appropriate processing elements for carrying out the corresponding functions and protocols. Typically, however, a credit server would be a card-sized system in the possession of the owner of the credit. The credit server is coupled to a repository and would interact via financial transactions as described below. Interactions with a financial institution may occur via protocols established by the financial institutions themselves.

In the currently preferred embodiment credit servers associated with both the server and the repository report the financial transaction to the billing clearinghouse. For example, when a digital work is copied by one repository to another for a fee, credit servers coupled to each of the repositories will report the transaction to the billing clearinghouse. This is desirable in that it insures that a transaction will be accounted for in the event of some break in the communication between a credit server and the billing clearinghouse. However, some implementations may embody only a single credit server reporting the transaction to minimize transaction processing at the risk of losing some transactions.

USAGE RIGHTS LANGUAGE

The present invention uses statements in a high level "usage rights language" to define rights associated with digital works and their parts. Usage rights statements are interpreted by repositories and are used to determine what transactions can be successfully carried out for a digital work and also to determine parameters for those transactions. For example, sentences in the language determine whether a given digital work can be copied, when and how it can be used, and what fees (if any) are to be charged for that use. Once the usage rights statements are generated, they are encoded in a suitable form for accessing during the processing of transactions.

Defining usage rights in terms of a language in combination with the hierarchical representation of a digital work enables the support of a wide variety of distribution and fee schemes. An example is the ability to attach multiple versions of a right to a work. So a creator may attach a PRINT right to make 5 copies for $10.00 and a PRINT right to make unlimited copies for $100.00. A purchaser may then choose which option best fits his needs. Another example is that rights and fees are additive. So in the case of a composite work, the rights and fees of each of the components works is used in determining the rights and fees for the work as a whole.

The basic contents of a right are illustrated in Figure 14. Referring to Figure 14, a right 1450 has a transactional component 1451 and a specifications component 1452. A right 1450 has a label (e.g. COPY or PRINT) which indicates the use or distribution privileges that are embodied by the right. The transactional component 1451 corresponds to a particular way in which a digital work may be used or distributed. The transactional component 1451 is typically embodied in software instructions in a repository which implement the use or distribution privileges for the right. The specifications components 1452 are used to specify conditions which must be satisfied prior to the right being exercised or to designate various transaction related parameters. In the currently preferred embodiment, these specifications include copy count 1453, Fees and Incentives 1454, Time 1455, Access and Security 1456 and Control 1457. Each of these specifications will be described in greater detail below with respect to the language grammar elements.

The usage rights language is based on the grammar described below. A grammar is a convenient means for defining valid sequence of symbols for a language. In describing the grammar the notation "[a| b |c]" is used to indicate distinct choices among alternatives. In this example, a sentence can have either an "a", "b" or "c". It must include exactly one of them. The braces {} are used to indicate optional items. Note that brackets, bars and braces are used to describe the language of usage rights sentences but do not appear in actual sentences in the language.

In contrast, parentheses are part of the usage rights language. Parentheses are used to group items together in lists. The notation (x*) is used to indicate a variable length list, that is, a list containing one or more items of type x. The notation (x)* is used to indicate a variable number of lists containing x

Keywords in the grammar are words followed by colons. Keywords are a common and very special case in the language. They are often used to indicate a single value typically an identifier. In many cases the keyword and the parameter are entirely optional. When a keyword is given, it often takes a single identifier as its value. In some cases the keyword takes a list of identifiers.

In the usage rights language, time is specified in an hours:minutes:seconds (or hh:mm:ss) representation. Time zone indicators, e.g. PDT for Pacific Daylight Time, may also be specified. Dates are represented as year/month/day (or YYYY/MMM/DD). Note that these time and date representations may specify moments in time or units of time. Money units are specified in terms of dollars.

Finally, in the usage rights language, various "things" will need to interact with each other. For example, an instance of a usage right may specify a bank account, a digital ticket, etc.. Such things need to be identified and are specified herein using the suffix "-ID."

The Usage Rights Grammar is listed in its entirety in Figure 15 and is described below.

Grammar element 1501 **"Digital Work Rights: = (Rights*)"** define the digital work rights as a set of rights. The set of rights attached to a digital work define how that digital work may be transferred, used, performed or played. A set of rights will attach to the entire digital work and in the case of compound digital works, each of the components of the digital work. The usage rights of components of a digital may be different.

Grammar element 1502 **"Right : = (Right-Code {Copy-Count} {Control-Spec} {Time-Spec} {Access-Spec} {Fee-Spec})"** enumerates the content of a right. Each usage right must specify a right code. Each right may also optionally specify conditions which must be satisfied before the right can be exercised. These conditions are copy count, control, time, access and fee conditions. In the currently preferred embodiment, for the optional elements, the following defaults apply: copy count equals 1, no time limit on the use of the right, no access tests or a security level required to use the right and no fee is required. These conditions will each be described in greater detail below.

It is important to note that a digital work may have multiple versions of a right, each having the same right code. The multiple version would provide alternative conditions and fees for accessing the digital work.

Grammar element 1503 **"Right-Code : = Render-Code | Transport-Code | File-Management-Code| Derivative-Works- Code Configuration-Code"** distinguishes each of the specific rights into a particular right type (although each right is identified by distinct right codes). In this way, the grammar provides a catalog of possible rights that can be associated with parts of digital works. In the following, rights are divided into categories for convenience in describing them.

Grammar element 1504 **"Render-Code : = [Play : {Player: Player-ID}|Print: {Printer: Printer-ID}]"** lists a category of rights all involving the making of ephemeral, transitory, or non-digital copies of the digital work. After use the copies are erased.

- Play — A process of rendering or performing a digital work on some processor. This includes such things as playing digital movies, playing digital music, playing a video game, running a computer program, or displaying a document on a display.
- Print — To render the work in a medium that is not further protected by usage rights, such as printing on paper.

Grammar element 1505 **"Transport-Code : = [Copy | Transfer | Loan {Remaining-Rights: Next-Set-of-Rights}] {(Next-Copy-Rights: Next-Set of Rights)}"** lists a category of rights involving the making of persistent, usable copies of the digital work on other repositories. The optional Next-Copy-Rights determine the rights on the work after it is transported. If this is not specified, then the rights on the transported copy are the same as on the original. The optional Remaining-Rights specify the rights that remain with a digital work when it is loaned out. If this is not specified, then the default is that no rights can be exercised when it is loaned out.

- Copy — Make a new copy of a work
- Transfer — Moving a work from one repository to another.
- Loan — Temporarily loaning a copy to another repository for a specified period of time.

Grammar element 1506 **"File-Management-Code : = Backup {Back-Up-Copy-Rights: Next-Set -of Rights}| Restore | Delete | Folder | Directory {Name:Hide-Local | Hide-Remote}{Parts:Hide-Local|Hide-Remote}"** lists a category of rights involving operations for file management, such as the making of backup copies to protect the copy owner against catastrophic equipment failure.

Many software licenses and also copyright law give a copy owner the right to make backup copies to protect against catastrophic failure of equipment. However, the making of uncontrolled backup copies is inherently at odds with the ability to control usage, since an uncontrolled backup copy can be kept and then restored even after the authorized copy was sold.

The File management rights enable the making and restoring of backup copies in a way that respects usage rights,

honoring the requirements of both the copy owner and the rights grantor and revenue owner. Backup copies of work descriptions (including usage rights and fee data) can be sent under appropriate protocol and usage rights control to other document repositories of sufficiently high security. Further rights permit organization of digital works into folders which themselves are treated as digital works and whose contents may be "hidden" from a party seeking to determine the contents of a repository.

- Backup — To make a backup copy of a digital work as protection against media failure
- Restore — To restore a backup copy of a digital work.
- Delete — To delete or erase a copy of a digital work.
- Folder — To create and name folders, and to move files and folders between folders.
- Directory — To hide a folder or its contents.

Grammar element 1507 **"Derivative-Works-Code : [Extract|Embed|Edit{Process: Process-ID}] {Next-Copy-Rights : Next-Set-of Rights}"** lists a category of rights involving the use of a digital work to create new works.

- Extract — To remove a portion of a work, for the purposes of creating a new work.
- Embed — To include a work in an existing work.
- Edit — To alter a digital work by copying, selecting and modifying portions of an existing digital work.

Grammar element 1508 **"Configuration-Code : = Install | Uninstall"** lists a category of rights for installing and uninstalling software on a repository (typically a rendering repository.) This would typically occur for the installation of a new type of player within the rendering repository.

- Install: — To install new software on a repository.
- Uninstall: — To remove existing software from a repository.

Grammar element 1509 **"Next-Set-of-Rights : = {(Add: Set-Of-Rights)} {(Delete: Set-Of-Rights)} {(Replace: Set-Of-Rights)} {(Keep: Set-Of-Rights)}"** defines how rights are carried forward for a copy of a digital work. If the Next-Copy-Rights is not specified, the rights for the next copy are the same as those of the current copy. Otherwise, the set of rights for the next copy can be specified. Versions of rights after Add: are added to the current set of rights Rights after Delete: are deleted from the current set of rights. If only right codes are listed after Delete:, then all versions of rights with those codes are deleted. Versions of rights after Replace: subsume all versions of rights of the same type in the current set of rights.

If Remaining-Rights is not specified, then there are no rights for the original after all Loan copies are loaned out. If Remaining-Rights is specified, then the Keep: token can be used to simplify the expression of what rights to keep behind. A list of right codes following keep means that all of the versions of those listed rights are kept in the remaining copy. This specification can be overridden by subsequent Delete: or Replace: specifications.

Copy Count Specification

For various transactions, it may be desirable to provide some limit as to the number of "copies" of the work which may be exercised simultaneously for the right. For example, it may be desirable to limit the number of copies of a digital work that may be loaned out at a time or viewed at a time.

Grammar element 1510 **"Copy-Count : = (Copies: positive-integer | 0 | unlimited)"** provides a condition which defines the number of "copies" of a work subject to the right . A copy count can be 0, a fixed number, or unlimited. The copy-count is associated with each right, as opposed to there being just a single copy-count for the digital work. The Copy-Count for a right is decremented each time that a right is exercised. When the Copy-Count equals zero, the right can no longer be exercised. If the Copy-Count is not specified, the default is one.

Control Specification

Rights and fees depend in general on rights granted by the creator as well as further restrictions imposed by later distributors. Control specifications deal with interactions between the creators and their distributors governing the imposition of further restrictions and fees. For example, a distributor of a digital work may not want an end consumer of a digital work to add fees or otherwise profit by commercially exploiting the purchased digital work.

Grammar element 1511 **"Control-Spec : = (Control: {Restrictable | Unrestrictable} {Unchargeable | Chargeable})"** provides a condition to specify the effect of usage rights and fees of parents on the exercise of the right A digital work is restrictable if higher level d-blocks can impose further restrictions (time specifications and access spec-

ifications) on the right. It is unrestrictable if no further restrictions can be imposed. The default setting is restrictable. A right is unchargeable if no more fees can be imposed on the use of the right. It is chargeable if more fees can be imposed. The default is chargeable.

Time Specification

It is often desirable to assign a start date or specify some duration as to when a right may be exercised. Grammar element 1512 **"Time-Spec : = ({Fixed-Interval | Sliding-Interval | Meter-Time} Until: Expiration-Date)"** provides for specification of time conditions on the exercise of a right. Rights may be granted for a specified time. Different kinds of time specifications are appropriate for different kinds of rights. Some rights may be exercised during a fixed and predetermined duration. Some rights may be exercised for an interval that starts the first time that the right is invoked by some transaction. Some rights may be exercised or are charged according to some kind of metered time, which may be split into separate intervals. For example, a right to view a picture for an hour might be split into six ten minute viewings or four fifteen minute viewings or twenty three minute viewings.

The terms "time" and "date" are used synonymously to refer to a moment in time. There are several kinds of time specifications. Each specification represents some limitation on the times over which the usage right applies. The Expiration-Date specifies the moment at which the usage right ends. For example, if the Expiration-Date is "Jan 1, 1995," then the right ends at the first moment of 1995. If the Expiration-Date is specified as "forever", then the rights are interpreted as continuing without end. If only an expiration date is given, then the right can be exercised as often as desired until the expiration date.

Grammar element 1513 **"Fixed-Interval : = From: Start-Time"** is used to define a predetermined interval that runs from the start time to the expiration date.

Grammar element 1514 **"Sliding-Interval : = Interval: Use-Duration"** is used to define an indeterminate (or "open") start time. It sets limits on a continuous period of time over which the contents are accessible. The period starts on the first access and ends after the duration has passed or the expiration date is reached, whichever comes first. For example, if the right gives 10 hours of continuous access, the use-duration would begin when the first access was made and end 10 hours later.

Grammar element 1515 **"Meter-Time : = Time-Remaining: Remaining-Use"** is used to define a "meter time," that is, a measure of the time that the right is actually exercised. It differs from the Sliding-Interval specification in that the time that the digital work is in use need not be continuous. For example, if the rights guarantee three days of access, those days could be spread out over a month. With this specification, the rights can be exercised until the meter time is exhausted or the expiration date is reached, whichever comes first.

Remaining-Use: = Time-Unit
Start-Time: = Time-Unit
Use-Duration: = Time-Unit

All of the time specifications include time-unit specifications in their ultimate instantiation.

Security Class and Authorization Specification

The present invention provides for various security mechanisms to be introduced into a distribution or use scheme. Grammar element 1516 **"Access-Spec : = ({SC: Security-Class} {Authorization: Authorization-ID*} {Other-Authorization: Authorization-ID*} {Ticket: Ticket-ID})"** provides a means for restricting access and transmission. Access specifications can specify a required security class for a repository to exercise a right or a required authorization test that must be satisfied.

The keyword **"SC:"** is used to specify a minimum security level for the repositories involved in the access. If "SC: " is not specified, the lowest security level is acceptable.

The optional **"Authorization:"** keyword is used to specify required authorizations on the same repository as the work. The optional **"Other-Authorization:"** keyword is used to specify required authorizations on the other repository in the transaction.

The optional **"Ticket:"** keyword specifies the identity of a ticket required for the transaction. A transaction involving digital tickets must locate an appropriate digital ticket agent who can "punch" or otherwise validate the ticket before the transaction can proceed. Tickets are described in greater detail below.

In a transaction involving a repository and a document server, some usage rights may require that the repository have a particular authorization, that the server have some authorization, or that both repositories have (possibly different) authorizations. Authorizations themselves are digital works (hereinafter referred to as an authorization object) that can be moved between repositories in the same manner as other digital works. Their copying and transferring is

subject to the same rights and fees as other digital works. A repository is said to have an authorization if that authorization object is contained within the repository.

In some cases, an authorization may be required from a source other than the document server and repository. An authorization object referenced by an Authorization-ID can contain digital address information to be used to set up a communications link between a repository and the authorization source. These are analogous to phone numbers. For such access tests, the communication would need to be established and authorization obtained before the right could be exercised.

For one-time usage rights, a variant on this scheme is to have a digital ticket. A ticket is presented to a digital ticket agent, whose type is specified on the ticket. In the simplest case, a certified generic ticket agent, available on all repositories, is available to "punch" the ticket. In other cases, the ticket may contain addressing information for locating a "special" ticket agent. Once a ticket has been punched, it cannot be used again for the same kind of transaction (unless it is unpunched or refreshed in the manner described below.) Punching includes marking the ticket with a timestamp of the date and time it was used. Tickets are digital works and can be copied or transferred between repositories according to their usage rights.

In the currently preferred embodiment, a "punched" ticket becomes "unpunched" or "refreshed" when it is copied or extracted. The Copy and Extract operations save the date and time as a property of the digital ticket. When a ticket agent is given a ticket, it can simply check whether the digital copy was made after the last time that it was punched. Of course, the digital ticket must have the copy or extract usage rights attached thereto.

The capability to unpunch a ticket is inportant in the following cases:

- A digital work is circulated at low cost with a limitation that it can be used only once.
- A digital work is circulated with a ticket that can be used once to give discounts on purchases of other works.
- A digital work is circulated with a ticket (included in the purchase price and possibly embedded in the work) that can be used for a future upgrade.

In each of these cases, if a paid copy is made of the digital work (including the ticket) the new owner would expect to get a fresh (unpunched) ticket, whether the copy seller has used the work or not. In contrast, loaning a work or simply transferring it to another repository should not revitalize the ticket.

Usage Fees and Incentives Specification

The billing for use of a digital work is fundamental to a commercial distribution system. Grammar Element 1517 **"Fee-Spec: = {Scheduled-Discount} Regular-Fee-Spec | Scheduled-Fee-Spec | Markup-Spec"** provides a range of options for billing for the use of digital works.

A key feature of this approach is the development of low-overhead billing for transactions in potentially small amounts. Thus, it becomes feasible to collect fees of only a few cents each for thousands of transactions.

The grammar differentiates between uses where the charge is per use from those where it is metered by the time unit. Transactions can support fees that the user pays for using a digital work as well as incentives paid by the right grantor to users to induce them to use or distribute the digital work.

The optional scheduled discount refers to the rest of the fee specification--discounting it by a percentage over time. If it is not specified, then there is no scheduled discount. Regular fee specifications are constant over time. Scheduled fee specifications give a schedule of dates over which the fee specifications change. Markup specifications are used in d-blocks for adding a percentage to the fees already being charged.

Grammar Element 1518 **"Scheduled-Discount:= (Scheduled-Discount: (Time-Spec Percentage)*)"** A Scheduled-Discount is a essentially a scheduled modifier of any other fee specification for this version of the right of the digital work. (It does not refer to children or parent digital works or to other versions of rights.). It is a list of pairs of times and percentages. The most recent time in the list that has not yet passed at the time of the transaction is the one in effect. The percentage gives the discount percentage. For example, the number 10 refers to a 10% discount.

Grammar Element 1519 **"Regular-Fee-Spec : = ({Fee: | Incentive: } [Per-Use-Spec | Metered-Rate-Spec | Best-Price-Spec | Call-For-Price-Spec] {Min: Money-Unit Per: Time-Spec}{Max: Money-Unit Per: Time-Spec} To: Account-ID)"** provides for several kinds of fee specifications.

Fees are paid by the copy-owner/user to the revenue-owner if **Fee:** is specified. Incentives are paid by the revenue-owner to the user if **Incentive:** is specified. If the **Min:** specification is given, then there is a minimum fee to be charged per time-spec unit for its use. If the **Max:** specification is given, then there is a maximum fee to be charged per time-spec for its use. When **Fee:** is specified, **Account-ID** identifies the account to which the fee is to be paid. When **Incentive:** is specified, **Account-ID** identifies the account from which the fee is to be paid.

Grammar element 1520 **"Per-Use-Spec:= Per-Use: Money-unit"** defines a simple fee to be paid every time the right is exercised, regardless of how much time the transaction takes.

Grammar element 1521 **"Metered-Rate-Spec : = Metered: Money-Unit Per: Time**-Spec" defines a metered-rate fee paid according to how long the right is exercised. Thus, the time it takes to complete the transaction determines the fee.

Grammar element 1522 **"Best-Price-Spec : = Best-Price: Money-unit Max: Money**-unit" is used to specify a best-price that is determined when the account is settled. This specification is to accommodate special deals, rebates, and pricing that depends on information that is not available to the repository. All fee specifications can be combined with tickets or authorizations that could indicate that the consumer is a wholesaler or that he is a preferred customer or that the seller be authorized in some way. The amount of money in the **Max:** field is the maximum amount that the use will cost. This is the amount that is tentatively debited from the credit server. However, when the transaction is ultimately reconciled, any excess amount will be returned to the consumer in a separate transaction.

Grammar element 1523 **"Call-For-Price-Spec : = Call-For-Price "** is similar to a **"Best-Price-Spec"** in that it is intended to accommodate cases where prices are dynamic. A **Call-For-Price Spec** requires a communication with a dealer to determine the price. This option cannot be exercised if the repository cannot communicate with a dealer at the time that the right is exercised. It is based on a secure transaction whereby the dealer names a price to exercise the right and passes along a deal certificate which is referenced or included in the billing process.

Grammar element 1524 **"Scheduled-Fee-Spec: = (Schedule: (Time-Spec Regular-Fee-Spec)*)"** is used to provide a schedule of dates over which the fee specifications change. The fee specification with the most recent date not in the future is the one that is in effect. This is similar to but more general than the scheduled discount. It is more general, because it provides a means to vary the fee agreement for each time period.

Grammar element 1525 **"Markup-Spec: = Markup: percentage To: Account-ID"** is provided for adding a percentage to the fees already being charged. For example, a 5% markup means that a fee of 5% of cumulative fee so far will be allocated to the distributor. A markup specification can be applied to all of the other kinds of fee specifications. It is typically used in a shell provided by a distributor. It refers to fees associated with d-blocks that are parts of the current d-block. This might be a convenient specification for use in taxes, or in distributor overhead.

REPOSITORY TRANSACTIONS

When a user requests access to a digital work, the repository will initiate various transactions. The combination of transactions invoked will depend on the specifications assigned for a usage right. There are three basic types of transactions, Session Initiation Transactions, Financial Transactions and Usage Transactions. Generally, session initiation transactions are initiated first to establish a valid session. When a valid session is established, transactions corresponding to the various usage rights are invoked. Finally, request specific transactions are performed.

Transactions occur between two repositories (one acting as a server), between a repository and a document playback platform (e.g. for executing or viewing), between a repository and a credit server or between a repository and an authorization server. When transactions occur between more than one repository, it is assumed that there is a reliable communication channel between the repositories. For example, this could be a TCP/IP channel or any other commercially available channel that has built-in capabilities for detecting and correcting transmission errors. However, it is not assumed that the communication channel is secure. Provisions for security and privacy are part of the requirements for specifying and implementing repositories and thus form the need for various transactions.

Message Transmission

Transactions require that there be some communication between repositories. Communication between repositories occurs in units termed as messages. Because the communication line is assumed to be unsecure, all communications with repositories that are above the lowest security class are encrypted utilizing a public key encryption technique. Public key encryption is a well known technique in the encryption arts. The term key refers to a numeric code that is used with encryption and decryption algorithms. Keys come in pairs, where "writing keys" are used to encrypt data and "checking keys" are used to decrypt data. Both writing and checking keys may be public or private. Public keys are those that are distributed to others. Private keys are maintained in confidence.

Key management and security is instrumental in the success of a public key encryption system. In the currently preferred embodiment, one or more master repositories maintain the keys and create the identification certificates used by the repositories.

When a sending repository transmits a message to a receiving repository, the sending repository encrypts all of its data using the public writing key of the receiving repository. The sending repository includes its name, the name of the receiving repository, a session identifier such as a nonce (described below), and a message counter in each message.

In this way, the communication can only be read (to a high probability) by the receiving repository, which holds the private checking key for decryption. The auxiliary data is used to guard against various replay attacks to security. If

messages ever arrive with the wrong counter or an old nonce the repositories can assume that someone is interfering with communication and the transaction terminated

The respective public keys for the repositories to be used for encryption are obtained in the registration transaction described below.

Session Initiation Transactions

A usage transaction is carried out in a session between repositories. For usage transactions involving more than one repository, or for financial transactions between a repository and a credit server, a registration transaction is performed. A second transaction termed a login transaction, may also be needed to initiate the session. The goal of the registration transaction is to establish a secure channel between two repositories who know each others identities. As it is assumed that the communication channel between the repositories is reliable but not secure, there is a risk that a non-repository may mimic the protocol in order to gain illegitimate access to a repository.

The registration transaction between two repositories is described with respect to Figures 16 and 17. The steps described are from the perspective of a "repository-1" registering its identity with a "repository-2". The registration must be symmetrical so the same set of steps will be repeated for repository-2 registering its identity with repository-1. Referring to Figure 16, repository-1 first generates an encrypted registration identifier, step 1601 and then generates a registration message, step 1602. A registration message is comprised of an identifier of a master repository, the identification certificate for the repository-1 and an encrypted random registration identifier. The identification certificate is encrypted by the master repository in its private key and attests to the fact that the repository (here repository-1) is a bona fide repository. The identification certificate also contains a public key for the repository, the repository security level and a timestamp (indicating a time after which the certificate is no longer valid.) The registration identifier is a number generated by the repository for this registration. The registration identifier is unique to the session and is encrypted in repository-1's private key. The registration identifier is used to improve security of authentication by detecting certain kinds of communications based attacks. Repository-1 then transmits the registration message to repository-2, step 1603.

Upon receiving the registration message, repository-2 determines if it has the needed public key for the master repository, step 1604. If repository-2 does not have the needed public key to decrypt the identification certificate, the registration transaction terminates in an error, step 1618.

Assuming that repository-2 has the proper public key the identification certificate is decrypted, step 1605. Repository-2 saves the encrypted registration identifier, step 1606, and extracts the repository identifier, step 1607. The extracted repository identifier is checked against a "hotlist" of compromised document repositories, step 1608. In the currently preferred embodiment, each repository will contain "hotlists" of compromised repositories. If the repository is on the "hotlist", the registration transaction terminates in an error per step 1618. Repositories can be removed from the hotlist when their certificates expire, so that the list does not need to grow without bound. Also, by keeping a short list of hotlist certificates that it has previously received, a repository can avoid the work of actually going through the list. These lists would be encrypted by a master repository. A minor variation on the approach to improve efficiency would have the repositories first exchange lists of names of hotlist certificates, ultimately exchanging only those lists that they had not previously received. The "hotlists" are maintained and distributed by Master repositories.

Note that rather than terminating in error, the transaction could request that another registration message be sent based on an identification certificate created by another master repository. This may be repeated until a satisfactory identification certificate is found, or it is determined that trust cannot be established.

Assuming that the repository is not on the hotlist, the repository identification needs to be verified. In other words, repository-2 needs to validate that the repository on the other end is really repository-1. This is termed performance testing and is performed in order to avoid invalid access to the repository via a counterfeit repository replaying a recording of a prior session initiation between repository-1 and repository-2. Performance testing is initiated by repository-2 generating a performance message, step 1609. The performance message consists of a nonce, the names of the respective repositories, the time and the registration identifier received from repository-1. A nonce is a generated message based on some random and variable information (e.g. the time or the temperature.) The nonce is used to check whether repository-1 can actually exhibit correct encrypting of a message using the private keys it claims to have, on a message that it has never seen before. The performance message is encrypted using the public key specified in the registration message of repository-1. The performance message is transmitted to repository-1, step 1610, where it is decrypted by repository-1 using its private key, step 1611. Repository-1 then checks to make sure that the names of the two repositories are correct, step 1612, that the time is accurate, step 1613 and that the registration identifier corresponds to the one it sent, step 1614. If any of these tests fails, the transaction is terminated per step 1616. Assuming that the tests are passed, repository-1 transmits the nonce to repository-2 in the clear, step 1615. Repository-2 then compares the received nonce to the original nonce, step 1617. If they are not identical, the registration transaction terminates in an error per step 1618. If they are the same, the registration transaction has successfully completed.

At this point, assuming that the transaction has not terminated the repositories exchange messages containing session keys to be used in all communications during the session and synchronize their clocks. Figure 17 illustrates the session information exchange and clock synchronization steps (again from the perspective of repository-1). Referring to Figure 17, repository-1 creates a session key pair, step 1701. A first key is kept private and is used by repository-1 to encrypt messages. The second key is a public key used by repository-2 to decrypt messages. The second key is encrypted using the public key of repository-2, step 1702 and is sent to repository-2, step 1703. Upon receipt, repository-2 decrypts the second key, step 1704. The second key is used to decrypt messages in subsequent communications. When each repository has completed this step, they are both convinced that the other repository is bona fide and that they are communicating with the original. Each repository has given the other a key to be used in decrypting further communications during the session. Since that key is itself transmitted in the public key of the receiving repository only it will be able to decrypt the key which is used to decrypt subsequent messages.

After the session information is exchanged, the repositories must synchronize their clocks. Clock synchronization is used by the repositories to establish an agreed upon time base for the financial records of their mutual transactions. Referring back to Figure 17, repository-2 initiates clock synchronization by generating a time stamp exchange message, step 1705, and transmits it to repository-1, step 1706. Upon receipt, repository-1 generates its own time stamp message, step 1707 and transmits it back to repository-2, step 1708. Repository-2 notes the current time, step 1709 and stores the time received from repository-1, step 1710. The current time is compared to the time received from repository-1, step 1711. The difference is then checked to see if it exceeds a predetermined tolerance (e.g. one minute), step 1712. If it does, repository-2 terminates the transaction as this may indicate tampering with the repository, step 1713. If not repository-2 computes an adjusted time delta, step 1714. The adjusted time delta is the difference between the clock time of repository-2 and the average of the times from repository-1 and repository-2.

To achieve greater accuracy, repository-2 can request the time again up to a fixed number of times (e.g. five times), repeat the clock synchronization steps, and average the results.

A second session initiation transaction is a Login transaction. The Login transaction is used to check the authenticity of a user requesting a transaction. A Login transaction is particularly prudent for the authorization of financial transactions that will be charged to a credit server. The Login transaction involves an interaction between the user at a user interface and the credit server associated with a repository. The information exchanged here is a login string supplied by the repository/credit server to identify itself to the user, and a Personal Identification Number (PIN) provided by the user to identify himself to the credit server. In the event that the user is accessing a credit server on a repository different from the one on which the user interface resides, exchange of the information would be encrypted using the public and private keys of the respective repositories.

Billing Transactions

Billing Transactions are concerned with monetary transactions with a credit server. Billing Transactions are carried out when all other conditions are satisfied and a usage fee is required for granting the request. For the most part, billing transactions are well understood in the state of the art. These transactions are between a repository and a credit server, or between a credit server and a billing clearinghouse. Briefly, the required transactions include the following:

- Registration and LOGIN transactions by which the repository and user establish their bona fides to a credit server. These transactions would be entirely internal in cases where the repository and credit server are implemented as a single system.
- Registration and LOG IN transactions, by which a credit server establishes its bona fides to a billing clearinghouse.
- An Assign-fee transaction to assign a charge. The information in this transaction would include a transaction identifier, the identities of the repositories in the transaction, and a list of charges from the parts of the digital work. If there has been any unusual event in the transaction such as an interruption of communications, that information is included as well.
- A Begin-charges transaction to assign a charge. This transaction is much the same as an assign-fee transaction except that it is used for metered use. It includes the same information as the assign-fee transaction as well as the usage fee information. The credit-server is then responsible for running a clock.
- An End-charges transaction to end a charge for metered use. (In a variation on this approach, the repositories would exchange periodic charge information for each block of time.)
- A report-charges transaction between a personal credit server and a billing clearinghouse. This transaction is invoked at least once per billing period. It is used to pass along information about charges. On debit and credit cards, this transaction would also be used to update balance information and credit limits as needed.

All billing transactions are given a transaction ID and are reported to the credit severs by both the server and the client. This reduces possible loss of billing information if one of the parties to a transaction loses a banking card and

provides a check against tampering with the system.

Usage Transactions

After the session initiation transactions have been completed, the usage request may then be processed. To simplify the description of the steps carried out in processing a usage request, the term requester is used to refer to a repository in the requester mode which is initiating a request, and the term server is used to refer to a repository in the server mode and which contains the desired digital work. In many cases such as requests to print or view a work, the requester and server may be the same device and the transactions described in the following would be entirely internal. In such instances, certain transaction steps, such as the registration transaction, need not be performed.

There are some common steps that are part of the semantics of all of the usage rights transactions. These steps are referred to as the common transaction steps. There are two sets -the "opening" steps and the "closing" steps. For simplicity, these are listed here rather than repeating them in the descriptions of all of the usage rights transactions.

Transactions can refer to a part of a digital work, a complete digital work, or a Digital work containing other digital works. Although not described in detail herein, a transaction may even refer to a folder comprised of a plurality of digital works. The term "work" is used to refer to what ever portion or set of digital works is being accessed.

Many of the steps here involve determining if certain conditions are satisfied. Recall that each usage right may have one or more conditions which must be satisfied before the right can be exercised. Digital works have parts and parts have parts. Different parts can have different rights and fees. Thus, it is necessary to verify that the requirements are met for ALL of the parts that are involved in a transaction. For brevity, when reference is made to checking whether the rights exist and conditions for exercising are satisfied, it is meant that all such checking takes place for each of the relevant parts of the work.

Figure 18 illustrates the initial common opening and closing steps for a transaction. At this point it is assumed that registration has occurred and that a "trusted" session is in place. General tests are tests on usage rights associated with the folder containing the work or some containing folder higher in the file system hierarchy. These tests correspond to requirements imposed on the work as a consequence of its being on the particular repository, as opposed to being attached to the work itself. Referring to Figure 18, prior to initiating a usage transaction, the requester performs any general tests that are required before the right associated with the transaction can be exercised, step, 1801. For example, install, uninstall and delete rights may be implemented to require that a requester have an authorization certificate before the right can be exercised. Another example is the requirement that a digital ticket be present and punched before a digital work may be copied to a requester. If any of the general tests fail, the transaction is not initiated, step, 1802. Assuming that such required tests are passed, upon receiving the usage request, the server generates a transaction identifier that is used in records or reports of the transaction, step 1803. The server then checks whether the digital work has been granted the right corresponding to the requested transaction, step 1804. If the digital work has not been granted the right corresponding to the request, the transaction terminates, step 1805. If the digital work has been granted the requested right, the server then determines if the various conditions for exercising the right are satisfied. Time based conditions are examined, step 1806. These conditions are checked by examining the time specification for the the version of the right. If any of the conditions are not satisfied, the transaction terminates per step 1805.

Assuming that the time based conditions are satisfied, the server checks security and access conditions, step 1807. Such security and access conditions are satisfied if: 1) the requester is at the specified security class, or a higher security class, 2) the server satisfies any specified authorization test and 3) the requester satisfies any specified authorization tests and has any required digital tickets. If any of the conditions are not satisfied, the transaction terminates per step 1805.

Assuming that the security and access conditions are all satisfied, the server checks the copy count condition, step 1808. If the copy count equals zero, then the transaction cannot be completed and the transaction terminates per step 1805.

Assuming that the copy count does not equal zero, the server checks if the copies in use for the requested right is greater than or equal to any copy count for the requested right (or relevant parts), step 1809. If the copies in use is greater than or equal to the copy count, this indicates that usage rights for the version of the transaction have been exhausted. Accordingly, the server terminates the transaction, step 1805. If the copy count is less than the copies in use for the transaction the transaction can continue, and the copies in use would be incremented by the number of digital works requested in the transaction, step 1810.

The server then checks if the digital work has a "Loan" access right, step 1811. The "Loan" access right is a special case since remaining rights may be present even though all copies are loaned out. If the digital work has the "Loan" access right, a check is made to see if all copies have been loaned out, step 1812. The number of copies that could be loaned is the sum of the Copy-Counts for all of the versions of the loan right of the digital work. For a composite work the relevant figure is the minimal such sum of each of the components of the composite work. If all copies have been loaned out, the remaining rights are determined, step 1813. The remaining-rights is determined from the remaining

rights specifications from the versions of the Loan right. If there is only one version of the Loan right, then the determination is simple. The remaining rights are the ones specified in that version of the Loan right, or none if Remaining-Rights: is not specified. If there are multiple versions of the Loan right and all copies of all of the versions are loaned out, then the remaining rights is taken as the minimum set (intersection) of remaining rights across all of the versions of the loan right. The server then determines if the requested right is in the set of remaining rights, step 1814. If the requested right is not in the set of remaining rights, the server terminates the transaction, step 1805.

If Loan is not a usage right for the digital work or if all copies have not been loaned out or the requested right is in the set of remaining rights, fee conditions for the right are then checked, step 1815. This will initiate various financial transactions between the repository and associated credit server. Further, any metering of usage of a digital work will commence. If any financial transaction fails, the transaction terminates per step 1805.

It should be noted that the order in which the conditions are checked need not follow the order of steps 1806-1815.

At this point, right specific steps are now performed and are represented here as step 1816. The right specific steps are described in greater detail below.

The common closing transaction steps are now performed. Each of the closing transaction steps are performed by the server after a successful completion of a transaction. Referring back to Figure 18, the copies in use value for the requested right is decremented by the number of copies involved in the transaction, step 1817. Next, if the right had a metered usage fee specification, the server subtracts the elapsed time from the Remaining-Use-Time associated with the right for every part involved in the transaction, step 1818. Finally, if there are fee specifications associated with the right, the server initiates End-Charge financial transaction to confirm billing, step 1819.

Transmission Protocol

An important area to consider is the transmission of the digital work from the server to the requester. The transmission protocol described herein refers to events occurring after a valid session has been created. The transmission protocol must handle the case of disruption in the communications between the repositories. It is assumed that interference such as injecting noise on the communication channel can be detected by the integrity checks (e.g., parity, checksum, etc.) that are built into the transport protocol and are not discussed in detail herein.

The underlying goal in the transmission protocol is to preclude certain failure modes, such as malicious or accidental interference on the communications channel. Suppose, for example, that a user pulls a card with the credit server at a specific time near the end of a transaction. There should not be a vulnerable time at which "pulling the card" causes the repositories to fail to correctly account for the number of copies of the work that have been created. Restated, there should be no time at which a party can break a connection as a means to avoid payment after using a digital work.

If a transaction is interrupted (and fails), both repositories restore the digital works and accounts to their state prior to the failure, modulo records of the failure itself.

Figure 19 is a state diagram showing steps in the process of transmitting information during a transaction. Each box represents a state of a repository in either the server mode (above the central dotted line 1901) or in the requester mode (below the dotted line 1901). Solid arrows stand for transitions between states. Dashed arrows stand for message communications between the repositories. A dashed message arrow pointing to a solid transition arrow is interpreted as meaning that the transition takes place when the message is received. Unlabeled transition arrows take place unconditionally. Other labels on state transition arrows describe conditions that trigger the transition.

Referring now to Figure 19, the server is initially in a state 1902 where a new transaction is initiated via start message 1903. This message includes transaction information including a transaction identifier and a count of the blocks of data to be transferred. The requester, initially in a wait state 1904 then enters a data wait state 1905.

The server enters a data transmit state 1906 and transmits a block of data 1907 and then enters a wait for acknowledgement state 1908. As the data is received, the requester enters a data receive state 1909 and when the data blocks are completely received it enters an acknowledgement state 1910 and transmits an Acknowledgement message 1911 to the server.

If there are more blocks to send, the server waits until receiving an Acknowledgement message from the requester. When an Acknowledgement message is received it sends the next block to the requester and again waits for acknowledgement. The requester also repeats the same cycle of states.

If the server detects a communications failure before sending the last block, it enters a cancellation state 1912 wherein the transaction is cancelled. Similarly, if the requester detects a communications failure before receiving the last block it enters a cancellation state 1913.

If there are no more blocks to send, the server commits to the transaction and waits for the final Acknowledgement in state 1914. If there is a communications failure before the server receives the final Acknowledgement message, it still commits to the transaction but includes a report about the event to its credit server in state 1915. This report serves two purposes. It will help legitimize any claims by a user of having been billed for receiving digital works that were not completely received. Also it helps to identify repositories and communications lines that have suspicious patterns of

use and interruption The server then enters its completion state 1916.

On the requester side. when there are no more blocks to receive the requester commits to the transaction in state 1917. If the requester detects a communications failure at this state. it reports the failure to its credit server in state 1918. but still commits to the transaction. When it has committed. it sends an acknowledgement message to the server The server then enters its completion state 1919.

The key property is that both the server and the requester cancel a transaction if it is interrupted before all of the data blocks are delivered. and commits to it if all of the data blocks have been delivered

There is a possibility that the server will have sent all of the data blocks (and committed) but the requester will not have received all of them and will cancel the transaction. In this case. both repositories will presumably detect a communications failure and report it to their credit server. This case will probably be rare since it depends on very precise timing of the communications failure. The only consequence will be that the user at the requester repository may want to request a refund from the credit services -- and the case for that refund will be documented by reports by both repositories.

To prevent loss of data, the server should not delete any transferred digital work until receiving the final acknowledgement from the requester. But it also should not use the file. A well known way to deal with this situation is called "two-phase commit" or 2PC.

Two-phase commit works as follows. The first phase works the same as the method described above. The server sends all of the data to the requester. Both repositories mark the transaction (and appropriate files) as uncommitted. The server sends a ready-to-commit message to the requester. The requester sends back an acknowledgement. The server then commits and sends the requester a commit message. When the requester receives the commit message. it commits the file.

If there is a communication failure or other crash. the requester must check back with the server to determine the status of the transaction. The server has the last word on this. The requester may have received all of the data. but if it did not get the final message. it has not committed. The server can go ahead and delete files (except for transaction records) once it commits, since the files are known to have been fully transmitted before starting the 2PC cycle.

There are variations known in the art which can be used to achieve the same effect. For example. the server could use an additional level of encryption when transmitting a work to a client. Only after the client sends a message acknowledging receipt does it send the key. The client then agrees to pay for the digital work. The point of this variation is that it provides a clear audit trail that the client received the work. For trusted systems. however. this variation adds a level of encryption for no real gain in accountability

The transaction for specific usage rights are now discussed.

The Copy Transaction

A Copy transaction is a request to make one or more independent copies of the work with the same or lesser usage rights. Copy differs from the extraction right discussed later in that it refers to entire digital works or entire folders containing digital works. A copy operation cannot be used to remove a portion of a digital work.

- The requester sends the server a message to initiate the Copy Transaction. This message indicates the work to be copied. the version of the copy right to be used for the transaction. the destination address information (location in a folder) for placing the work, the file data for the work (including its size), and the number of copies requested.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the client according to the transmission protocol. If a Next-Set-Of-Rights has been provided in the version of the right. those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted. In any event. the Copy-Count field for the copy of the digital work being sent right is set to the number-of-copies requested.
- The requester records the work contents. data. and usage rights and stores the work. It records the date and time that the copy was made in the properties of the digital work.
- The repositories perform the common closing transaction steps.

The Transfer Transaction

A Transfer transaction is a request to move copies of the work with the same or lesser usage rights to another repository. In contrast with a copy transaction, this results in removing the work copies from the server.

- The requester sends the server a message to initiate the Transfer Transaction. This message indicates the work to be transferred, the version of the transfer right to be used in the transaction, the destination address information for placing the work, the file data for the work, and the number of copies involved

- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted.

In either case, the Copy-Count field for the transmitted rights are set to the number-of-copies requested.

- The requester records the work contents, data, and usage rights and stores the work.
- The server decrements its copy count by the number of copies involved in the transaction.
- The repositories perform the common closing transaction steps.
- If the number of copies remaining in the server is now zero, it erases the digital work from its memory.

The Loan Transaction

A loan transaction is a mechanism for loaning copies of a digital work. The maximum duration of the loan is determined by an internal parameter of the digital work. Works are automatically returned after a predetermined time period.

- The requester sends the server a message to initiate the Transfer Transaction. This message indicates the work to be loaned, the version of the loan right to be used in the transaction, the destination address information for placing the work, the number of copies involved, the file data for the work, and the period of the loan.
- The server checks the validity of the requested loan period, and ends with an error if the period is not valid. Loans for a loaned copy cannot extend beyond the period of the original loan to the server.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted, as modified to reflect the loan period.
- The requester records the digital work contents, data, usage rights, and loan period and stores the work.
- The server updates the usage rights information in the digital work to reflect the number of copies loaned out.
- The repositories perform the common closing transaction steps.
- The server updates the usage rights data for the digital work. This may preclude use of the work until it is returned from the loan. The user on the requester platform can now use the transferred copies of the digital work. A user accessing the original repository cannot use the digital work, unless there are copies remaining. What happens next depends on the order of events in time.

Case 1. If the time of the loan period is not yet exhausted and the requester sends the repository a Return message.

- The return message includes the requester identification, and the transaction ID.
- The server decrements the copies-in-use field by the number of copies that were returned. (If the number of digital works returned is greater than the number actually borrowed, this is treated as an error.) This step may now make the work available at the server for other users.
- The requester deactivates its copies and removes the contents from its memory.

Case 2. If the time of the loan period is exhausted and the requester has not yet sent a Return message.

- The server decrements the copies-in-use field by the number digital works that were borrowed.
- The requester automatically deactivates its copies of the digital work. It terminates all current uses and erases the digital work copies from memory. One question is why a requester would ever return a work earlier than the period of the loan, since it would be returned automatically anyway. One reason for early return is that there may be a metered fee which determines the cost of the loan. Returning early may reduce that fee.

The Play Transaction

A play transaction is a request to use the contents of a work. Typically, to "play" a work is to send the digital work through some kind of transducer, such as a speaker or a display device. The request implies the intention that the contents will not be communicated digitally to any other system. For example, they will not be sent to a printer, recorded on any digital medium, retained after the transaction or sent to another repository.

This term "play" is natural for examples like playing music, playing a movie, or playing a video game. The general form of play means that a "player" is used to use the digital work. However, the term play covers all media and kinds of recordings. Thus one would "play" a digital work, meaning, to render it for reading, or play a computer program,

meaning to execute it. For a digital ticket the player would be a digital ticket agent.

- The requester sends the server a message to initiate the play transaction. This message indicates the work to be played, the version of the play right to be used in the transaction, the identity of the player being used, and the file data for the work.
- The server checks the validity of the player identification and the compatibility of the player identification with the player specification in the right. It ends with an error if these are not satisfactory.
- The repositories perform the common opening transaction steps.
- The server and requester read and write the blocks of data as requested by the player according to the transmission protocol. The requester plays the work contents, using the player
- When the player is finished, the player and the requester remove the contents from their memory.
- The repositories perform the common closing transaction steps.

The Print Transaction

A Print transaction is a request to obtain the contents of a work for the purpose of rendering them on a "printer." We use the term "printer" to include the common case of writing with ink on paper. However, the key aspect of "printing" in our use of the term is that it makes a copy of the digital work in a place outside of the protection of usage rights. As with all rights, this may require particular authorization certificates.

Once a digital work is printed, the publisher and user are bound by whatever copyright laws are in effect. However, printing moves the contents outside the control of repositories. For example, absent any other enforcement mechanisms, once a digital work is printed on paper, it can be copied on ordinary photocopying machines without intervention by a repository to collect usage fees. If the printer to a digital disk is permitted, then that digital copy is outside of the control of usage rights. Both the creator and the user know this, although the creator does not necessarily give tacit consent to such copying, which may violate copyright laws.

- The requester sends the server a message to initiate a Print transaction. This message indicates the work to be played, the identity of the printer being used, the file data for the work, and the number of copies in the request.
- The server checks the validity of the printer identification and the compatibility of the printer identification with the printer specification in the right. It ends with an error if these are not satisfactory
- The repositories perform the common opening transaction steps.
- The server transmits blocks of data according to the transmission protocol.
- The requester prints the work contents, using the printer.
- When the printer is finished, the printer and the requester remove the contents from their memory.
- The repositories perform the common closing transaction steps.

The Backup Transaction

A Backup transaction is a request to make a backup copy of a digital work, as a protection against media failure. In the context of repositories, secure backup copies differ from other copies in three ways: (1) they are made under the control of a Backup transaction rather than a Copy transaction, (2) they do not count as regular copies, and (3) they are not usable as regular copies. Generally, backup copies are encrypted.

Although backup copies may be transferred or copied, depending on their assigned rights, the only way to make them useful for playing, printing or embedding is to restore them.

The output of a Backup operation is both an encrypted data file that contains the contents and description of a work, and a restoration file with an encryption key for restoring the encrypted contents. In many cases, the encrypted data file would have rights for "printing" it to a disk outside of the protection system, relying just on its encryption for security. Such files could be stored anywhere that was physically safe and convenient. The restoration file would be held in the repository. This file is necessary for the restoration of a backup copy. It may have rights for transfer between repositories.

- The requester sends the server a message to initiate a backup transaction. This message indicates the work to be backed up, the version of the backup right to be used in the transaction, the destination address information for placing the backup copy, the file data for the work.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, a set of default rights for backup files of the original are transmitted by the server.

- The requester records the work contents, data, and usage rights. It then creates a one-time key and encrypts the contents file. It saves the key information in a restoration file
- The repositories perform the common closing transaction steps.

In some cases, it is convenient to be able to archive the large, encrypted contents file to secure offline storage such as a magneto-optical storage system or magnetic tape. This creation of a non-repository archive file is as secure as the encryption process. Such non-repository archive storage is considered a form of "printing" and is controlled by a print right with a specified "archive-printer." An archive-printer device is programmed to save the encrypted contents file (but not the description file) offline in such a way that it can be retrieved

The Restore Transaction

A Restore transaction is a request to convert an encrypted backup copy of a digital work into a usable copy. A restore operation is intended to be used to compensate for catastrophic media failure. Like all usage rights, restoration rights can include fees and access tests including authorization checks.

- The requester sends the server a message to initiate a Restore transaction. This message indicates the work to be restored, the version of the restore right for the transaction, the destination address information for placing the work, and the file data for the work.
- The server verifies that the contents file is available (i.e. a digital work corresponding to the request has been backed-up.) If it is not, it ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The server retrieves the key from the restoration file. It decrypts the work contents, data, and usage rights.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, a set of default rights for backup files of the original are transmitted by the server.
- The requester stores the digital work.
- The repositories perform the common closing transaction steps.

The Delete Transaction

A Delete transaction deletes a digital work or a number of copies of a digital work from a repository. Practically all digital works would have delete rights.

- The requester sends the server a message to initiate a delete transaction. This message indicates the work to be deleted, the version of the delete right for the transaction.
- The repositories perform the common opening transaction steps.
- The server deletes the file, erasing it from the file system.
- The repositories perform the common closing transaction steps.

The Directory Transaction

A Directory transaction is a request for information about folders, digital works, and their parts. This amounts to roughly the same idea as protection codes in a conventional file system like TENEX, except that it is generalized to the full power of the access specifications of the usage rights language.

The Directory transaction has the important role of passing along descriptions of the rights and fees associated with a digital work. When a user wants to exercise a right, the user interface of his repository implicitly makes a directory request to determine the versions of the right that are available. Typically these are presented to the user -- such as with different choices of billing for exercising a right. Thus, many directory transactions are invisible to the user and are exercised as part of the normal process of exercising all rights.

- The requester sends the server a message to initiate a Directory transaction. This message indicates the file or folder that is the root of the directory request and the version of the directory right used for the transaction.
- The server verifies that the information is accessible to the requester. In particular, it does not return the names of any files that have a HIDE-NAME status in their directory specifications, and it does not return the parts of any folders or files that have HIDE-PARTS in their specification. If the information is not accessible, the server ends the transaction with an error
- The repositories perform the common opening transaction steps.

- The server sends the requested data to the requester according to the transmission protocol
- The requester records the data
- The repositories perform the common closing transaction steps.

The Folder Transaction

A Folder transaction is a request to create or rename a folder, or to move a work between folders. Together with Directory rights, Folder rights control the degree to which organization of a repository can be accessed or modified from another repository.

- The requester sends the server a message to initiate a Folder transaction. This message indicates the folder that is the root of the folder request, the version of the folder right for the transaction, an operation, and data. The operation can be one of create, rename, and move file. The data are the specifications required for the operation, such as a specification of a folder or digital work and a name.
- The repositories perform the common opening transaction steps.
- The server performs the requested operation -- creating a folder, renaming a folder, or moving a work between folders.
- The repositories perform the common closing transaction steps.

The Extract Transaction

A extract transaction is a request to copy a part of a digital work and to create a new work containing it. The extraction operation differs from copying in that it can be used to separate a part of a digital work from d-blocks or shells that place additional restrictions or fees on it. The extraction operation differs from the edit operation in that it does not change the contents of a work, only its embedding in d-blocks. Extraction creates a new digital work.

- The requester sends the server a message to initiate an Extract transaction. This message indicates the part of the work to be extracted, the version of the extract right to be used in the transaction, the destination address information for placing the part as a new work, the file data for the work, and the number of copies involved.
- The repositories perform the common opening transaction steps
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the new work. Otherwise, the rights of the original are transmitted. The Copy-Count field for this right is set to the number-of-copies requested.
- The requester records the contents, data, and usage rights and stores the work. It records the date and time that new work was made in the properties of the work.
- The repositories perform the common closing transaction steps.

The Embed Transaction

An embed transaction is a request to make a digital work become a part of another digital work or to add a shell d-block to enable the adding of fees by a distributor of the work.

- The requester sends the server a message to initiate an Embed transaction. This message indicates the work to be embedded, the version of the embed right to be used in the transaction, the destination address information for placing the part as a a work, the file data for the work, and the number of copies involved.
- The server checks the control specifications for all of the rights in the part and the destination. If they are incompatible, the server ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the new work. Otherwise, the rights of the original are transmitted. The Copy-Count field for this right is set to the number-of-copies requested.
- The requester records the contents, data, and usage rights and embeds the work in the destination file.
- The repositories perform the common closing transaction steps.

The Edit Transaction

An Edit transaction is a request to make a new digital work by copying, selecting and modifying portions of an existing digital work. This operation can actually change the contents of a digital work. The kinds of changes that are

permitted depend on the process being used. Like the extraction operation, edit operates on portions of a digital work. In contrast with the extract operation, edit does not affect the rights or location of the work. It only changes the contents. The kinds of changes permitted are determined by the type specification of the processor specified in the rights. In the currently preferred embodiment, an edit transaction changes the work itself and does not make a new work. However it would be a reasonable variation to cause a new copy of the work to be made.

- The requester sends the server a message to initiate an Edit transaction. This message indicates the work to be edited, the version of the edit right to be used in the transaction, the file data for the work (including its size), the process-ID for the process, and the number of copies involved.
- The server checks the compatibility of the process-ID to be used by the requester against any process-ID specification in the right. If they are incompatible, it ends the transaction with an error.
- The repositories perform the common opening transaction steps.
- The requester uses the process to change the contents of the digital work as desired. (For example, it can select and duplicate parts of it; combine it with other information; or compute functions based on the information. This can amount to editing text, music, or pictures or taking whatever other steps are useful in creating a derivative work.)
- The repositories perform the common closing transaction steps.

The edit transaction is used to cover a wide range of kinds of works. The category describes a process that takes as its input any portion of a digital work and then modifies the input in some way. For example, for text, a process for editing the text would require edit rights. A process for "summarizing" or counting words in the text would also be considered editing. For a music file, processing could involve changing the pitch or tempo, or adding reverberations, or any other audio effect. For digital video works, anything which alters the image would require edit rights. Examples would be colorizing, scaling, extracting still photos, selecting and combining frames into story boards, sharpening with signal processing, and so on.

Some creators may want to protect the authenticity of their works by limiting the kinds of processes that can be performed on them. If there are no edit rights, then no processing is allowed at all. A processor identifier can be included to specify what kind of process is allowed. If no process identifier is specified, then arbitrary processors can be used. For an example of a specific process, a photographer may want to allow use of his photograph but may not want it to be colorized. A musician may want to allow extraction of portions of his work but not changing of the tonality.

Authorization Transactions

There are many ways that authorization transactions can be defined. In the following, our preferred way is to simply define them in terms of other transactions that we already need for repositories. Thus, it is convenient sometimes to speak of "authorization transactions," but they are actually made up of other transactions that repositories already have.

A usage right can specify an authorization-ID, which identifies an authorization object (a digital work in a file of a standard format) that the repository must have and which it must process. The authorization is given to the generic authorization (or ticket) server of the repository which begins to interpret the authorization.

As described earlier, the authorization contains a server identifier, which may just be the generic authorization server or it may be another server. When a remote authorization server is required, it must contain a digital address. It may also contain a digital certificate.

If a remote authorization server is required, then the authorization process first performs the following steps:

- The generic authorization server attempts to set up the communications channel. (If the channel cannot be set up, then authorization fails with an error.)
- When the channel is set up, it performs a registration process with the remote repository. (If registration fails, then the authorization fails with an error.)
- When registration is complete, the generic authorization server invokes a "Play" transaction with the remote repository, supplying the authorization document as the digital work to be played, and the remote authorization server (a program) as the "player." (If the player cannot be found or has some other error, then the authorization fails with an error.)
- The authorization server then "plays" the authorization. This involves decrypting it using either the public key of the master repository that issued the certificate or the session key from the repository that transmitted it. The authorization server then performs various tests. These tests vary according to the authorization server. They include such steps as checking issue and validity dates of the authorization and checking any hot-lists of known invalid authorizations. The authorization server may require carrying out any other transactions on the repository as well, such as checking directories, getting some person to supply a password, or playing some other digital work. It may also invoke some special process for checking information about locations or recent events. The

"script" for such steps is contained within the authorization server.

- If all of the required steps are completed satisfactorily the authorization server completes the transaction normally signaling that authorization is granted.

The Install Transaction

An Install transaction is a request to install a digital work as runnable software on a repository. In a typical case the requester repository is a rendering repository and the software would be a new kind or new version of a player. Also in a typical case, the software would be copied to file system of the requester repository before it is installed.

- The requester sends the server an Install message. This message indicates the work to be installed, the version of the Install right being invoked, and the file data for the work (including its size).
- The repositories perform the common opening transaction steps.
- The requester extracts a copy of the digital certificate for the software. If the certificate cannot be found or the master repository for the certificate is not known to the requester the transaction ends with an error.
- The requester decrypts the digital certificate using the public key of the master repository, recording the identity of the supplier and creator, a key for decrypting the software, the compatibility information, and a tamper-checking code. (This step certifies the software.)
- The requester decrypts the software using the key from the certificate and computes a check code on it using a 1-way hash function. If the check-code does not match the tamper-checking code from the certificate the installation transaction ends with an error. (This step assures that the contents of the software, including the various scripts, have not been tampered with.)
- The requester retrieves the instructions in the compatibility-checking script and follows them. If the software is not compatible with the repository, the installation transaction ends with an error. (This step checks platform compatibility.)
- The requester retrieves the instructions in the installation script and follows them. If there is an error in this process (such as insufficient resources), then the transaction ends with an error. Note that the installation process puts the runnable software in a place in the repository where it is no longer accessible as a work for exercising any usage rights other than the execution of the software as part of repository operations in carrying out other transactions.
- The repositories perform the common closing transaction steps

The Uninstall Transaction

An Uninstall transaction is a request to remove software from a repository. Since uncontrolled or incorrect removal of software from a repository could compromise its behavioral integrity, this step is controlled.

- The requester sends the server an Uninstall message. This message indicates the work to be uninstalled, the version of the Uninstall right being invoked, and the file data for the work (including its size).
- The repositories perform the common opening transaction steps.
- The requester extracts a copy of the digital certificate for the software. If the certificate cannot be found or the master repository for the certificate is not known to the requester the transaction ends with an error.
- The requester checks whether the software is installed. If the software is not installed the transaction ends with an error.
- The requester decrypts the digital certificate using the public key of the master repository, recording the identity of the supplier and creator, a key for decrypting the software, the compatibility information, and a tamper-checking code. (This step authenticates the certification of the software, including the script for uninstalling it.)
- The requester decrypts the software using the key from the certificate and computes a check code on it using a 1-way hash function. If the check-code does not match the tamper-checking code from the certificate, the installation transaction ends with an error. (This step assures that the contents of the software, including the various scripts, have not been tampered with.)
- The requester retrieves the instructions in the uninstallation script and follows them. If there is an error in this process (such as insufficient resources), then the transaction ends with an error.
- The repositories perform the common closing transaction steps.

Claims

1. A method for controlling access to and distribution of a composite digital work, said composite digital work com-

prising a plurality of parts, said method comprising the steps of:

a) creating a composite digital work.
b) creating a description structure for said composite digital work, said description structure comprising a plurality of description blocks, each of said description blocks storing access information for at least one of said plurality of parts of said composite digital work;
c) storing said description structure and said composite digital work in a repository;
d) said repository receiving a request to access said composite digital work, said request having one or more request attributes; and
e) said repository determining if said request may be granted by examining the access information for each description block of said description structure of said composite digital work with respect to said one or more request attributes of said request.

2. The method as recited in Claim 1 wherein said step of creating a composite digital work is further comprised of the steps of:

a1) creating a first part of said digital work;
a2) creating a first description block for said first part of said composite digital work;
a3) obtaining an existing second part for said composite digital work, said second part of the digital work having a second description block.
a4) combining said first part and said second part to form said composite digital work, and
a5) creating a third description block for said composite digital work.

3. The method as recited in Claim 2 wherein said step of creating a description structure for said composite digital work is further comprised of the step of linking said first description block, said second description block and said third description block to correspond to the organization of said composite digital work.

4. The method as recited in Claim 3 wherein said step of storing said description structure and said composite digital work in a repository is further comprised of the steps of storing said description structure in a first storage means and said composite digital work in a second storage means.

5. The method as recited in Claim 4 wherein each of said first description block, said second description block and said third description block is comprised of a pointer to a corresponding part of said composite digital work stored in said second storage means and a control information part for storing usage rights for said corresponding part of said digital work and said step of creating a first description block for said first part of said composite digital work is further comprised of the step of specifying a first set of usage rights and storing in said control information part of said first description block.

6. The method as recited in Claim 1 wherein said step of creating a description structure for said composite digital work is further comprised of the step of adding a shell description block for specifying usage rights and fees of a distributor of said composite digital work

7. The method as recited in Claim 2 wherein said step of obtaining an existing second part for said composite digital work is further comprised of the step of extracting said second part from an existing digital work.

8. A repository for storing and controlling access to composite digital works comprising:

an interface means for receiving requests to access digital works stored therein.
a first storage unit for storing digital data representing digital works.
a second storage unit for storing description structures for digital works stored in said first storage unit, said description structure comprising a plurality of description blocks, each of said description blocks comprising: a pointer to a parent description block, one or more pointers to children description blocks, a pointer to a corresponding part of a digital work stored in said first storage unit and a usage rights part for storing one or more usage rights, each of said usage rights specifying an instance of how said part may be used;
a transactions processor for processing requests to access a digital work, said transactions processor comprising a means for identifying a usage right from a request to access said digital work, and a means for determining if a description block contains an identified usage right.

9. The repository as recited in claim 8 wherein said transaction processor is further comprised of usage rights conflict resolution means for resolving usage rights conflicts between different description blocks

10. A system for controlling access to and usage of composite digital works, said composite digital work comprising a plurality of digital works, said system comprising:

means for attaching usage rights to digital works, said usage rights indicating how a recipient of a digital work may use and subsequently distribute said digital work;
means for creating a description structure for said composite digital work, said description structure comprising a description block for each digital work of said composite digital work, said description block comprising said usage rights for said digital work and addressing information for said digital work;
a plurality of repositories for managing exchange of digital works based on usage rights attached to said digital works, each of said plurality of document repositories comprising a storage means for storing digital works, a processor having a first server mode of operation for processing access requests to said digital works and a second requester mode of operation for initiating requests to access digital works, a timekeeping means and a connection means;
a rendering system for rendering of digital works, said rendering system comprising a rendering repository for secure receipt of composite digital works and a rendering device having means for converting digital signals to signals suitable for rendering of said digital works.

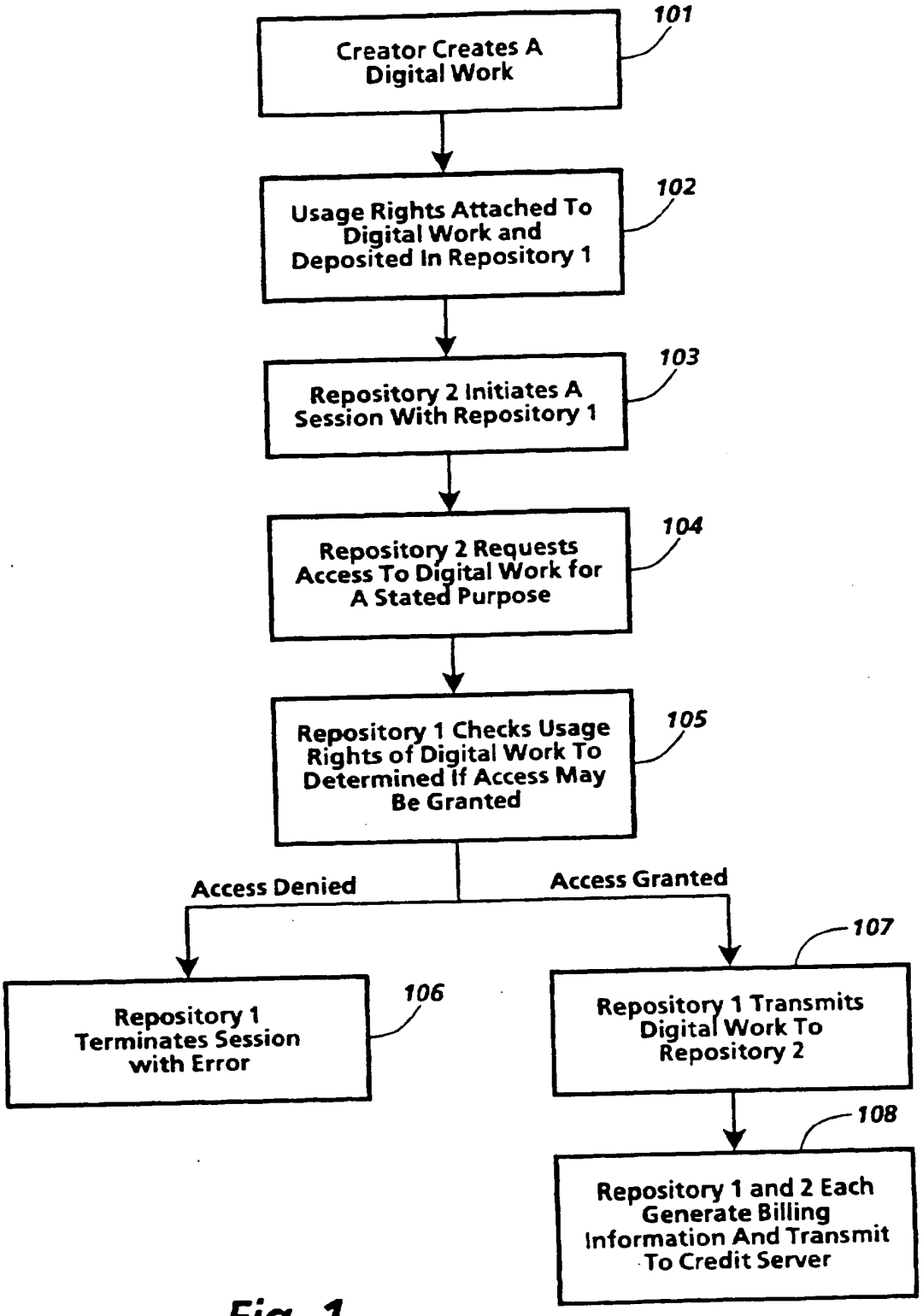


Fig. 1

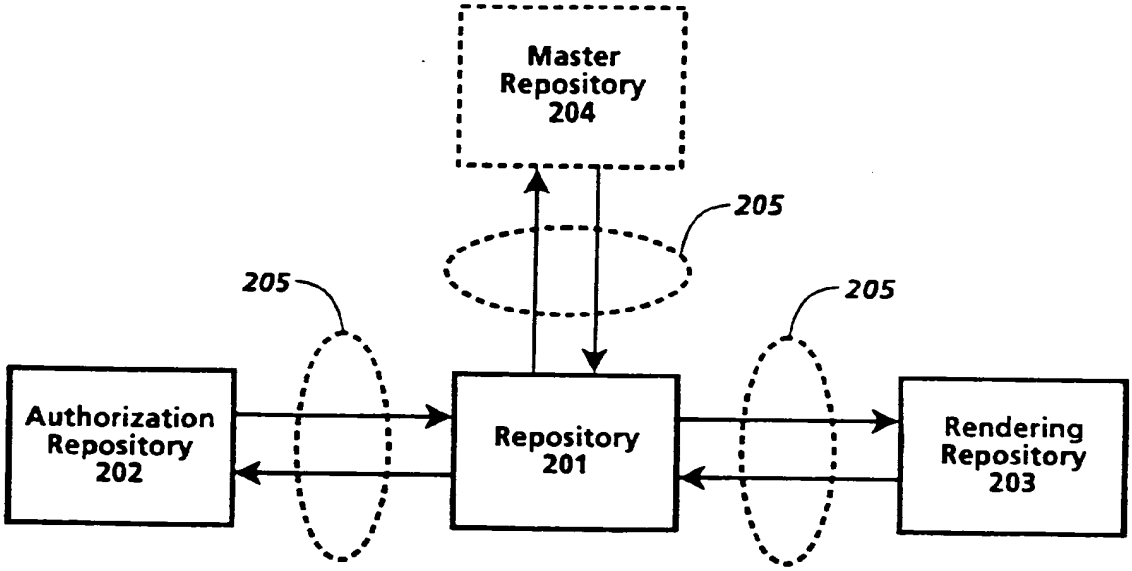


Fig. 2

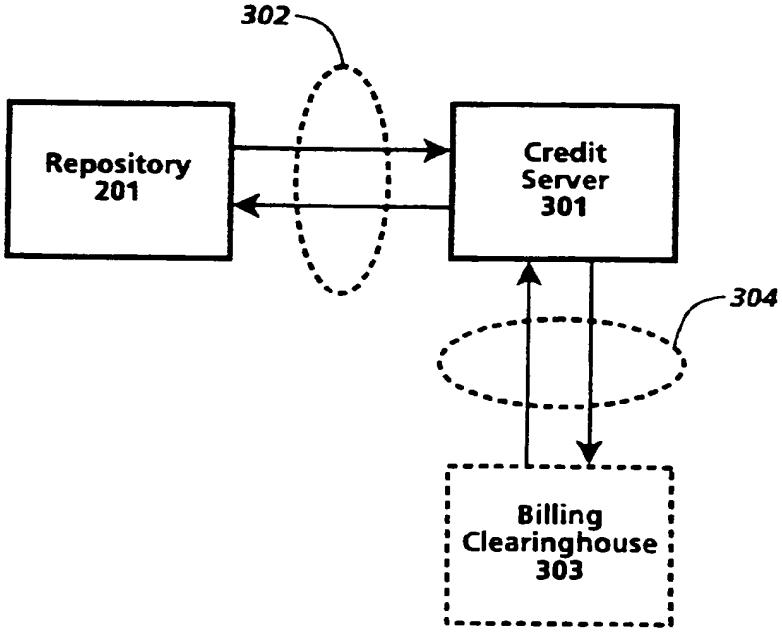


Fig. 3

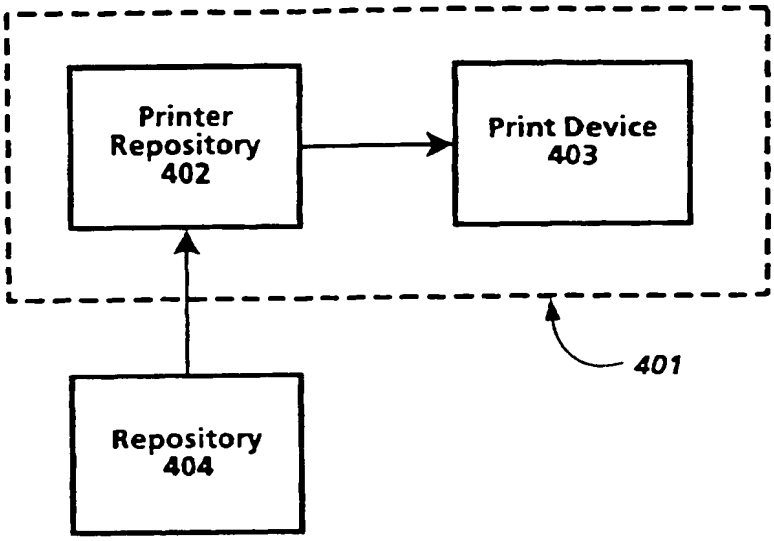


Fig. 4a

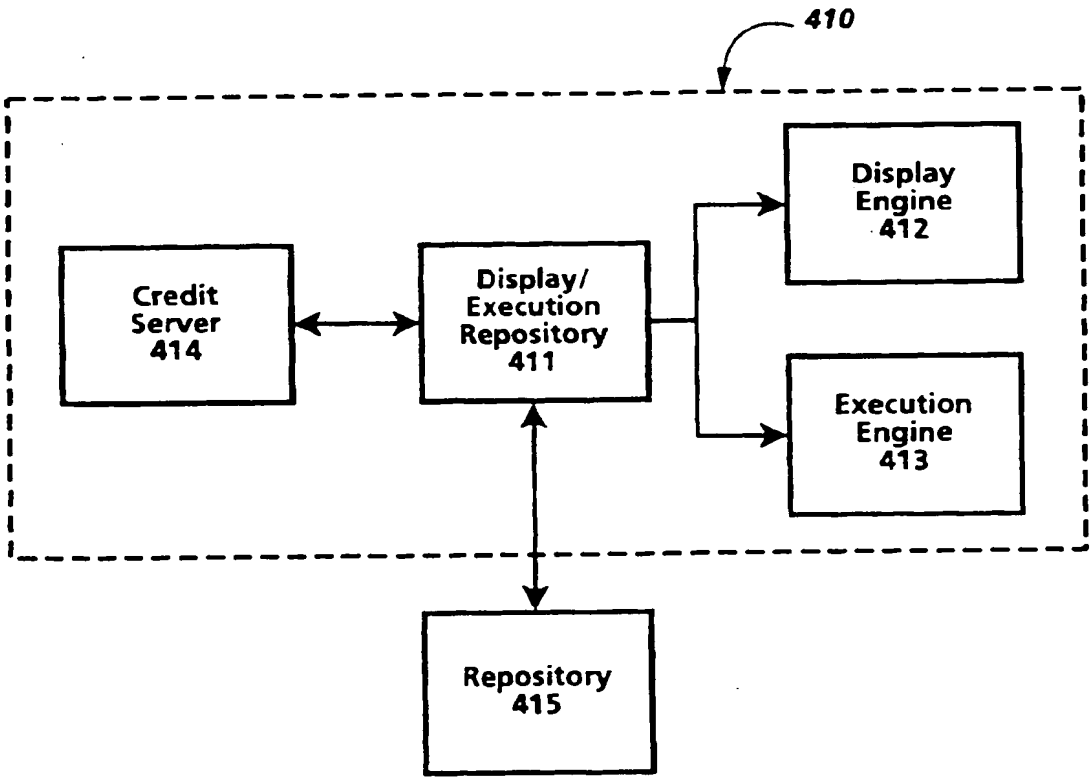


Fig. 4b

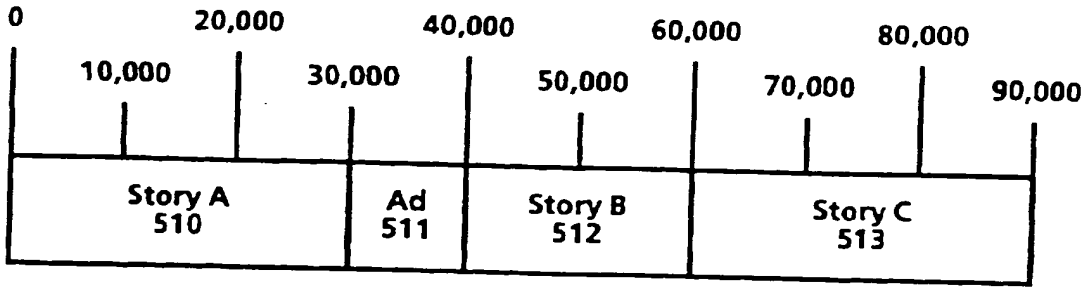


Fig. 5

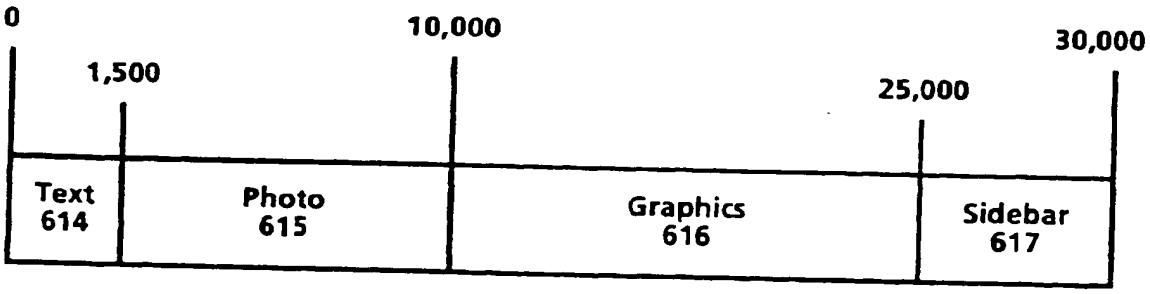


Fig. 6

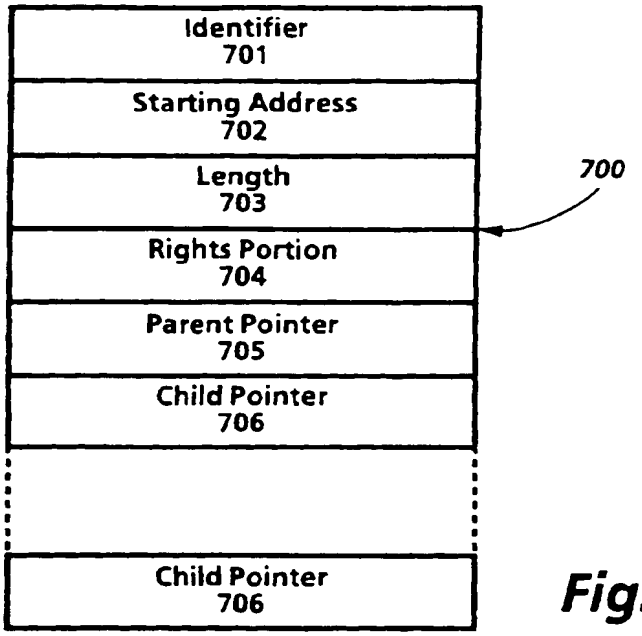


Fig. 7

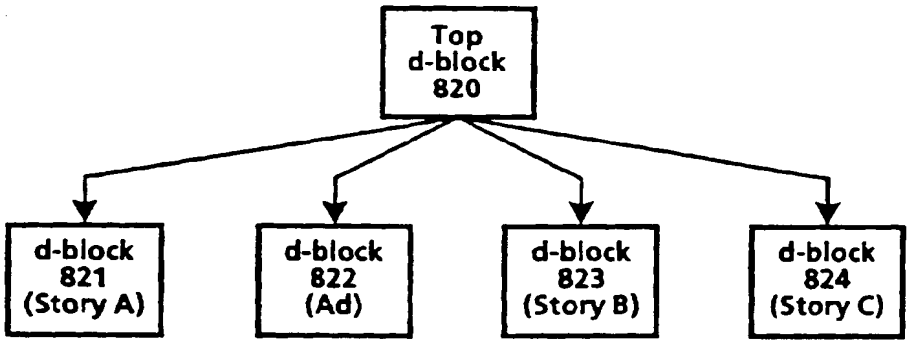


Fig. 8

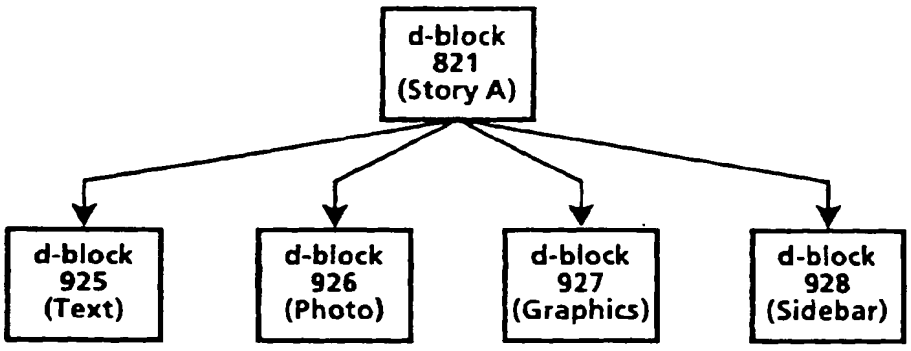


Fig. 9

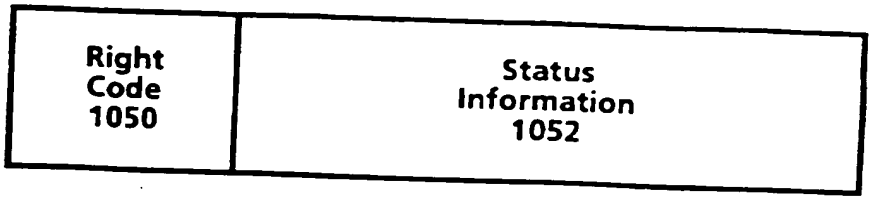


Fig.10

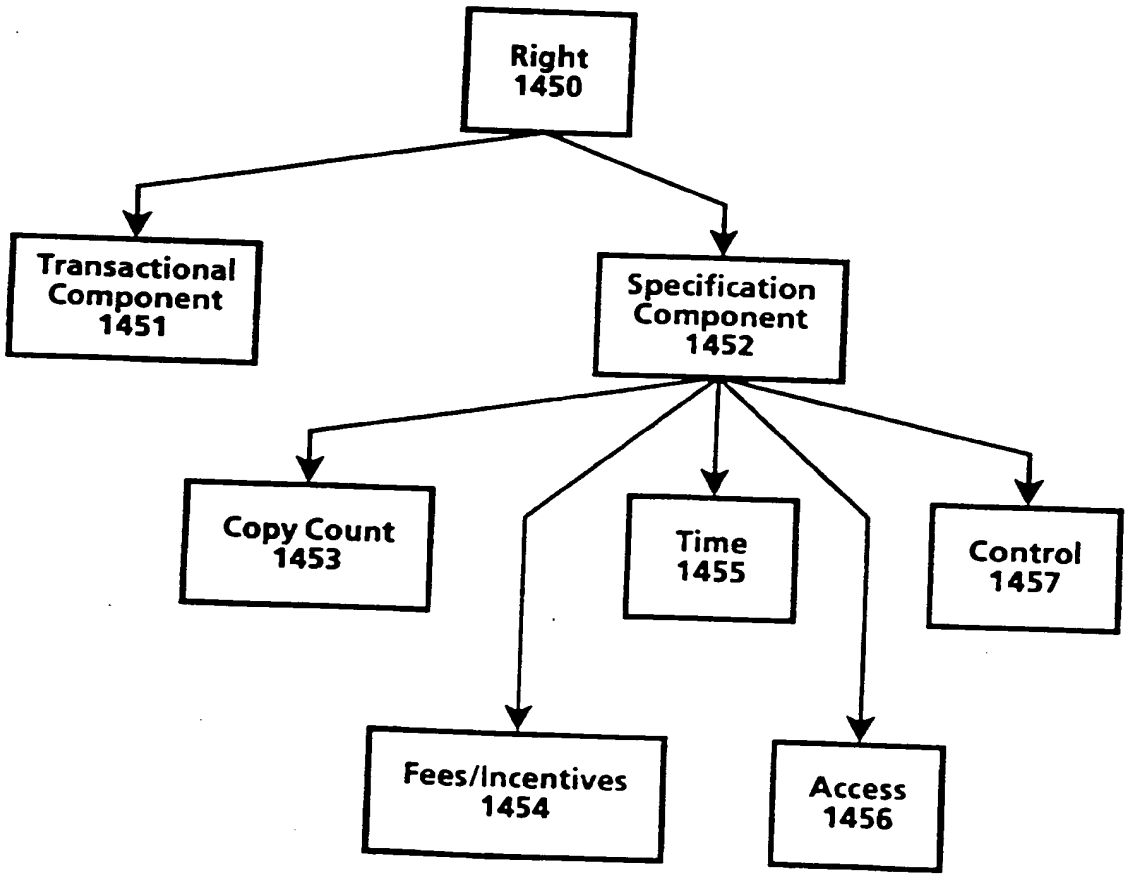


Fig.14

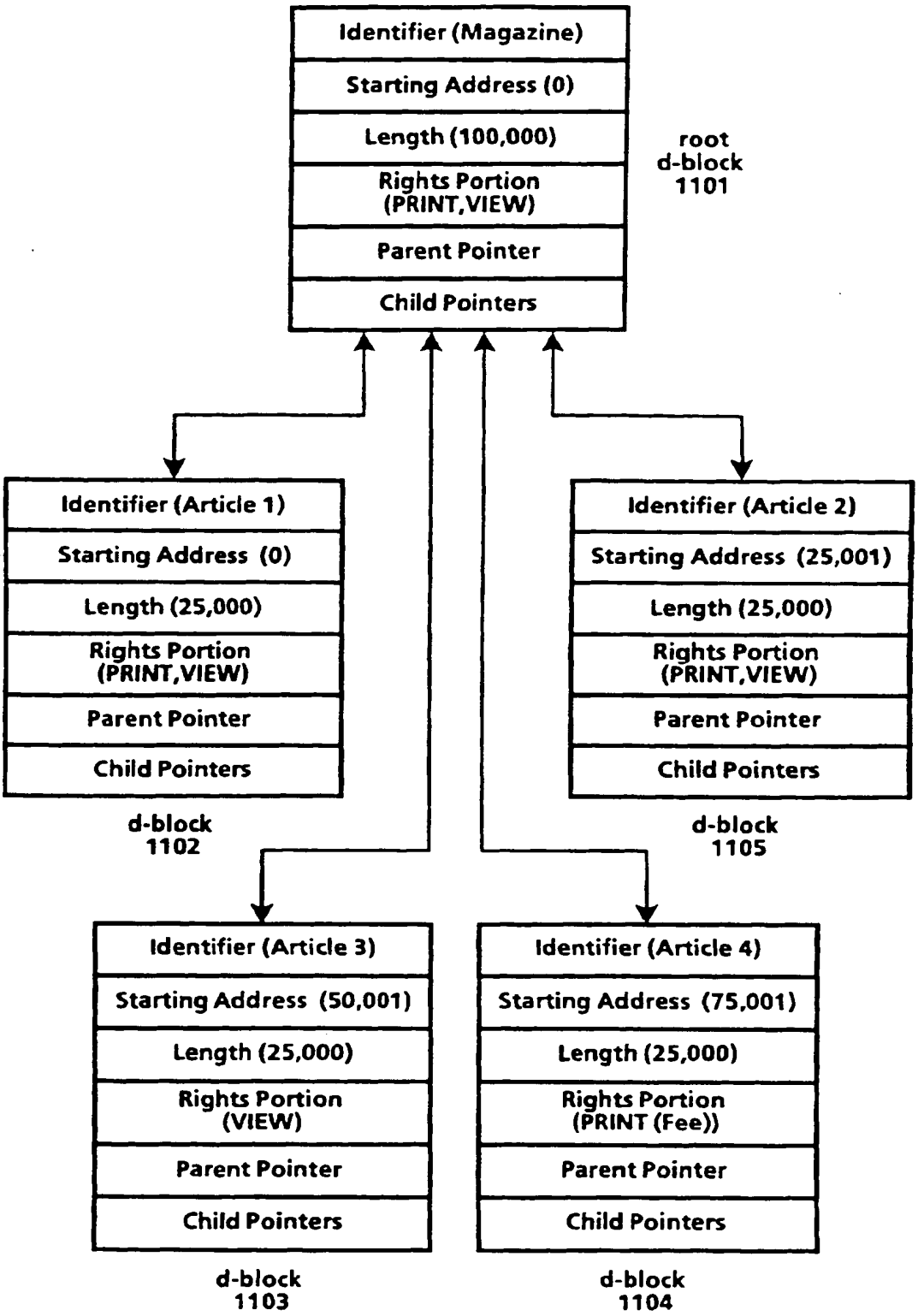


Fig.11

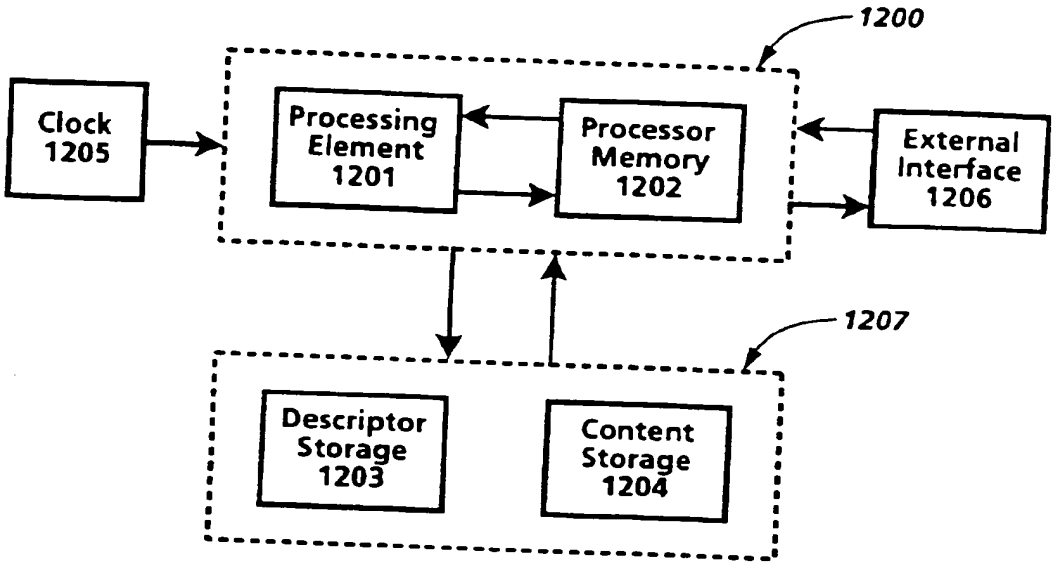


Fig.12

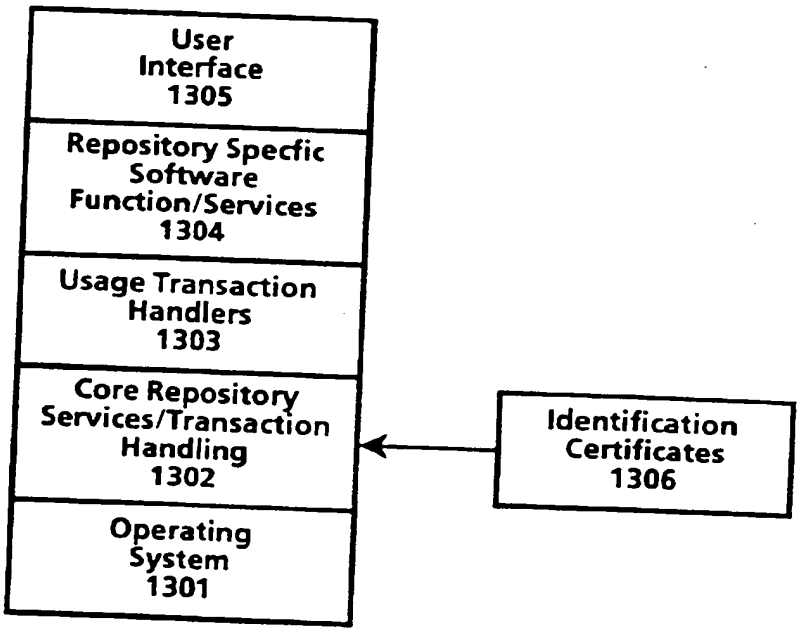


Fig.13

1501 ~ **Digital Work Rights:** = (Rights*)
1502 ~ **Right** : = (Right-Code {Copy-Count} {Control-Spec} {Time-Spec } {Access-Spec} {Fee-Spec})
1503 ~ **Right-Code** : = Render-Code | Transport-Code | File-Management-Code| Derivative-Works- Code | Configuration-Code
1504 ~ **Render-Code** : = [Play : {Player: Player-ID} | Print: {Printer: Printer-ID}]
1505 ~ **Transport-Code** : = [Copy | Transfer | Loan {Remaining-Rights: Next-Set-of-Rights}]{(Next-Copy-Rights: Next-Set-of-Rights)}
1506 ~ **File-Management-Code** := Backup {Back-Up-Copy-Rights: Next-Set-of-Rights} | Restore | Delete | Folder | Directory {Name: Hide-Local | Hide-Remote} {Parts: Hide-Local | Hide-Remote}
1507 ~ **Derivative-Works-Code** : = [Extract | Embed | Edit{Process: Process-ID}] {Next-Copy-Rights : Next-Set-of Rights}
1508 ~ **Configuration-Code** : = Install | Uninstall
1509 ~ **Next-Set-of-Rights** : = {(Add: Set-Of-Rights)} {(Delete: Set-Of-Rights)} {(Replace: Set-Of-Rights)}{(Keep: Set-Of-Rights)}
1510 ~ **Copy-Count** : = (Copies:positive-integer | 0 | Unlimited)
1511 ~ **Control-Spec** : = (Control: {Restrictable | Unrestrictable} {Unchargeable | Chargeable})
1512 ~ **Time-Spec** : = ({Fixed-Interval | Sliding-Interval | Meter-Time} Until: Expiration-Date)
1513 ~ **Fixed-Interval** : = From: Start-Time
1514 ~ **Sliding-Interval** : = Interval: Use-Duration
1515 ~ **Meter-Time**: = Time-Remaining: Remaining-Use
1516 ~ **Access-Spec** : = ({SC: Security-Class} {Authorization: Authorization-ID*} {Other-Authorization: Authorization-ID*} {Ticket: Ticket-ID})
1517 ~ **Fee-Spec**: = {Scheduled-Discount} Regular-Fee-Spec | Scheduled-Fee-Spec | Markup-Spec
1518 ~ **Scheduled-Discount**: = Scheduled-Discount: (Scheduled-Discount: (Time-Spec Percentage)*)
1519 ~ **Regular-Fee-Spec** : = ({Fee: | Incentive: } [Per-Use-Spec | Metered-Rate-Spec | Best-Price-Spec | Call-For-Price-Spec] {Min: Money-Unit Per: Time-Spec}{Max: Money-Unit Per: Time-Spec} To: Account-ID)
1520 ~ **Per-Use-Spec**: = Per-Use: Money-unit
1521 ~ **Metered-Rate-Spec** : = Metered: Money-Unit Per: Time-Spec
1522 ~ **Best-Price-Spec** : = Best-Price: Money-unit Max: Money-unit
1523 ~ **Call-For-Price-Spec** : = Call-For -Price
1524 ~ **Scheduled-Fee-Spec**: = (Schedule: (Time-Spec Regular-Fee-Spec)*)
1525 ~ **Markup-Spec**: = Markup: percentage To: Account-ID

Fig.15

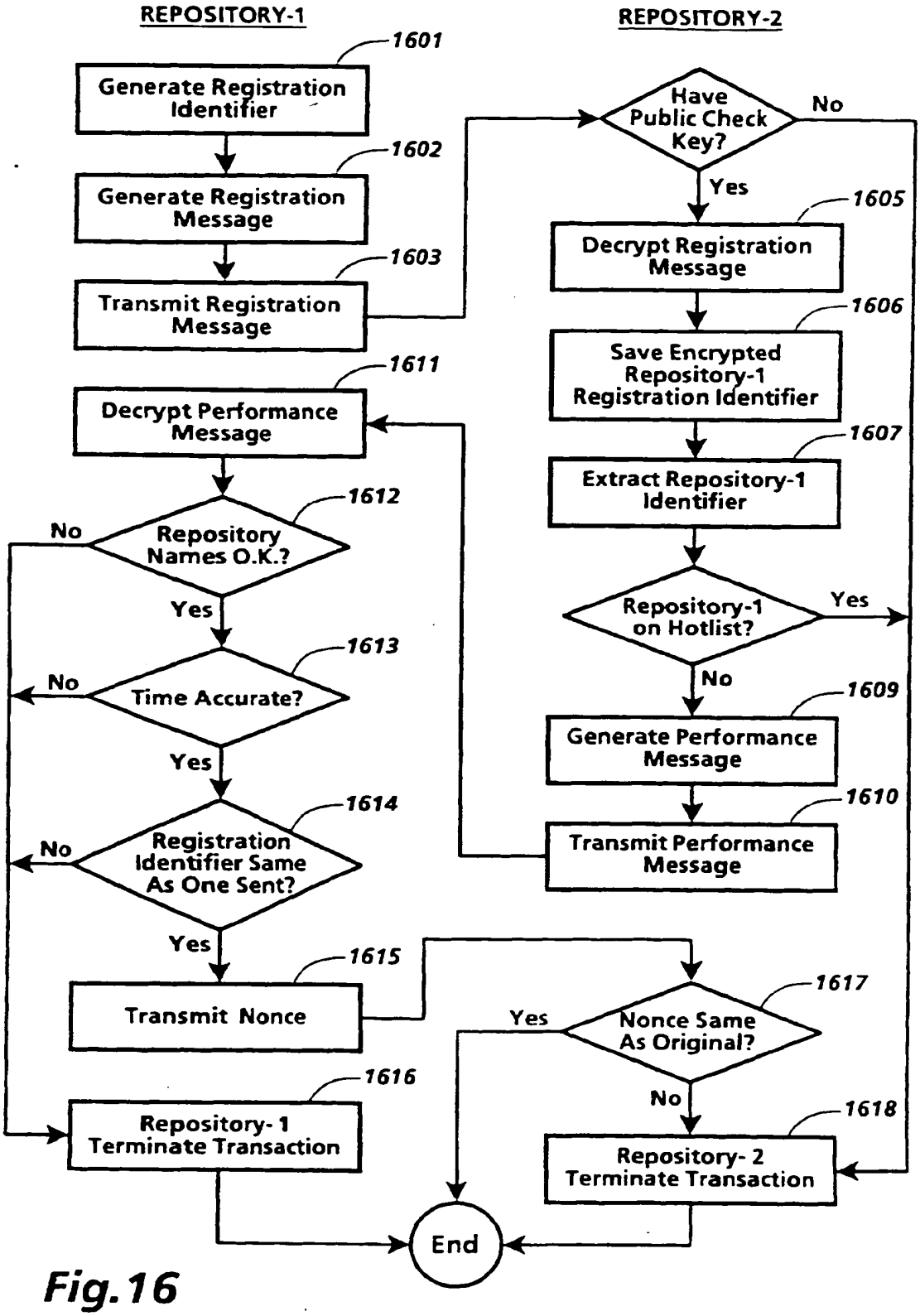


Fig.16

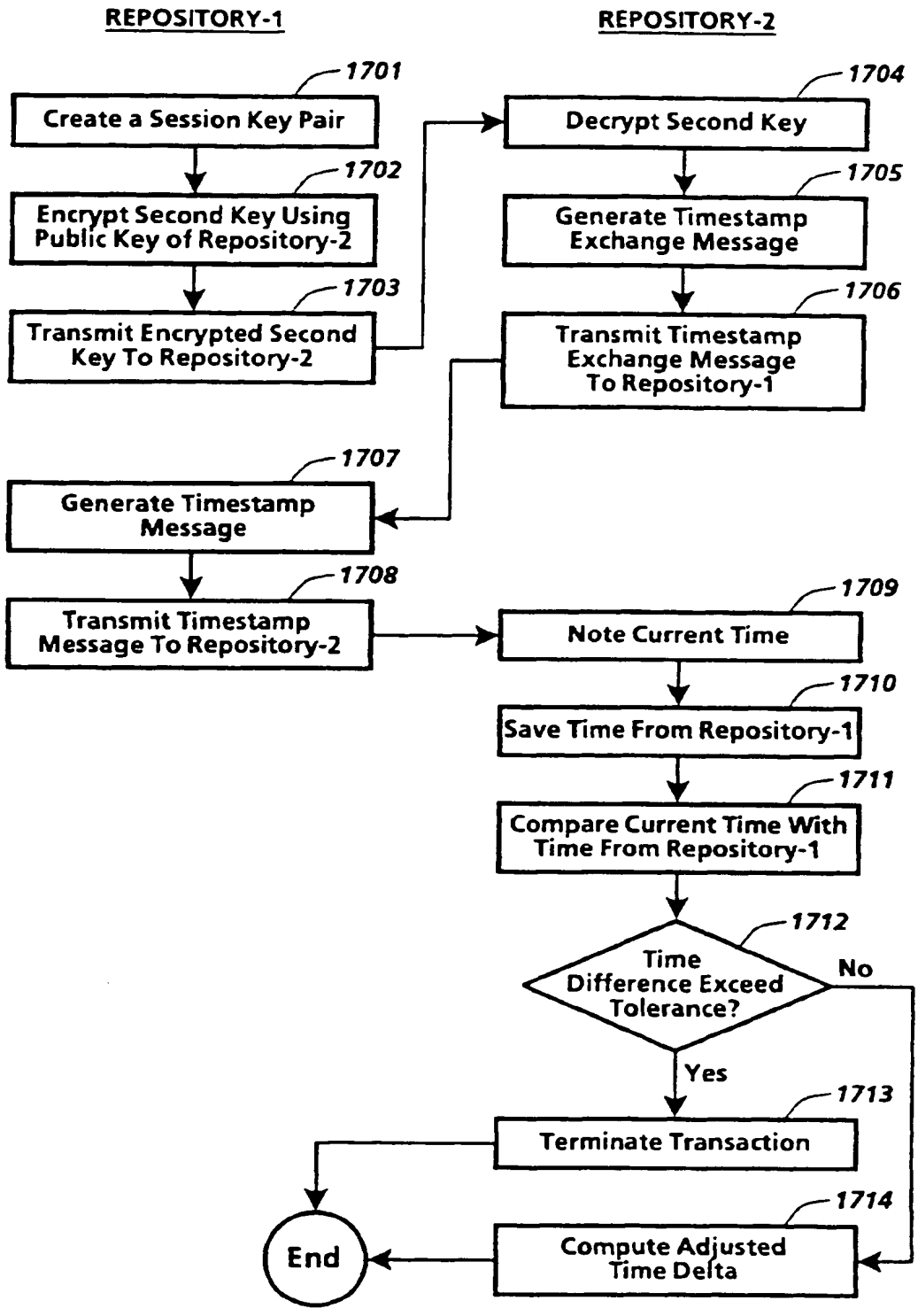


Fig.17

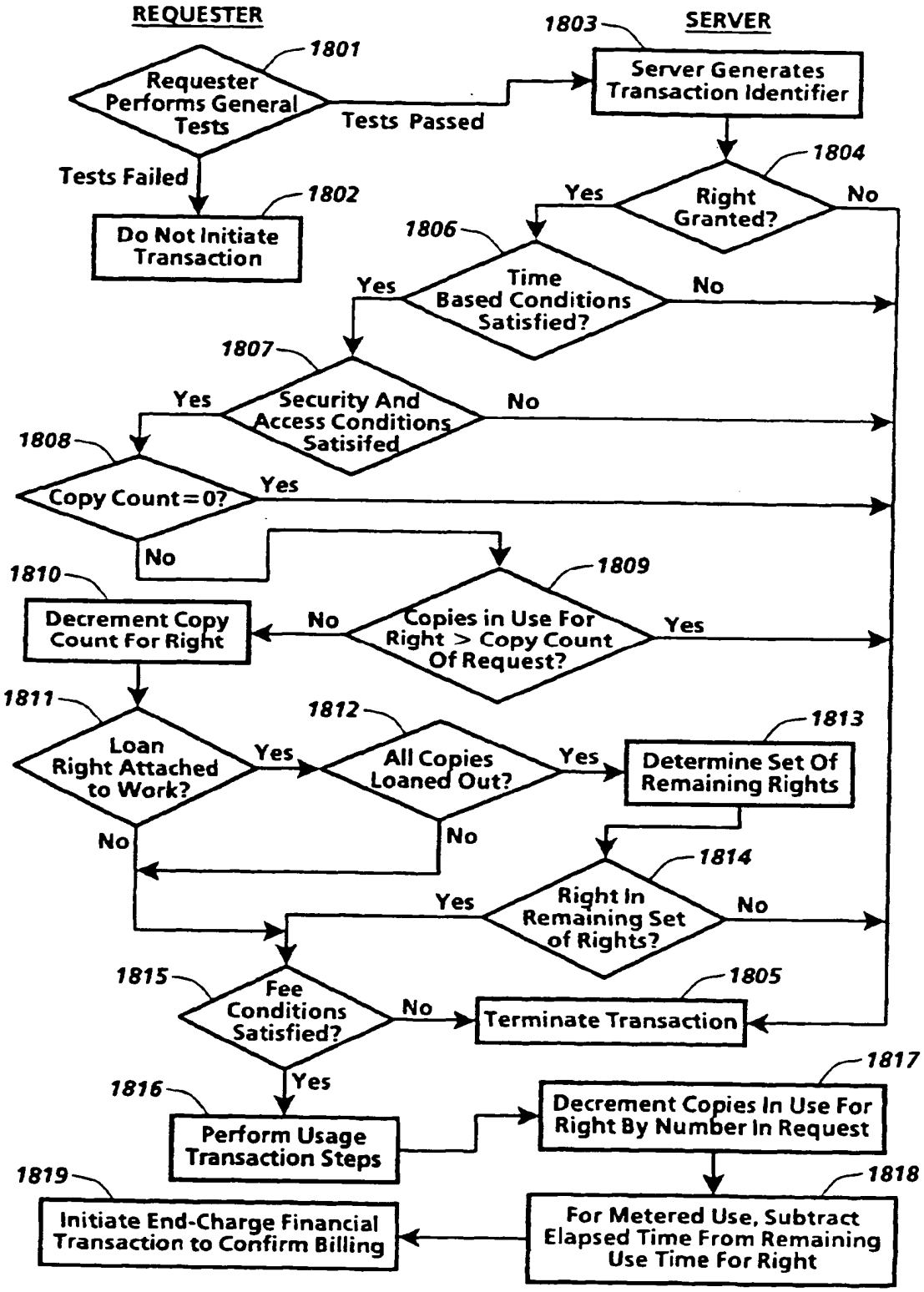


Fig. 18

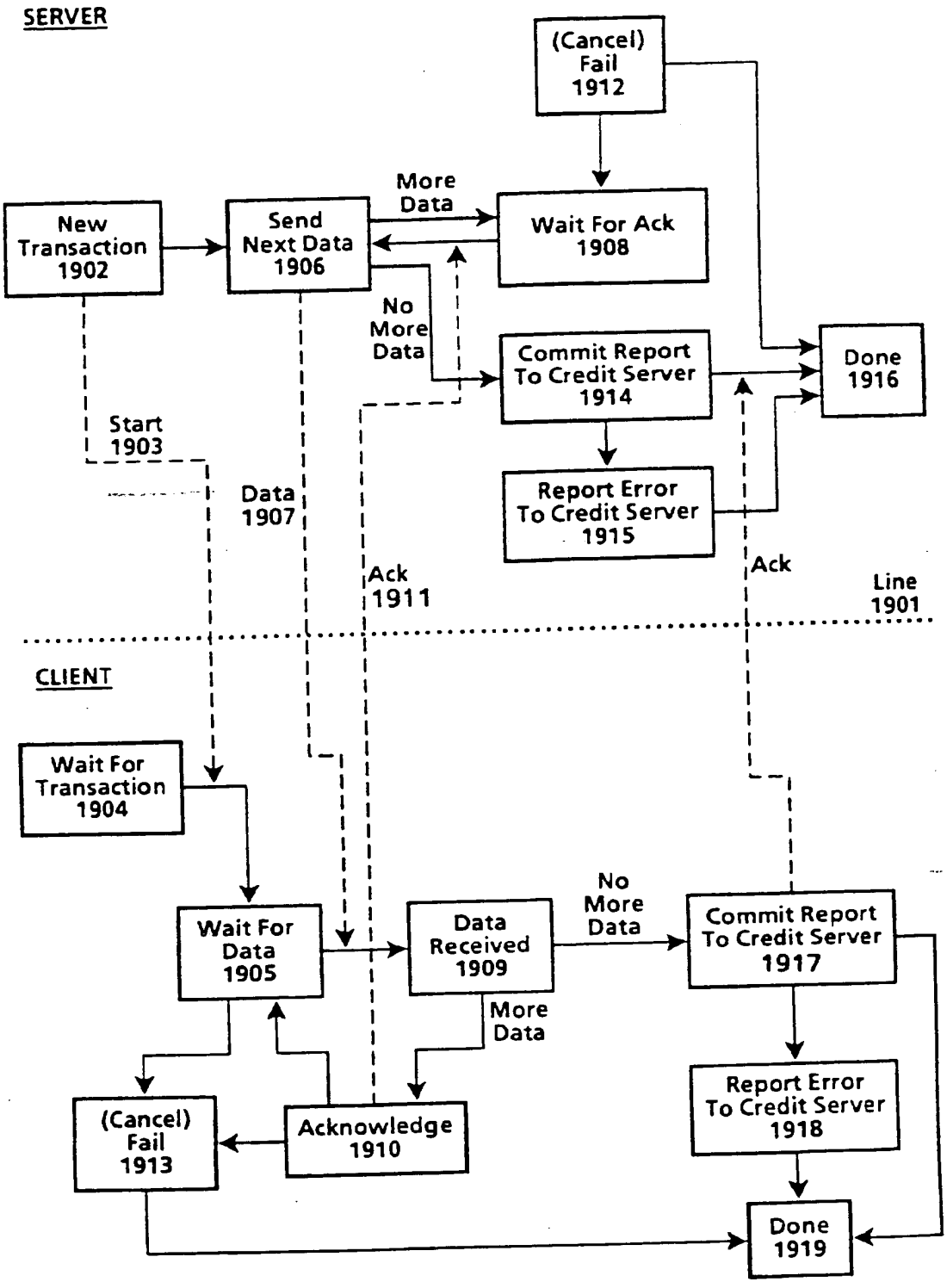


Fig.19

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 95 30 8421

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-A-0 332 707 (MIYOSHI ET ALL) * column 43, line 22 - column 46, line 38; figures 2,28-31 *	1,8	G06F1/00
A	COMPUTERS AND GRAPHICS, vol. 10, no. 2, 1986 OXFORD GB, pages 119-131, U. FLASCHE ET AL. 'DECENTRALIZED PROCESSING OF DOCUMENTS' * page 120, right column, line 14 - page 121, left column, line 57 * * page 125, left column, line 8 - page 126, left column, line 10; figures 1,5-7 *	1,8,10	
A	WO-A-92 20022 (DIGITAL EQUIPMENT CORP.) * page 45, line 10 - page 64, line 17 *	1,8,10	
A	US-A-5 291 596 (MITA) * the whole document *	1,8,10	

TECHNICAL FIELDS SEARCHED (Int.Cl.6)
G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	1 April 1996	Moens, R

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1501 01.82 (P04C01)

EP 25665 (3)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 February 2001 (22.02.2001)

PCT

(10) International Publication Number
WO 01/13198 A1

(51) International Patent Classification⁷: G06F 1/00

(21) International Application Number: PCT/GB00/03095

(22) International Filing Date: 11 August 2000 (11.08.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
99306415.3 13 August 1999 (13.08.1999) EP
9922669.8 25 September 1999 (25.09.1999) GB

(71) Applicant *(for all designated States except US)*: **HEWLETT-PACKARD COMPANY** [US/US]; 3000 Hanover Street, Palo Alto, CA 94304 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **PEARSON, Siani, Lynne** [GB/GB]; 35 Sandyleaze, Westbury-on-Trym, Bristol BS9 3PZ (GB). **PROUDLER, Graeme, John** [GB/GB]; 5 Touchstone Avenue, Stoke Gifford, Bristol BS34 8XQ (GB).

(74) Agent: **LAWRENCE, Richard, Anthony**; Hewlett-Packard Limited, Intellectual Property Section, Filton Road, Stoke Gifford, Bristol BS34 8QZ (GB).

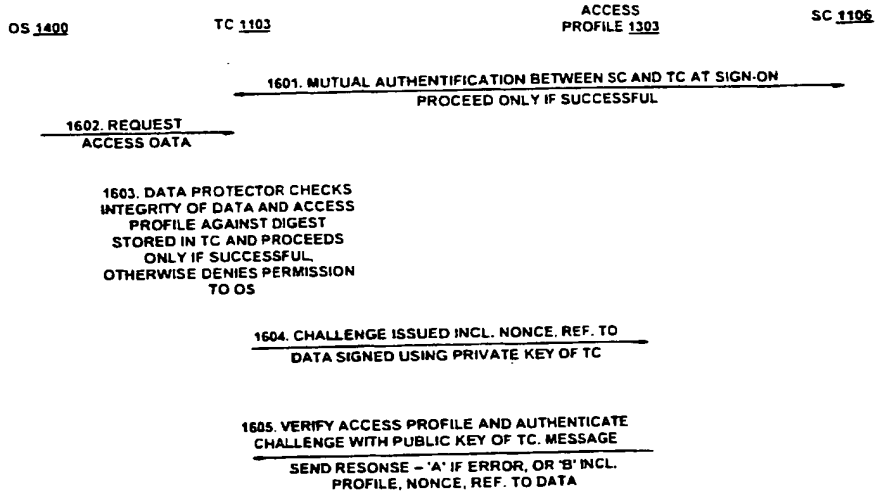
(81) Designated States *(national)*: JP, US.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

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.*

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.

(54) Title: ENFORCING RESTRICTIONS ON THE USE OF STORED DATA



(57) Abstract: A computer system is adapted to restrict operations on data. The computer system comprises a computer platform having a secure operator for checking whether a user of the platform is licensed to perform a requested operation on the data and for enabling use of the data; a portable trusted module containing a user identity, wherein a trusted module is a component adapted to behave in an expected manner and resistant to unauthorised external modification; and an access profile specifying license permissions of users with respect to the data. Advantageously the computer platform contains a platform trusted module, which engages in mutual authentication with the portable trusted module and which contains the secure operator. The secure operator is adapted to check the access profile to determine whether a requested operation is licensed for used identity contained in the portable trusted module. The secure operator prevents the requested operation if a license is required and not present.

WO 01/13198 A1

Enforcing Restrictions On The Use Of Stored Data

This invention relates to computer platforms and their method of operation, and is more particularly concerned with controlling and/or metering the use of data on computer platforms, particularly computer platforms that are available to a number of mobile users.

In this document, 'data' signifies anything that can be formatted digitally, such as images, software and streaming media.

In the future, computer systems will be able to achieve a more secure booting, together with integrity checks on other code to ensure that viruses or other unauthorised modifications have not been made to the operating systems and mounted software. In addition, a new generation of tamper-proof devices are already appearing or will soon appear on the market and include both external or portable components (such as smart cards) and internal components (embedded processors, semi-embedded processors or co-processors with security functionality, i.e. including motherboard, USB and ISA implementations). These tamper-proof components will be used to check that the hardware of the system has not been tampered with, and to provide a more reliable form of machine identity than currently available (for example, the machine's Ethernet name). Applicant's co-pending International Patent Application No. PCT/GB00/00528, filed on 15 February 2000, "Trusted Computing Platform", the entire contents of which are hereby incorporated herein by reference, describes a system adapted to enable verification of the integrity of a computer platform by the reliable measurement and reliable reporting of integrity metrics. This enables the verification of the integrity of a platform by either a local user or a remote entity.

The existence of such tamper-proof components and the possibility of secure booting does not by itself remove all security problems related to computing platform use. In particular, the counteraction of piracy, and the licencing and metering of software use in a manner that is acceptable to software developers and end-users, still provide major problems.

Software licensing is subject to hacking and piracy, and all the current software licensing methods used have problems associated with them. Software implementations of licensing (such as "licence management systems") are flexible, but not especially secure or fast. In particular, they suffer from a lack of security (for example, being subject to a generic "hack") and difficulty in genuine replacement of software. Conversely, hardware implementations ("dongles") are faster and

generally more secure than software implementations, but inflexible. They are tailored only for a particular piece of software and are inconvenient for end-users.

A common technique in the field of licence protection is to use a software wrapper to encode information relating to licensing and other protection measures. Data wrappers, or cryptographic containers, are commonly used within software-only and hybrid methods of data protection, but are not at present a very secure method of protection because they are vulnerable to alteration and removal, even if an integrity check is contained within the wrapper. In particular, the data wrapper is a prime target for hackers since it may contain a profile defined by the data's developer that governs the way in which the data may be executed, or other sensitive information which should not be altered. Authentication, encryption and integrity checks may be used to protect the wrapper from being modified en route to its being downloaded and stored onto the client platform. However, there is a major risk that it could be modified or deleted by a malicious entity, or by accident, once the data and associated wrapper are stored (for example, on a hard disk) within the client platform. Once modified, the data could then be used on the client platform in a way that is outside the scope of the profile defined in the original, unmodified wrapper.

One system to address such difficulties has been proposed in "Persistent Access Control to Prevent Piracy of Digital Information" Paul B. Schneck, Proceedings of the IEEE, Volume 87, No. 7, July 1999, PP1239-1250, which uses access control software to check licensing information before access to data. That system, however, only considered the case where a generic licence was operable for all users. Access control mechanisms cannot provide a complete solution to this problem because they can be bypassed and, moreover, they focus on controls specified by the user's administrator rather than the developer of the data. There are many situations where a platform is shared by users who have different software permissions. Existing systems do not satisfactorily address this issue, which is likely to become increasingly significant.

Summary of the Invention

Accordingly, the present invention provides a computer system adapted to restrict operations on data, comprising: a computer platform having a secure operator for checking whether a user of the platform is licensed to perform a requested operation on the data and for enabling use of the data; a portable trusted module containing a user identity, wherein a trusted module is a component adapted to behave in an expected manner and resistant to unauthorised external modification; and an access profile specifying license permissions of users with respect to the data; wherein the secure operator

is adapted to check the access profile to determine whether a requested operation is licensed for the user identity contained in the portable trusted module and prevent the requested operation if a license is required and not present. Preferably, the computer platform further comprises a platform trusted module, and wherein the platform trusted module and the portable trusted module are adapted for mutual authentication.

The present invention is applicable to software wrappers or other types of data licence used to protect and qualify the operations that may be performed upon data, such as copying, modification, or execution. A particularly preferred form of the present invention uses two trusted modules (TCs): the first is a portable TC typically held on a smart card, and the second is part of a computer platform. These are used in conjunction with software, preferably running within the TC, to ensure that data can only be used by the owner of the portable TC in ways specified by the developer.

In preferred embodiments of the system, some or all of the data is within the portable trusted module or in a device containing the portable trusted module, and the portable trusted module or the device containing the portable trusted module further comprises a data protector for checking data integrity before a processor of the computer platform carries out operations on the data. This arrangement has the advantage of providing a check that the associated access profile or other type of wrapper has not been altered or deleted after the data (together with access profile or wrapper) have been stored on the trusted client platform.

The present invention differs from Schneck's system in that it uses a client system where access checks are made according to the user identity (derived from a removable TC such as a smart card), but the checks themselves are made using the access control software mounted on the client PC or other client platform. Furthermore, the following are possible: (a) a licence can be associated with each end-user, instead of or as well as with the PC TC, which is necessary for certain types of access control; (b) it is not essential (although it is preferable) that the data is encrypted (c); to prevent modification of the licence (cf. access profile), a digest is stored in the TC upon loading and consulted before data access; (d) the access control code is protected at BIS (BOOT Integrity Service) and preferably runs within the TC, or else, there is a dedicated communications path between the code and TC that is inaccessible to other parts of the computer platform; (e) logs are made within the TC; and (f) the licence can have a more proactive role.

The motivation for this particular invention is that more complex models of data usage dictate greater flexibility, which can only be brought about effectively by using multiple TCs in the client

platform. In particular, hot-desking in an office environment or accessing information or services in from a shared terminal in a public place such as an airport can be modelled by having a TC in a shared client machine, and each user being issued with at least one portable TC that identifies this user. Use of appropriate embodiments of the present invention allows mobile users to have universal data access on trusted computer platforms by enforcing restrictions on the use of stored data via a user's licence associated with the data together with software that checks the validity of operations carried out on the data. The user's licence can be stored on or issued with a portable trusted device such as a smart card, downloaded together with the data, or sent separately from the data. There is an option to perform integrity checks on the data to ensure that the data has not been modified since installation. Hence, unauthorised operations on data such as copying can be prevented, together with modification of data or its associated licence on the same platform, while users can benefit from a hot-desking model of data access.

Embodiments of the present invention allows individual users to pay for data access that only they will benefit from. Users may capture such a licence on a tamper-resistant portable device that they can carry around with them, and use on any trusted platform, no matter where its location. Alternatively, individual licences held on trusted platforms can be customised to refer to a secure ID of the portable module. If the data itself if also captured on the portable device, it is not necessary for the data to be installed on the client machine. Optionally, the data can be downloaded as required, if not already installed.

In one aspect, the invention provides a computer platform having a secure operator for checking whether a user of the platform is licensed to perform a requested operation on the data and for enabling use of the data and an access profile specifying license permissions of users with respect to the data; wherein the secure operator is adapted to check the access profile to determine whether a requested operation is licensed for a user identity contained in a portable trusted module in communication with the computer platform, wherein a trusted module is a component adapted to behave in an expected manner and resistant to unauthorised external modification, and prevent the requested operation if a license is required and not present.

In a further aspect, the invention provides a portable trusted module containing a user identity, wherein a trusted module is a component adapted to behave in an expected manner and resistant to unauthorised external modification; the portable trusted module containing a user access license specifying access rights to data associated with the removable trusted module.

A particularly preferred embodiment of the present invention uses a trusted module (TC) of a computer platform (that is possibly shared by several users) in conjunction with software, preferably running within the TC, that ensures that restriction on the usage by each individual end-user of stored data specified by the developer must be adhered to, that the different end-users can have different access profiles and in addition that data cannot be used on the platform if the data or associated wrapper or licence has been modified since the initial download onto the platform. The host CPU requests a policy check before acting upon data, by sending the name of the target data plus the intended operation to a TC. The TC checks the ID of the user that is logged in (via the ID of the portable TC), and the restrictions corresponding to this current end-user that are associated with the target data. Those restrictions could be on who may access the data, on the number of times the data can be used, the operations which may not be carried out, and so on, or the restrictions might have deliberately been loaded as 'NULL'. The TC checks the proposed usage with the restrictions. If no appropriate permission of found, the TC checks for a licence on the portable TC (advantageously a smart card) and for valid permission for the data usage within this. The TC then replies to the CPU with or without the access permission, as appropriate. The CPU is not able to carry out certain operations on target data such as copy, edit, add section, replace section, execute, delete, print, open, scan, rename, move location, send to or read without obtaining the appropriate permission from the TC in such a manner. Preferably, the integrity of the target data and restrictions is checked before the operation is carried out to ensure that they have not been illegally or accidentally modified on the platform. Alternatively, the checking can be carried out on the portable TC itself.

A significant component of the system is the access profile associated with each piece of application software or data, which specifies the data to be protected and specifies the type of operations that the developer wishes to be carried out upon that particular software or data. Optionally, the access profile specifies any other particular information to be checked for in carrying out certain operations, such as a particular TC ID or a secret which is to be checked for in the TC or current signed-on smart card. Another possibility is for the access profile to run, preferably together with the data, within a TC or smart card (suitably segmented). The access profile can be thought of as a form of licence or cryptographic container associated with the data.

In a further aspect, the invention provides a computer system adapted to restrict operations on data, comprising: a computer platform having an access profile for specifying license permissions of users with respect to the data and for enabling use of the data; a portable trusted module containing a user identity, wherein a trusted module is a component adapted to behave in an expected manner and

resistant to unauthorised external modification; wherein the access profile is adapted to determine whether a requested operation is licensed for the user identity contained in the portable trusted module and prevent the requested operation if a license is required and not present.

In a still further aspect, the invention provides a method of restricting operations on data in a system comprising: a computer platform having an access profile specifying license permissions of users with respect to the data; and for enabling use of the data; a portable trusted module containing a user identity, wherein a trusted module is a component adapted to behave in an expected manner and resistant to unauthorised external modification; the method comprising a request for a policy check by the operating system of the computer platform to the access profile before acting upon the data, by sending to the access profile the name of the target data plus the intended operation the access profile checking the restrictions associated with the target data to determine whether the data may be operated upon; and replying to the operating system.

In these aspects of the invention, the access profile takes a more proactive role. The access profile, rather than the secure operator, takes the role of controlling the operating system's ability to execute the restricted data.

In a still further aspect, the invention provides a method of restricting operations on data in a system comprising: a computer platform having a secure operator for checking whether a user of the platform is licensed to perform a requested operation on the data and for enabling use of the data; a portable trusted module containing a user identity, wherein a trusted module is a component adapted to behave in an expected manner and resistant to unauthorised external modification; and an access profile specifying license permissions of users with respect to the data; the method comprising a request for a policy check by the operating system of the computer platform to the secure operator before acting upon the data, by sending to the secure operator the name of the target data plus the intended operation the secure operator checking the restrictions associated with the target data in the access profile to determine whether the data may be operated upon; and the secure operator checking the proposed usage with the restrictions, and replying to the operating system.

In a preferred method of operation according to the invention, upon sign-on, the removable module and the PC TC mutually authenticate and the TC stores the identifier of the removable module. Before protected data can be used, the secure operator or access profile associated with the data (depending upon the particular model used) need to give permission to the OS to carry out the particular operation. Upon checking the restrictions relating to the data, the secure operator or

access profile is operable to perform the restrictions check with reference to the user identity. If the licence is stored in the smart card, the secure operator needs to retrieve the licence into a store held on the TC PC that it can consult in future, or else consult the smart card each time to find out the details of the licence. This user licence may be updated as a result of the data access: for instance, if an operation permission is qualified by being for a fixed number of uses.

The developer can issue (user) licences on smart cards, which would then be sent out to end-users after registration, or the licence can be downloaded electronically either to the smart card or to the PC TC. Data can be downloaded at the same time, or transferred separately, possibly by non-electronic means such as CD-ROM.

Description of the Figures

Figure 1 is a diagram that illustrates a system capable of implementing embodiments of the present invention;

Figure 2 is a diagram which illustrates a motherboard including a trusted device arranged to communicate with a smart card via a smart card reader and with a group of modules;

Figure 3 is a diagram that illustrates the trusted device in more detail;

Figure 4 is a flow diagram which illustrates the steps involved in acquiring an integrity metric of the computing apparatus;

Figure 5 is a flow diagram which illustrates the steps involved in establishing communications between a trusted computing platform and a remote platform including the trusted platform verifying its integrity;

Figure 6 is a diagram that illustrates the operational parts of a user smart card for use in accordance with embodiments of the present invention;

Figure 7 is a flow diagram which illustrates the process of mutually authenticating a smart card and a host platform;

Figure 8 is a schematic block diagram of a trusted module in the system of Figure 15;

Figures 9 to 12 show parts of the system of Figure 15 to illustrate various communication methods employed therein;

Figure 13 illustrates the format of a protocol data unit used in the system of Figure 15;

Figure 14 shows a modification to the system of Figure 15, which will be used to describe a specific embodiment of the present invention;

Figure 15 is a schematic block diagram of a host computer system which is the subject of another patent application (International Patent Application No. PCT/GB00/00504, filed on 15 February 2000);

Figure 16 is a diagram of the logical components of a trusted module in the system of Figure 14;

Figure 17 illustrates the structure of protected software or data in the system of Figure 14;

Figure 18 is a flow chart illustrating installing or upgrading of software or other data on the system of Figure 14;

Figure 19 is a diagram illustrating the relationship between a portable trusted device and a trusted platform in a system according to embodiments of the invention;

Figure 20 is a flow chart illustrating the use of protected data or software in the system of Figure 14 so as to enforce licensing restrictions;

Figure 21 is a flow chart illustrating installation and use of software or other data on the system of Figure 14 in a further embodiment of the invention.

Specific Embodiments of the Invention

Preferred embodiments of the invention will now be described, by way of example.

Before describing the embodiment of the present invention, a computing platform incorporating a trusted device (as described in International Patent Application No. PCT/GB00/00528) and suitable for use in embodiments of the invention will be described with reference to Figures 1 to 7. Also described as suitable for use in embodiments of the invention is a trusted token device personal to a user of the computer platform - in preferred examples, this token device is a smart card.

What is described is the incorporation into a computing platform of a physical trusted device or module whose function is to bind the identity of the platform to reliably measured data that provides an integrity metric of the platform, thereby forming a "trusted platform". The identity and the integrity metric are compared with expected values provided by a trusted party (TP) that is prepared to vouch for the trustworthiness of the platform. If there is a match, the implication is that at least part of the platform is operating correctly, depending on the scope of the integrity metric.

In this specification, the term "trusted" when used in relation to a physical or logical component, is used to mean that the physical or logical component always behaves in an expected manner. The behavior of that component is predictable and known. Trusted components have a high degree of resistance to unauthorized modification.

In this specification, the term "computing platform" (or "computer platform") is used to refer to at least one data processor and at least one data storage means, usually but not essentially with associated communications facilities e.g. a plurality of drivers, associated applications and data files, and which may be capable of interacting with external entities e.g. a user or another computer platform, for example by means of connection to the internet, connection to an external network, or by having an input port capable of receiving data stored on a data storage medium, e.g. a CD ROM, floppy disk, ribbon tape or the like.

A user verifies the correct operation of the platform before exchanging other data with the platform. A user does this by requesting the trusted device to provide its identity and an integrity metric. (Optionally the trusted device will refuse to provide evidence of identity if it itself was unable to verify correct operation of the platform.) The user receives the proof of identity and the integrity metric, and compares them against values which it believes to be true. Those proper values are provided by the TP or another entity that is trusted by the user. If data reported by the trusted device is the same as that provided by the TP, the user trusts the platform. This is because the user trusts the entity. The entity trusts the platform because it has previously validated the identity and determined the proper integrity metric of the platform.

A user of a computing entity may, for example, establish a level of trust with the computer entity by use of such a trusted token device. The trusted token device is a personal and portable device having a data processing capability and in which the user has a high level of confidence. It may also be used by the trusted platform to identify the user. The trusted token device may perform the functions of:

- verifying a correct operation of a computing platform in a manner which is readily apparent to the user, for example by audio or visual display;
- challenging a monitoring component to provide evidence of a correct operation of a computer platform with which the monitoring component is associated; and
- establishing a level of interaction of the token device with a computing platform, depending on whether a monitoring component has provided satisfactory evidence of a correct operation of the computing entity, and withholding specific interactions with the computer entity if such evidence of correct operation is not received by the token device.

Once a user has established trusted operation of the platform, he exchanges other data with the platform. For a local user, the exchange might be by interacting with some software application running on the platform. For a remote user, the exchange might involve a secure transaction. In

either case, the data exchanged is 'signed' by the trusted device. The user can then have greater confidence that data is being exchanged with a platform whose behaviour can be trusted.

The trusted device uses cryptographic processes but does not necessarily provide an external interface to those cryptographic processes. Also, a most desirable implementation would be to make the trusted device tamperproof, to protect secrets by making them inaccessible to other platform functions and provide an environment that is substantially immune to unauthorised modification. Since tamper-proofing is impossible, the best approximation is a trusted device that is tamper-resistant, or tamper-detecting. The trusted device, therefore, preferably consists of one physical component that is tamper-resistant.

Techniques relevant to tamper-resistance are well known to those skilled in the art of security. These techniques include methods for resisting tampering (such as appropriate encapsulation of the trusted device), methods for detecting tampering (such as detection of out of specification voltages, X-rays, or loss of physical integrity in the trusted device casing), and methods for eliminating data when tampering is detected. Further discussion of appropriate techniques can be found at http://www.cl.cam.ac.uk/~mgk25/tamper.html. It will be appreciated that, although tamper-proofing is a most desirable feature of the present invention, it does not enter into the normal operation of the invention and, as such, is beyond the scope of the present invention and will not be described in any detail herein.

The trusted device is preferably a physical one because it must be difficult to forge. It is most preferably tamper-resistant because it must be hard to counterfeit. It typically has an engine capable of using cryptographic processes because it is required to prove identity, both locally and at a distance, and it contains at least one method of measuring some integrity metric of the platform with which it is associated.

A trusted platform 10 is illustrated in the diagram in Figure 1. The platform 10 includes the standard features of a keyboard 14 (which provides a user's confirmation key), mouse 16 and monitor 18, which provide the physical 'user interface' of the platform. This embodiment of a trusted platform also contains a smart card reader 12. Along side the smart card reader 12, there is illustrated a smart card 19 to allow trusted user interaction with the trusted platform as shall be described further below. In the platform 10, there are a plurality of modules 15: these are other functional elements of the trusted platform of essentially any kind appropriate to that platform.. The functional significance of such elements is not relevant to the present invention and will not be discussed further herein. Additional components of the trusted computer entity will typically include one or more local area network (LAN) ports, one or more modem ports, and one or more power supplies, cooling fans and the like.

As illustrated in Figure 2, the motherboard 20 of the trusted computing platform 10 includes (among other standard components) a main processor 21, main memory 22, a trusted device 24, a data bus 26 and respective control lines 27 and lines 28, BIOS memory 29 containing the BIOS program for the platform 10 and an Input/Output (IO) device 23, which controls interaction between the components of the motherboard and the smart card reader 12, the keyboard 14, the mouse 16 and the monitor 18 (and any additional peripheral devices such as a modem, printer, scanner or the like). The main memory 22 is typically random access memory (RAM). In operation, the platform 10 loads the operating system, for example Windows NT™, into RAM from hard disk (not shown). Additionally, in operation, the platform 10 loads the processes or applications that may be executed by the platform 10 into RAM from hard disk (not shown).

The computer entity can be considered to have a logical, as well as a physical, architecture. The logical architecture has a same basic division between the computer platform, and the trusted component, as is present with the physical architecture described in Figs. 1 to 4 herein. That is to say, the trusted component is logically distinct from the computer platform to which it is physically related. The computer entity comprises a user space being a logical space which is physically resident on the computer platform (the first processor and first data storage means) and a trusted component space being a logical space which is physically resident on the trusted component. In the user space are one or a plurality of drivers, one or a plurality of applications programs, a file storage area; smart card reader; smart card interface; and a software agent which can perform operations in the user space and report back to trusted component. The trusted component space is a logical area based upon and physically resident in the trusted component, supported by the second data processor and second memory area of the trusted component. Monitor 18 receives images directly from the trusted component space. External to the computer entity are external communications networks e.g. the Internet, and various local area networks, wide area networks which are connected to the user space via the drivers (which may include one or more modem ports). An external user smart card inputs into smart card reader in the user space.

Typically, in a personal computer the BIOS program is located in a special reserved memory area, the upper 64K of the first megabyte do the system memory (addresses FØØØh to FFFFh), and the main processor is arranged to look at this memory location first, in accordance with an industry wide standard.

The significant difference between the platform and a conventional platform is that, after reset, the main processor is initially controlled by the trusted device, which then hands control over to the platform-specific BIOS program, which in turn initialises all input/output devices as normal. After the BIOS program has executed, control is handed over as normal by the BIOS program to an operating system program, such as Windows NT (TM), which is typically loaded into main memory 22 from a hard disk drive (not shown).

Clearly, this change from the normal procedure requires a modification to the implementation of the industry standard, whereby the main processor 21 is directed to address the trusted device 24 to receive its first instructions. This change may be made simply by hard-coding a different address into the main processor 21. Alternatively, the trusted device 24 may be assigned the standard BIOS program address, in which case there is no need to modify the main processor configuration.

It is highly desirable for the BIOS boot block to be contained within the trusted device 24. This prevents subversion of the obtaining of the integrity metric (which could otherwise occur if rogue software processes are present) and prevents rogue software processes creating a situation in which the BIOS (even if correct) fails to build the proper environment for the operating system.

Although, in the preferred embodiment to be described, the trusted device 24 is a single, discrete component, it is envisaged that the functions of the trusted device 24 may alternatively be split into multiple devices on the motherboard, or even integrated into one or more of the existing standard devices of the platform. For example, it is feasible to integrate one or more of the functions of the trusted device into the main processor itself, provided that the functions and their communications cannot be subverted. This, however, would probably require separate leads on the processor for sole use by the trusted functions. Additionally or alternatively, although in the present embodiment the trusted device is a hardware device that is adapted for integration into the motherboard 20, it is anticipated that a trusted device may be implemented as a 'removable' device, such as a dongle, which could be attached to a platform when required. Whether the trusted device is integrated or removable is a matter of design choice. However, where the trusted device is separable, a mechanism for providing a logical binding between the trusted device and the platform should be present.

The trusted device 24 comprises a number of blocks, as illustrated in Figure 3. After system reset, the trusted device 24 performs a secure boot process to ensure that the operating system of the platform 10 (including the system clock and the display on the monitor) is running properly and in a

secure manner. During the secure boot process, the trusted device 24 acquires an integrity metric of the computing platform 10. The trusted device 24 can also perform secure data transfer and, for example, authentication between it and a smart card via encryption/decryption and signature/verification. The trusted device 24 can also securely enforce various security control policies, such as locking of the user interface.

Specifically, the trusted device comprises: a controller 30 programmed to control the overall operation of the trusted device 24, and interact with the other functions on the trusted device 24 and with the other devices on the motherboard 20; a measurement function 31 for acquiring the integrity metric from the platform 10; a cryptographic function 32 for signing, encrypting or decrypting specified data; an authentication function 33 for authenticating a smart card; and interface circuitry 34 having appropriate ports (36, 37 & 38) for connecting the trusted device 24 respectively to the data bus 26, control lines 27 and address lines 28 of the motherboard 20. Each of the blocks in the trusted device 24 has access (typically via the controller 30) to appropriate volatile memory areas 4 and/or non-volatile memory areas 3 of the trusted device 24. Additionally, the trusted device 24 is designed, in a known manner, to be tamper resistant.

For reasons of performance, the trusted device 24 may be implemented as an application specific integrated circuit (ASIC). However, for flexibility, the trusted device 24 is preferably an appropriately programmed micro-controller. Both ASICs and micro-controllers are well known in the art of microelectronics and will not be considered herein in any further detail.

One item of data stored in the non-volatile memory 3 of the trusted device 24 is a certificate 350. The certificate 350 contains at least a public key 351 of the trusted device 24 and an authenticated value 352 of the platform integrity metric measured by a trusted party (TP). The certificate 350 is signed by the TP using the TP's private key prior to it being stored in the trusted device 24. In later communications sessions, a user of the platform 10 can verify the integrity of the platform 10 by comparing the acquired integrity metric with the authentic integrity metric 352. If there is a match, the user can be confident that the platform 10 has not been subverted. Knowledge of the TP's generally-available public key enables simple verification of the certificate 350. The non-volatile memory 35 also contains an identity (ID) label 353. The ID label 353 is a conventional ID label, for example a serial number, that is unique within some context. The ID label 353 is generally used for indexing and labelling of data relevant to the trusted device 24, but is insufficient in itself to prove the identity of the platform 10 under trusted conditions.

The trusted device 24 is equipped with at least one method of reliably measuring or acquiring the integrity metric of the computing platform 10 with which it is associated. In the present embodiment, the integrity metric is acquired by the measurement function 31 by generating a digest of the BIOS instructions in the BIOS memory. Such an acquired integrity metric, if verified as described above, gives a potential user of the platform 10 a high level of confidence that the platform 10 has not been subverted at a hardware, or BIOS program, level. Other known processes, for example virus checkers, will typically be in place to check that the operating system and application program code has not been subverted.

The measurement function 31 has access to: non-volatile memory 3 for storing a hash program 354 and a private key 355 of the trusted device 24, and volatile memory 4 for storing acquired integrity metric in the form of a digest 361. In appropriate embodiments, the volatile memory 4 may also be used to store the public keys and associated ID labels 360a-360n of one or more authentic smart cards 19s that can be used to gain access to the platform 10.

In one preferred implementation, as well as the digest, the integrity metric includes a Boolean value, which is stored in volatile memory 4 by the measurement function 31, for reasons that will become apparent.

A preferred process for acquiring an integrity metric will now be described with reference to Figure 4.

In step 400, at switch-on, the measurement function 31 monitors the activity of the main processor 21 on the data, control and address lines (26, 27 & 28) to determine whether the trusted device 24 is the first memory accessed. Under conventional operation, a main processor would first be directed to the BIOS memory first in order to execute the BIOS program. However, in accordance with the present embodiment, the main processor 21 is directed to the trusted device 24, which acts as a memory. In step 405, if the trusted device 24 is the first memory accessed, in step 410, the measurement function 31 writes to non-volatile memory 3 a Boolean value, which indicates that the trusted device 24 was the first memory accessed. Otherwise, in step 415, the measurement function writes a Boolean value which indicates that the trusted device 24 was not the first memory accessed.

In the event the trusted device 24 is not the first accessed, there is of course a chance that the trusted device 24 will not be accessed at all. This would be the case, for example, if the main processor 21 were manipulated to run the BIOS program first. Under these circumstances, the platform would

operate, but would be unable to verify its integrity on demand, since the integrity metric would not be available. Further, if the trusted device 24 were accessed after the BIOS program had been accessed, the Boolean value would clearly indicate lack of integrity of the platform.

In step 420, when (or if) accessed as a memory by the main processor 21, the main processor 21 reads the stored native hash instructions 354 from the measurement function 31 in step 425. The hash instructions 354 are passed for processing by the main processor 21 over the data bus 26. In step 430, main processor 21 executes the hash instructions 354 and uses them, in step 435, to compute a digest of the BIOS memory 29, by reading the contents of the BIOS memory 29 and processing those contents according to the hash program. In step 440, the main processor 21 writes the computed digest 361 to the appropriate non-volatile memory location 4 in the trusted device 24. The measurement function 31, in step 445, then calls the BIOS program in the BIOS memory 29, and execution continues in a conventional manner.

Clearly, there are a number of different ways in which the integrity metric may be calculated, depending upon the scope of the trust required. The measurement of the BIOS program's integrity provides a fundamental check on the integrity of a platform's underlying processing environment. The integrity metric should be of such a form that it will enable reasoning about the validity of the boot process - the value of the integrity metric can be used to verify whether the platform booted using the correct BIOS. Optionally, individual functional blocks within the BIOS could have their own digest values, with an ensemble BIOS digest being a digest of these individual digests. This enables a policy to state which parts of BIOS operation are critical for an intended purpose, and which are irrelevant (in which case the individual digests must be stored in such a manner that validity of operation under the policy can be established).

Other integrity checks could involve establishing that various other devices, components or apparatus attached to the platform are present and in correct working order. In one example, the BIOS programs associated with a SCSI controller could be verified to ensure communications with peripheral equipment could be trusted. In another example, the integrity of other devices, for example memory devices or co-processors, on the platform could be verified by enacting fixed challenge/response interactions to ensure consistent results. Where the trusted device 24 is a separable component, some such form of interaction is desirable to provide an appropriate logical binding between the trusted device 24 and the platform. Also, although in the present embodiment the trusted device 24 utilises the data bus as its main means of communication with other parts of the platform, it would be feasible, although not so convenient, to provide alternative communications

paths, such as hard-wired paths or optical paths. Further, although in the present embodiment the trusted device 24 instructs the main processor 21 to calculate the integrity metric, it is anticipated that, in other embodiments, the trusted device itself is arranged to measure one or more integrity metrics.

Preferably, the BIOS boot process includes mechanisms to verify the integrity of the boot process itself. Such mechanisms are already known from, for example, Intel's draft "Wired for Management baseline specification v 2.0 - BOOT Integrity Service", and involve calculating digests of software or firmware before loading that software or firmware. Such a computed digest is compared with a value stored in a certificate provided by a trusted entity, whose public key is known to the BIOS. The software/firmware is then loaded only if the computed value matches the expected value from the certificate, and the certificate has been proven valid by use of the trusted entity's public key. Otherwise, an appropriate exception handling routine is invoked.

Optionally, after receiving the computed BIOS digest, the trusted device 24 may inspect the proper value of the BIOS digest in the certificate and not pass control to the BIOS if the computed digest does not match the proper value. Additionally, or alternatively, the trusted device 24 may inspect the Boolean value and not pass control back to the BIOS if the trusted device 24 was not the first memory accessed. In either of these cases, an appropriate exception handling routine may be invoked.

Figure 5 illustrates the flow of actions by a TP, the trusted device 24 incorporated into a platform, and a user (of a remote platform) who wants to verify the integrity of the trusted platform. It will be appreciated that substantially the same steps as are depicted in Figure 5 are involved when the user is a local user. In either case, the user would typically rely on some form of software application to enact the verification. It would be possible to run the software application on the remote platform or the trusted platform. However, there is a chance that, even on the remote platform, the software application could be subverted in some way. Therefore, it is preferred that, for a high level of integrity, the software application would reside on a smart card of the user, who would insert the smart card into an appropriate reader for the purposes of verification. Particular embodiments relate to such an arrangement.

At the first instance, a TP, which vouches for trusted platforms, will inspect the type of the platform to decide whether to vouch for it or not. This will be a matter of policy. If all is well, in step 500, the TP measures the value of integrity metric of the platform. Then, the TP generates a certificate,

in step 505, for the platform. The certificate is generated by the TP by appending the trusted device's public key, and optionally its ID label, to the measured integrity metric, and signing the string with the TP's private key.

The trusted device 24 can subsequently prove its identity by using its private key to process some input data received from the user and produce output data, such that the input/output pair is statistically impossible to produce without knowledge of the private key. Hence, knowledge of the private key forms the basis of identity in this case. Clearly, it would be feasible to use symmetric encryption to form the basis of identity. However, the disadvantage of using symmetric encryption is that the user would need to share his secret with the trusted device. Further, as a result of the need to share the secret with the user, while symmetric encryption would in principle be sufficient to prove identity to the user, it would insufficient to prove identity to a third party, who could not be entirely sure the verification originated from the trusted device or the user.

In step 510, the trusted device 24 is initialised by writing the certificate 350 into the appropriate non-volatile memory locations 3 of the trusted device 24. This is done, preferably, by secure communication with the trusted device 24 after it is installed in the motherboard 20. The method of writing the certificate to the trusted device 24 is analogous to the method used to initialise smart cards by writing private keys thereto. The secure communications is supported by a 'master key', known only to the TP, that is written to the trusted device (or smart card) during manufacture, and used to enable the writing of data to the trusted device 24; writing of data to the trusted device 24 without knowledge of the master key is not possible.

At some later point during operation of the platform, for example when it is switched on or reset, in step 515, the trusted device 24 acquires and stores the integrity metric 361 of the platform.

When a user wishes to communicate with the platform, in step 520, he creates a nonce, such as a random number, and, in step 525, challenges the trusted device 24 (the operating system of the platform, or an appropriate software application, is arranged to recognise the challenge and pass it to the trusted device 24, typically via a BIOS-type call, in an appropriate fashion). The nonce is used to protect the user from deception caused by replay of old but genuine signatures (called a 'replay attack') by untrustworthy platforms. The process of providing a nonce and verifying the response is an example of the well-known 'challenge/response' process.

In step 530, the trusted device 24 receives the challenge and creates an appropriate response. This may be a digest of the measured integrity metric and the nonce, and optionally its ID label. Then, in step 535, the trusted device 24 signs the digest, using its private key, and returns the signed digest, accompanied by the certificate 350, to the user.

In step 540, the user receives the challenge response and verifies the certificate using the well known public key of the TP. The user then, in step 550, extracts the trusted device's 24 public key from the certificate and uses it to decrypt the signed digest from the challenge response. Then, in step 560, the user verifies the nonce inside the challenge response. Next, in step 570, the user compares the computed integrity metric, which it extracts from the challenge response, with the proper platform integrity metric, which it extracts from the certificate. If any of the foregoing verification steps fails, in steps 545, 555, 565 or 575, the whole process ends in step 580 with no further communications taking place.

Assuming all is well, in steps 585 and 590, the user and the trusted platform use other protocols to set up secure communications for other data, where the data from the platform is preferably signed by the trusted device 24.

Further refinements of this verification process are possible. It is desirable that the challenger becomes aware, through the challenge, both of the value of the platform integrity metric and also of the method by which it was obtained. Both these pieces of information are desirable to allow the challenger to make a proper decision about the integrity of the platform. The challenger also has many different options available - it may accept that the integrity metric is recognised as valid in the trusted device 24, or may alternatively only accept that the platform has the relevant level of integrity if the value of the integrity metric is equal to a value held by the challenger (or may hold there to be different levels of trust in these two cases).

The techniques of signing, using certificates, and challenge/response, and using them to prove identity, are well known to those skilled in the art of security and therefore need not be described in any more detail herein.

The user's smart card 19 is a token device, separate from the computing entity, which interacts with the computing entity via the smart card reader port 19. A user may have several different smart cards issued by several different vendors or service providers, and may gain access to the internet or a plurality of network computers from any one of a plurality of computing entities as described

herein, which are provided with a trusted component and smart card reader. A user's trust in the individual computing entity to which s/he is using is derived from the interaction between the user's trusted smart card token and the trusted component of the computing entity. The user relies on their trusted smart card token to verify the trustworthiness of the trusted component.

A processing part 60 of a user smart card 19 is illustrated in Figure 6. As shown, the user smart card 19 processing part 60 has the standard features of a processor 61, memory 62 and interface contacts 63. The processor 61 is programmed for simple challenge/response operations involving authentication of the user smart card 19 and verification of the platform 10, as will be described below. The memory 62 contains its private key 620, its public key 628, (optionally) a user profile 621, the public key 622 of the TP and an identity 627. The user profile 621 lists the allowable auxiliary smart cards 20 AC1-ACn usable by the user, and the individual security policy 624 for the user. For each auxiliary smart card 20, the user profile includes respective identification information 623, the trust structure 625 between the smart cards (if one exists) and, optionally, the type or make 626 of the smart card.

In the user profile 621, each auxiliary smart card 20 entry AC1-ACn includes associated identification information 623, which varies in dependence upon the type of card. For example, identification information for a cash card typically includes a simple serial number, whereas, for a crypto card, the identification information typically comprises the public key (or certificate) of the crypto card (the private key being stored secretly on the crypto card itself).

The 'security policy' 624 dictates the permissions that the user has on the platform 10 while using an auxiliary smart card 20. For example, the user interface may be locked or unlocked while an auxiliary smart card 20 is in use, depending on the function of the auxiliary smart card 20. Additionally, or alternatively, certain files or executable programs on the platform 10 may be made accessible or not, depending on how trusted a particular auxiliary smart card 20 is. Further, the security policy 624 may specify a particular mode of operation for the auxiliary smart card 20, such as 'credit receipt' or 'temporary delegation', as will be described below.

A 'trust structure' 625 defines whether an auxiliary smart card 20 can itself 'introduce' further auxiliary smart cards 20 into the system without first re-using the user smart card 19. In the embodiments described in detail herein, the only defined trust structure is between the user smart card 19 and the auxiliary smart cards 20 that can be introduced to the platform 10 by the user smart card 19. Introduction may be 'single session' or 'multi-session', as will be described below.

However, there is no reason why certain auxiliary smart cards 20 could not in practice introduce further auxiliary smart cards 20. This would require an auxiliary smart card 20 to have an equivalent of a user profile listing the or each auxiliary smart card that it is able to introduce.

Use of auxiliary smart cards 20 is not a necessary feature of the present invention, and is not described further in the present application. Use of auxiliary smart cards is the subject of the present applicant's copending International Patent Application No. PCT/GB00/00751dated 5 March 2000 and entitled "Computing Apparatus and Methods of Operating Computing Apparatus", which is incorporated by reference herein.

A preferred process for authentication between a user smart card 19 and a platform 10 will now be described with reference to the flow diagram in Figure 7. As will be described, the process conveniently implements a challenge/response routine. There exist many available challenge/response mechanisms. The implementation of an authentication protocol used in the present embodiment is mutual (or 3-step) authentication, as described in ISO/IEC 9798-3. Of course, there is no reason why other authentication procedures cannot be used, for example 2-step or 4-step, as also described in ISO/IEC 9798-3.

Initially, the user inserts their user smart card 19 into the smart card reader 12 of the platform 10 in step 700. Beforehand, the platform 10 will typically be operating under the control of its standard operating system and executing the authentication process, which waits for a user to insert their user smart card 19. Apart from the smart card reader 12 being active in this way, the platform 10 is typically rendered inaccessible to users by 'locking' the user interface (i.e. the screen, keyboard and mouse).

When the user smart card 19 is inserted into the smart card reader 12, the trusted device 24 is triggered to attempt mutual authentication in step by generating and transmitting a nonce A to the user smart card 19 in step 705. A nonce, such as a random number, is used to protect the originator from deception caused by replay of old but genuine responses (called a 'replay attack') by untrustworthy third parties.

In response, in step 710, the user smart card 19 generates and returns a response comprising the concatenation of: the plain text of the nonce A, a new nonce B generated by the user smart card 19, the ID 353 of the trusted device 24 and some redundancy; the signature of the plain text, generated

by signing the plain text with the private key of the user smart card 19; and a certificate containing the ID and the public key of the user smart card 19.

The trusted device 24 authenticates the response by using the public key in the certificate to verify the signature of the plain text in step 715. If the response is not authentic, the process ends in step 720. If the response is authentic, in step 725 the trusted device 24 generates and sends a further response including the concatenation of: the plain text of the nonce A, the nonce B, the ID 627 of the user smart card 19 and the acquired integrity metric; the signature of the plain text, generated by signing the plain text using the private key of the trusted device 24; and the certificate comprising the public key of the trusted device 24 and the authentic integrity metric, both signed by the private key of the TP.

The user smart card 19 authenticates this response by using the public key of the TP and comparing the acquired integrity metric with the authentic integrity metric, where a match indicates successful verification, in step 730. If the further response is not authentic, the process ends in step 735.

If the procedure is successful, both the trusted device 24 has authenticated the user smart card 19 and the user smart card 19 has verified the integrity of the trusted platform 10 and, in step 740, the authentication process executes the secure process for the user. Then, the authentication process sets an interval timer in step 745. Thereafter, using appropriate operating system interrupt routines, the authentication process services the interval timer periodically to detect when the timer meets or exceeds a pre-determined timeout period in step 750.

Clearly, the authentication process and the interval timer run in parallel with the secure process. When the timeout period is met or exceeded, the authentication process triggers the trusted device 24 to re-authenticate the user smart card 19, by transmitting a challenge for the user smart card 19 to identify itself in step 760. The user smart card 19 returns a certificate including its ID 627 and its public key 628 in step 765. In step 770, if there is no response (for example, as a result of the user smart card 19 having been removed) or the certificate is no longer valid for some reason (for example, the user smart card has been replaced with a different smart card), the session is terminated by the trusted device 24 in step 775. Otherwise, in step 770, the process from step 745 repeats by resetting the interval timer.

The techniques of signing, using certificates, and challenge/response, and using them to prove identity, are well known to those skilled in the art of security and will, thus, not be described in any more detail herein.

Referring now to Figures 21 and 8 to 13, a specific embodiment of the system which is the subject of International Patent Application No. PCT/GB00/00504, filed on 15 February 2000, will now be described. This system is particularly appropriate for application of the present invention. In Figure 21, a host computer 100 has a main CPU 102, a hard disk drive 104, a PCI network interface card 106 and DRAM memory 108 with conventional ("normal") communications paths 110 (such as ISA, EISA, PCI, USB) therebetween. The network interface card 106 also has an external communication path 112 with the world outside the host computer 100.

The network interface card 106 is logically divided into "red" and "black" data zones 114,116 with an interface 118 therebetween. In the red zone 114, data is usually plain text and is sensitive and vulnerable to undetectable alteration and undesired eavesdropping. In the black data zone 116, data is protected from undetected alteration and undesired eavesdropping (preferably encrypted by standard crypto mechanisms). The interface 118 ensures that red information does not leak into the black zone 116. The interface 118 preferably uses standard crypto methods and electronic isolation techniques to separate the red and black zones 114,116. The design and construction of such red and black zones 114,116 and the interface 118 is well known to those skilled in the art of security and electronics, particularly in the military field. The normal communication path 110 and external communication path 112 connect with the black zone 116 of the network interface card 106.

The host computer 100 also includes a trusted module 120 which is connected, not only to the normal communication paths 110, but also by mutually separate additional communication paths 122 (sub-referenced 122a,122b,122c) to the CPU 102, hard disk drive 104 and the red zone 114 of the network interface card 106. By way of example, the trusted module 120 does not have such a separate additional communication path 122 with the memory 108.

The trusted module 120 can communicate with the CPU 102, hard disk drive 104 and red zone 114 of the network interface card 106 *via* the additional communication paths 122a,b,c, respectively. It can also communicate with the CPU 102, hard disk drive 104, black zone 116 of the network interface card 106 and the memory 108 *via* the normal communication paths 110. The trusted module 120 can also act as a 100VG switching centre to route certain information between the CPU 102, hard disk drive 104 and the red zone 114 of the network interface card 106, *via* the trusted module 120 and the additional communication paths 122, under control of a policy stored in the trusted module. The trusted module 120 can also generate cryptographic keys and distribute those

keys to the CPU 102, the hard disk drive 104, and the red zone 114 of the network interface card 106 *via* the additional communication paths 122a,b,c, respectively.

Figure 8 illustrates the physical architecture of the trusted module 120. A first switching engine 124 is connected separately to the additional communication paths 122a,b,c and also to an internal communication path 126 of the trusted module 120. This switching engine 124 is under control of a policy loaded into the trusted module 120. Other components of the trusted module 120 are:

- a computing engine 128 that manages the trusted module 120 and performs general purpose computing for the trusted module 120;
- volatile memory 130 that stores temporary data;
- non-volatile memory 132 that stores long term data;
- cryptographic engines 134 that perform specialist crypto functions such as encryption and key generation;
- a random number source 136 used primarily in crypto operations;
- a second switching engine 138 that connects the trusted module 120 to the normal communication paths 110; and
- tamper detection mechanisms 140,

all connected to the internal communication path 126 of the trusted module 120.

The trusted module 120 is based on a trusted device or module 24 as described in more detail above with reference to Figures 1 to 7.

With regard to crypto key generation and distribution, the trusted module 120 generates cryptographic keys, using the random number generator 136, a hash algorithm, and other algorithms, all of which are well known, *per se*, to those skilled in the art of security. The trusted module 120 distributes selected keys to the CPU 102, hard disk drive 104 and the red zone 114 of the network interface card 106 using the additional communication paths 122a,b,c, respectively, rather than the normal communications paths 110. Keys may be used for communications between the internal modules 102,104,106,120 of the platform over the normal communication paths 110. Other temporary keys may be used (by the network interface card 106 or CPU 102) for bulk encryption or decryption of external data using the SSL protocol after the trusted module 120 has completed the SSL handshaking phase that uses long term identity secrets that must not be revealed outside the trusted module 120. Other temporary keys may be used (by the hard disk drive 104 or CPU 102) for bulk encryption or decryption of data stored on the hard disk drive 104 after those temporary keys have been created or revealed inside the trusted module 120 using long term secrets that must not be revealed outside the trusted module 120.

The trusted module 120 enforces policy control over communications between modules by the selective distribution of encryption keys. The trusted module 120 enforces a policy ban on communications between given pairs of modules by refusing to issue keys that enable secure communications over the shared infrastructure 110 between those pairs of modules.

Figure 9 illustrates a process by which the trusted module 120 can perform a watchdog function and 'ping' the modules 102,104,106 connected to the additional communication paths 122. The trusted module generates a challenge 142 and sends it to the CPU 102, hard disk drive 104 and red zone 114 of the network interface card 106 using the additional communication paths 122a,b,c, respectively. Each of the CPU 102, hard disk drive 104 and network interface card 106 responds with a response 144a,b,c, respectively, on the respective additional communication path 122a,b,c to say whether the respective module is active, and preferably that the module is acting properly. The trusted module 120 notes the responses 144a,b,c and uses them as metrics in its responses to integrity challenges that are described above with reference to Figures 1 to 7.

Figure 10 illustrates the process by which incoming external secure messages are processed when the trusted module 120 is the only module in the platform with cryptographic capabilities. An external message 146 is received by the black zone 116 of the network interface card 106 using the external communication path 112. The network interface card 106 sends a protocol data unit 148 (to be described in further detail later) containing some data and a request for an authentication and integrity check to the trusted module 120 using the normal communication paths 110. The trusted module 120 performs the authentication and integrity checks using the long term keys inside the trusted module 120 that must not revealed outside the trusted module 120, and sends a protocol data unit 150 containing an 'OK' indication to the red zone 114 of the network interface card 106 using the additional communication path 122c. The network interface card 106 then sends a protocol data unit 152 containing some data and a request for decryption to the trusted module 120 using the normal communication paths 110. The trusted module 120 decrypts the data using either temporary or long term keys inside the trusted module 120, and sends a protocol data unit 154 containing the decrypted data to the CPU 102 using the additional communication path 122a. The CPU then takes appropriate action.

Figure 11 illustrates the process by which the CPU 102 requests a policy decision from the trusted module 120. This could be used, for example, when the CPU 102 must determine whether policy allows certain data to be manipulated or an application to be executed. This will be described in more later with reference to Figures 14 to 20. The CPU 102 sends a protocol data unit 156 containing a request to the trusted module 120 using the normal communication paths 110. The trusted module 120 processes the request 156 according to the policy stored inside the trusted

module 120. The trusted module 120 sends a protocol data unit 158 containing a reply to the CPU 102 using the additional communication path 122a, in order that the CPU 102 can be sure that authorisation came from the trusted module 120. If the action is authorised, the CPU 102 takes the necessary action. Otherwise, it abandons the process.

Figure 12 illustrates an example of the control of policy over protected communications between the modules 102,104,106. All of the communications in this example use the additional communication paths 122. The red zone 114 of the network interface card 106 sends a protocol data unit 160 that is destined for the hard disk drive 104 to the trusted module 120 on the additional data path 122c. In the case where the policy does not permit this, the trusted module 120 denies the request by sending a protocol data unit 162 containing a denial to the network interface card 106 on the additional data path 122c. Later, the CPU 102 requests sensitive data from the hard disk drive 104 by sending a protocol data unit 164 addressed to the hard disk drive, but sent on the additional data path 122a to the trusted module 120. The trusted module 120 checks that the policy allows this. In the case where it does, the trusted module 120 relays the protocol data unit 164 to the hard disk drive 104 on the additional data path 122b. The hard disk drive 104 provides the data and sends it in a protocol data unit 166 on the additional data path 122b back to the trusted module 120 addressed to the CPU 102. The trusted module 120 checks that the policy allows this, and, in the case where it does, relays the protocol data unit 166 to the CPU 102 on the additional data path 122a.

Figure 13 illustrates the format of the data protocol units 178 by which data is passed over the additional communication paths 122. The data protocol unit 178 has:-

- an identifier field 168 indicating the type of the protocol data unit;
- a length field 170 indicating the length of the protocol data unit;
- a source field 172 indicating the source of the protocol data unit;
- a destination field 174 indicating the destination of the protocol data unit;
- and so on, including in many cases a data field 176.

Not all fields are always necessary. For example, assuming the policy of the trusted module 120 forbids it to relay key protocol data units that that did not originate within the trusted module 120, the CPU 102, hard disk drive 104 and network interface card 106 can therefore assume that keys are always from the trusted module 120. Hence, source and destination fields are unnecessary in key protocol data units - such protocol data units are implicitly authenticated. The design and construction and use, *per se*, of protocol data units is well known to those skilled in the art of communications.

Specific embodiments of the present invention will now be described for use in a system employing trusted computing platforms and portable trusted modules (typically smart cards) as described above. Figure 14 illustrates a particularly appropriate form of trusted computing platform for the purpose, the platform being a development of the system described above with reference to Figures 15 and 7 to 13. In Figure 14, a display 121 is connected to the trusted module 120 by means of one 122d of the additional communications paths as described above. This enables the trusted module 120 to reliably write to the display, without fear of subversion from normal software, including the operating system. Also, the host computer 100 is connected to a keyboard 101 that has a built-in smart card reader 103, both of which are connected to the normal communications paths 110. A smart card which is inserted into the smart card reader 103 can be considered to be an additional trusted module and is therefore able to communicate securely with the trusted module 120.

There are several stages in which a system for restriction of access to data in accordance with the invention can be constructed: these stages may be considered as progressing from one to another. The first stage is to use generic operation protection software that performs checks upon operations applied to data and checks against unauthorised alteration and is protected against bypassing by integrity checking. Such operation protection software need not run within the trusted module itself. A preferred stage is the logical extension of such a system in which the operation protection software runs within the trusted module. A request to perform an operation upon some data will be sent to the trusted module, preferably from the access profile. The operation protection software in the trusted module will evaluate such a request and decide whether to allow this, based on the restrictions defined within the access profile. Preferably, the trusted module and an operating system of the platform have a dedicated communications path between them which is inaccessible to other parts of the computer platform (as in the Figure 14 structure). In the preferred model, the request from the secure operator to the operating system to access the data is preferably supplied via the dedicated communications path.

Architectures appropriate for operation according to the invention have now been described. Methods for implementing embodiments of the invention in such architectures will now be described below. Certain of these methods are analogous to or have features in common with methods for license checking described in applicant's copending International Patent Application of even date to the present application entitled "Computer Platforms and Their Methods of Operation", this copending International Patent Application claiming the priority of European Patent Application No. 99306415.3, filed on 13 August 1999.

The procedures by which the system operates depend very much upon the particular trusted relationships in force between the developer, client computing platform (holding a trusted component or TC) and a trusted portable module such as a smart card (hereafter termed TPM). In the most general case, the TC must be registered with the data-provider in order that the data can be sent to the TC (or analogously, to the TPM, if the data is to be sent to the TPM). The TPM must also be registered with the licence-provider (very probably the same entity as the data-provider), in order that the user ID of the TPM can be incorporated into the licence before it is issued to the TC. This would be a suitable model for circumstances such as when given users share a PC, in an office environment for example. However, in scenarios where the users of a client machine will not be known in advance, such as where machines are available in public places such as airports, this approach is not possible. Instead, the licence needs to be customised to the user ID of the TPM, and given to the end-user either by a new TPM being issued by the licence-provider, or by this information being downloaded into one already held by the end-user. The licence will contain a reference to the name and version of the software, if appropriate, and the ways in which that software can be used by the end-user. When the data is installed into such a public shared trusted terminal, optionally a different access profile can be installed that can specify default restrictions upon the data installed upon it, or overriding restrictions, or a combination of both. For example, copying of a document could be forbidden unless an end-user specifically had this permission in their personal licence (held on their TPM). After the access profile has been transferred (preferably encrypted), preferably integrity checks are carried out and a digest of the profile is stored within the local TC.

Figure 16 illustrates a logical diagram of the components of the TC 1103. These comprise operation protection software components 1211 and other operation protection data components 1210 within the trusted component 1103. The following components of the invention are operation protection code 1211 that should be run within a protected environment, as previously described, and preferably within the TC 1103 itself (though the skilled person will appreciate that an appropriate protected environment can be provided outside the TC 1103): the secure operator 1206, and the data protector 1207. Operation protection data components stored on the TC include the private key of the TC 1201, the public key certificate 1202 of a trusted entity, the developer's public key certificate 1203, a log 1204, and a hashed version 1205 of the secure operator 1206 and the data protector 1207, signed by the trusted entity. The operation of these logical components will be described further below.

Whenever data is to be installed onto the trusted platform, integrity and other checks should be carried out in order to safely download or upgrade data from a third party. Data installation will only proceed via the operating system ('OS') if such the expected integrity values match. If such checks succeed (in the sense of the data or wrapper not having been altered), the data protector will store in the TC (e.g. smart card) the digest of the data (and any access profile) which was appended to the data itself, together with a reference to the stored data. Optionally, an alternative data form used in integrity checking such as the integrity checksum of the data is stored instead in the TC (e.g. smart card).

Figure 17 illustrates the structure of protected software or data 1306 within the client computer. Digital data 1304 on the client computer is associated with a access profile 1303, within which is stored the public key of the TC 1302. This data structure 1301 is stored together with a hashed version 1305 of the data structure 1301. This hashed version 1305 is signed with the clearinghouse or developer's private key. Preferably, the hashed version 1305 is stored within the TC itself (this is carried out during the installation process by the data protector 1207).

Figure 18 illustrates the flowchart for loading or upgrading software or other data onto the client platform. The steps shown in Figure 18 apply for the general case in which data protector 1207 may not be running within the TC 1103 itself, but can readily be adapted to the (simpler) case in which the data protector is running within the TC 1103.

The data to be installed is hashed and signed with the sender's private key, and this is appended to the data itself by the sender. Prior to sending of the data, it would be normal for the sender to require an integrity check of the trusted computing platform (as described above).

When the operating system 1400 of the trusted computing platform requests that the data should be installed in step 1401, the data protector 1207 receives the request in step 1402, and checks the signature of this message in step 1402, using the public key certificate corresponding to the sender, thereby checking authentication of the sender.

If authentication fails (step 1404), the data protector 1207 sends an error message to the operating system (step 1405) and the operating system 1400 causes an appropriate message to be displayed.

If authentication succeeds (step 1407), the data protector 1207 computes the hash of the message by using the cryptographic capabilities available within the TC 1103 and compares it to the message hash that is associated with the data (step 1408). This checks for integrity of the message.

If the hashes are the same (step 1409), the data protector 1207 saves a hash 1305 of the message 1304 and the corresponding access control data 1303 within the TC (step 1411) and indicates that the operating system 1400 can install the data as normal (step 1410). The TC makes a log of the installation and adds it to the relevant log file 1204 (step 1412).

If the hashes are not the same (step 1413), this indicates that the data has been altered, and that it should not be installed. The data protector 1207 sends an error message (step 1414) to the operating system 1400, which displays an appropriate message to the user (step 1415).

An alternative possibility is for data is sent to a user's smart card or other PTM for subsequent execution on a trusted computing system. This would a smart card as shown in Figure 6 also to contain within its trusted part code similar to data protector 1207. Again, an integrity check of the PTM would typically be required before installation of data, and the smart card would need capacity not only to store the data but also to store a digest of the data and access control data preferably within its trusted part. Installation of data may otherwise be essentially as described in Figure 18. It would be desirable in such an arrangement for an equivalent to operation log 1204 to be held on the PTM, preferably in the trusted part.

Figure 19 illustrates the relationships between a PTM 1106 and a TC 1103 relevant in embodiments of the present invention to execution of restricted code. There is mutual authentication at sign-on and the TC checks the PTM's ID (preferably via a certificate of the SC's public key) - this may be as shown in Figure 7. The user then asks to access data which is restricted. Before the secure operator 1206 on the TC refuses permission to the operating system 1400 of the trusted computing platform to access this data, the TC makes a check for a relevant user licence on the PTM. The operating system 1400 should therefore have a trusted input/output process for data: the skilled person will appreciate that this can be achieved in several ways, of which a particularly advantageous one is a secure hardware communications path between the secure operator software 1206 and the operating system 1400 that is not accessible to other software - this is achievable by the communication paths present in the system of Figure 14. The relevant part of the operating system will be checked upon BIS: optionally, the system integrity check to fail if the integrity check on this part of the operating system fails.

A typical approach to user access of restricted data may be as follows. When the user wishes to access particular data, perhaps via another program, the secure operator 1206 uses the access profile associated with the data (alternatives to use of access profiles may employ use of any licence-related information stored locally in order to see whether the user ID obtained during the last sign-on allows permission to carry out the required permission, or else whether there is generic permission to do so (irrespective or identity)). Preferably, the data protector 1206 will also check the integrity of the profile, and of the data. If an effective permission is found for the PTM user ID (or a general permission exists), permission will be given to the operating system 1400 to access the data. If not, the secure operator will query the PTM 1106 to find out if a licence is stored on the PTM 1106 relating to the data in question. If not, permission will be denied to the operating system to carry out the operation. However, if a license is stored on the PTM 1106 itself, the licensing information will be retrieved by being encrypted via a shared session key and integrity checked (and possibly stored). Even if there is a license on the PTM 1106, a check may need to be made to see whether the current operation is valid. If so, permission will be given to the operating system to access the data; if not, permission will be denied. Preferably, before the operation takes place, the TC will also check that the PTM corresponding to that user ID is still inserted in the smart card reader.

If the access profile and data is altered, it may not be possible to match the data against the digest stored within the TC, as it might not be clear what is the corresponding entry. Hence, again, preferably the data protector will not allow any data to be executed if there is not a corresponding correct digest stored within it.

Specific approaches to operation restriction according to these principles will now be discussed.

Figure 20 shows a flowchart for operation restriction using a model of checking where the operating system 1400 communicates with a secure operator 1206 which is in the TC 1103, and the with an access profile (outside the TC) associated with a piece of data which specifies the operations allowed by the developer upon that data. This is appropriate where, as is preferable, all operation protection software is mounted within the TC 1103. Communications between the operating system 1400, the operation protection software 1211 and the TC 1103 need to be protected against modification or spoofing. As indicated above, one option is to make part of the operating system trusted (the part dealing with data input and output), and this part of the OS can be integrity checked as part of the BIS procedure. If this part has been modified, then the platform integrity will fail. Another option is to use a trusted communication path (such as is shown in Figure 14) from the TC to the CPU when communicating with the operating system.

This approach is effective where individual users each have their own unique PTM. However, it can also be used where a smart card or other appropriate PTM is duplicated or shared amongst members of a group.

The steps shown in Figure 20, with appropriate interactions between operating system 1400, trusted component 1103, portable trusted module 1106 (here shown as a smart card, abbreviated to SC) and the access profile 1303, are as follows:

Upon sign-on using the smart card, there is mutual authentication between the TC and the smart card (step 1601). The TC stores the (current) smart card ID, which is preferably the certificate of the smart card public key.

When the user wishes to carry out an operation on some digital data, in general the operating system 1400 sends a message to the data protector 1207 (step 1602), which then checks (step 1603) whether there is a hash or checksum corresponding to the data or to the issuer of the data and access profile stored within the TC.

If there is no such hash or checksum, the data protector 1207 relays a message to the operating system 1400 and the data is not executed.

If there is such a hash or checksum present, the secure operator 1206 issues a challenge/response to the access profile 1303 corresponding to that piece of data, by means of sending a random number (nonce), together with a reference to the data (e.g. its title), signed using the private key of the TC 1103 (step 1604). Such a challenge/response protocol is well understood within this art (and has been described, for example, with respect to Figures 5 and 7 above).

The access profile 1303 verifies and authenticates the secure operator's challenge using the public key of the TC 1103, and returns a message (step 1604). If authentication is successful, the response incorporates the nonce and reference to the data. The nonce is included to give protection against replay attacks. If authentication is not successful, or if the access profile signals an error because it does not wish the data operation to be carried out on this particular machine by any user, the secure operator relays a message to the operating system and the data is not operated upon (step 1606).

The secure operator will make the appropriate operating check dependent upon the information contained within the access profile, with reference to the smart card ID and the TC ID. If no further information is required, the secure operator allows the operating system to carry out the data access (step 1607).

If there is no access profile associated with the data, the secure operator requires a model of how to proceed. This may be to allow for license checking on the smart card. A default model previously set within it by an administrator may be brought into operation (either after or instead of license checking against the smart card). This may simply be to deny data access, or may be more sophisticated - for example, the administrator may wish to stipulate that deletion can occur by default, but that copying more than a certain number of times cannot.

If no explicit access permissions are given, or perhaps if the access profile contains a flag that licences can be checked for this type of data access, the secure operator issues a challenge/response to the smart card, by means of sending a nonce, together with a reference to the data, signed using the private key of the TC (step 1308). The smart card then verifies and authenticates the secure operator's challenge using the public key of the TC (step 1309), and returns a message (step 1610). If the smart card contains a relevant license, this message will incorporate the nonce, reference to data and user access licence information. The secure operator then checks for appropriate permission within this licence to carry out the data access operation.

If there is no valid permission within the access profile, the secure operator asks the operating system 1400 to notify the end-user appropriately and the data is not operated upon (step 1611).

If there is a valid permission resulting from license checking against the smart card, the secure operator asks the operating system to carry out the data operation (step 1612).

Where the data operation has been allowed, the TC 1103 may be adapted to take a metering record of the transaction and store it in operation log 1204.

In order to counter software piracy, by giving protection against use of copied versions of the data outside the trusted platform, there are several approaches, corresponding to techniques used within dongle technology today. First, the data itself can be transmitted and stored encrypted, with the decryption key stored in the access profile or within the licence stored on the smart card. Secondly, API calls could be inserted into the data, if the source code were available, to check for the TC ID

or the smart card ID, or a key stored within the TC or SC, before data access permission is given and/or during the data access operation. Such measures are not necessary if the only aim is to ensure that the data cannot be accessed on trusted platforms in a manner outside the licence agreements with the developer.

In a preferred mechanism for enforcing checks on permission to execute digital data, the trusted module 120 (now considering Figure 14) includes the hardware and/or stores the software used to check permission. In particular, the trusted module 120 acts as a bridge between an application and the operating system (OS) of the computer platform. The OS preferably ignores all requests to load or run applications except those from the trusted module 120, given via a communications path 122 between the trusted module 120 and the CPU 102 of the computer platform that is preferably inaccessible to ordinary applications and non-OS software. The processes operating on the host computer are as follows. First, there is an initial request to the relevant operation protection code in the trusted module 120 to execute an application or other data, usually in response to some action by the end-user. The secure operator within the trusted module 120 will carry out appropriate licence checking, as detailed above. If the result of this checking is that it is appropriate to execute the data, the secure operator will convey this information to the OS via a communications path 122 to the CPU 102, which is preferably inaccessible to ordinary applications and non-OS software. The OS then starts a process on the host to execute the application or data. An analogous process will be carried out when the data protector communicates with the OS to indicate that data installation is appropriate.

Preferably the trusted module is operable to log the request to the operating system to use the data. The security and reliability of metering of data usage is enhanced by securely logging data usage within the trusted module. Logging of data manipulation activity is carried out and recorded securely in the TC. There is the option to carry this out at a number of different stages. The most common would be at the stage at which the data was opened, copied, printed or allowed to run by the secure operator. Another common point would be at the stage at which the data protector has successfully completed its integrity checks on the data to be installed, and has successfully installed this data onto the client machine. Since the access profile, secure operator and data protector are protected by integrity checks, some protection is given against hackers trying to bypass or edit the logging process. Such logs would provide both secure auditing information and the possibility of flexible licensing and payment models. Such audit logs would form the basis for usage reports and information accessible to third parties such as the machine user's IT department or company auditors.

Advantageously, API calls may be used to the trusted module or to the operation protection code to check for information relevant to data restriction, such as the presence of a secret in the trusted module, the identity and presence of the trusted module, or the user ID associated with a portable trusted module. In addition, the trusted module can be made to execute part of the code. Strong authentication of the trusted module is possible by using the trusted module's private cryptographic key, and standard authentication protocols.

There are benefits for the developer in the use of API calls in this way (over, say, using API calls to a conventional dongle). The normal benefit of addition of API calls to the software are that the software is customised for a particular user, and hence not immediately of benefit for another authorised user, even if the executable or source code were obtained in clear. However, in the conventional arrangement, this can require substantial effort on the part of the developer. By the only difference being a different trusted module ID, with protection via integrity-checking of code, substantial protection can be gained with very little effort by the developer, as running part of the code within the trusted module itself does not require individual customisation of code.

In this case the developer can insert API calls into the software to check for the presence of a secret in the platform trusted module (e.g. the user ID in the portable trusted module). Typically, the secure operator would generally only instigate a check at runtime; further API calls within the code can be made at various stages during execution of the code if desired. This could be done in a general way for the software (i.e. each customer will receive the same version), and customised details such as the exact trusted module ID can be added later.

The role of the data protector is to ensure that the data is securely installed (as described with reference to Figure 18) and also to check the integrity of both the data and any associated access profile before relevant operations on the data. There are logical extensions to this role which provide further benefits. The integrity check of the data may be such that the data protector can prevent data without a wrapper from being executed, to give further protection against the data being executed if the wrapper is removed. Optionally, checks are also made by the data protector to ensure that multiple copies of the data are not in existence; this prevents for example unauthorised extension of usage of data protected by licensing models involving a set number of uses, or executing for a given time. If multiple copies are found, the user would be given the option to delete all copies except one, in order to allow execution of this copy.

A key component of the system is the access profile associated with the data. This specifies the data to be protected, and also the operations that the user is allowed to carry out on that particular software or data. In operation of aspects of the present invention, the access profile specifies that a user ID held in the portable trusted module be checked to allow certain (or any) operations on the data. The access profile may also allow the portable trusted module to be checked for a user license (or this may not be allowed for particular data). The access profile may be located within the platform trusted module, or elsewhere on the trusted platform, provided that the integrity of the access profile can be checked by the platform trusted component. The access profile is similar to a license or cryptographic container associated with the data.

There is a variation on this procedure in which the profile is more proactive, and the operating system 1400 contacts the access profile directly to request an operation on data, and the access profile responds with permission only in the case where the operation does not counter the profile specification. Similarly, the user licence on the smart card can initiate the appropriate checks.

In such an arrangement, it is the access profile, rather than the secure operator, which controls the operating system in respect of operations on the data. In this case it is advantageous for the access profile to be located fully or partly within the platform trusted module (preferably with a secure communication path to the operating system). Installation of data onto the computer platform and subsequent execution of data by a user having a portable trusted module will now be described with reference to Figure 22.

Upon registration and/or payment for the data, in step 2201 the clearinghouse or developer (according to the exact payment model) authorises the licence corresponding to a smart card ID (to be stored in due course on a portable trusted module) and data to be updated, according to the data purchased. (Prior to this, there will be mutual authentication (possibly off-line), and public key certificates between these bodies will have been exchanged, or else the developer will actually issue the smart card containing the portable trusted module). The clearinghouse or developer sends the data (step 2202), associated with a (customised) access profile, to the client. The access profile is customised such that the public key of the portable trusted module is inserted into the access profile (alternatively, a shared key is set up between the secure operator and the smart card portable trusted module). Both the data and the access profile are hashed and signed with the clearinghouse/developer's private key, and the public key corresponding to this is stored in the portable trusted module on the smart card. The contents of any message which is to be protected are encrypted using a randomly generated secret key (such as a DES key), and transferred together with

the symmetric key which is public key (e.g. RSA)-encrypted using the public key of the intended recipient, according to a standard protocol. If the data is transferred to the computer platform, an analogous process is carried out for transferral of the data, with the public key of the developer being sent to the trusted module of the computer platform.

The data protector checks the integrity of the data whenever this is transferred to the computer platform: upon installation (step 2203), the package is verified by hashing and comparison with the decrypted signature (using the public key in the platform trusted component), and a hash is stored in the platform trusted component. Neither the data, not the access profile, is loaded if the digital signature it bears is not that which is expected.

The preceding steps relate to installation of the data: the following steps relate to use of the system to restrict access to the data. Upon sign-on using the smart card, there is mutual authentication between the platform trusted component and the smart card portable trusted module (step 2204).

The platform trusted component receives and stores the (current) smart card user ID (step 2205).

When the user wishes to use the data, the operating system of the computer platform requires action from the access profile corresponding to that data. The access profile issues a challenge/response to the secure operator (step 2206), by means of sending a random number (nonce), together with a reference to the data.

The secure operator makes an appropriate check on the data, using the smart card ID, or else by obtaining some information stored on the smart card (step 2207). For example, the secure operator may check in the profile stored within the platform trusted component whether the data is licensed to be used according to the user ID of the smart card which has been inserted, or may check whether the data is licensed to be used on the trusted platform itself (regardless of user) according to a profile stored within the platform trusted component, or may consult the smart card to obtain further details of any licence stored therein associated with the data to be accessed. In this case, the secure operator issues a challenge/response to the smart card, by means of sending a nonce, together with a reference to the data, signed using the private key of the platform trusted component. The smart card then verifies and authenticates the secure operator's challenge using the public key of the platform trusted component, and returns a message incorporating the nonce, reference to data and user access licence information. The secure operator then checks for appropriate permission within this licence to carry out the data access operation.

If there is no valid licence, the secure operator returns an error message (step 2208), from which the access profile can determine the exact type of problem with licensing and notify the operating system appropriately. If there is a valid licence, the secure operator returns a message incorporating the nonce and data reference, signed and encrypted using the computer platform trusted component private key.

The access profile verifies if the secure operator's reply is correct using the public key of the computer platform trusted component (step 2209), and either passes the call to the operating system to execute the data or sends an error message to the operating system as appropriate.

The access to the data is logged (step 2210). The log is preferably held within the computer platform trusted component, but could in addition or instead be held in the smart card, and is updated appropriately.

The skilled man will readily appreciate that many variations may be made to the embodiments described above without departing from the scope of the invention as claimed.

Claims

1. A computer system adapted to restrict operations on data, comprising:

a computer platform having a secure operator for checking whether a user of the platform is licensed to perform a requested operation on the data and for enabling use of the data;

a portable trusted module containing a user identity, wherein a trusted module is a component adapted to behave in an expected manner and resistant to unauthorised external modification;

and an access profile specifying license permissions of users with respect to the data;

wherein the secure operator is adapted to check the access profile to determine whether a requested operation is licensed for the user identity contained in the portable trusted module and prevent the requested operation if a license is required and not present.

2. A computer system as claimed in claim 1, wherein the computer platform further comprises a platform trusted module, and wherein the platform trusted module and the portable trusted module are adapted for mutual authentication.

3. A computer system as claimed in claim 2, wherein some or all of the functionality of the secure operator is within the platform trusted module.

4. A computer system as claimed in any preceding claim, wherein the access profile is within the computer platform.

5. A computer system as claimed in any preceding claim, wherein some or all of the data is within the computer platform, and the computer platform further comprises a data protector for checking data integrity before a processor of the computer platform carries out operations on the data.

6. A computer system as claimed in any of claims 1 to 4, wherein some or all of the data is within the portable trusted module or in a device containing the portable trusted module, and the portable trusted module or the device containing the portable trusted module further comprises a

data protector for checking data integrity before a processor of the computer platform carries out operations on the data.

7. A computer system as claimed in claim 5 or claim 6, wherein the data protector is within the relevant trusted module.

8. A computer system as claimed in any of claims 5 to 7, wherein the data protector is adapted to check installation of data and to load a digest of protected data and/or any associated access profile into the relevant trusted component.

9. A computer system as claimed in any preceding claim, wherein the trusted platform is adapted at boot to check the integrity of operation protection code comprising the secure operator and, if present, the data protector.

10. A computer system as claimed in claim 9 where dependent on claim 2, wherein the computer platform is adapted to perform the integrity check by reading and hashing the operation protection code to produce a first hash, reading and decrypting a stored signed version of a secure operation protection code hash using a public key certificate of a third party stored in the platform trusted module to produce a second hash, and comparing the first hash and the second hash.

11. A computer system as claimed in any preceding claim, wherein the portable trusted module contains a user access license specifying access rights to the data associated with the removable trusted module, whereby unless prevented by the access profile, the secure operator is adapted to check the user access license to determine whether a requested operation is licensed for the user identity contained in the portable trusted module.

12. A computer system as claimed in any preceding claim where dependent on claim 2, wherein the computer platform comprises a secure communication path between the platform trusted module and the operating system of the computer platform.

13. A computer system as claimed in any preceding claim, wherein the computer platform is adapted such that:

the operating system requests a policy check from the secure operator before acting upon the data, by sending the name of the target data plus the intended operation

the secure operator checks the restrictions associated with the target data in the access profile, to determine whether the data may be operated upon; and

the secure operator checks the proposed usage with the restrictions, and replies to the operating system

14. A computer system as claimed in claim 13 where dependent on claim 3, wherein on request by the operating system for permission to operate on the data, the secure operator sends a message to the access profile signed with a private key of the platform trusted module, wherein the access profile has access to the public key of the platform trusted module and can verify and authenticate the signed message with said public key, whereby if satisfied the access profile sends access profile data to the secure operator, whereupon the secure operator tests the access profile data and if appropriate requests the operating system to carry out the operation requested.

15. A computer system as claimed in claim 13 or claim 14 where dependent on claim 2 and any of claims 5 to 7, wherein the relevant trusted component contains a secure result of a one-way function on the data and associated access profile, and the data protector prevents the operation from being carried out if calculation of the one-way function provides a result different from the secure result.

16. A computer system as claimed in any preceding claim where dependent on claim 2, wherein the platform trusted component is adapted to log requests to the operating system to perform particular operations on the data.

17. A computer system as claimed in claim 6, wherein the portable trusted component is adapted to log requests to the operating system to perform particular operations on the data.

18. A computer system adapted to restrict operations on data, comprising:

a computer platform having an access profile for specifying license permissions of users with respect to the data and for enabling use of the data;

a portable trusted module containing a user identity, wherein a trusted module is a component adapted to behave in an expected manner and resistant to unauthorised external modification;

wherein the access profile is adapted to determine whether a requested operation is licensed for the user identity contained in the portable trusted module and prevent the requested operation if a license is required and not present.

19. A computer system as claimed in claim 18, wherein the operating system of the computer platform is adapted to request a policy check from the access profile before carrying out certain operations on the data, whereupon the access profile checks restrictions applying to the data to determine whether the data may be operated on, and replies to the operating system accordingly.

20. A method of restricting operations on data in a system comprising:

a computer platform having a secure operator for checking whether a user of the platform is licensed to perform a requested operation on the data and for enabling use of the data;

a portable trusted module containing a user identity, wherein a trusted module is a component adapted to behave in an expected manner and resistant to unauthorised external modification;

and an access profile specifying license permissions of users with respect to the data;

the method comprising a request for a policy check by the operating system of the computer platform to the secure operator before acting upon the data, by sending to the secure operator the name of the target data plus the intended operation

the secure operator checking the restrictions associated with the target data in the access profile to determine whether the data may be operated upon; and

the secure operator checking the proposed usage with the restrictions, and replying to the operating system.

21. A method as claimed in claim 20, wherein the computer platform further comprises a platform trusted module, and wherein some or all of the functionality of the secure operator is within the platform trusted module, and whereby on request by the operating system for permission to operate on the data, the secure operator sends a message to the access profile signed with a private key of the platform trusted module, wherein the access profile has access to the public key of the platform trusted module and can verify and authenticate the signed message with said public key,

whereby if satisfied the access profile sends access profile data to the secure operator, whereupon the secure operator tests the access profile data and if appropriate requests the operating system to carry out the operation requested.

22. A method as claimed in claim 21, wherein the the computer platform further comprises a data protector for checking data integrity before a processor of the computer platform carries out operations on the data, and wherein wherein the platform trusted component contains a secure result of a one-way function on the data and associated access profile, and the data protector prevents the operation from being carried out if calculation of the one-way function provides a result different from the secure result.

23. A method as claimed in claim 21, wherein before execution of the data, the data protector checks that there are not multiple copies of the data stored within the computer platform and prevents data execution if there are multiple copies.

24. A method as claimed in claim 21, wherein the computer platform comprises a secure communication path between the platform trusted component and the operating system, and whereby the request from the secure operator to the operating system to use the data is provided on the secure communication path.

25. A method as claimed in claim 21, wherein the platform trusted module is adapted to log any request to the operating system to perform a particular operation on the data.

26. A method of installing data on to a computer platform for restricted use thereon, the computer platform comprising: a computer platform having a secure operator for checking whether a user of the platform is licensed to perform a requested operation on the data and for enabling use of the data, a platform trusted module wherein a trusted module is a component adapted to behave in an expected manner and resistant to unauthorised external modification, and a data protector for checking data integrity before a processor of the computer platform carries out operations on the data; the method comprising verification of the reliability of the data before installation of the data and an associated access profile, and loading of a digest of protected data and an associated access profile into the platform trusted module, whereby the digest is used by the data protector and/or secure operator before execution of the data.

27. A computer platform having a secure operator for checking whether a user of the platform is licensed to perform a requested operation on the data and for enabling use of the data and an access profile specifying license permissions of users with respect to the data; wherein the secure operator is adapted to check the access profile to determine whether a requested operation is licensed for a user identity contained in a portable trusted module in communication with the computer platform, wherein a trusted module is a component adapted to behave in an expected manner and resistant to unauthorised external modification, and prevent the requested operation if a license is required and not present.

28. A computer platform as claimed in claim 27, further comprising a platform trusted module, and wherein the platform trusted module and the portable trusted module are adapted for mutual authentication.

29. A computer platform as claimed in claim 28, wherein some or all of the functionality of the secure operator is within the platform trusted module.

30. A computer platform as claimed in any of claims 27 to 29, wherein the access profile is within the platform trusted module.

31. A computer platform as claimed in any of claims 27 to 30, wherein some or all of the data is within the computer platform, and the computer platform further comprises a data protector for checking data integrity before a processor of the computer platform carries out operations on the data.

32. A computer platform as claimed in claim 31, wherein the data protector is within the platform trusted module.

33. A computer platform as claimed in claim 31 or claim 32, wherein the data protector is adapted to check installation of data and to load a digest of protected data and/or any associated access profile into the platform trusted component.

34. A computer platform as claimed in any of claims 27 to 33, wherein the computer platform is adapted at boot to check the integrity of operation protection code comprising the secure operator and, if present, the data protector.

35. A computer platform as claimed in claim 28, further comprising a secure communication path between the platform trusted module and the operating system of the computer platform.

36. A computer platform as claimed in any of claims 27 to 35, adapted such that:

the operating system requests a policy check from the secure operator before acting upon the data, by sending the name of the data plus the intended operation

the secure operator checks the restrictions associated with the target data in the access profile, to determine whether the data may be operated upon; and

the secure operator checks the proposed usage with the restrictions, and replies to the operating system

37. A computer platform as claimed in claim 36 where dependent on claim 28, wherein on request by the operating system for permission to operate on the data, the secure operator sends a message to the access profile signed with a private key of the platform trusted module, wherein the access profile has access to the public key of the platform trusted module and can verify and authenticate the signed message with said public key, whereby if satisfied the access profile sends access profile data to the secure operator, whereupon the secure operator tests the access profile data and if appropriate requests the operating system to carry out the operation requested.

38. A computer platform as claimed in claim 31 where dependent on claim 28, wherein the platform trusted component contains a secure result of a one-way function on the data and associated access profile, and the data protector prevents the operation from being carried out if calculation of the one-way function provides a result different from the secure result.

39. A portable trusted module containing a user identity, wherein a trusted module is a component adapted to behave in an expected manner and resistant to unauthorised external modification; the portable trusted module containing a user access license specifying access rights to data associated with the removable trusted module.

40. A portable trusted module as claimed in claim 39 and located within a smart card.

41. A method of restricting operations on data in a system comprising:

a computer platform having an access profile specifying license permissions of users with respect to the data; and for enabling use of the data;

a portable trusted module containing a user identity, wherein a trusted module is a component adapted to behave in an expected manner and resistant to unauthorised external modification;

the method comprising a request for a policy check by the operating system of the computer platform to the access profile before acting upon the data, by sending to the access profile the name of the target data plus the intended operation

the access profile checking the restrictions associated with the target data to determine whether the data may be operated upon; and replying to the operating system.

42. A method as claimed in claim 41, wherein the computer platform further comprises a platform trusted module, and wherein some or all of the functionality of the access profile is within the platform trusted module.

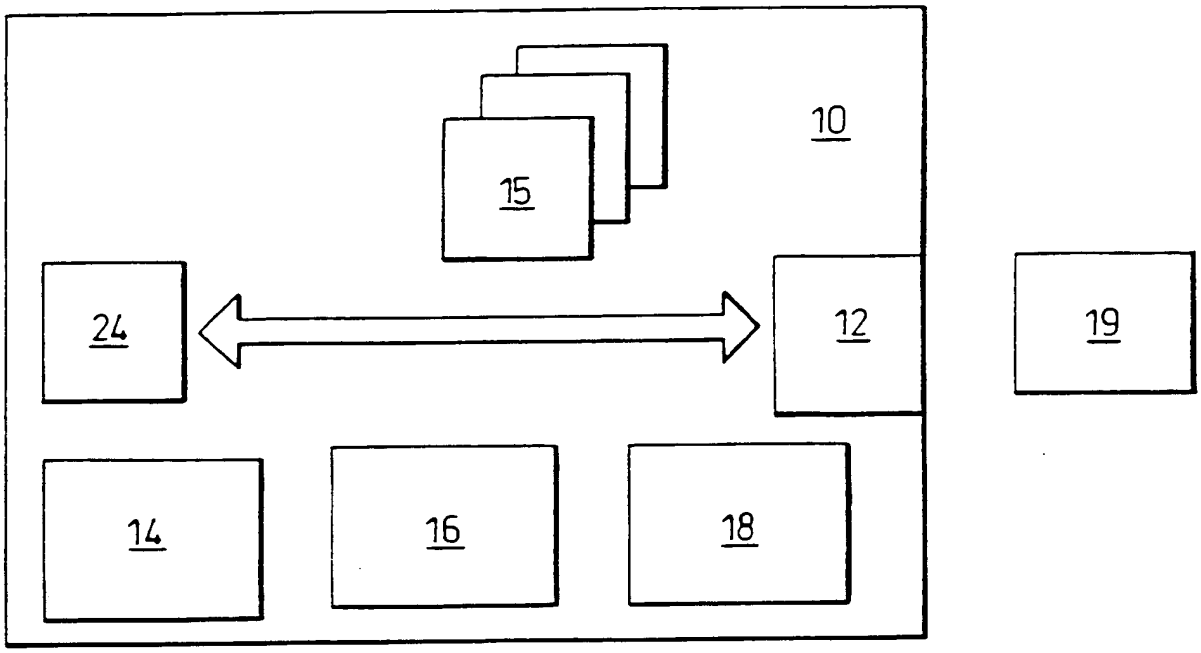


Fig. 1

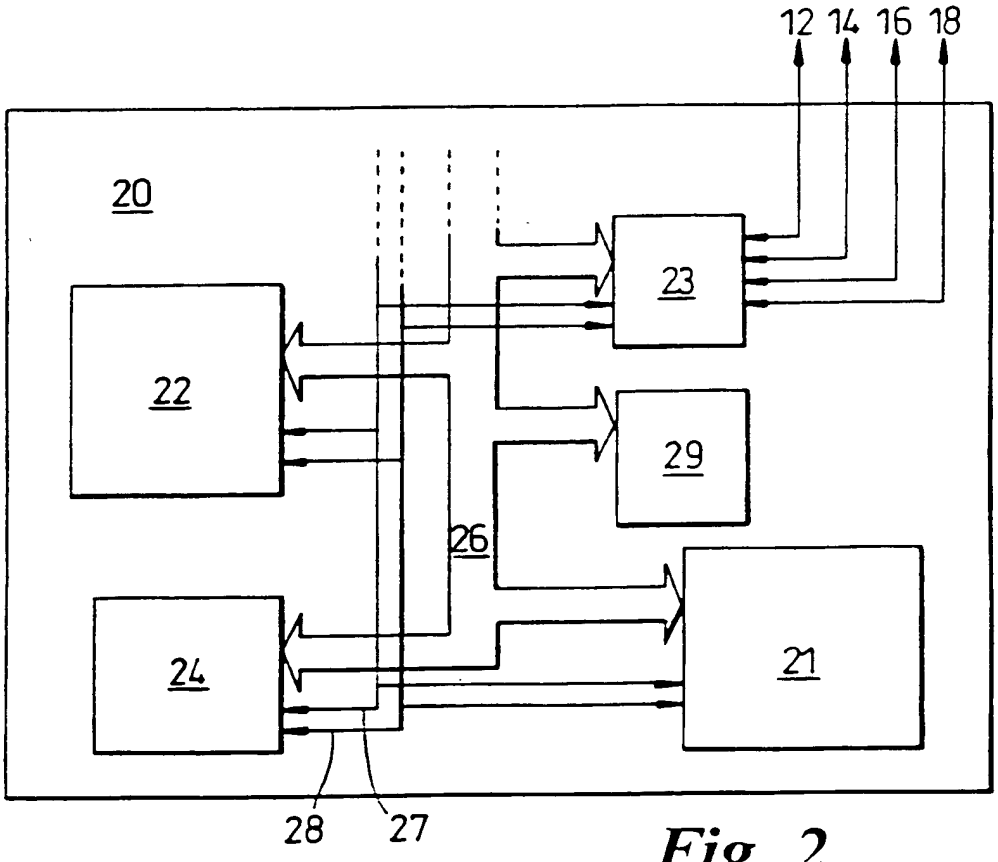


Fig. 2

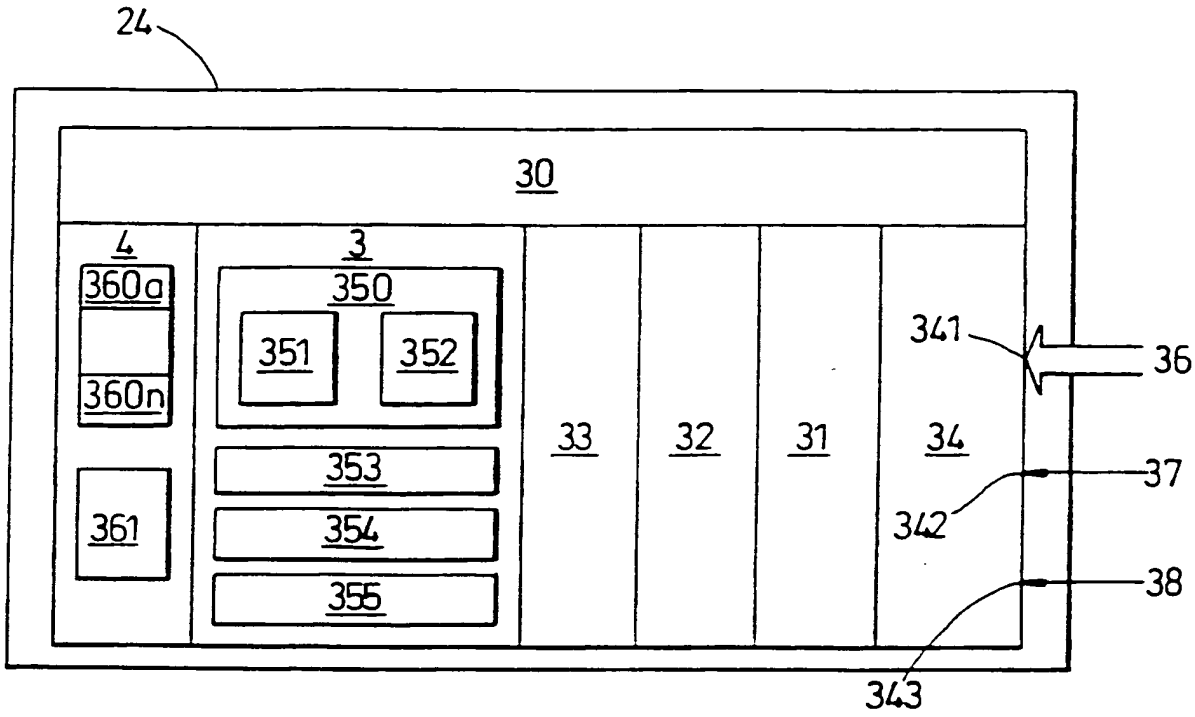


Fig. 3

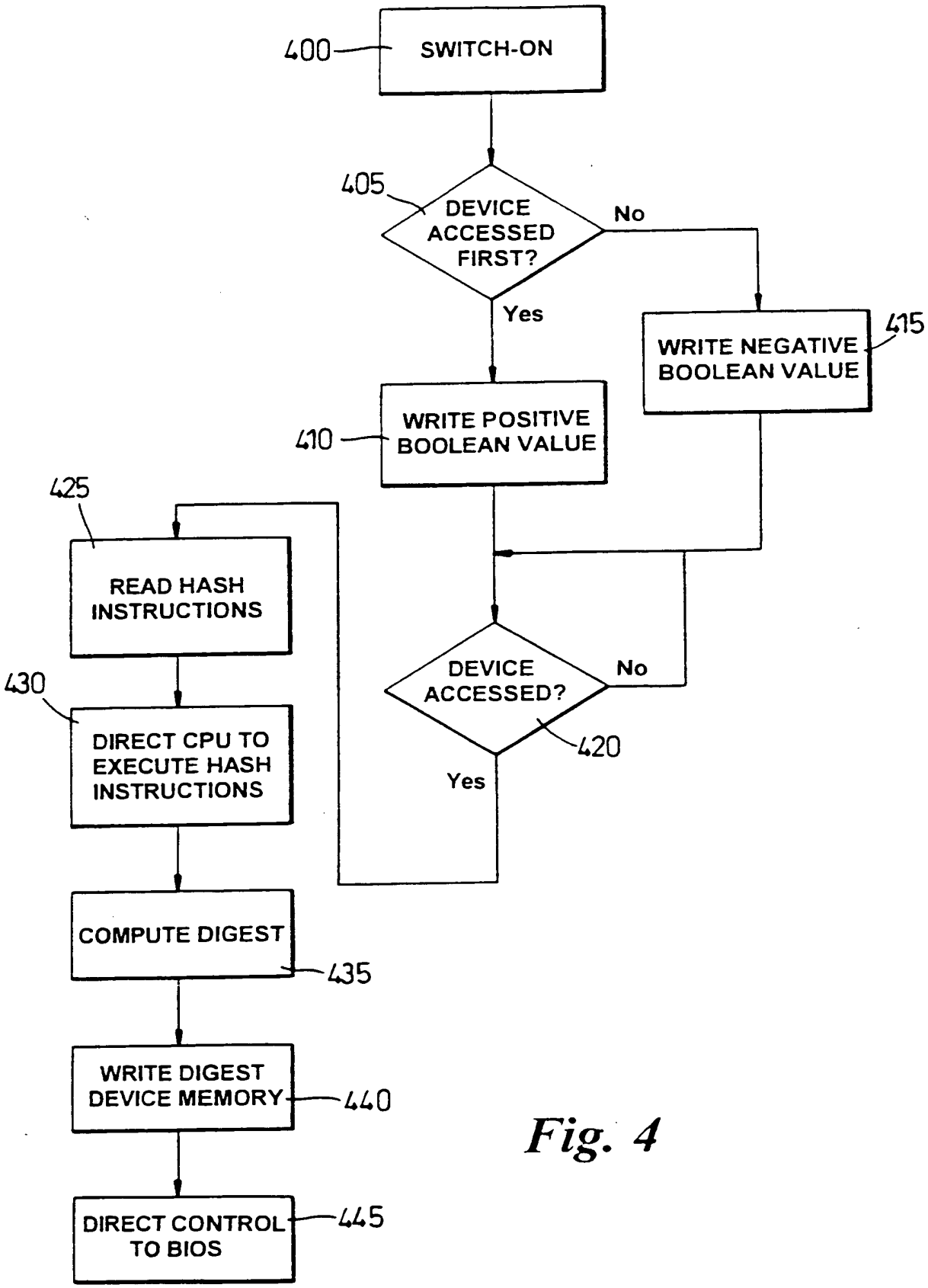


Fig. 4

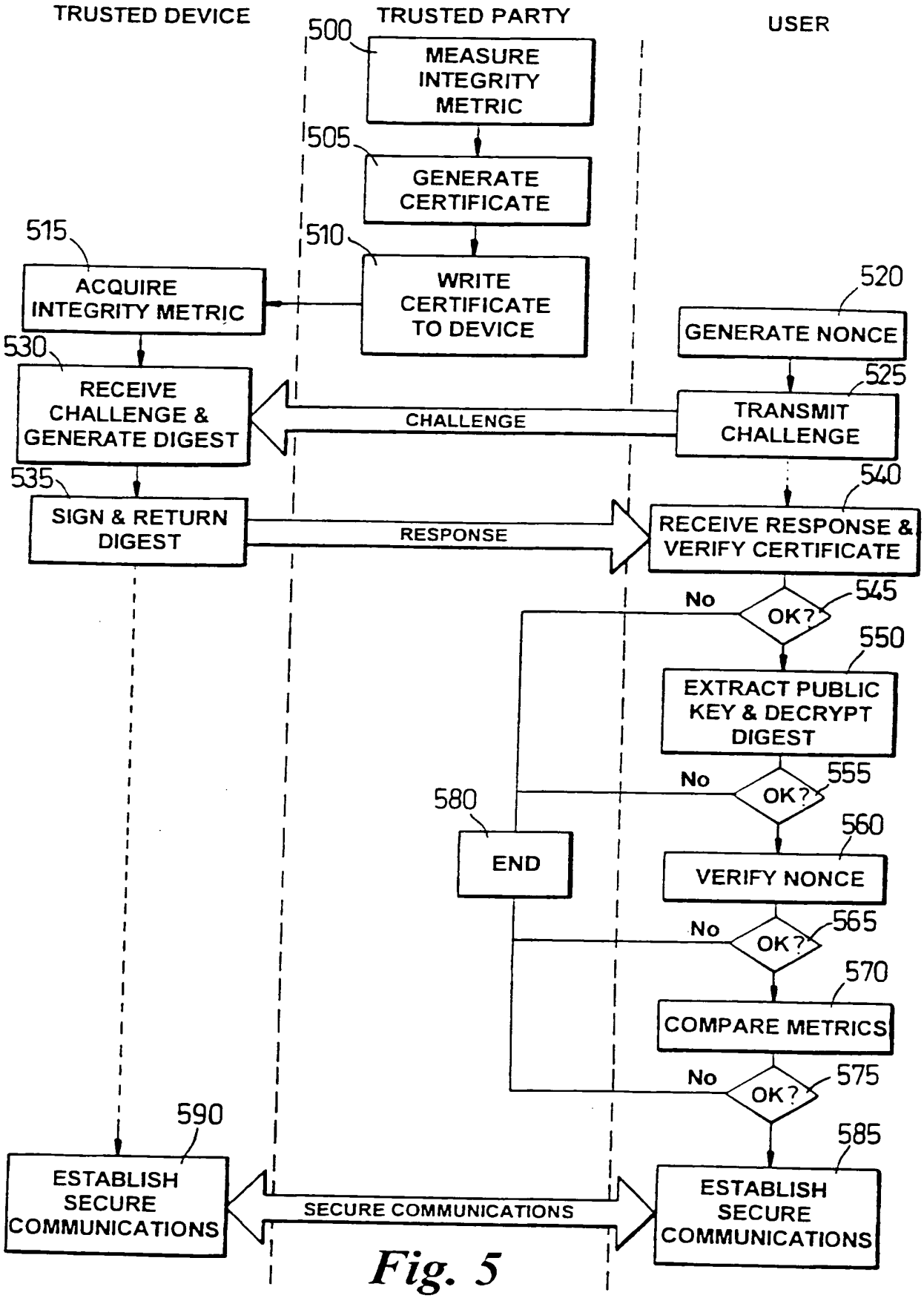


Fig. 5

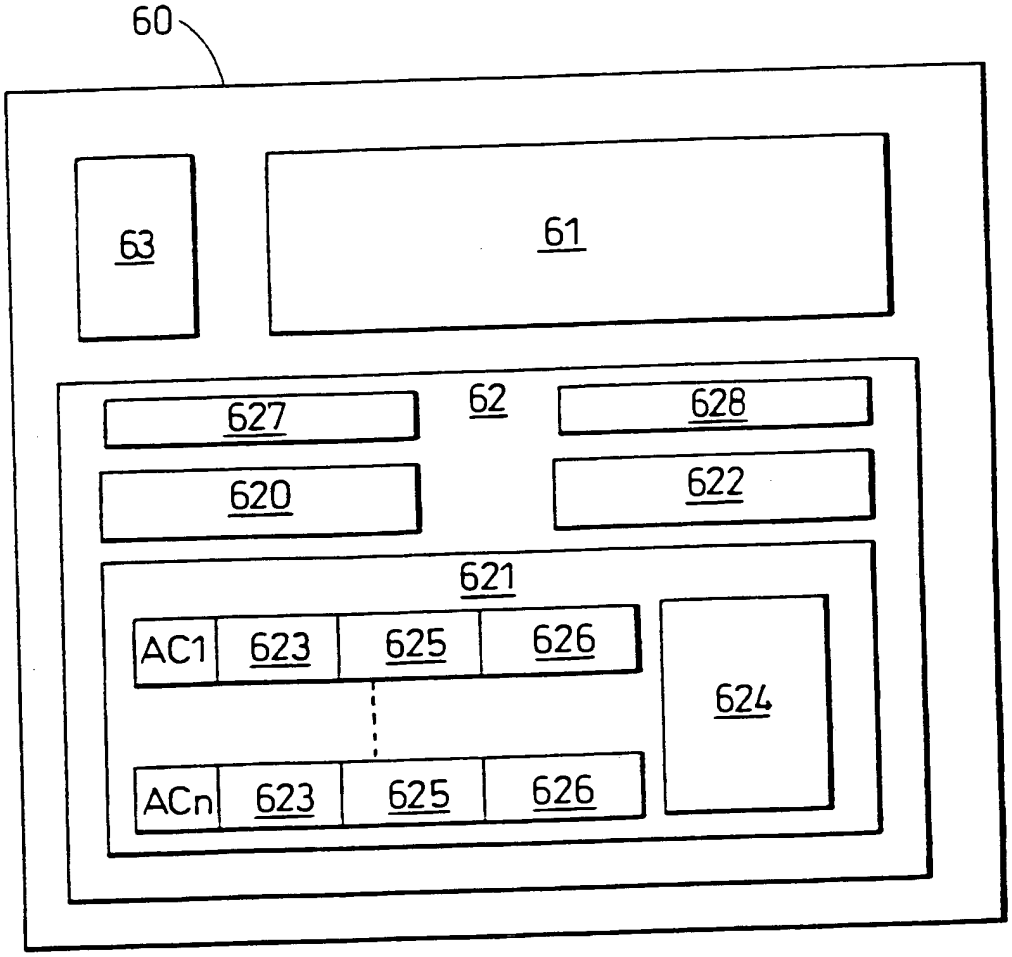


Fig. 6

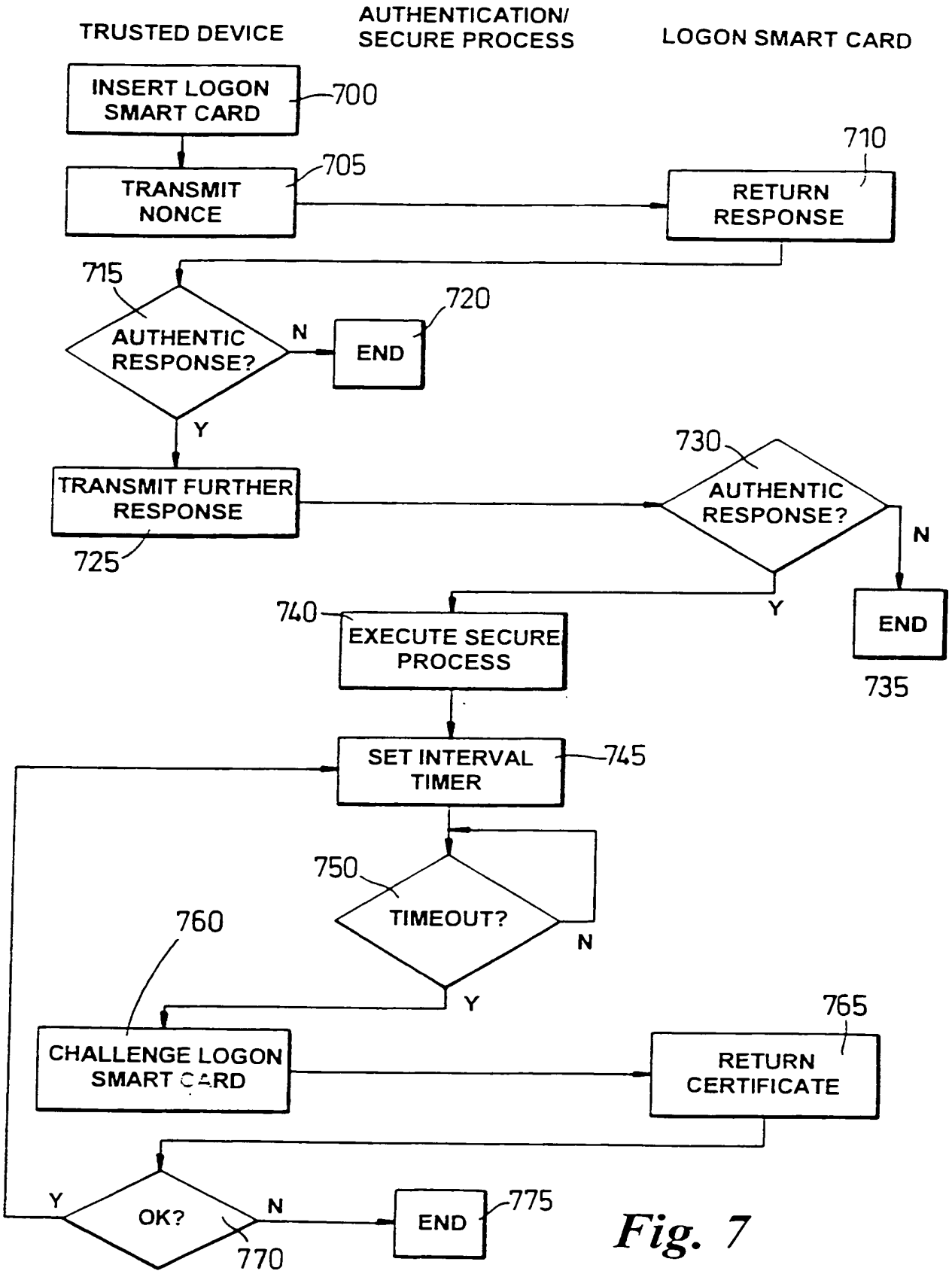


Fig. 7

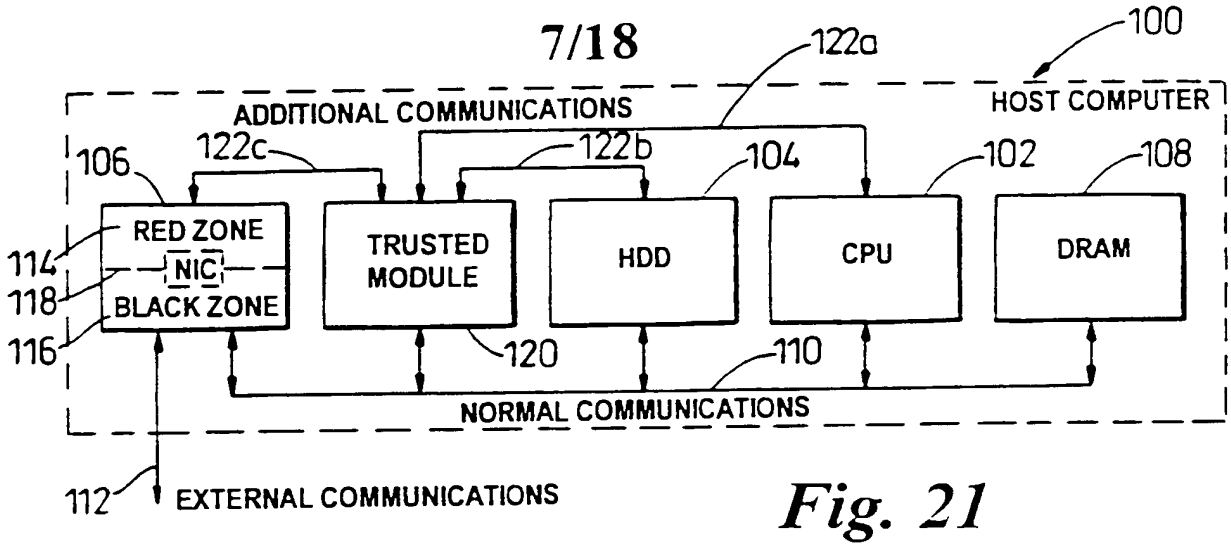


Fig. 21

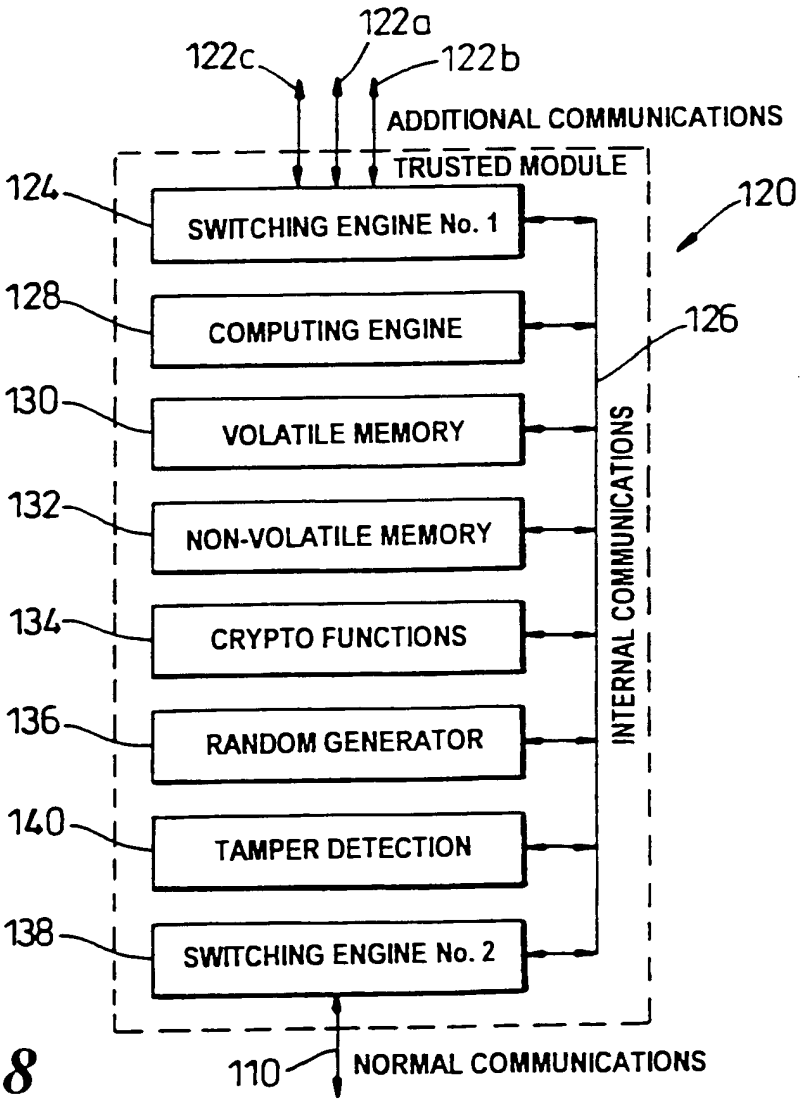


Fig. 8

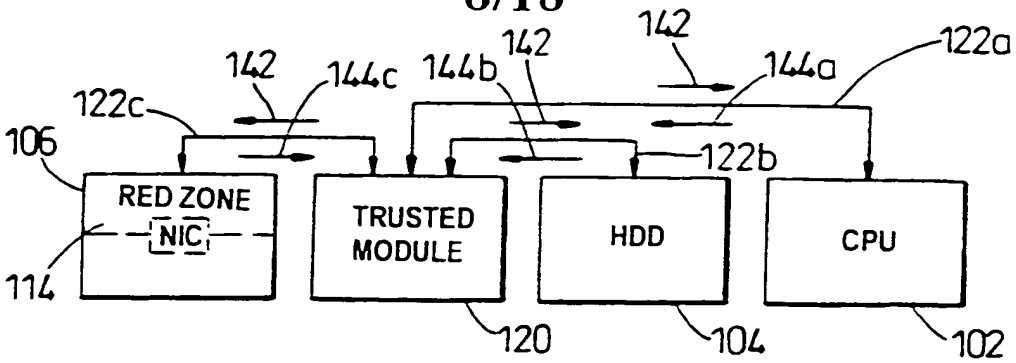


Fig. 9

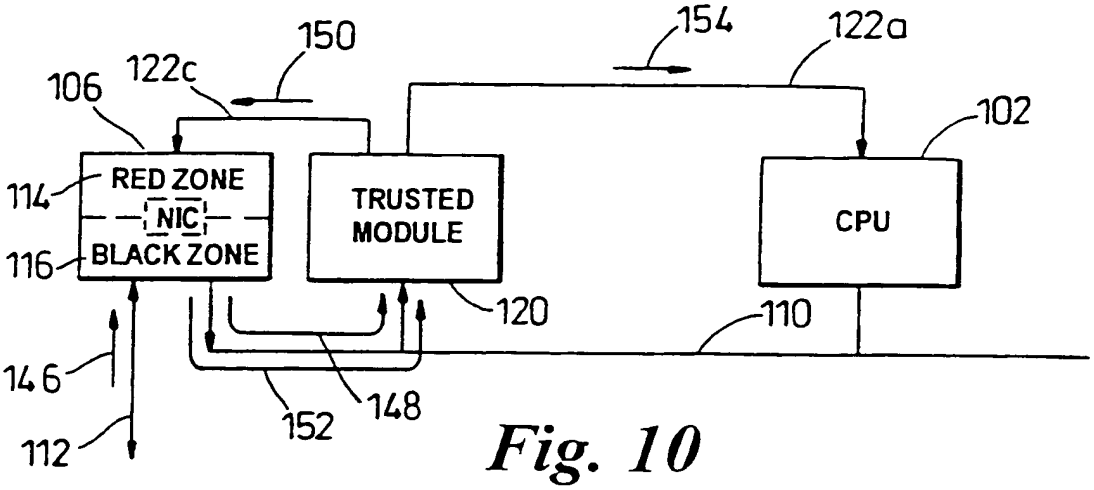


Fig. 10

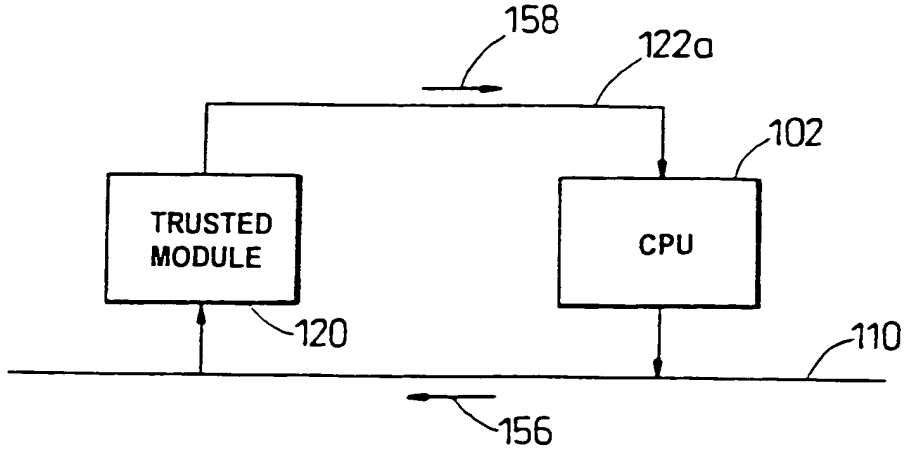


Fig. 11

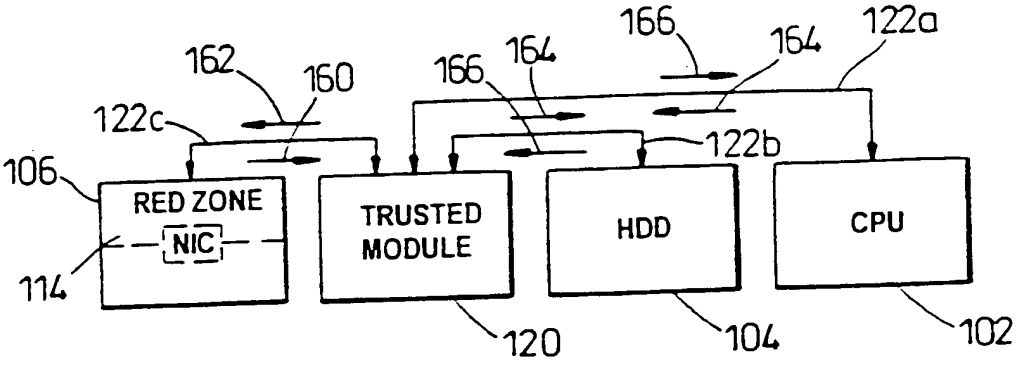


Fig. 12

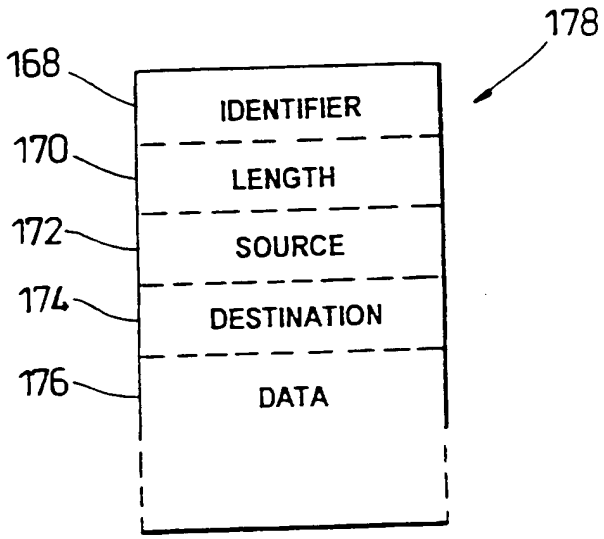


Fig. 13

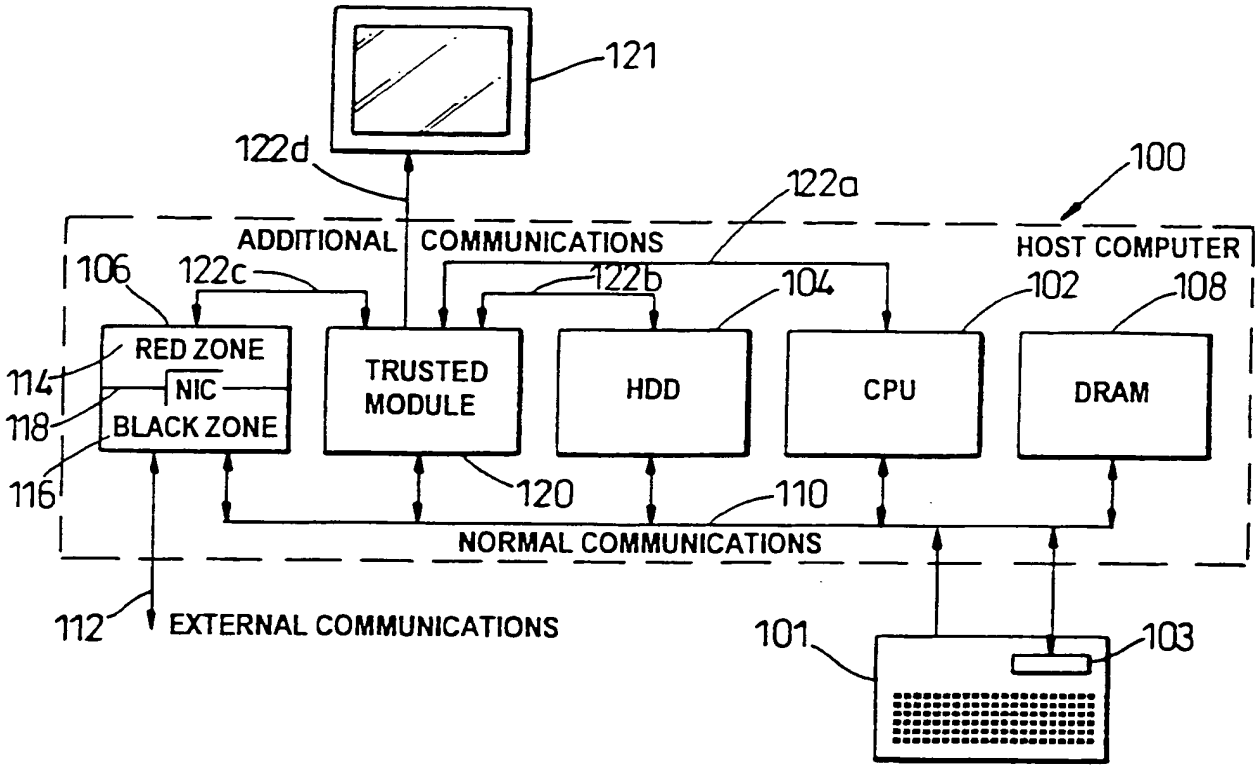
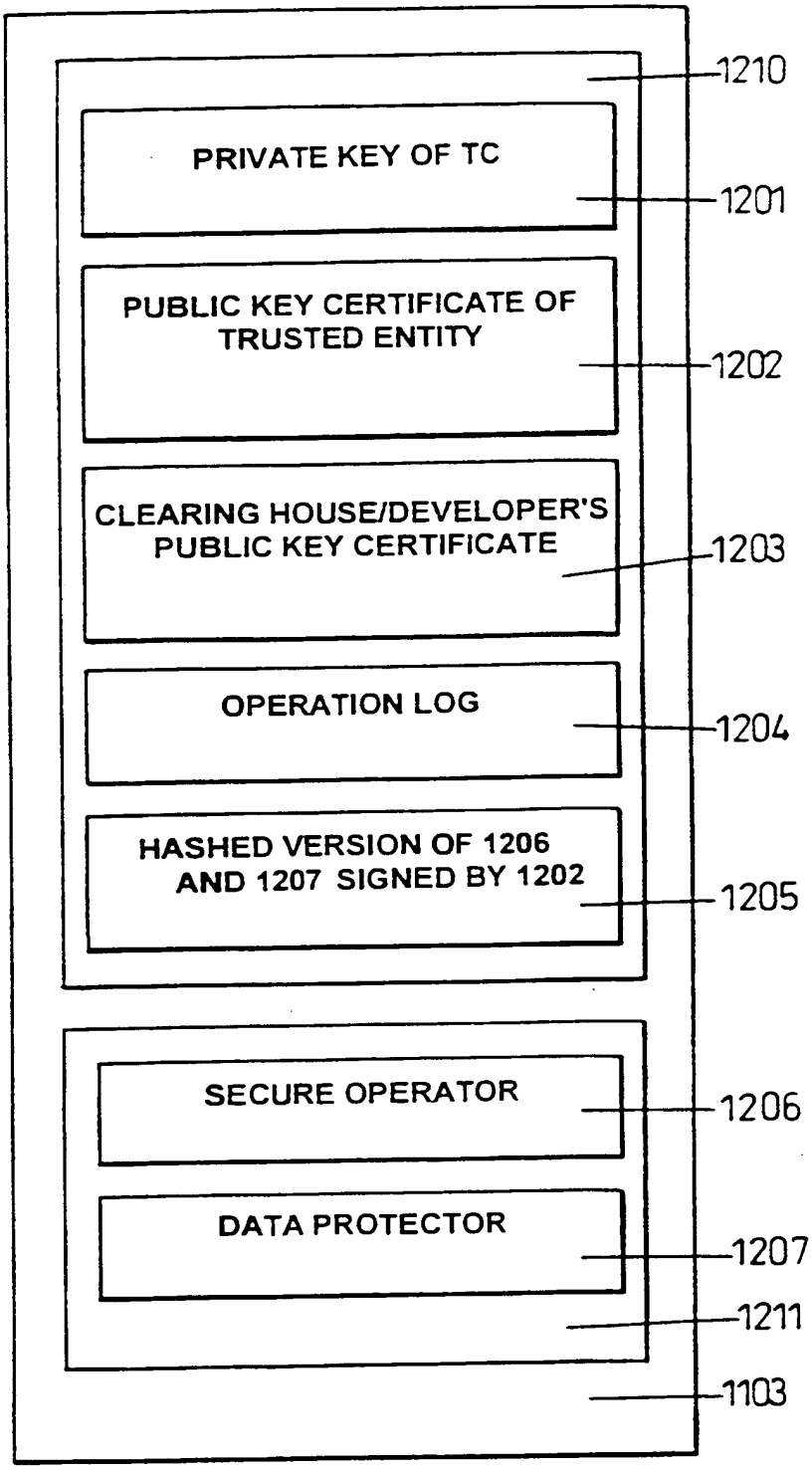
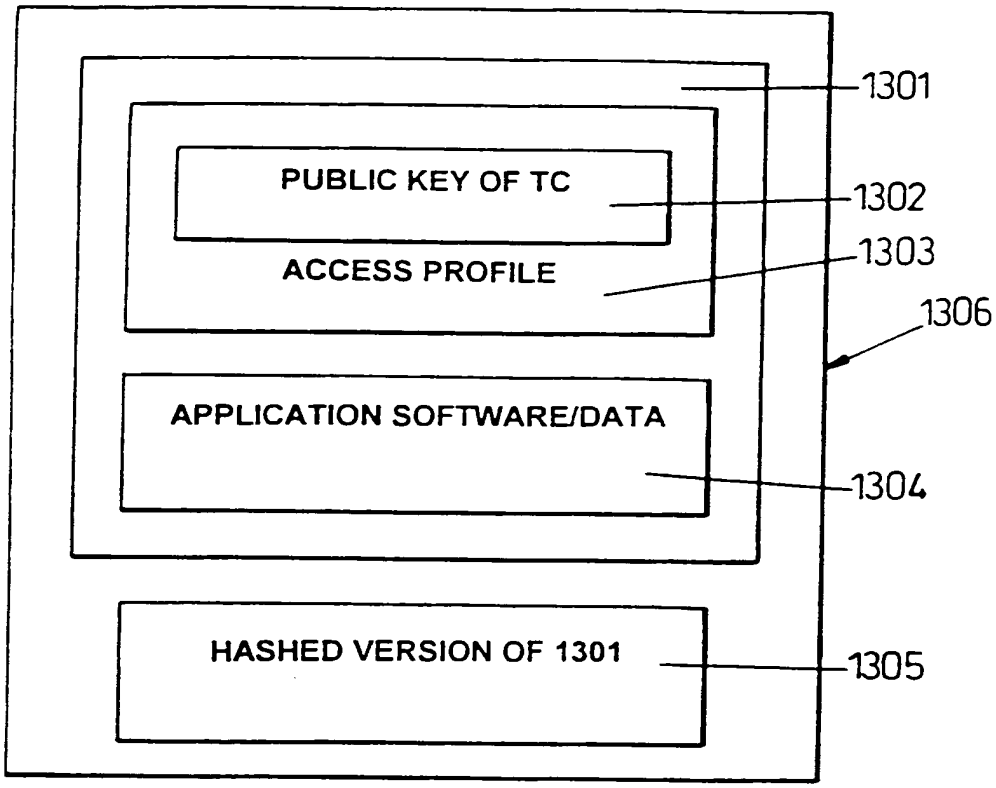


Fig. 14



LOGICAL DIAGRAM OF TC

Fig. 16



LOGICAL DIAGRAM OF APPLICATION SOFTWARE/DATA MOUNTED ON CLIENT PC

Fig. 17

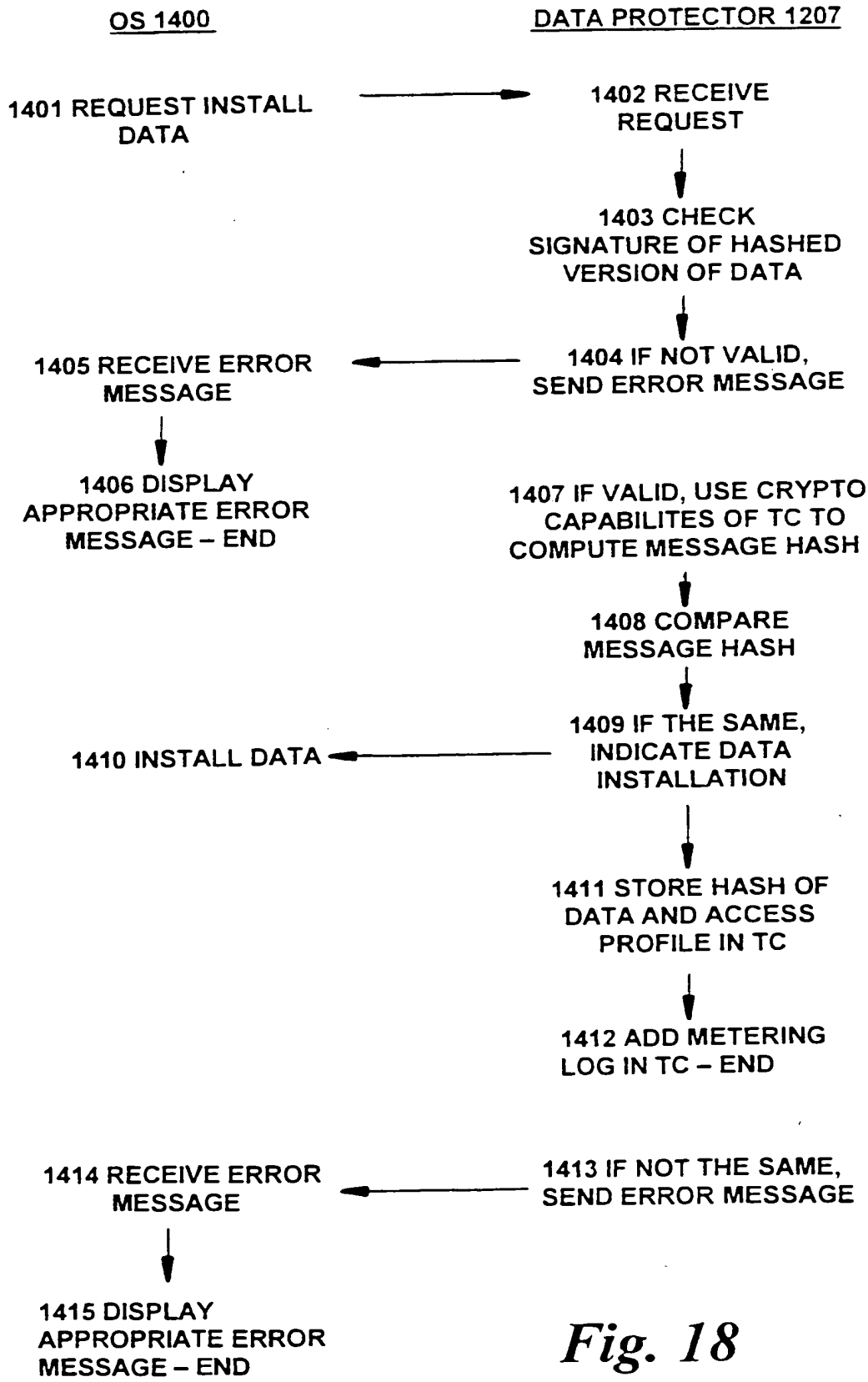
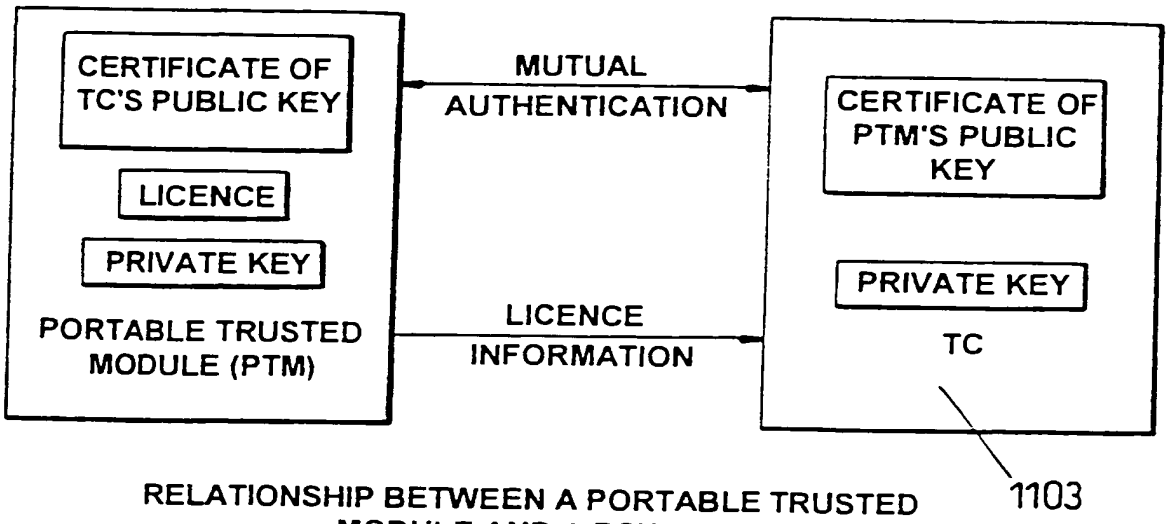


Fig. 18



RELATIONSHIP BETWEEN A PORTABLE TRUSTED MODULE AND A PC'S TC

Fig. 19

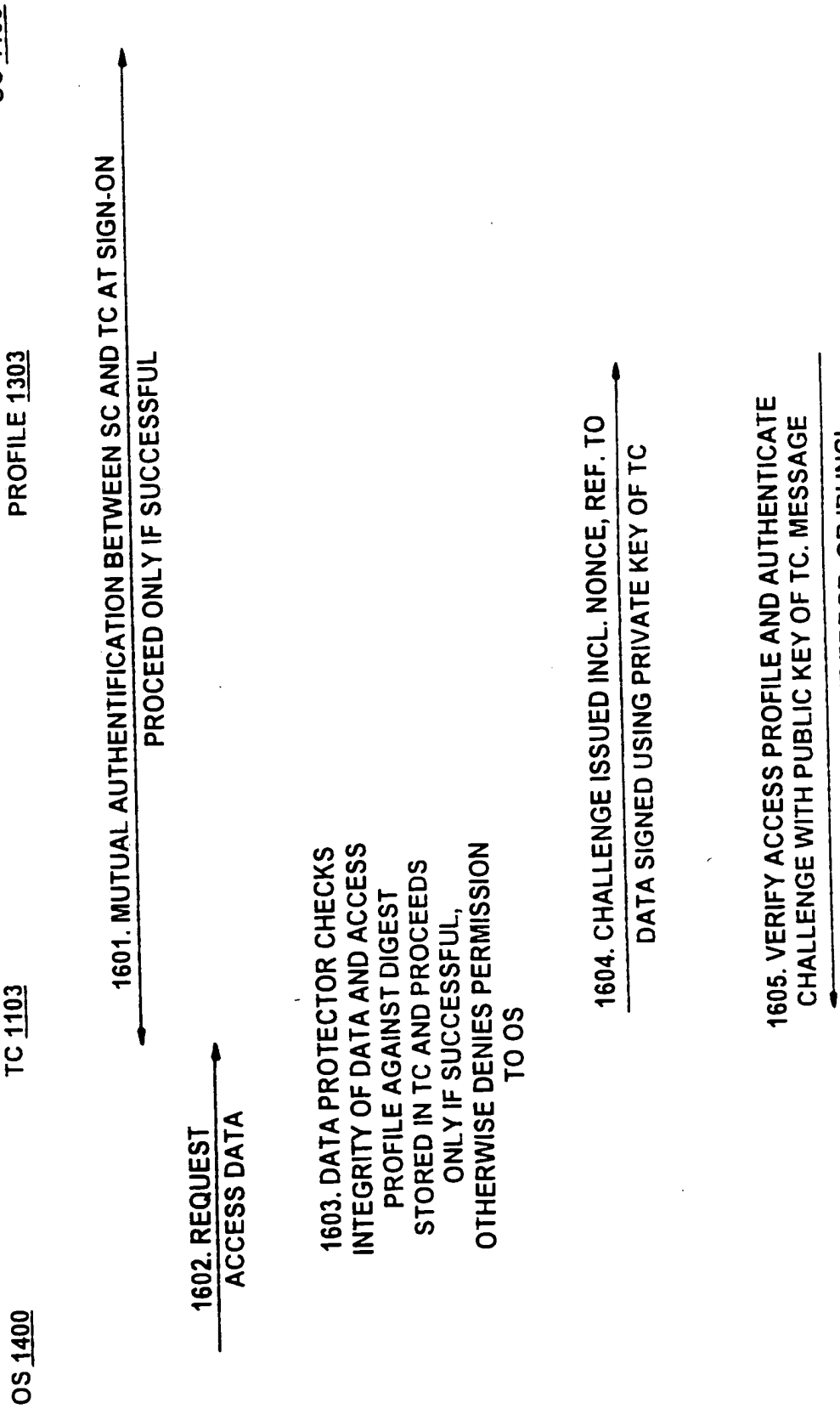


Fig. 20

Fig. 20 (Cont.)

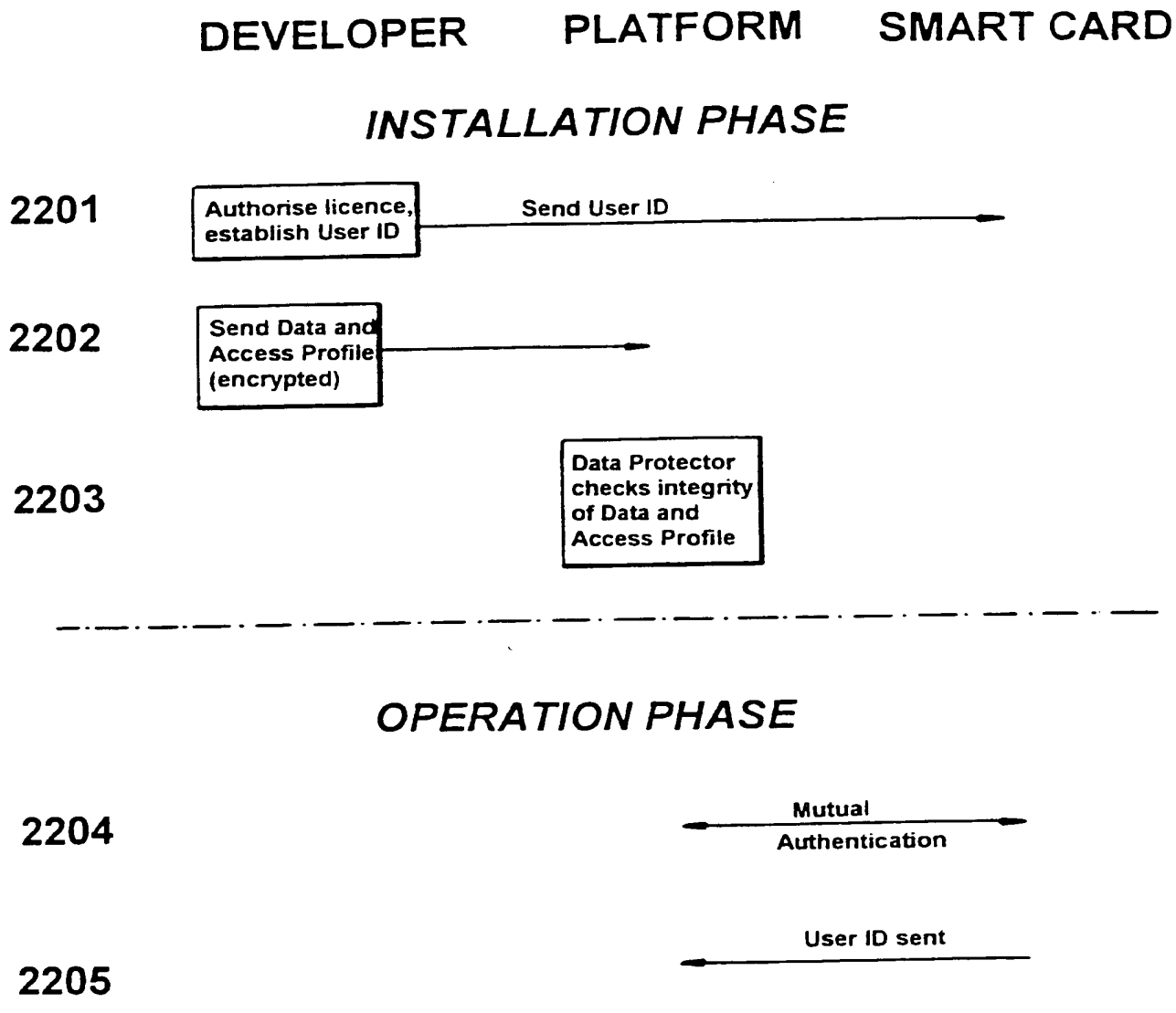


Fig. 22

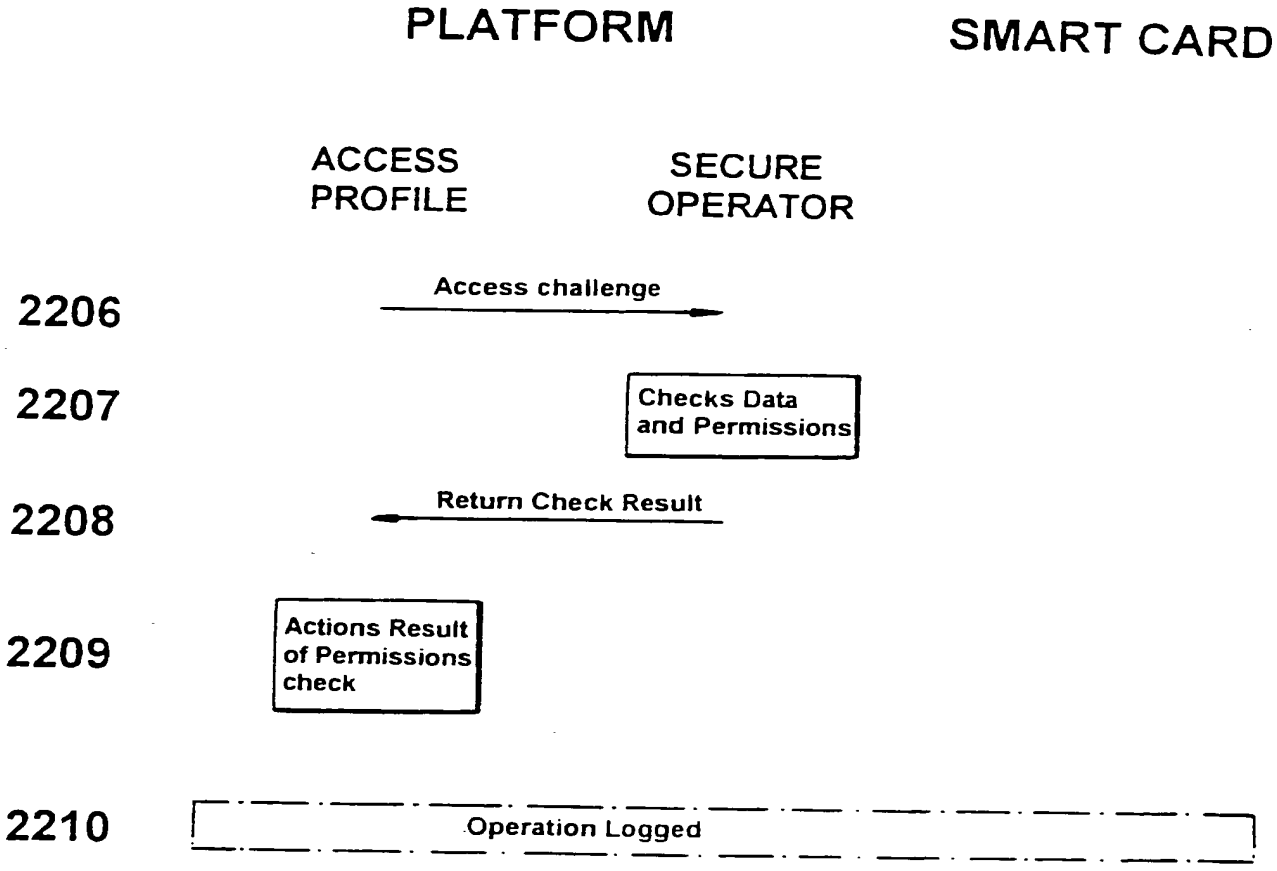


Fig. 22(Cont.)

INTERNATIONAL SEARCH REPORT

Ir. ational Application No
PCT/GB 00/03095

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 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 7 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)
EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 933 498 A (ABRAMS MARSHALL D ET AL) 3 August 1999 (1999-08-03) abstract; figures 3,14; table I column 7, line 1 - line 45 column 15, line 20 -column 17, line 12 column 21, line 10 - line 25 column 22, line 62 -column 25, line 5	1,4-8, 11,13, 18-20, 26,27, 30-32, 36,39-42
Y		2,3,9, 10,12, 14-17, 21,22, 24,25, 28,29, 33-35, 37,38

-/--

[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

Date of the actual completion of the international search	Date of mailing of the international search report
22 January 2001	30/01/2001

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

2

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 00/03095

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No
X	SCHNECK P B: "PERSISTENT ACCESS CONTROL TO PREVENT PIRACY OF DIGITAL INFORMATION" PROCEEDINGS OF THE IEEE.IEEE. NEW YORK.US. vol. 87, no. 7, July 1999 (1999-07), pages 1239-1250. XP000955318 ISSN: 0018-9219 cited in the application page 1243, left-hand column, line 6 -page 1244, right-hand column, last line page 1246, right-hand column, line 13 - line 44 page 1249, left-hand column, line 12 - line 25	1,18,27, 39,40
Y	US 5 473 692 A (DAVIS DEREK L) 5 December 1995 (1995-12-05) the whole document	2,3,10, 12, 14-17, 21,22, 24,25, 28,29, 33,35, 37,38
Y	US 5 680 547 A (CHANG STEVE MING-JANG) 21 October 1997 (1997-10-21) the whole document	9,34
A		26
A	EP 0 421 409 A (IBM) 10 April 1991 (1991-04-10)	

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/GB 00/03095

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5933498 A	03-08-1999	AU 1690597 A	01-08-1997
		CA 2242596 A	17-07-1997
		EP 0880840 A	02-12-1998
		JP 2000503154 T	14-03-2000
		WO 9725798 A	17-07-1997
US 5473692 A	05-12-1995	AU 3583295 A	27-03-1996
		EP 0780039 A	25-06-1997
		JP 10507324 T	14-07-1998
		WO 9608092 A	14-03-1996
		US 5568552 A	22-10-1996
US 5680547 A	21-10-1997	US 5444850 A	22-08-1995
		AU 1042895 A	15-05-1996
		JP 10511783 T	10-11-1998
		WO 9613002 A	02-05-1996
EP 0421409 A	10-04-1991	US 5048085 A	10-09-1991
		CA 2026739 A,C	07-04-1991
		JP 3237551 A	23-10-1991
		US 5148481 A	15-09-1992

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

(11) EP 0 818 748 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
14.01.1998 Bulletin 1998/03

(51) Int Cl 6: **G06F 17/60**

(21) Application number: **97304946.3**

(22) Date of filing: **07.07.1997**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: **08.07.1996 JP 178130/96**
21.05.1997 JP 130626/97

(71) Applicant: **Murakoshi, Hiromasa**
Koriyama-shi, Fukushima, 963 (JP)

(72) Inventor: **Kanno, Kazuhiro**
Koriyama-shi, Fukushima, 963-02 (JP)

(74) Representative:
Cross, Rupert Edward Blount et al
BOULT WADE TENNANT,
27 Furnival Street
London EC4A 1PQ (GB)

(54) Software management system and method

(57) An operation management system for managing the operation of a managed software product. When a management target function is executed, reference is made to a battery value and, if the value is zero or greater, the function is allowed to be executed. The battery value is decremented as the function is executed. A charge value is supplied on a charge disk, such as a floppy disk, to allow the user to increase the battery value and to extend the usage period of the managed software product. The charge value may be supplied over a communication line.

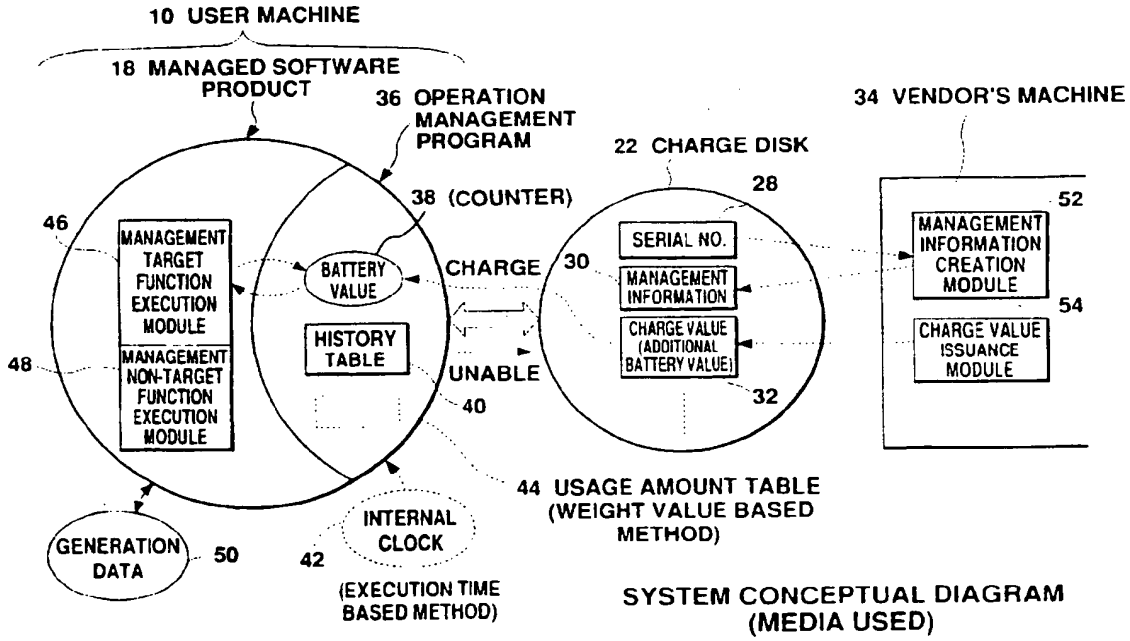


Fig. 3

EP 0 818 748 A2

Printed by Jouve 75001 PARIS (FR)

Description

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates to an operation management system and an operation management method, and more particularly to software operation management or execution management.

Description of the Related Art

As computers and computer use become more common, more advanced technology is introduced and a variety of software products are developed for use in various fields. However, in many cases, the user finds it difficult to select a product from among a variety of software products that seem to meet the user's requirements: often, the user cannot find the best tool for his needs.

To reduce such a risk, a service has been available that supplies the user with a trial-use software product free of charge. However, most of these trial-use software products contain only function descriptions or provide the user with limited functions (e.g., save function and/or output function is/are not included). This makes it difficult for the user to evaluate the actual product (all the functions) correctly.

A sales system which charges the user according to how long the user actually uses a software product (including a trial use) would allow him to buy the product anytime he wants, to fully evaluate the product, and to precisely determine the requirements for continued use (including payment for it). Many users would find this type of sales system appealing and economical.

In Japanese Patent Laid-Open Publication No. Sho 59-41061 and Japanese Patent Laid-Open Publication No. Sho 63-153633, a system is disclosed that automatically prevents a program from being used when the usage count reaches a specified value. In Japanese Patent Laid-Open Publication No. Hei 1-147622 a system is disclosed which accumulates program execution time (total program execution time) and prevents the program from being used when the accumulation time reaches a specified amount. However, these systems do not disclose means for extending the program usage period. Japanese Patent Laid-Open Publication No. Hei 5-134949 discloses a system in which a program and expiry of the program are downloaded from a host computer to a user computer via a communication line. Also disclosed is a system in which a new expiry of the program is downloaded from the host computer to the user computer in order to update the expiry. However, the system only measures the execution time taken for executing the entire program, and does not include any means for changing the expiry on the user computer.

In Japanese Patent Laid-Open Publication No. Hei 7-234785 a system is disclosed that relates to a software rental system. This system connects a computer in a rental company to a user computer on which a rental software product is running over a communication line. When the time elapsed from the rental start time reaches the rental limit time, the system makes the program unavailable for use. (For example, the program is deleted.) To allow the user to update the rental period, the rental company sends a rental period extension program to the user's computer over a communication line. The user runs this program to extend the rental period of the program. A drawback of this system is that the user must pay for the software product regardless of whether the user has used it frequently or not. This means that the amount of money the user has to pay depends, not on how often he has used it, but on how long he has used it.

In Japanese Patent Laid-Open Publication No. Hei 7-244585, a system is disclosed that manages the program usage period. This system assigns a usage limit date to a program and, when the current date becomes greater than the limit date, the program product is made unavailable. To extend the usage limit date, the system reads update limit data from a recording medium containing that data and re-assigns a usage limit date based on the update limit data. This system is not reasonable because the amount of money the user has to pay does not depend on whether or not the user actually uses the program.

For example, during execution of a Computer Aided Design (CAD) software product, the user often spends much time thinking without entering data. In the system disclosed by the above mentioned Japanese Patent Laid-Open Publication No. Hei 7-234785 or Japanese Patent Laid-Open Publication No. Hei 7-244585, the user must pay for this thinking time. This places unwanted pressure on the user, especially when he must think carefully during program execution.

SUMMARY OF THE INVENTION

The present invention seeks to solve the problems associated with the art described above. In view of the foregoing, it is an object of the present invention to provide an operation management system and method which reasonably manage the operation of a managed software product.

It is another object of the present invention to provide an operation management system and method which levy a charge according to the actual usage amount of the managed software product (or the amount of the result generated by the managed software product).

It is still another object of the present invention to provide an operation management system and method which manage the operation according to the property of each function of the managed software product

(1) To achieve the above objects, an operation manage-

ment system for managing the operation of a managed software product according to the present invention comprises: battery value management means for decrementing a battery value according to the operation amount of the managed software product: operation limit means for limiting the operation of the managed software product when the battery value has decreased to a specified limit value: and charge means for adding a charge value to the current battery value when the charge value is entered from external means.

The "battery value" mentioned above is a "virtual battery" which drives a managed software product. This battery value is preferably the value of a counter.

The battery value management means decrement the battery value according to the operation amount of the managed software product. When the battery value has reached a specified limit value (for example. 0). the operation limit means limit all of or a part of the operation of the managed software product. Upon receiving a charge value (additional battery value) from the external means. the charge means add the received value to the current battery value. thus extending the operation period. That is. the battery value is incremented. just as a battery is charged. to allow the continued use of the managed software product.

The managed software product described above is preferably a packaged application software program including a CAD program. game program. video program. language processor. music program. communication program. or a measurement program.

The battery value management means. operation management means. and charge means described above should be implemented preferably as software programs (management software programs) that run on a computer. The managed software product and the management software product may be separate. or the whole or a part of the management software product may be included in the managed software product.

A system according to the present invention is implemented on a general-purpose computer or special-purpose computer having such peripheral units as a disk drive. display. and input unit. The external means described above include recording media such as a magnetic disk or an optical disk and other host computers connected over a network.

(2) An operation management system according to the present invention may be applied to an application software product sales system. The following explains an example.

A vendor sells an application software product containing the operation management program according to the present invention. The operation management program has a battery value defined as the initial value. In addition to this product. the vendor sells recording media containing charge values (e.g.. floppy disk (FD)). In this case. it is desirable that a variety of recording media. each containing a unique charge value. be supplied

On the other hand a user who bought the application software product may use the product until the battery value reaches zero This allows the user to fully evaluate and examine the product. A user who wants to use the product after the battery value becomes zero must buy a recording medium containing a charge value to charge the battery. This enables him to add a charge value to the battery value and to use the product continuously.

If the specifications of the application software product do not satisfy the user's request. the user does not buy the recording medium. This prevents additional charges and reduces the cost to the user.

Considering an increase in the sales profit in recording media that will be produced in the future. a combination of a managed software product and the operation management program will lower prices significantly The operation management system according to the present invention will increase the profits of both the user and the vendor. making it possible to build a very reasonable. economical system.

(3) In a preferred embodiment of the present invention. the battery value management means calculate the operation amount of each function of the managed software product. and subtracts a value corresponding to the operation amount from the battery value.

A continuous decrease in the battery value during execution of a managed software product. as in a conventional system. decrements the value even when the user is idle (input wait time). which places pressure on the user.

Calculating the operation amount of each function during execution of a managed software product, as in a system according to the present invention. decreases the battery value only when the managed software product is actually used. enabling the user to do operation without having to worry about time elapsed while thinking.

(4) In a preferred embodiment of the present invention, function category determination means are also available which determine if an execution instruction from the user activates a management target function or a management non-target function. And. the battery value management means decrement the battery value only when the management target function is executed.

For example. with the data generation function defined as a management target function and with other functions as management non-target functions. a cost can be levied only when new data are generated.

(5) In a preferred embodiment of the present invention, the battery value management means have a weight table containing an operation amount weight value for each of the management target functions. When any of the management target functions is executed, the battery value management means decrement the battery value by the weight value corresponding to the management target function

In a preferred embodiment of the present invention,

the battery value management means measure the execution time of each of the management target functions and decrement the battery value by the value corresponding to the execution time.

This weight value system is able to calculate the operation amount regardless of the computer speed, which may differ among computers. In addition by measuring time in this manner, the execution time is directly monitored and therefore the operation mount becomes proportional to the CPU load.

(6) In a preferred embodiment of the present invention, the operation limit means prevent only the management target functions from being executed when the battery value has decreased to a specified limit value: management non-target functions are executed.

For example, forcing a game program used at home to terminate when the battery value has reached a specified value does not cause a serious problem.

However, for a CAD program used in an office, forced termination when the battery value has reached a specified value may make already-produced data unavailable, possibly interrupting a job. Therefore, considering user's advantage and convenience, the embodiment keeps some functions operable even when the battery value has reached a specified value.

(7) A preferred embodiment of the present invention has remainder warning means for issuing a remainder warning message when the battery value has decremented to a specified warning value because a sudden inoperable condition in the managed software product without prior notice may cause the user unexpected damage. The remainder warning means alert the user to that condition before it occurs. In other words, the warning message prompts the user to determine whether to charge the battery value.

A preferred embodiment of the present invention has remainder display means for displaying the battery value on the screen during execution of the managed software product. This remainder display information keeps the user informed of the amount by which the managed software product will be able to continue operation without being charged.

It is also possible to program the system so that, upon detecting that the battery value has been charged to a specified value, the system can automatically disable operation management through the battery value to allow the user to use the product indefinitely.

(8) To achieve the above objects, a method for managing the operation of a managed software product according to the present invention comprises: a count value management step for changing a count value according to the operation amount of the managed software product: an operation limit step for limiting the operation of the managed software product when the count value has reached a specified limit value: and a charge step for charging the current count value or the limit value when a charge value is entered from external means

The above count value is incremented or decremented according to the operation amount of the managed software product. When the count value is incremented, a charge value is added to the limit value when the count value is decremented, a charge value is added to the current count value. In either case, the usage period is extended by charging the battery value

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagram showing a user machine used in the operation management system according to the present invention.

Fig. 2 is a diagram showing the data structure of a charge disk.

Fig. 3 is a diagram showing the concept of the operation management system according to the present invention.

Fig. 4 is a diagram showing an example of the history table.

Fig. 5 is a diagram showing an example of the usage amount table.

Fig. 6 is a flowchart showing the processing of the system when a management target function is executed in the execution time based method.

Fig. 7 is a flowchart showing the processing of the system when a management target function is executed in the weight value based method.

Fig. 8 is a flowchart showing the charge disk read processing

Fig. 9 is a flowchart showing the charge processing

Fig. 10 is a diagram showing a user machine used in another embodiment.

Fig. 11 is a diagram showing the structure of data sent from the host machine to a user machine.

Fig. 12 is a diagram showing the concept of the system in another embodiment.

Fig. 13 is a diagram showing an example of the user registration table.

Fig. 14 is a flowchart showing the operation of the user machine and a user machine in another embodiment.

Fig. 15 is a diagram showing another configuration of the system.

Fig. 16 is a diagram showing an example of an application according to the present invention.

Fig. 17 is a flowchart showing the function category determination processing

DESCRIPTION OF PREFERRED EMBODIMENTS

Fig. 1 shows a user machine 10. This user machine 10 is a computer which executes various types of application programs under control of the operation system (OS). The user machine 10 is composed of a system unit 12, display 14, keyboard (not shown in the figure), output unit (not shown in the figure) such as a printer or plotter, and so forth The system unit 12 contains a CD-ROM disk drive 16 which accesses a CD-ROM and

reads data from it and a floppy disk drive 20 which accesses a floppy disk (FD) and reads data from it.

The CD-ROM shown in Fig. 1 contains a managed software product 18. In this embodiment, the managed software product 18 such as a CAD software product, has an operation management program built in. The operation management program, designed for managing the operation of the managed software product 18, manages the operation using a "battery value" which will be described below. In the example shown in Fig. 1, the managed software product 18 is installed from the CD-ROM to the user machine 10; it may be installed from any other recording medium or via a communication line.

A charge disk 22, containing specified data (including a charge value) on a floppy disk, functions as a battery value charger. Inserting this charge disk 22 into the floppy disk drive 20 causes a charge value to be read and enables the user to extend the allowable operation period of the managed software product 18. In this embodiment, several charge disks 22, each containing a unique charge value, are supplied to allow the user to select or buy a desired charge disk 22 to add a desired charge value to the battery value.

The managed software product 18 and the charge disk 22 are usually supplied from the same vendor. In this embodiment, the managed software product 18 includes the operation management program. Of course, the managed software product 18 and the operation management program may be separately loaded into the user machine 10.

In Fig. 1, the display 14 has a remainder information area 24 where remainder information is displayed and a remainder warning area 26 where a warning message is displayed when the remainder drops below the specified amount. These areas will be described later.

Fig. 2 shows the data structure of the charge disk 22. As shown in Fig. 2, the charge disk 22 contains a serial number 28, management information 30, and charge value (additional battery value) 32. The serial number 28 is a unique identification number that is assigned when the floppy disk is formatted. Usually, this number is not copied when the disk is copied. The management information 30 is created when the serial number 28 is encrypted. This management information 30 is copied when the disk is copied. Therefore, when the disk is copied illegally, the serial number 28 and the management information 30 do not match, thereby making it easy to determine that the disk is copied illegally. Of course, any other conventional security system may also be used instead of this method.

The charge value 32 is an additional charge value to be added to the battery value that is decremented as the user uses the managed software product 18. Charging the battery value with this charge value enables the user to extend the usage period.

When the battery value is managed in the "execution time based method" in which the battery value is decremented by the execution time of each function, an additional time is recorded as the charge value 32. On the other hand, when the battery value is managed in the "weight value based method" in which the battery value is decremented by the weight value of each function, the additional value is recorded as the charge value 32. These methods will be described in more detail later.

Although a floppy disk is used as the charge disk 22 in the embodiment shown in Fig. 1, other types of recording media may also be used. Also, as shown in another embodiment that will be explained later, a charge value may be sent over a communication line.

Fig. 3 shows the concept of the operation management system which uses the charge disk 22. The system is composed primarily of the user machine 10, charge disk 22, and vendor's machine 34. In this embodiment, the managed software product 18 including the operation management program 36 is installed in the user machine 10.

The charge disk 22 is generated on the vendor's machine 34 owned by the vendor which sold the managed software product 18. More specifically, the vendor's machine 34 has two software modules: the management information creation module 52 and the charge value issuance module 54. The management information creation module 52 encrypts the serial number 28 recorded on the charge disk 22, and writes the resulting management information 30 back onto the charge disk 22. Note that the operation management program 36, which contains the encryption condition or the decryption condition, can check whether or not the serial number 28 agrees with the management information 30. The charge value issuance module 54 records the charge value 32, which has been set by the vendor, onto the charge disk 22. In the execution time based method, the charge value 32 is recorded, for example, as 100 hours, 200 hours, or 500 hours. Note that the operation management program 36 contains an initial battery value (for example, 100 hours).

The operation management program 36 has a counter 38 which decrements the battery value (battery value management function). In this embodiment, the operation management program 36 decrements the counter 38 each time a "management target function" provided by the managed software product 18 is executed. When the battery value, i.e., the counter value, has decremented to the limit value of 0, the operation management program 36 prevents management target functions from being executed. That is, in this embodiment, when the battery value has reached a specified limit value, the execution of the managed software product 18 is limited and, when the battery value is charged with the charge value 32 contained on the charge disk 22, the charge value is added to the battery value and the resulting value is used as a new battery value. The usage period of the managed software product 18 is thus extended.

A history table 40 managed by the operation man-

agement program 36 contains history information on charge values recorded on the charge disk 22. Fig. 4 shows an example. As shown in Fig. 4, the history table 40 is composed of three columns: FD serial number column 40A, charge data/time column 40B, and charge value column 40C. The table may have other columns as necessary.

Referring to Fig. 3 again, the following explains how the battery value is managed. When the battery value is managed in the "execution time based method" described above, the execution time of each management target function, measured based on the internal clock 42, is subtracted from the battery value. On the other hand, when the "weight value based method" described above is used, the battery value is managed based on the usage amount table 44. Fig. 5 shows an example of the usage amount table 44. In this embodiment, the table contains entries, each consisting of a function name 44A and the corresponding usage amount 44B. It should be noted that each usage amount is used as a weight value. For example, a weight value is pre-defined according to the processing time of each function. Therefore, when a management target function is executed, the corresponding usage amount (weight value) is subtracted from the battery value.

The managed software product 18 shown in Fig. 3 has many user interface programs as well as many internal functions and common functions used by the programs. These functions are classified roughly into two: management target functions and management non-target functions. Whenever the managed software product 18 attempts to execute a management target function, the operation management program 36 references the battery value and, when it is zero or greater, allows the managed software product 18 to execute that function. When the managed software product 18 attempts to execute a management non-target function, the operation management program 36 does not check the battery value. For example, when input/output function for processing generated data 50 from the managed software product 18 is defined as a management non-target function, the input/output processing is always executed on the generated data 50, even if the usage period of the managed software product 18 has expired. This ensures that the generated data 50 are always processed, thus protecting user assets. Examples of management non-target functions include the data display function, data print function, and data plotter output function.

Management target functions include the data generation function. For example, when the managed software product is a CAD software product, the data generation function includes the straight-line drawing function, curved-line drawing function, circle drawing function, area fill-in function, area hatching function, and character insertion function.

Fig. 3 conceptually shows management target function execution module 46 which executes management target functions and management non-target function execution module 48 which executes management non-target functions. In this embodiment, the battery value is decremented only when a management target function is activated. Note that the battery may be decremented when both a management target function and a management non-target function are activated.

In addition to the data described above, the charge disk 22 may contain other types of data. For example, it may contain the name of the managed software product 18 which accepts a charge value. In this case, the name of the managed software product 18 is used as follows. When the charge disk 22 is read, the operation management program 36 checks whether or not the name of the managed software recorded on the charge disk 22 matches that of the managed software product 18 installed in the user machine 10 and, only when they match, accepts the charge value 32.

The battery value described above is stored on the hard disk and then copied into the computer's RAM. The battery value in the RAM is decremented whenever a management target function is executed. Also, at an interval or as necessary, the battery value in the RAM replaces the battery value on the hard disk. This means that, even when the computer fails, the battery value is not erased. The battery value may also be maintained in some other way.

Fig. 17 is a flowchart showing how the operation management program operates when it accepts an instruction requesting the execution of a managed software product function. The following explains this processing in more detail.

Upon receiving from a user an instruction requesting the execution of a function of the managed software product while the managed software product is in execution (S601), the operation management program checks whether the requested function is a management target function or a management non-target function (S602). When the function is a management target function (S603), the operation management program performs the processing shown in Fig. 6 or Fig. 7 (S604). When the function is a management non-target function (S603), the program executes the function immediately (S605). This processing is repeated whenever an execution instruction is received.

Next, referring to Fig. 3, the execution of a management target function in the execution time based method is explained with the use of Fig. 6.

When the user requests the execution of a management target function while the managed software product 18 shown in Fig. 3 is in execution, the routine shown in Fig. 6 is started. First, the management target function execution module 46 or the operation management program 36 reads the battery value to check if it is greater than zero. If the battery value is zero or less, the routine is terminated. That is, the requested management target function cannot be started. Note that a management non-target function is started even if the battery value is

zero.

In S102, the routine gets the start time from the internal clock 42 before starting the requested management target function and. in S103. starts the management target function. In S104. the routine gets the end time from the internal clock 42 and. in S105. subtracts the start time from the end time to calculate the processing time (execution time) of the processing executed in S103.

In S106. the routine subtracts the processing time calculated in S105 from the battery value. In S107. the routine checks if the resulting battery value is equal to or less than the warning value and, if so displays a message in the remainder warning area 26 shown in Fig. 1. If the resulting battery value is greater than the warning value, the routine does not display the message As shown in Fig. 1, the remainder information area 24 is displayed during execution of the managed software product 18 (see Fig. 1) to allow the user to check the remaining amount. This helps the user determine how long he can execute the managed software product 18.

Fig. 7 shows the processing of a management target function in the weight value based method.

When the execution of a management target function is requested as described above, the routine references the battery value in S201 to check if it is equal to or greater than 0. If it is, the routine executes the requested management target function in S202 and. in S203, references the usage amount table 44 shown in Fig. 5 to find the usage amount (weight value) of the executed management target function. Then, in S204. the routine subtracts the processing amount found in S203 from the battery value to find a new battery value. In S205. the routine checks if the battery value is less than the warning value and, if so, displays a message in the remainder warning area 26 in S206.

The "execution time based method" shown in Fig. 6 allows the user to manage operation using a physical amount that is easy to understand. In addition, the user can manage operation in a relatively simple configuration. On the other hand, the "weight value based method" shown in Fig. 7 gives the user the same result regardless of the CPU speed of the user's machine.

Next, referring to Fig. 3, the charge disk 22 read processing is explained with the use of Fig. 8.

This processing is started when the charge disk 22 is inserted into the floppy disk drive 20 as shown in Fig. 1. The routine reads the serial number in S301, and the management information in S302, both from the charge disk 22. In S303, the routine encrypts the serial number according to the encryption condition, or decrypts the management information according to the decryption condition, and compares the serial number with the management information. This comparison determines whether or not the charge disk 22 is legal. For example, when the disk is illegally copied, the management information 30 is copied. but the serial number 28 is not copied but replaced. This results in a mismatch between the serial number 28 and the management information 30 thereby making it possible to find an illegal copy

In S304. the routine checks if the charge disk 22 is valid and. if it is not valid. terminates processing in S309. If it is valid. the routine references the history table 40 containing past charge history data. in S305 to check the validity of the charge value 32 recorded on the charge disk 22 To do so. the routine first checks to see if the serial number 28 of the charge disk 22 is in the history table 40. If the serial number is found. the routine takes the following steps to check if the charge value 32 recorded on the charge disk 22 is valid. The routine finds the charge value initially recorded on the charge disk 22 and. from that initial value. subtracts the actual charge value to find the remainder. The next time the battery value is charged. the routine compares the remainder with the charge value currently recorded on the charge disk. If the charge value on the charge disk 22 is greater than the remainder. the routine determines in S306 that the charge disk is not valid and terminates processing in S308. If the routine finds that the charge value 32 on the charge disk 22 is valid. it performs the charge processing. shown in Fig. 9. in S307.

Fig. 9 shows an example of charge processing. In S401. the routine references the counter 38 to read the current battery value and. in S402. reads the charge value from the charge disk 22. In S403. the routine asks the user to type an actual charge value that does not exceed the charge value 32 recorded on the charge disk 22 The user types the charge value. for example. from the keyboard. In S404. the routine checks that the specified charge value is less than the charge value on the charge disk 22. If the specified charge value is greater than the charge value on the charge disk 22. the routine asks the user to retype the charge value.

In S405. the routine adds the specified charge value to the battery value, thus charging the battery value. In S406. the routine subtracts the specified charge value from the initial charge value and writes the resulting value on the charge disk 22 as a new charge value 32 . If the initial charge value 32 is exhausted, the routine writes the value of 0 on the charge disk 22 to virtually erase the charge value. The value of 0 prevents the charge disk 22 from being re-used. In S407, a record relating to the charge processing is added to the history table 40.

In the above embodiment. the user specifies an actual charge value. Instead of having the user specify a value. a pre-defined charge value may be added to the battery value at that time.

Fig. 10 shows another embodiment according to the present invention. In the embodiment described above. the battery value is charged using a recording medium. In this embodiment. the battery value is charged via a communication line 60. For the same components as those used in the above embodiment. the same numbers are assigned and their descriptions are omitted.

The user machine 10 in Fig 10 is connected to the

host machine 62 via the communication line 60. From this host machine 62, send data 64 shown in Fig. 11 are sent to the user machine 10 to charge the battery value.

In Fig. 11, address information 68 specifies the address of the user machine 10. Management information 70 is created by encrypting the serial number on the recording medium containing the managed software product 18. A charge value 72, a value to be added to the battery value as with the above embodiment, is an additional period of time in the execution time based method, and is an additional amount in the weight value based method.

Fig. 12 illustrates the system concept of this embodiment.

As described above, the user machine 10 is connected to the host machine 62 via the communication line 60. That is, this host machine 62 is connected to each of a number of user machines 10 for integrated operation management. This host machine 62 has a management information creation module 76, charge value issuance module 78, user registration table 80, and billing module 82. The management information creation module 76 creates the management information 70 shown in Fig. 11, and the charge value issuance module 78 issues a charge value 72 in response to a request from the user machine 10. As shown in Fig. 13, the user registration table 80 is composed primarily of the user ID column 80A, user name column 80B, and request charge value column 80C. The billing module 82 references the user registration table 80 to automatically issue a bill for a requested amount whenever a charge value is issued, or at some specified interval.

Next, referring to Fig. 12, the operation of this embodiment is explained with the use of Fig. 14. The operation of the user machine 10 is shown in the left side of Fig. 14, while that of the host machine 62 is shown on the right.

First, in S501 and S502, the user machine 10 is connected to the host machine 62 via a communication line. In S503, the user machine 10 generates a request for a charge value that will be sent to the host machine 62. In this case, the request contains at least the serial number of the CD-ROM containing the managed software product 18 and information on the charge value. In S504, the user machine sends the request to the host machine and, in S505, the host machine receives the request.

In S506, the host machine checks the user registration table 80. If the host machine finds, in S507, that the requesting user is registered in the host machine 62, the management information creation module 76 creates management information based on the serial number in S508, and the charge value issuance module 78 generates a charge value in response to the request from the user. In S509, the host machine 62 sends the management information and the charge value to the user machine 10 as the send data 64 shown in Fig. 11. In S510, the user machine 10 receives the send data 64. In S511 and S512, the user machine 10 and the host machine 62 are disconnected.

In S513, the operation management program 36 compares the serial number 74 with the management information 70 to check to see if the data received by the user machine 10 are valid. This prevents the user from illegally charging the battery value. If it is found in S514 that the send data are valid, the charge processing is performed in S515. This charge processing is the same as that in Fig. 9.

As shown in Fig. 12, this embodiment may also use the execution time based method or the weight value based method in order to manage the battery value.

Although the battery value is charged over a communication line such as a telephone line in the above embodiment, it may also be charged over a communication satellite (satellite line).

In the above embodiments, the operation management program 36 is included in the managed software product 18. Of course, an external program can manage the operation of the managed software product 18. Fig. 15 shows the concept of such an embodiment.

As shown in Fig. 15, the operation system (OS) 83 is located between the hardware 81 and each of application programs 84, 86, and 88. The operation management program 36 according to the present invention may be located between the operation system 83 and the application program 84.

Operation management program 36 therefore functions as an interface program. Messages are exchanged between the operation management program 36 and the application program 84 according to some specific rule. Messages are also exchanged between the operation management program 36 and the operation system 83 according to a specific rule.

To execute a management target function in this configuration, the operation management program 36 references the battery value when it receives an execution request from the application program 84. If the battery value is not zero, the operation management program 36 sends an instruction to the operation system 83 while simultaneously decrementing the battery value by a value corresponding to the function. If the battery value is zero, the operation management program 36 sends a message back to the application program 84, indicating that the instruction cannot be executed.

To execute a management non-target function, the operation management program 36 does not reference the battery value when it receives an execution request from the application program 84 but instead sends the instruction directly to the operation system 83.

The battery value is decremented as management target functions are executed. Charging the battery value allows the user to extend the usage period of the application program 84, which may be supplied separately from the application program 84.

In the above embodiments, one operation management program manages one operation management program. It is also possible for one operation manage-

ment program to manage several application programs.

Fig 16 shows an application of the present invention. The system shown in Fig. 16 is composed of one host machine 90 and several user machines 92. Within each user machine 92 are a managed software product 18 and the operation management program 36, which, in turn, contains the counter 38 where the battery value to be decremented is stored. In other words, the operation of the managed software product 18 is controlled by the value stored in the counter 38. To execute the managed software product 18 in this system, it is necessary to insert a battery disk 96 into the user machine 92 and to move the battery value from the battery disk 96 into the counter 38. The battery value is decremented as the operation of the managed software product 18 proceeds. When the user finishes the managed software product 18, a sequence of operations are executed to move the current counter value from the counter 38 to the battery disk 96. This initializes the counter 38 to zero just as it was before the battery disk 96 was inserted.

The host machine 90 has several disk drives into which a battery disk 96 is inserted to read the battery value that was returned to the battery disk 96. This host machine 90 is also used to charge the battery value on the battery disk 96.

Integrated management of the battery values on several battery disks 96 through the host machine 90 brings a benefit of integrally managing several managed software products 18.

This type of system may be used, for example, in a school or a business where many computers are installed. With an individual carrying his or her own portable battery disk 96, it is possible to check and control the software usage amount of each person. In this case, either the "execution time based method" or the "weight value based method" may be used.

Claims

1. An operation management system for managing the operation of a managed software product, comprising:

   battery value management means for decrementing a battery value according to the operation amount of said managed software product;
   operation limit means for limiting the operation of said managed software product when said battery value has decremented to a specified limit value; and
   charge means for adding a charge value to the current battery value when the charge value is entered from external means.

2. An operation management system according to claim 1 wherein said battery value management means find the operation amount for each execution of a function owned by said managed software product and subtract a value corresponding to said operation amount from said battery value

3. An operation management system according to claim 2, further comprising

   function category determination means for determining if a function to which an execution instruction is issued is a management target function or a management non-target function, wherein said battery value management means decrement said battery value only when said management target function is executed.

4. An operation management system according to claim 3, wherein

   said battery value management means has a weight table containing pairs of said management target function and a weight value representing said operation amount thereof, and
   said battery value management means subtract a weight value corresponding to said management target function from said battery value when said management target function is executed.

5. An operation management system according to claim 3, wherein, when said management target function is executed, said battery value management means measure the execution time and subtracts the execution time from said battery value.

6. An operation management system according to claim 3, wherein said operation limit means prevent said management target function from being executed but allows said management non-target function to be executed when said battery value has reached a limit value.

7. An operation management system according to claim 3, wherein said managed software product has a data generation function and a data output function and wherein said function category determination means determine said data generation function as said management target function and determine said data output function as said management non-target function.

8. An operation management system according to claim 1, further comprising remainder warning means for issuing a remainder warning when said battery value has decremented to a warning value.

9. An operation management system according to claim 1, further comprising remainder display

means for displaying said battery value during execution of said managed software product

10. An operation management system for managing the operation of a managed software product, comprising

battery value management means for decrementing a battery value according to the operation amount of said managed software product;
operation limit means for limiting the operation of said managed software product when said battery value has decremented to a specified limit value;
read means for reading a charge value from a recording medium containing the charge value thereon; and
charge means for adding said charge value to the current battery value.

11. An operation management system according to claim 10, further comprising erase means for erasing the charge value from said recording medium after said charge value is added.

12. An operation management system according to claim 10, further comprising:

specification means for allowing a user to specify an actual charge value by which the current battery value is to be actually charged, the actual charge value not exceeding the charge value recorded on said recording medium; and
rewrite means for rewriting the charge value on said recording medium with a remainder value after said actual charge value is added to the current battery value

13. An operation management system according to claim 10, in which said recording medium contains not only said charge value, but also the identification number of the recording medium and management information generated through encryption of the identification number, said operation management system further comprising:

validity determination means for comparing said identification number with said management information considering the condition of said encryption to determine the validity of said recording medium.

14. An operation management system comprising:

a managed machine containing a managed software product; and
a managing machine connected to said managed machine with a communication line,

wherein
said managed machine comprises
battery value management means for decrementing a battery value according to the operation amount of said managed software product;
operation limit means for limiting the operation of said managed software product when said battery value has decremented to a specified limit value;
charge value receive means for receiving a charge value from said managing machine, and
charge means for adding said charge value to the current battery value, and wherein
said managing machine comprises;
charge value send means for sending said charge value to said managed machine.

15. An operation management system according to claim 14, wherein said managed machine further comprises.

notification means for notifying said managing machine of the identification number of a portable recording medium initially containing said managed software product; and
validity determination means for comparing management information sent from said managing machine with said identification number to determine the validity of the recording medium; and wherein said managing machine further comprises;
management information creation means for creating said management information generated by encrypting said notified identification number and for sending the management information to said managed machine.

16. An operation management system comprising:

at least one managed machine containing a managed software product; and
a managing machine for managing the operation of said managed machine, wherein said managed machine comprises;
a counter containing a battery value changing according to the operation amount of said managed software product,
first charge means for reading a battery value from a portable recording medium to store the battery value into said counter; and
first return means for writing the current battery value on said recording medium, and wherein, said managing machine comprises;
second charge means for writing said battery value on said recording medium; and
second return means for reading said battery value from said recording medium.

17. An operation management method comprising:

a count value management step for changing a count value according to the operation amount of a managed software product:
an operation limit step for limiting the operation of said managed software product when said count value has reached a specified limit value: and
a charge step for charging the current count value or said limit value when a charge value is entered from external means.

18. A medium containing a management software product for managing the operation of a managed software product, wherein said managed software product and said management software product are executed on computers, said management software product comprising:

a module for changing a count value according to the operation amount of said managed software product:
a module for limiting the operation of said managed software product when said count value has reached a specified limit value; and
a module for charging the current count value or said limit value when a charge value is entered from external means.

19. A medium containing a charge value read by a management software product for use in managing the operation of a managed software product, wherein said managed software product and said management software product are executed on computers, said management software product comprising:

a module for changing a count value according to the operation amount of said managed software product:
a module for limiting the operation of said managed software product when said count value has reached a specified limit value: and
a module for charging the current count value or said limit value when said charge value is entered.

20. A computer system having an interface software product between an operation system and at least one application software product, wherein said interface software product comprises:

a module for changing a count value according to the operation amount of said application software product:
a module for limiting the operation of said application software product when said count value has reached a specified limit value: and
a module for charging the current count value or said limit value when a charge value is entered from external means

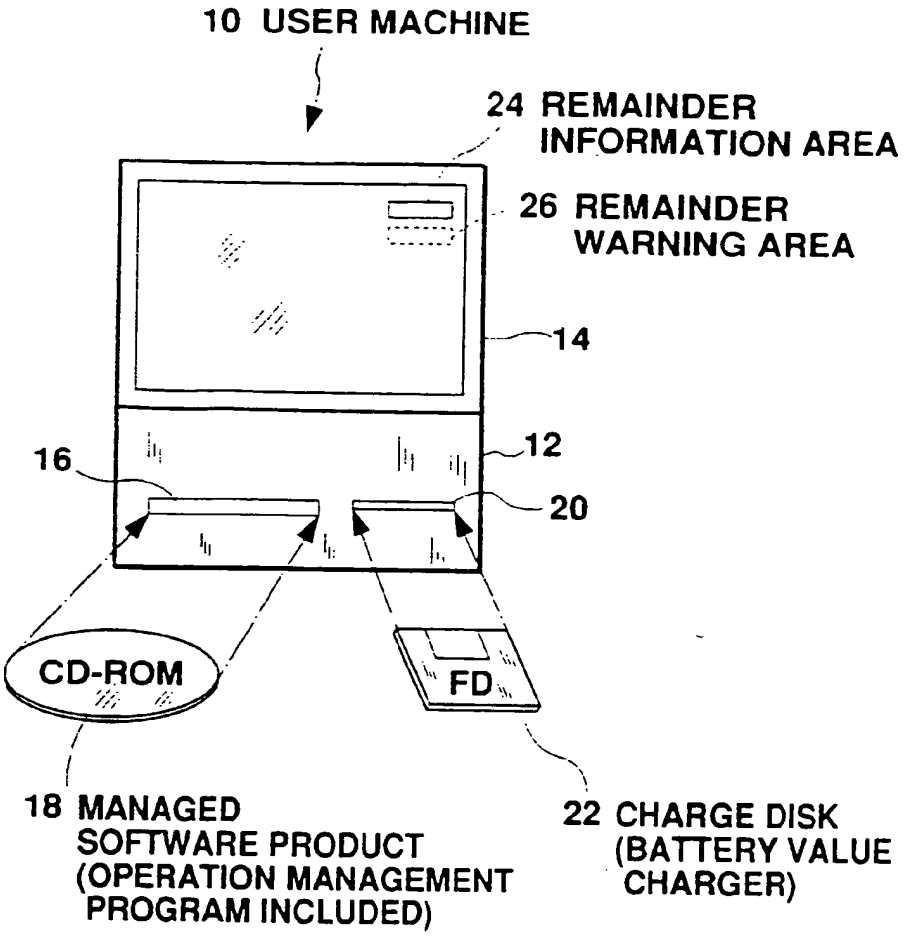


Fig. 1

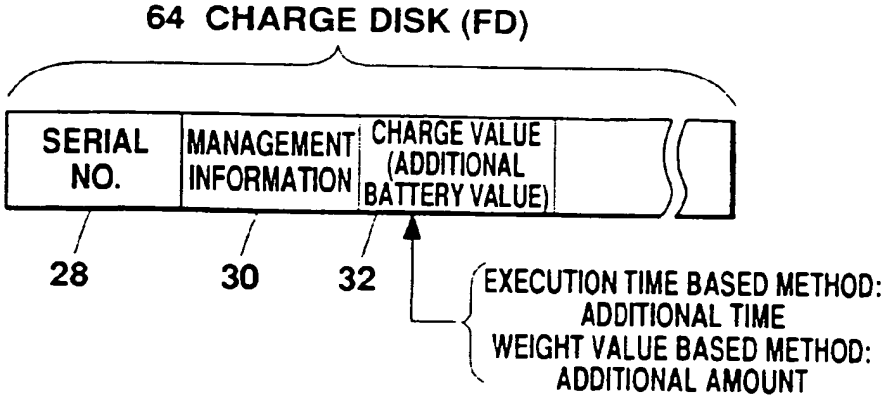


Fig. 2

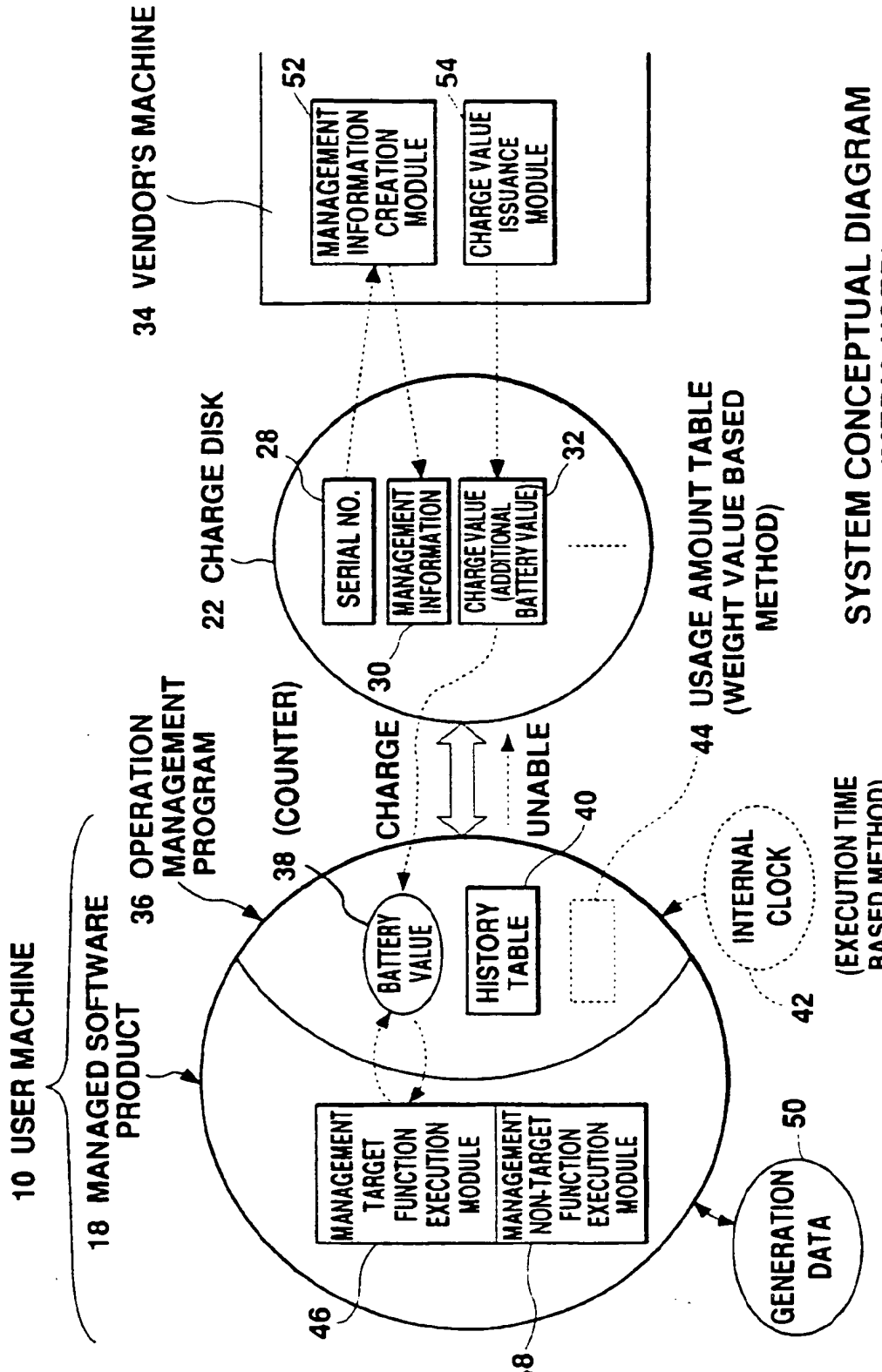


Fig. 3

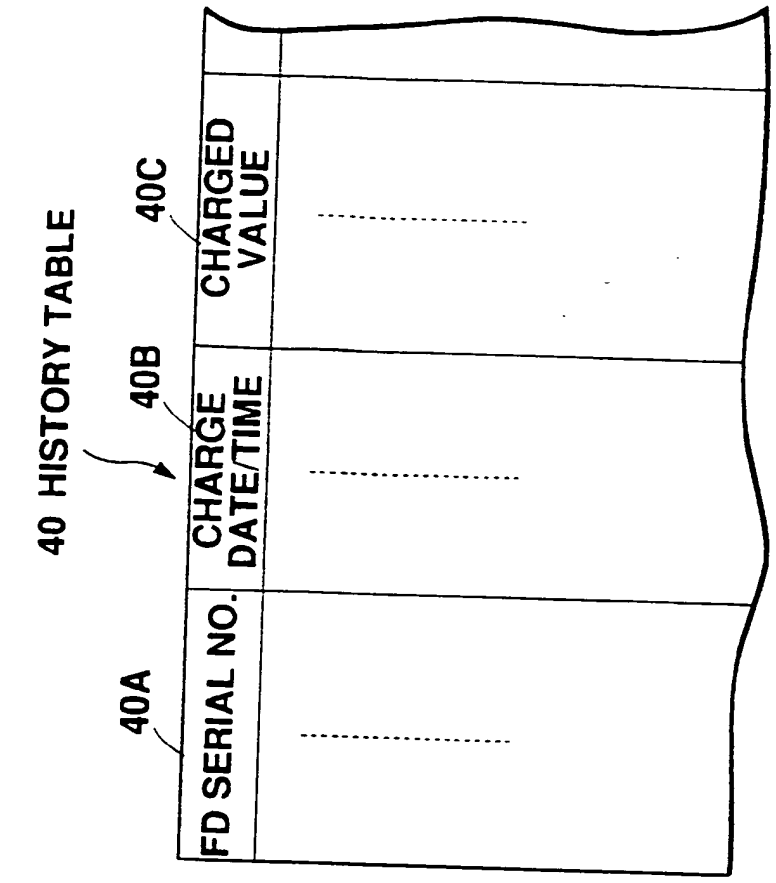


Fig. 5

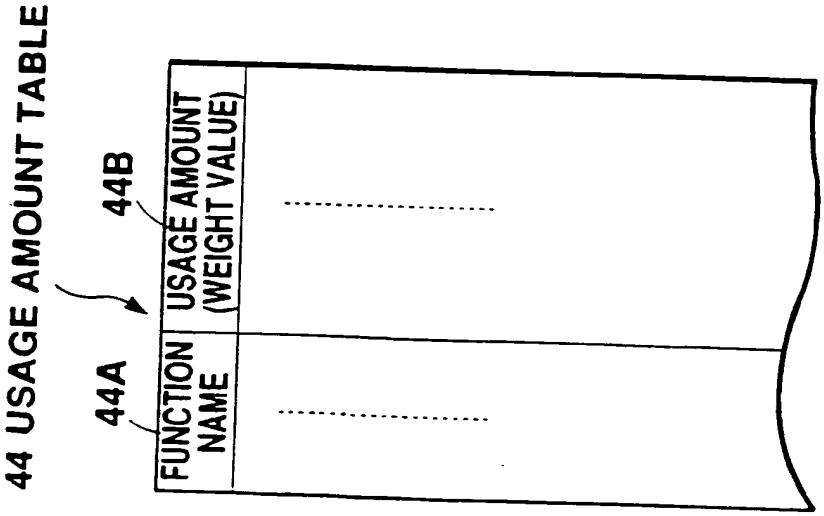


Fig. 4

MANAGEMENT TARGET FUNCTION EXECUTION
(EXECUTION TIME BASED METHOD)

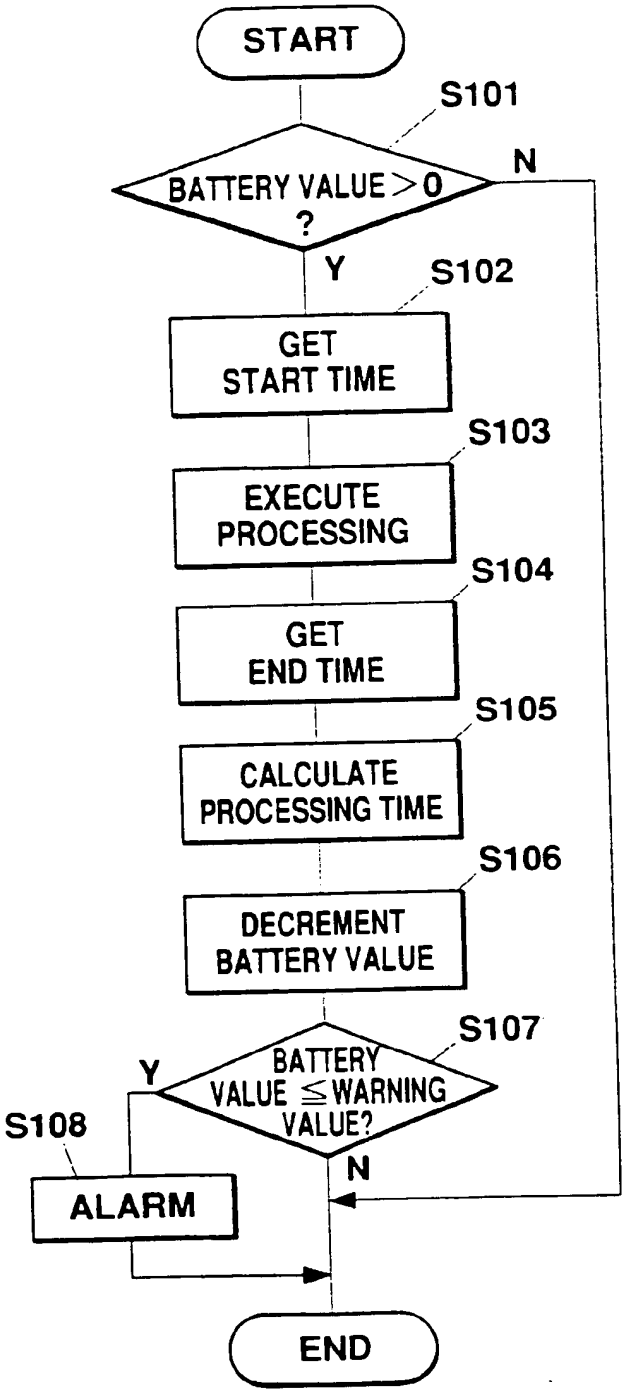


Fig. 6

MANAGEMENT TARGET FUNCTION EXECUTION
(WEIGHT VALUE BASED METHOD)

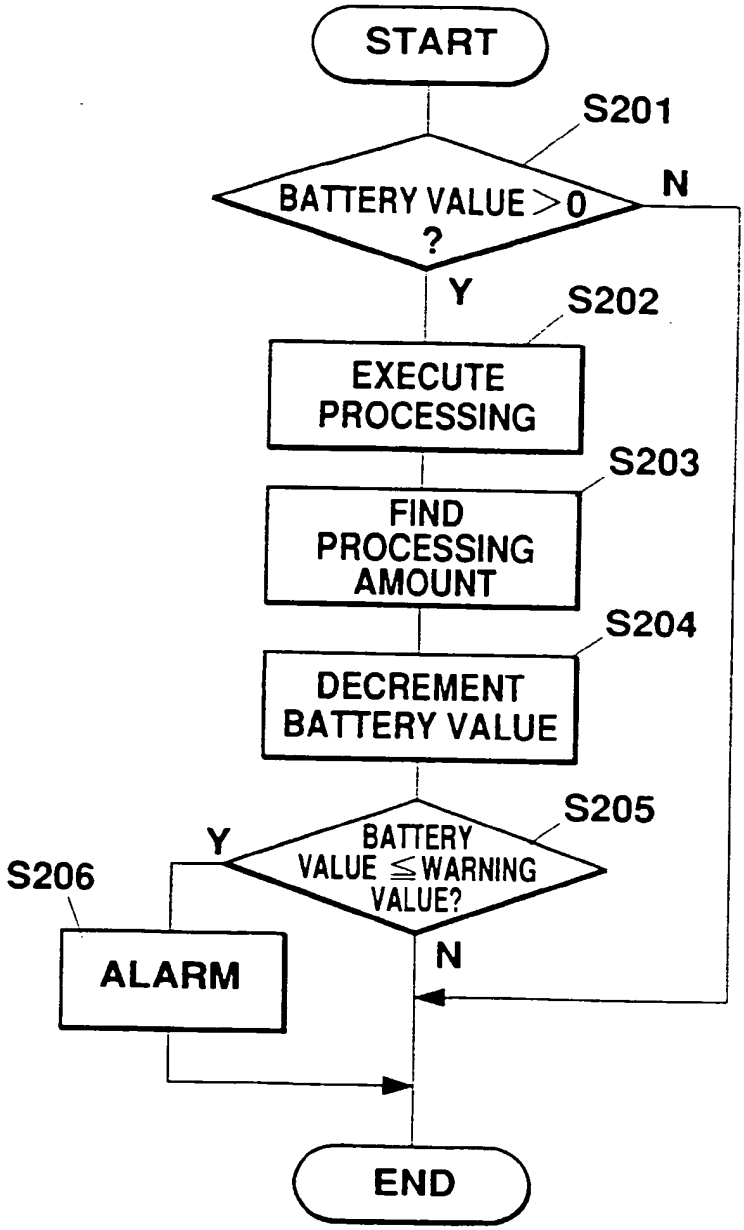


Fig. 7

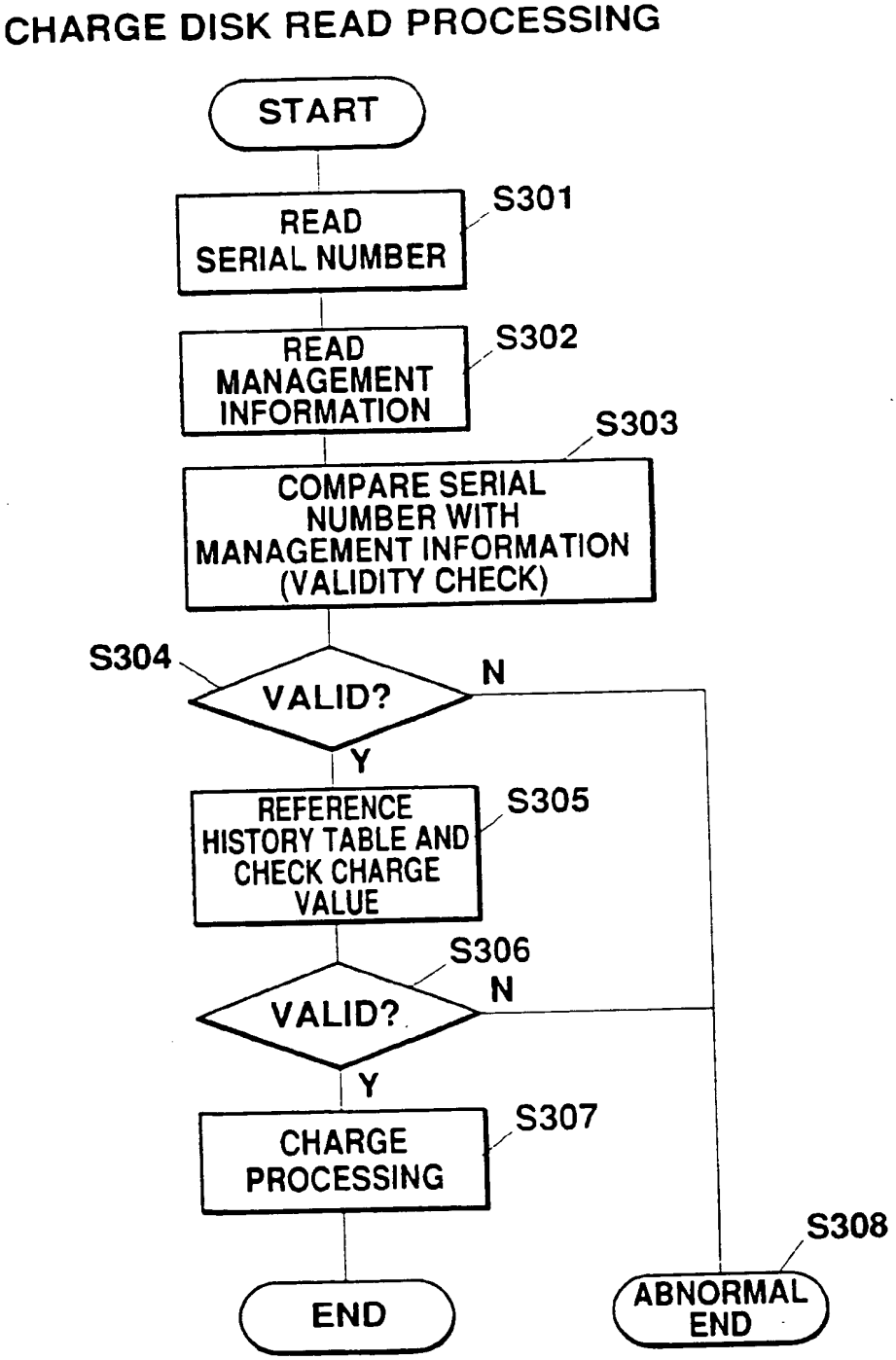


Fig. 8

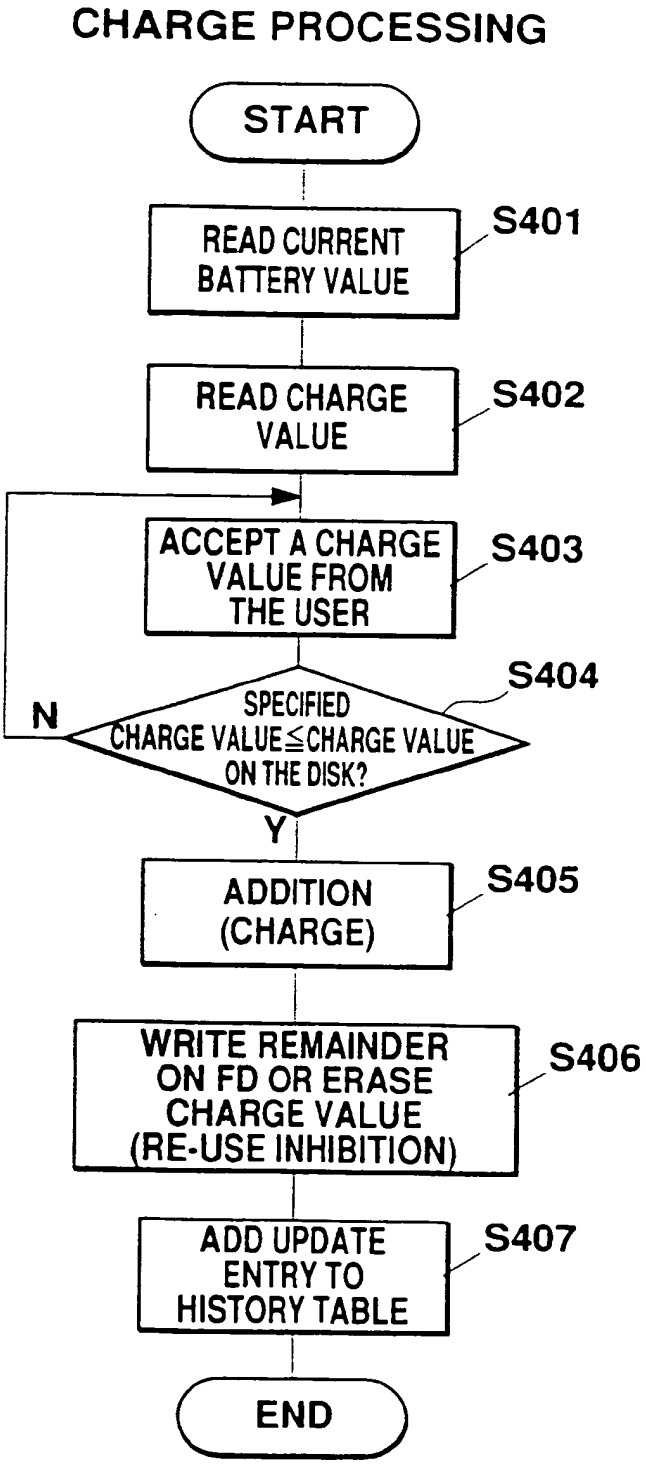


Fig. 9

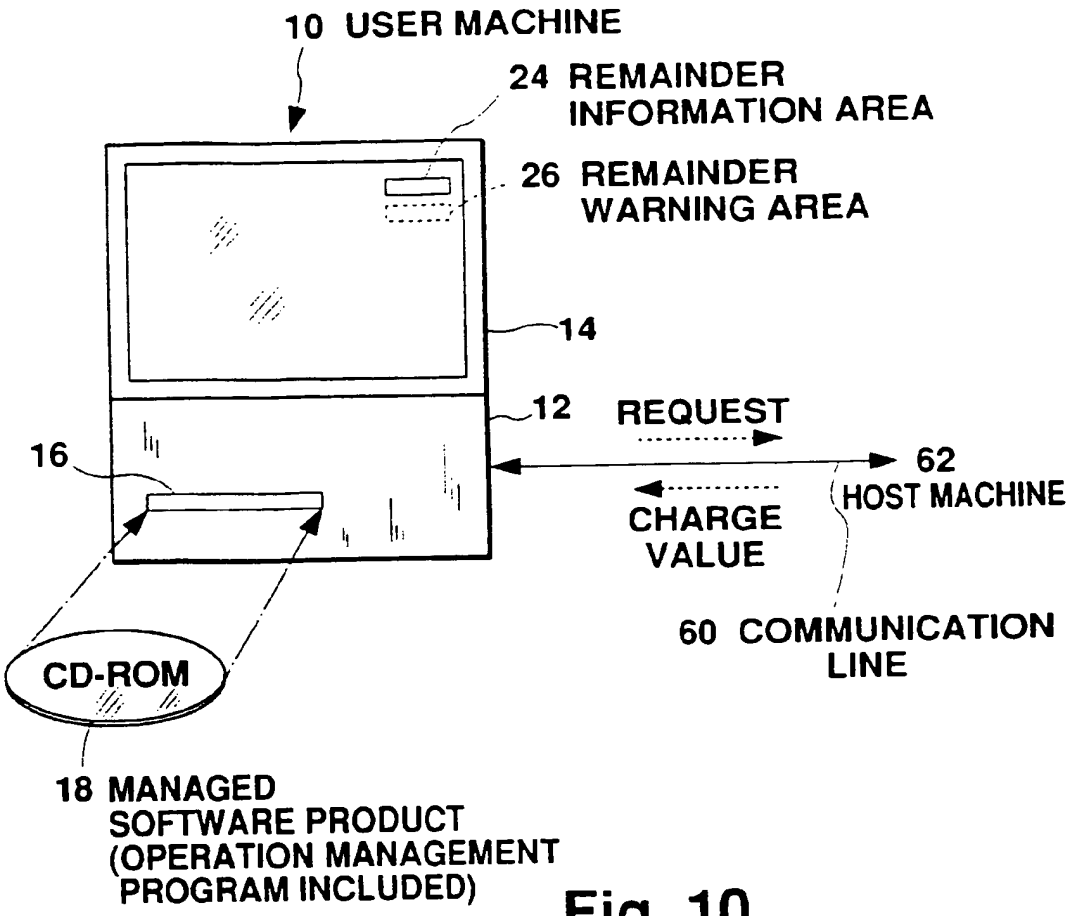


Fig. 10

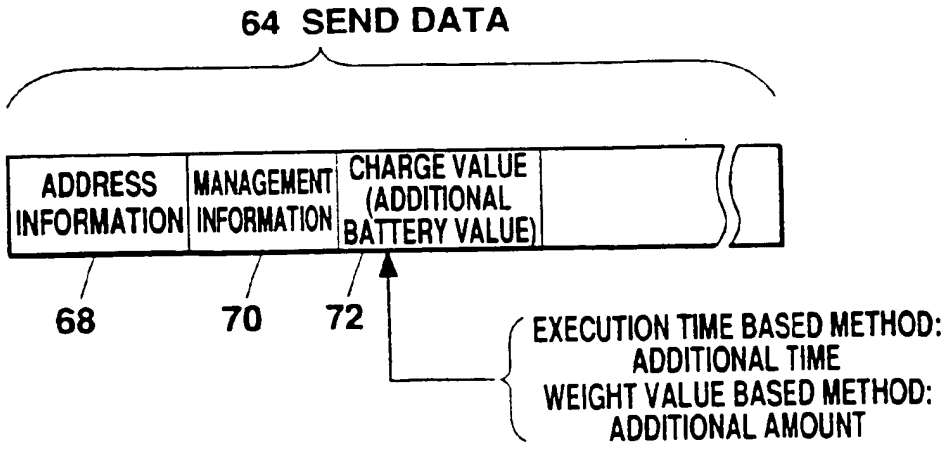
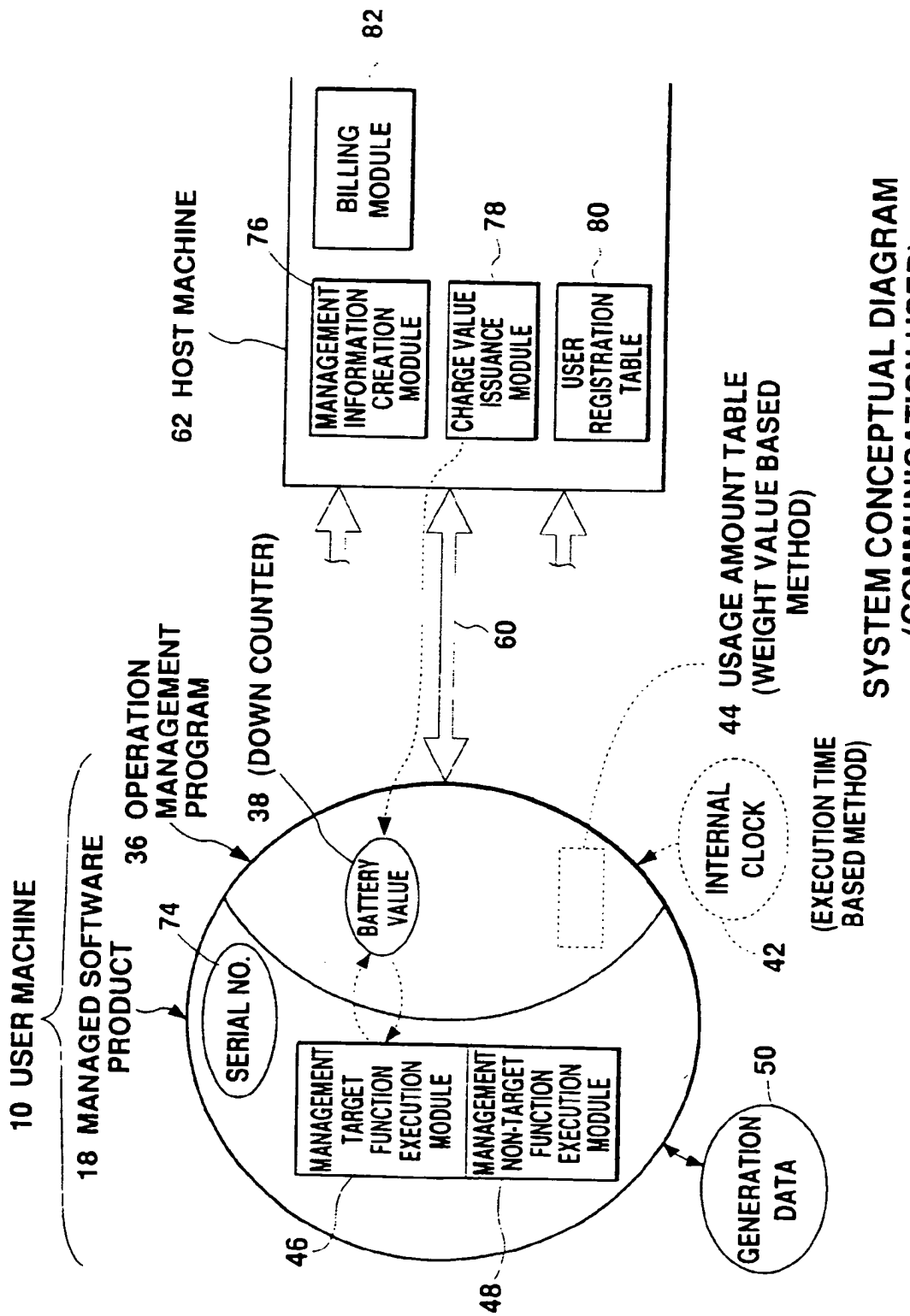


Fig. 11



SYSTEM CONCEPTUAL DIAGRAM (COMMUNICATION USED)

Fig. 12

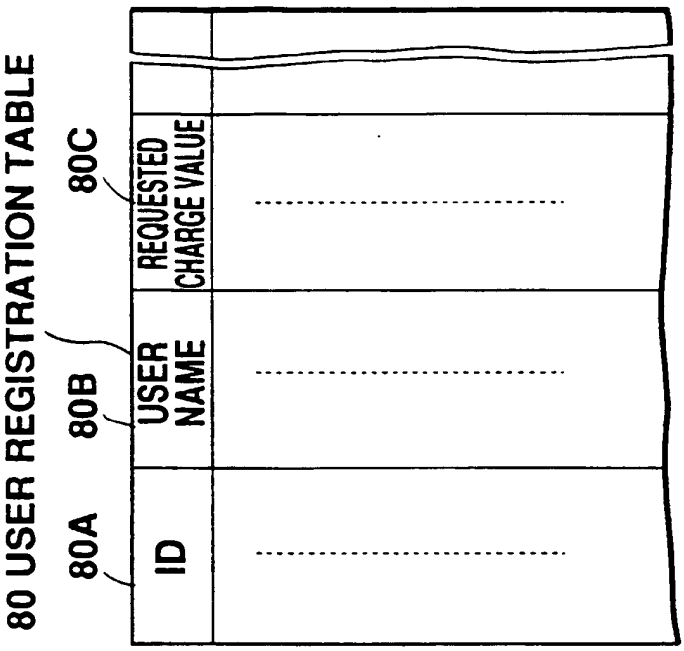


Fig. 13

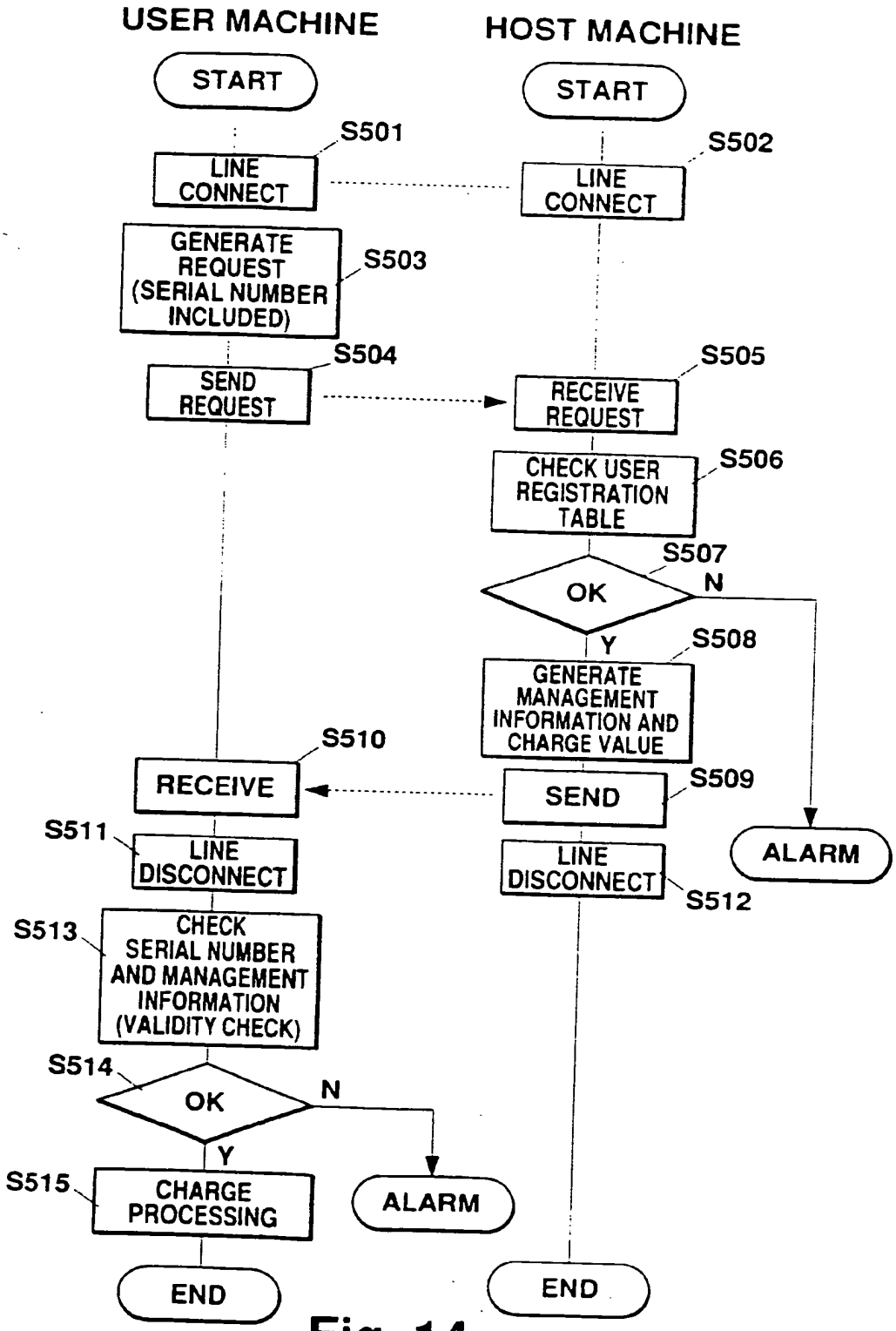


Fig. 14

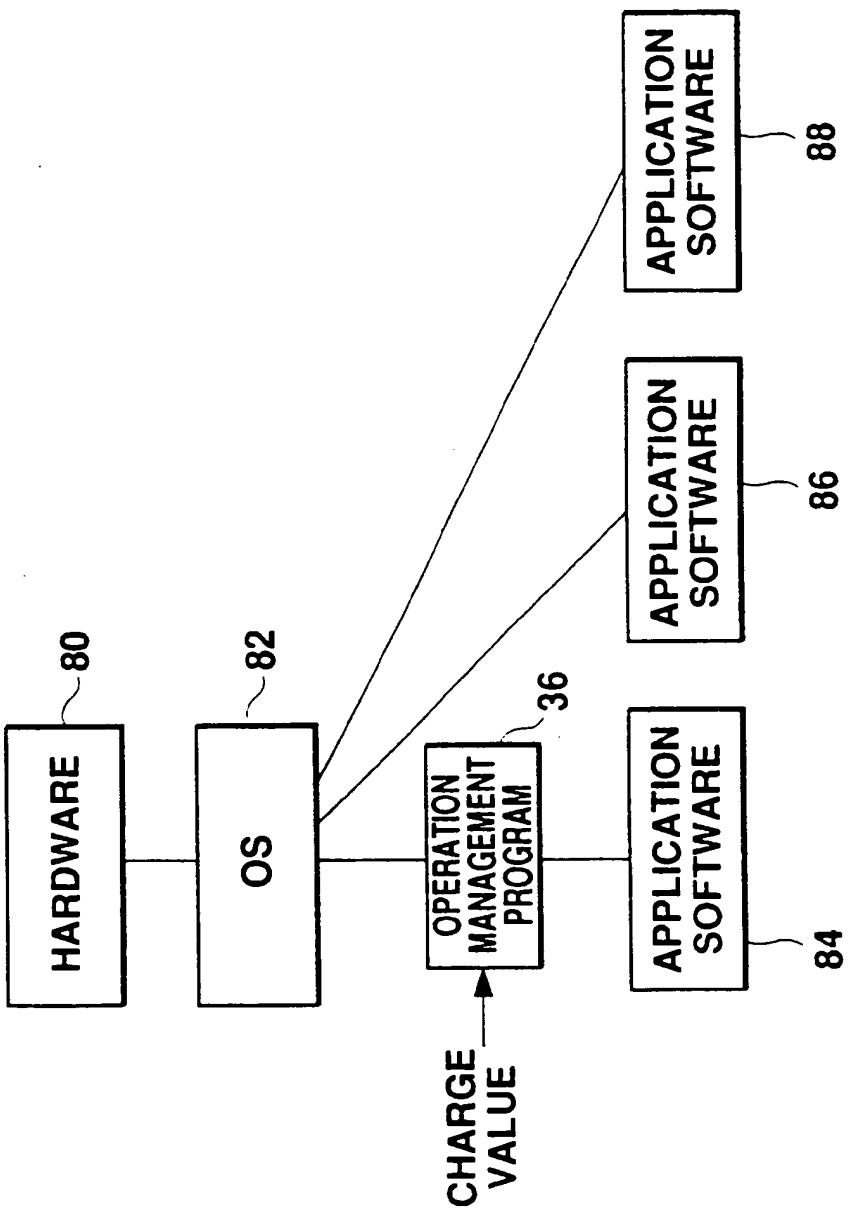


Fig. 15

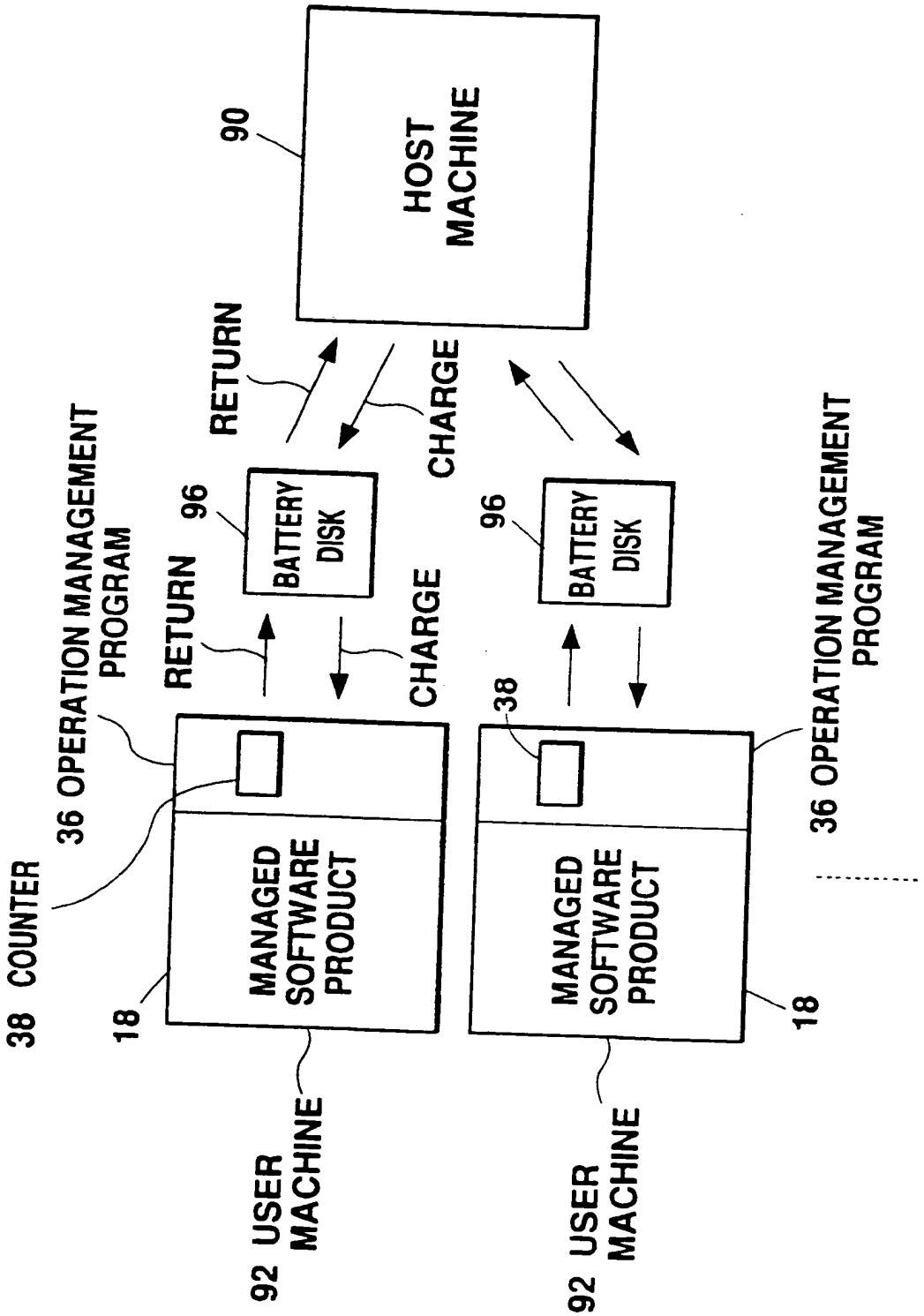


Fig. 16

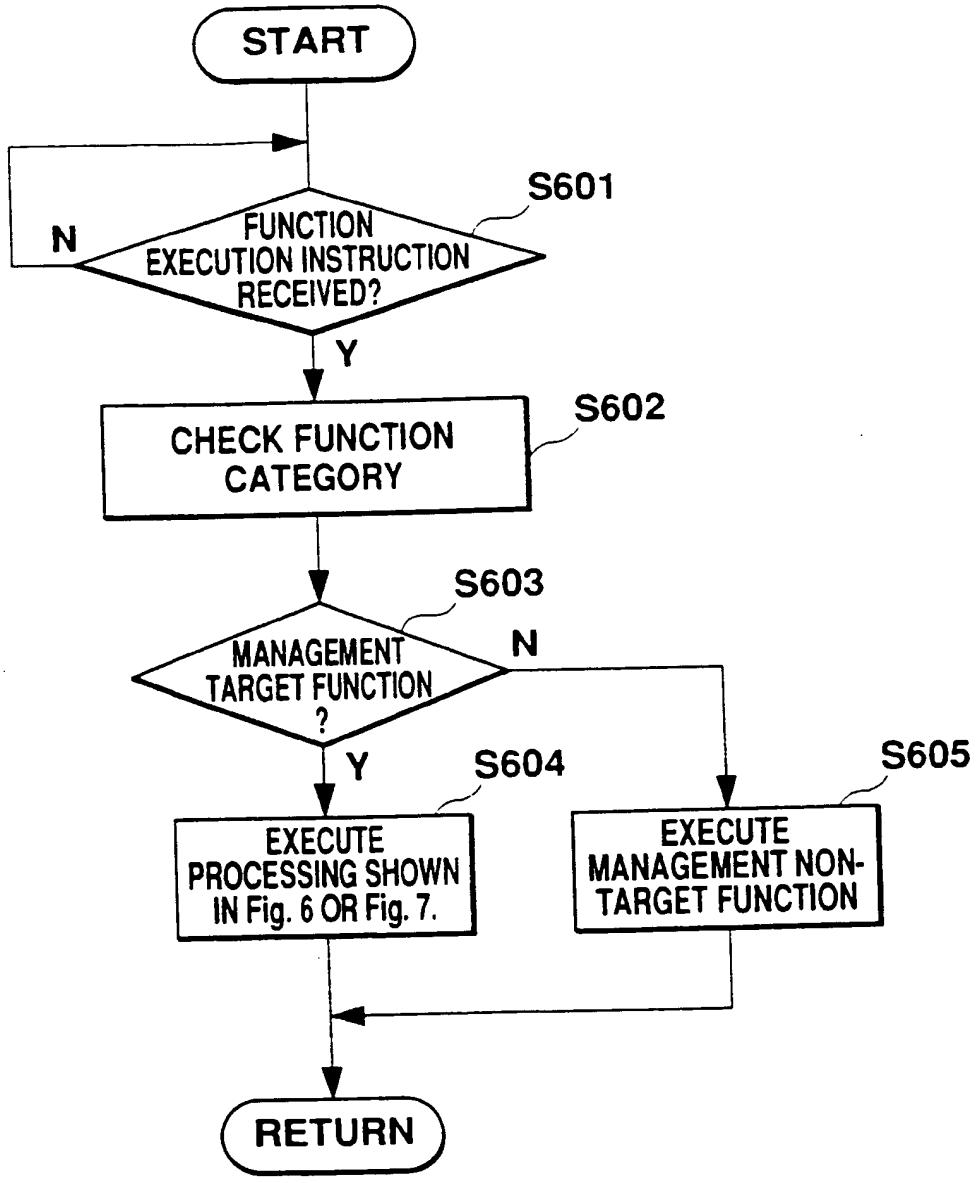


Fig. 17

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

(11) EP 0 818 748 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
15.11.2000 Bulletin 2000/46

(43) Date of publication A2:
14.01.1998 Bulletin 1998/03

(21) Application number: **97304946.3**

(22) Date of filing: **07.07.1997**

(51) Int Cl.[7]: **G06F 17/60**, G06F 1/00

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: **08.07.1996 JP 17813096**
21.05.1997 JP 13062697

(71) Applicant: **Murakoshi, Hiromasa**
Koriyama-shi, Fukushima, 963 (JP)

(72) Inventor: **Kanno, Kazuhiro**
Koriyama-shi, Fukushima, 963-02 (JP)

(74) Representative:
Cross, Rupert Edward Blount et al
BOULT WADE TENNANT,
Verulam Gardens
70 Gray's Inn Road
London WC1X 8BT (GB)

(54) Software management system and method

(57) An operation management system for managing the operation of a managed software product. When a management target function is executed, reference is made to a battery value and, if the value is zero or greater, the function is allowed to be executed. The battery value is decremented as the function is executed. A charge value is supplied on a charge disk, such as a floppy disk, to allow the user to increase the battery value and to extend the usage period of the managed software product. The charge value may be supplied over a communication line.

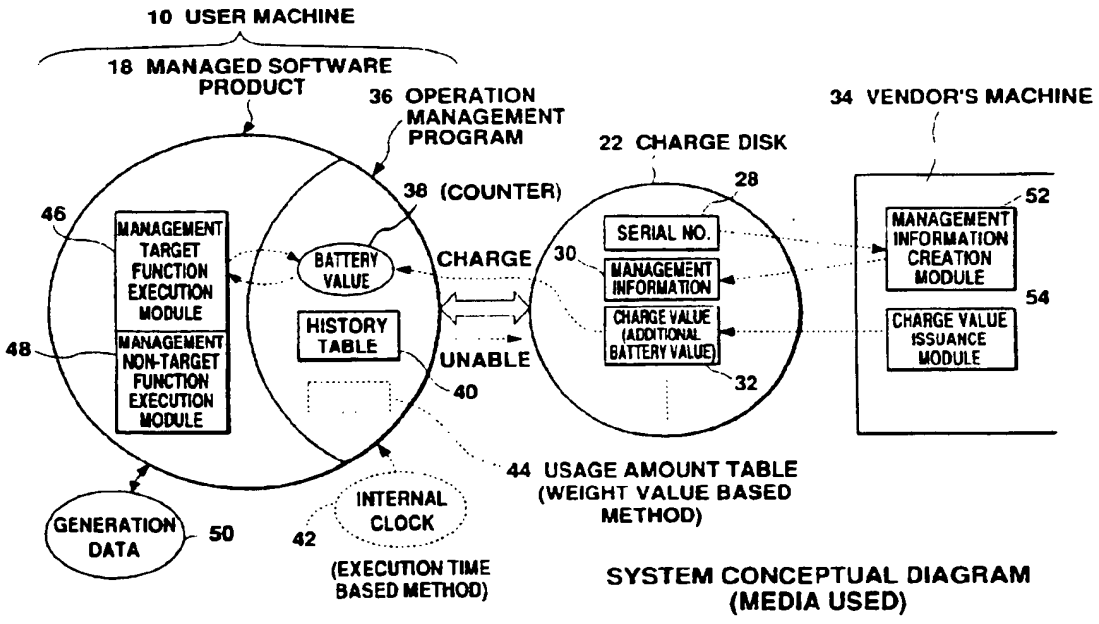


Fig. 3

EP 0 818 748 A3

Printed by Jouve, 75001 PARIS (FR)

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 97 30 4946

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	US 5 047 928 A (WIEDEMER JOHN D) 10 September 1991 (1991-09-10) * abstract; claims 1-5; figures 1,2 * * column 2, line 35 - column 3, line 6 * * column 4, line 22 - column 14, line 16 *	1-20	G06F17/60 G06F1/00
X	US 5 410 598 A (SHEAR VICTOR H) 25 April 1995 (1995-04-25) * abstract; claims 1-12; figures 1-6 * * column 3, line 5 - column 5, line 42 * * column 9, line 25 - line 46 *	1-20	
A	FR 2 697 358 A (GENTRALP INTERNATIONAL BV) 29 April 1994 (1994-04-29) * abstract; claims 1-3; figure 2 * * page 1, line 29 - page 2, line 38 * * page 5, line 24 - line 39 *	1-20	
A	EP 0 679 979 A (IBM) 2 November 1995 (1995-11-02) * abstract; figures 1,4,7,15 * * column 15, line 39 - column 17, line 42 *	1-20	
			TECHNICAL FIELDS SEARCHED (Int.Cl.6) G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	27 September 2000	Gardiner, A

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)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 97 30 4946

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

27-09-2000

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5047928 A	10-09-1991	US 4796181 A	03-01-1989
		CA 1281418 A	12-03-1991
		EP 0265183 A	27-04-1988
		JP 63191228 A	08-08-1988
		US 5155680 A	13-10-1992
US 5410598 A	25-04-1995	US 5272750 A	21-12-1993
		US 5050213 A	17-09-1991
		US 4977594 A	11-12-1990
		US 4827508 A	02-05-1989
		AT 133305 T	15-02-1996
		DE 3751678 D	29-02-1996
		DE 3751678 T	14-11-1996
		EP 0329681 A	30-08-1989
		WO 8802960 A	21-04-1988
FR 2697358 A	29-04-1994	NONE	
EP 0679979 A	02-11-1995	US 5689560 A	18-11-1997
		AU 1485695 A	02-11-1995
		BR 9501522 A	21-11-1995
		CA 2145925 A,C	26-10-1995
		CN 1115059 A	17-01-1996
		JP 7295803 A	10-11-1995
		KR 200444 B	15-06-1999

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/162,212	06/05/2002	Xin Wang	111325-104	3700

22204 7590 02/18/2005

NIXON PEABODY, LLP
401 9TH STREET, NW
SUITE 900
WASHINGTON, DC 20004-2128

EXAMINER	
AUGUSTIN, EVENS J	

ART UNIT	PAPER NUMBER
3621	

DATE MAILED: 02/18/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/162,212	WANG ET AL.	
	Examiner	Art Unit	
	Evens Augustin	3621	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 June 2002*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-28* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-28* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *8/30/2002*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

Status of Claims

1. Claims 1-28 have been examined.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) The invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

3. Claims 1-13, 15-18 and 20-28 are rejected under 35 U.S.C. 102(b) as being anticipated by Downs et al. (U.S 6,226,618).

As per claims 1-13, 15-18 and 20-28, Downs et al. discloses a system for Electronic Content Delivery, comprising of:

- Usage rights and other downstream rights (column 9, lines 33-35, column 10, lines 15-18). The system also presents the digital content offering to consumers (column 48, lines 32-36). The digital content owners receive requests from distributors to sale digital content (column 42, lines 65-67, column 43, lines 1-2). The content owners then license the original content to the distributors (Electronic Digital Content Store) (column 9, lines 5-13) – *Claims 1,3, 20*
- The content suppliers specify usage rights. They also specify that the distributors can customized those original usage rights and provide secondary usage rights (column 9, lines 33-36 - column 10, lines 13-18) - *Claim 2*

- Receiving a request for usage rights of digital content from a second consumer or end-user (column 18, step 136). The content stores can offer contents with their own customized usage conditions, as long as the customized conditions don't invalidate the original conditions set by the content provider (column 21, lines 23-36). Those secondary usage rights are then validated (column 21, lines 36-51) – *Claim 4, 21*
- The Secure Digital Content Electronic Distribution system uses multiple formats of digital content such as pictures, movies, videos, music, programs, multimedia and games (column 6, lines 45-48) – *Claim 5*
- The digital content owners receive requests from distributors to sale digital content (column 42, lines 65-67, column 43, lines 1-2). The content owners then license the original content to the distributors (Electronic Digital Content Store) (column 9, lines 5-13). The content stores offer their own customized usage conditions to end-users, as long as the customized conditions don't invalidate the original conditions set by the content provider (column 21, lines 23-36). The end-users don't get a license until the conditions are validated/authenticated throughout the supply chain (column 22, lines 26-52) – *Claim 6*
- The license comprises of transaction data (column 24, lines 37-38). The transaction data includes unique transaction ID (column 23, line 62). The license also comprises of digital signature (column 24, lines 45-47) and usage conditions (column 24, line 10-11) – *Claim 7*
- The first supplier as the content proprietor (column 9, lines 5-15). The first consumers are distributors such as electronic content stores (column 9, lines 63-65) – *Claims 8-9*

- Usage rights attached to contents offered to consumers (column 21, lines 30-33). The system verifies that the supplier has the authority to distribute the content (column 22, lines 45-50), and the supplier has the right to receive content (column 42, lines 52-55). The system also verifies that the requested usage conditions are in agreement with the allowable conditions (column 26, lines 20-23). If the conditions are not valid, the system terminates the request (column 10, lines 65-67). Otherwise, the system validates and sends license authorization to consumer (column 10, lines 60-64) – *Claims 10-12*
- If license is not validated or approved, the system determines if the user is entitled to the content, then authenticates and retransmits the content the user(negotiation) (column 48, lines 1-25) – *Claim 13*
- The supplier of digital content has an interface for multi users (column 49, lines 13-17). The supplier also generates the content for distribution (column 9, lines 15-20), with usage conditions (column 9, line 33), kept in a database (figure 1A, item 60). The consumer device also has a user interface (column 20, line 120), a license database (figure 1D, item 197) and is interconnected to a public network such as the internet (column 23, lines 5-15) – *Claim 15*
- Offer templates containing predetermined usage and meta-rights (column 26, line 62). The system also keeps the identity of the purchaser (column 20, lines 36-37) – *Claim 16*
- The consumer device has the ability to provide data/digital content originated from the content provider (column 79, lines 35-41) – *Claim 17*
- The end user devices such personal computers (column 79, line 16-17) and the packaged application provide means for the user to accept digital content (column 80, lines 20-25).

The system also provides means to specify and apply usage rights and to authenticate those rights (column 42, lines 35-56) – *Claim 18*

- The system uses watermarks to embed copy/play codes within the data (column 21, lines 64-65). For example, the watermarks keep track of the number of copies usage condition for the digital content, and if the number has been exhausted the, the system will not perform a particular request (column 7, lines 41-55) - *Claims 22-28*

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Downs et al. (U.S 6,226,618).

As per claims 14, Downs et al. discloses a system comprising of:

- The system currently uses audio data as an example and specifies usage rights accordingly (column 59, Lines 37-67).
- The system also supports other types of digital content such as pictures, movies, videos, music, programs, multimedia and games (column 6, lines 45-48).

Downs et al. did not explicitly describe a system that wherein the usage rights are associated with copy, transfer, loan, play, print, back-up, restore, delete, extract embed, edit,

authorize, install/un-install. However, Downs et al. discloses a system that supports digital content such as pictures, movies, videos, music, programs, multimedia and games (column 6, lines 47-48). Therefore, it would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to specify usage conditions for a particular digital content in order to include the rights of as many digital content formats as possible.

6. Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over Downs et al. (U.S 6,226,618) in view of Hitson et al. (US 20020010759)

As per claim 19, Downs et al. discloses a system comprising of:

- Usage rights and other downstream rights (column 9, lines 33-35, column 10, lines 15-18). The system also presents the digital content offering to consumers (column 48, lines 32-36). The digital content owners receive requests from distributors to sale digital content (column 42, lines 65-67, column 43, lines 1-2). The content owners then license the original content to the distributors (Electronic Digital Content Store) (column 9, lines 5-13)
- The content suppliers specify usage rights. They also specify that the distributors can customized those original usage rights and provide secondary usage rights (column 9, lines 33-36 - column 10, lines 13-18)
- Receiving a request for usage rights of digital content from a second consumer or end-user (c18, step 136). The content stores can then can offer content with their own customized usage conditions, as long as the customized conditions don't invalidate the original conditions set by the content provider (column 21, lines 23-36). The stores send

those usage conditions to the user and a clearinghouse for validation (column 21, lines 36-51)

Downs et al. did not explicitly describe a system in which conditions are filtered and applied, based on user preferences. However, Hitson et al. discloses a system and method for content distribution in which content is selected based on user preferences (page 1, paragraph 11). Therefore, it would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to design a system in which contents are filtered and applied, based on user preferences. It would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to design such system because it would provide personalized content delivery, and would provide economic incentives to content providers by supplying a means of target marketing to users based upon user content preferences page 1, paragraph 11).

Conclusion

7. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Danieli (US 6510513)

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Evens Augustin whose telephone number is 703-305-0267. The examiner can normally be reached on Monday thru Friday 8 to 5 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jim Trammel can be reached on 703-305-9768. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Any response to this action should be mailed to:

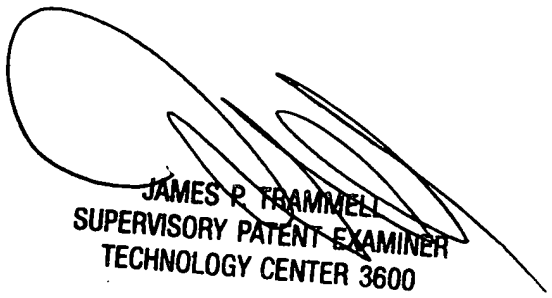
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Or faxed to:

(703) 305 – 5532 (for formal communications intended for entry and after-final communications), or (703) 746-5532 (for informal or draft communications, please label "PROPOSED" of "DRAFT")

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 308-1113.

Evens J. Augustin
October 28, 2004
Art Unit 3621



Form PTO-1449 (Rev. 8-83) Department of Commerce Patent and Trademark Office	Atty Docket 111325-104	Serial No. 10/162,212
INFORMATION DISCLOSURE STATEMENT	Applicants: Xin WANG	
	Filing Date: June 05, 2002	Group Art Unit:

U.S. PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date (if appropriate)
EA	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 [signature] Date Considered 1/27/05

*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.

08/27/2002

Form PTO-1449 (Rev. 8-83)	Department of Commerce Patent and Trademark Office	Atty Docket 111325-404	Serial No. 10/162,212
INFORMATION DISCLOSURE STATEMENT		Applicants: Xin WANG	
		Filing Date: June 05, 2002	Group Art Unit:

U.S. PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date (if appropriate)
CA	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 [signature] of [signature] Date Considered 1/27/05

*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.

08/27/2002

Form PTO-1449 (Rev. 8-83)	Department of Commerce Patent and Trademark Office	Atty Docket 111325-704	Serial No. 10/162,212
INFORMATION DISCLOSURE STATEMENT		Applicants: Xin WANG	
		Filing Date: June 05, 2002	Group Art Unit:

OIPE JC29 AUG 30 2002 PATENT & TRADEMARK OFFICE

U.S. PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date (if appropriate)
EP	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 EV — Date Considered 1/27/05

*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.

08/27/2002

Form PTO-1449 (Rev. 8-83)	Department of Commerce Patent and Trademark Office	Atty Docket 11132[illegible]04	Serial No. 10/162,212
INFORMATION DISCLOSURE STATEMENT		Applicants: Xin WANG	
		Filing Date: June 05, 2002	Group Art Unit:

U.S. PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date (if appropriate)
CS	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
CS	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	
CS	"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 [signature] Date Considered 1/27/05

*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.

08/27/2002

Form PTO-1449 (Rev. 8-83)	U.S. Department of Commerce Patent and Trademark Office	Atty Docket 11132[illegible]4	Serial No. 10/162,212
INFORMATION DISCLOSURE STATEMENT		Applicants: Xin WANG	
		Filing Date: June 05, 2002	Group Art Unit:

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

Examiner Initial	
GA	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 [signature]	Date Considered 1/27/05

*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 111325-[illegible]	Serial No. 10/162,212
INFORMATION DISCLOSURE STATEMENT	**Applicants:** Xin WANG	
	Filing Date: June 05, 2002	**Group Art Unit:**

Examiner [signature]	Date Considered 1/22/05

*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.

Notice of References Cited	Application/Control No. 10/162,212	Applicant(s)/Patent Under Reexamination WANG ET AL.	
	Examiner Evens Augustin	Art Unit 3621	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-5,671,412	09-1997	Christiano, Matt	707/104.1
	B	US-5,790,664	08-1998	Coley et al.	709/203
	C	US-5,925,127	07-1999	Ahmad, Arshad	713/200
	D	US-6,009,401	12-1999	Horstmann, Cay S.	705/1
	E	US-6,056,786	05-2000	Rivera et al.	717/168
	F	US-6,056,786	05-2000	Rivera et al.	717/168
	G	US-6,226,618	05-2001	Downs et al.	705/1
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	Hitson et al. - Patent Publication (US 20020010759)
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Bib Data Sheet

CONFIRMATION NO. 3700

SERIAL NUMBER	FILING DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
10/162,212	06/05/2002 RULE	705	3621	111325-104

APPLICANTS

Xin Wang, Torrance, CA;

Bijan Tadayon, Germantown, MD;

** CONTINUING DATA **********

This application is a CIP of 09/867,745 05/31/2001 PAT 6,754,642
which claims benefit of 60/296,113 06/07/2001
and claims benefit of 60/331,625 11/20/2001
and claims benefit of 60/331,624 11/20/2001

** FOREIGN APPLICATIONS **********

IF REQUIRED, FOREIGN FILING LICENSE GRANTED
** 06/28/2002

Foreign Priority claimed ☐ yes ☑ no 35 USC 119 (a-d) conditions met ☑ yes ☐ no ☐ Met after Allowance Verified and Acknowledged Examiner's Signature / Initials EA	STATE OR COUNTRY CA	SHEETS DRAWING 11	TOTAL CLAIMS 21	INDEPENDENT CLAIMS 3

ADDRESS
22204
NIXON PEABODY, LLP
401 9TH STREET, NW
SUITE 900
WASHINGTON , DC
20004-2128

TITLE
Rights offering and granting

FILING FEE	FEES: Authority has been given in Paper	☐ All Fees
RECEIVED 1014	No. _____ to charge/credit DEPOSIT ACCOUNT No. _____ for following:	☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue)

		☐ Other _____
		☐ Credit

Index of Claims

Application/Control No.	Applicant(s)/Patent under Reexamination
10/162,212	WANG ET AL.
Examiner	**Art Unit**
Evens Augustin	3621

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim		Date
Final	Original	1/27/05
	1	√
	2	√
	3	√
	4	√
	5	√
	6	√
	7	√
	8	√
	9	√
	10	√
	11	√
	12	√
	13	√
	14	√
	15	√
	16	√
	17	√
	18	√
	19	√
	20	√
	21	√
	22	√
	23	√
	24	√
	25	√
	26	√
	27	√
	28	√
	29	
	30	
	31	
	32	
	33	
	34	
	35	
	36	
	37	
	38	
	39	
	40	
	41	
	42	
	43	
	44	
	45	
	46	
	47	
	48	
	49	
	50	

Claim		Date
Final	Original	
	51	
	52	
	53	
	54	
	55	
	56	
	57	
	58	
	59	
	60	
	61	
	62	
	63	
	64	
	65	
	66	
	67	
	68	
	69	
	70	
	71	
	72	
	73	
	74	
	75	
	76	
	77	
	78	
	79	
	80	
	81	
	82	
	83	
	84	
	85	
	86	
	87	
	88	
	89	
	90	
	91	
	92	
	93	
	94	
	95	
	96	
	97	
	98	
	99	
	100	

Claim		Date
Final	Original	
	101	
	102	
	103	
	104	
	105	
	106	
	107	
	108	
	109	
	110	
	111	
	112	
	113	
	114	
	115	
	116	
	117	
	118	
	119	
	120	
	121	
	122	
	123	
	124	
	125	
	126	
	127	
	128	
	129	
	130	
	131	
	132	
	133	
	134	
	135	
	136	
	137	
	138	
	139	
	140	
	141	
	142	
	143	
	144	
	145	
	146	
	147	
	148	
	149	
	150	

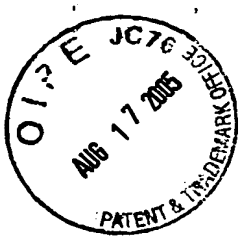
Search Notes

Application/Control No.	Applicant(s)/Patent under Reexamination	
10/162,212	WANG ET AL.	
Examiner	**Art Unit**	
Evens Augustin	3621	

SEARCHED			
Class	Subclass	Date	Examiner
705	1	1/27/2005	EA
705	51	1/27/2005	EA
705	54	1/27/2005	EA
705	75	1/27/2005	EA

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
Searched on EAST for Digital Rights Management	1/27/2005	EA



Docket No. 111325-104 (230300)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:	)	
Xin **WANG,** *et al.*	)	Examiner: Augustin, Evens J.
Serial No. 10/162,212	)	Group Art Unit: 3621
Filed: June 5, 2002	)	Confirmation No. 3700
For: **RIGHTS OFFERING AND GRANTING**	)	

U.S. Patent and Trademark Office
Customer Window, Mail Stop Non-Fee Amendment
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Sir:

AMENDMENT AFTER NON-FINAL REJECTION

In response to the non-final Office Action mailed **February 18, 2005**, and further to the personal interview conducted by Examiner Augustin on August 5, 2005, please amend the above-identified patent application, as follows.

08/19/2005 EFLORES 00000130 192380 10162212
02 FC:1202 150.00 DA

663769.1

Amendments to the Claims:

1. (Currently amended) A method for transferring usage rights adapted to be associated with items within a digital rights management system, said method comprising:

generating, by a supplier, at least one first offer ~~containing~~ including usage rights and meta-rights for the items, said usage rights defining a manner of use for the items, said meta-rights specifying rights to derive usage rights or other meta-rights for the items;

presenting, by the supplier, said offer to a first consumer in said system,

wherein the offer expresses what rights the consumer can acquire for the items;

receiving, by the supplier, a selection from the first consumer indicating desired usage rights and meta-rights; and [[,]]

generating, by the supplier, a first license granting to the first consumer the ~~desired~~ usage rights and meta-rights ~~to the first consumer~~ for the items,

wherein the first license grants the usage rights and meta-rights that are selected by the first consumer during the receiving step.

2. (Original) The method of claim 1, wherein said license specifies one or more conditions which must be satisfied in order for said usage right to be exercised and one or more conditions which must be satisfied in order for said meta-rights to be exercised.

3. (Original) The method of claim 1, further comprising the step of receiving a request for a license from the first consumer.

4. (Currently amended) The method of claim 1, further comprising:

receiving a request generated by a second consumer for a license ~~containing~~ including at least one of usage rights and meta-rights for the items;

generating, by a second supplier, a second offer ~~containing~~ including rights derived from said meta-rights ~~contained~~ included in the ~~second offer~~ first license, wherein the second supplier is the first consumer; and

generating, by [[a]] the second supplier, a second license ~~containing~~ including rights derived from said meta-rights ~~contained~~ included in the second offer ~~first license, wherein the second supplier is the first consumer~~.

5. (Original) The method of claim 1, wherein the item comprises digital content.

6. (Original) The method of claim 1, further comprising the steps of:
providing said first license as a customized draft license to the first consumer;
accepting a confirmation of said customized draft license from the first consumer; and
authenticating said draft license to create an authenticated license.

7. (Original) The method of claim 1, wherein said first license comprises a license identification, a digital signature, and at least one grant, said at least one grant including usage rights, meta-rights, a named principal designating the first consumer to whom rights are granted, and a condition list.

8. (Previously presented) The method of claim 1, wherein the first supplier is at least one of a provider, distributor, retailer, consumer, and a user.

9. (Previously presented) The method of claim 1, wherein the first consumer is at least one of a provider, distributor, retailer, consumer, and a user.

10. (Original) The method of claim 1, wherein the step of generating at least one offer comprises the steps of:
collecting usage rights and meta-rights available to be offered;
determining if the supplier has a right to offer the available usage rights and meta-rights;
terminating the generating of a set of offers, if a right to offer other usage and meta rights does not exist;
composing an offer based on available rights if the supplier has the right to offer other usage and meta rights; and
authenticating said offer.

11. (Original) The method of claim 10, wherein said composing step comprises:

determining if a consumer has requested an offer including specific usage rights and meta-rights;

applying the specific usage rights and meta-rights to the offer as a filter; and

determining if an offer template corresponds to the filtered offer and if so applying said offer template as an offer,

12. (Previously presented) The method of claim 6, wherein said step of generating a first license further comprises the steps of:

determining if the suppler has the right to grant the rights;

terminating the step of customizing a draft license, if the supplier does not have the right to grant the rights;

analyzing one or more choices received from the consumer;

determining if the choices are acceptable; and

creating a draft license based on the choices if the choices are acceptable.

13. (Currently amended) The method of claim 12, wherein said step of generating a first license further comprises: [[;]]

presenting the draft license to the consumer;

re-negotiating a license if the first license is not approved by the consumer; and

authenticating the draft license if the first consumer approves the draft license.

14. (Previously presented) The method of claim 1, wherein said usage rights specify rights to copy, transfer, loan, play, print, back-up, restore, delete, extract, embed, edit, authorize, install, or un-install the items.

15. (Currently amended) A system for transferring usage rights adapted to be associated with an item, to be licensed in multi-tier channels of distribution with downstream rights and conditions assigned [[at]] at least <u>at</u> one level, said system comprising:

a supplier component, comprising:

a supplier user interface module;

an offer generator module for generating an offer ~~containing~~ including at least usage rights and meta-rights for the item, the usage rights defining a manner of use for the item, the meta-rights specifying rights to derive usage rights or other meta-rights for the item;

a rights composer module for composing a draft license;

a repository for supplier's rights;

a supplier management database; and

a consumer component comprising:

a consumer user interface module;

an offer-consideration module configured to analyze the offers generated by the supplier component and select offers based on the analysis;

a repository for consumer's rights;

a consumer management database; and

a communication link coupling said supplier component and said consumer component,

wherein the rights composer module is configured to compose a license granting the usage rights and meta-rights that are selected by the offer-consideration module of the consumer component.

16. (Currently amended) A system as recited in claim 15, wherein the supplier component further comprises offer-templates and consumer profile information, wherein said offer-template ~~contains~~ includes one or more predetermined usage rights and meta-rights, and wherein said consumer profile information comprises at least one of consumer identity information, account information, purchase history information, consumer preferences information, and credit rating information.

17. (Original) A system as recited in claim 15, wherein said consumer component further comprises a supplier-preference module for providing supplier information.

18. (Original) The system of claim 15, wherein said offer-consideration module comprises:

means for determining if the consumer can accept an offer;

means for applying selection logic to the offer;

means for specifying contingencies; and

means for authenticating choices and providing the choices to said supplier component.

19. (Original) The system of claim 18, wherein said means for applying comprises:

means for parsing the offer and selecting preferred usage rights and meta-rights in the offer;

means for filtering offers based on supplier preferences;

means for applying consumer preferences; and

means for selecting options based on the output of said means for parsing, said means for filtering, and said means for applying consumer preferences.

20-28. (Cancelled)

29. (New) The method of claim 1, the method being for generating a license to digital content to be used within the system for at least one of managing use and distribution of the digital content, wherein the license permits the first consumer to exercise the at least one meta-right and permits the first consumer to offer at least one derived right from the at least one meta-right and generate a license including the at least one derived right.

30. (New) The method of claim 29, wherein the at least one derived right in the license is for a second consumer, the license includes usage rights to be exercised by the second consumer and meta-rights permitting derived rights to be offered to a third consumer.

31. (New) The method of claim 1, wherein said method is implemented with one or more hardware and/or software components configured to perform the steps of the method.

32. (New) The method of claim 1, wherein said method is implemented with one or more computer readable instructions embedded on a computer readable medium and configured to cause one or more computer processors to perform the steps of the method.

33. (New) The system of claim 15, wherein said system is implemented with one or more hardware and/or software components.

34. (New) The method of claim 1, wherein rights derived from said meta-rights include rights that revoke at least one of a usage right, and a meta-right.

35. (New) The method of claim 1, wherein rights derived from said meta-rights include rights that reduce or expand at least one of a usage right, and a meta-right.

36. (New) The system of claim 15, wherein rights derived from said meta-rights include rights that revoke at least one of a usage right, and a meta-right.

37. (New) The system of claim 15, wherein rights derived from said meta-rights include rights that reduce or expand at least one of a usage right, and a meta-right.

38. (New) A method for transferring usage rights adapted to be associated with an item within a digital rights management system, the method being performed by a consumer device within the system, the method comprising:

receiving, by the consumer device, from a supplier device within the system at least one offer including usage rights and meta-rights for the item,

wherein the usage rights define a manner of use for the item, and the meta-rights specify rights to derive usage rights or other meta-rights for the item, and

the offer expresses what rights the consumer device can acquire for the item;

selecting, by the consumer device, desired usage rights and meta-rights from the received offer,

wherein the selected rights express what rights the consumer device desires to acquire for the item; and

receiving, by the consumer device, a license from the supplier device,

wherein the received license grants the usage rights and meta-rights that are selected and provided by the consumer device.

39. (New) The method of claim 38, wherein said method is implemented with one or more hardware and/or software components configured to perform the steps of the method.

40. (New) The method of claim 38, wherein said method is implemented with one or more computer readable instructions embedded on a computer readable medium and configured to cause one or more computer processors to perform the steps of the method.

REMARKS

The following remarks are submitted to be fully responsive to the non-final Office Action of **February 18, 2005**. Reconsideration and allowance of this application are respectfully requested. Claims 1-19, and 29-40 are now pending in the application, with claims 1, 4, 13, 15, and 16 amended, with claims 20-28 cancelled, and with claims 29-40 added. No new matter is introduced (see, e.g., claims 1-28, as originally filed and as previously presented, Applicants' Published Specification, FIG. 4, and paragraphs [0010], [0040], [0041], [0046], [0059], and [0062]).

First, Applicants wish to thank Examiner Augustin for extending the courtesy of a personal interview with Applicants' representatives on August 5, 2005. During the interview, the claims, as substantially presented herewith, were discussed, and which patentably distinguish over the applied references, U.S. Patent No. 6,226,618 to *Downs et al.*, and U.S. Patent Application No. 20020010759 to *Hitson et al.*, as further set forth herein.

Referring now to the present Office Action, claims 1-13, 15-18 and 20-28 were rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 6,226,618 to *Downs et al.*, claim 14 was rejected under 35 U.S.C. §103(a) as being unpatentable over *Downs et al.*, and claim was rejected under 35 U.S.C. §103(a) as being unpatentable over *Downs et al.* in view of U.S. Patent Application No. 20020010759 to *Hitson et al.* Claims 1-28 are patentably distinguishable over *Downs et al.* and *Hitson et al.*, because *Downs et al.* and *Hitson et al.*, alone or in combination, fail to disclose, teach or suggest all of the features recited in the present claims. For example, independent claim 1, as amended, (emphasis added) recites:

> A method for transferring usage rights adapted to be associated with items within a digital rights management system, said method comprising:
>
> **generating, by a supplier, at least one first offer including usage rights and meta-rights for the items, said usage rights defining a manner of use for the items, said meta-rights specifying rights to derive usage rights or other meta-rights for the items;**
>
> presenting, by the supplier, said offer to a first consumer in the system,
>
> wherein the offer expresses what rights a consumer can acquire for the items;
>
> **receiving, by the supplier, a selection of from the first consumer indicating desired usage rights and meta-rights;** and
>
> generating, by the supplier, a first license granting to the first consumer the usage rights and meta-rights for the items,

wherein said first license grants the usage rights and meta-rights that are selected by the first consumer during said receiving step; and

independent claim 15, as amended, (emphasis added) recites:

A system for transferring usage rights adapted to be associated with an item, to be licensed in multi-tier channels of distribution with downstream rights and conditions assigned at least at one level, said system comprising:

a supplier component, comprising:

a supplier user interface module;

an offer generator module for generating an offer including at least usage rights and meta-rights for the item, the usage rights defining a manner of use for the item, the meta-rights specifying rights to derive usage rights or other meta-rights for the item;

a rights composer module for composing a draft license;

a repository for supplier's rights;

a supplier management database; and

a consumer component comprising:

a consumer user interface module;

an offer-consideration module configured to analyze the offers generated by the supplier component and select offers based on the analysis;

a repository for consumer's rights;

a consumer management database; and

a communication link coupling said supplier component and said consumer component,

wherein the rights composer module is configured to compose a license granting the usage rights and meta-rights that are selected by the offer-consideration module of the consumer component; and

new independent claim 38 (emphasis added) recites:

A method for transferring usage rights adapted to be associated with an item within a digital rights management system, the method being performed by a consumer device within the system, the method comprising:

receiving, by the consumer device, from a supplier device within the system at least one offer including usage rights and meta-rights for the item,

wherein the usage rights define a manner of use for the item, and the meta-rights specify rights to derive usage rights or other meta-rights for the item, and

the offer expresses what rights the consumer device can acquire for the item;

selecting, by the consumer device, desired usage rights and meta-rights from the received offer,

wherein the selected rights express what rights the consumer device desires to acquire for the item; and

receiving, by the consumer device, a license from the supplier device,

wherein the received license grants the usage rights and meta-rights that are selected and provided by the consumer device.

By contrast, *Downs et al.* is directed to a method and apparatus of securely providing data to a user's system, wherein the data is encrypted so as to only be decryptable by a data decrypting key, the data decrypting key being encrypted using a first public key, and the encrypted data being accessible to the user's system. The method includes transferring the encrypted data decrypting key to a clearing house that possesses a first private key, which corresponds to the first public key; decrypting the data decrypting key using the first private key; re-encrypting the data decrypting key using a second public key; transferring the re-encrypted data decrypting key to the user's system, the user's system possessing a second private key, which corresponds to the second public key; and decrypting the re-encrypted data decrypting key using the second private key. However, *Downs et al.* fails to disclose, teach or suggest the noted features recited in independent claims 1, 15 and 38.

Specifically, although the cited portions of *Downs et al.* in the present Office Action may generally pertain to usage conditions, such as wholesale price, retail price, copy restrictions, etc., *Downs et al.* fails to disclose, teach or suggest usage rights in the manner claimed, much less meta-rights which govern the exercising of other usage rights or meta-rights, as recited in independent claims 1, 15, and 38. Further, *Downs et al.* fails to disclose, teach or suggest the noted features, e.g., generating or receiving of an offer, selecting or receiving a selection of desired usage rights and meta-rights from the offer, generating a license based on the selected right, etc., as recited independent claims 1, 15, and 38.

Hitson et al. is directed to a system and method for allowing multimedia content to be delivered to a computer, personal desktop assistant, portable media player, or other electronic device, wherein the content may include advertisements, and distribution of content may emulate current television and/or radio broadcasts, the content may be encrypted or otherwise secured, and such security may restrict use of some content, including limiting content to specific devices, specific users, or a predefined number of playbacks. The users may also indicate a preference for a particular content type or types, and content may be chosen based on user preferences. Users may further refine content preferences as content is experienced, and may indicate a desire to purchase content or find out more about specific content, and such desires may be recorded for later review or action. However, *Hitson et al.* fails to cure the noted deficiencies in *Downs et al.* Accordingly, *Downs et al.* and *Hitson et al.*, alone or

in combination, fail to disclose, teach or suggest the noted features recited in independent claims 1, 15 and 38.

The invention recited in independent claims 1, 15 and 38 and claims dependent therefrom recognizes and solves the following problems:

> **[0010]** However, there are limitations associated with the above-mentioned paradigms wherein only usage rights and conditions associated with content are specified by content owners or other grantors of rights. Once purchased by an end user, a consumer, or a distributor, of content along with its associated usage rights and conditions has no means to be legally passed on to a next recipient in a distribution chain. Further the associated usage rights have no provision for specifying rights to derive other rights, i.e. rights to modify, transfer, offer, grant, obtain, transfer, delegate, track, surrender, exchange, transport, exercise, revoke, or the like. Common content distribution models often include a multi-tier distribution and usage chain. Known DRM systems do not facilitate the ability to prescribe rights and conditions for all participants along a content distribution and usage chain. Therefore, it is difficult for a content owner to commercially exploit content unless the owner has a relationship with each party in the distribution chain.

The invention recited in independent claims 1, 15 and 38 and claims dependent therefrom provides the following advantages:

> **[0063]** Overall, as can be seen in the description of FIGS. 6, 7, and 8 above, the consumer sends a request, and then a license is constructed. Either the supplier or the consumer could draft the content of the license, but in the example above the supplier does so. The request is a subset of an offer and the offer has one or more options. The supplier makes the offer available to the consumer sending the request (and to other consumers if that is the desire), and the consumer (including other consumers, if applicable) makes choices. Then, the supplier analyzes the choices, and constructs the license (i.e. a grant of rights). Note that the request can also be rejected, or a counter proposal could be made and the same process could then repeat for the counter proposal.

By contrast, *Downs et al.* and *Hitson et al.*, alone or in combination, fail disclose, teach or suggest the noted features, fail to recognize or solve the noted problems, and fail to provided the advantages thereof.

The dependent claims are allowable on their on merits and for at least the reasons as argued above with respect to independent claims 1, 15 and 38.

The prior art that has been cited, but not applied by the Examiner, has been taken into consideration during formulation of this response. However, since this art was not considered by the Examiner to be of sufficient relevance to apply against any of the claims, no detailed comments thereon are believed to be warranted at this time.

In view of the foregoing, it is submitted that the present application is in condition for allowance and a notice to that effect is respectfully requested. However, if the Examiner deems that any issue remains after considering this response, the Examiner is invited to contact the undersigned attorney to expedite the prosecution and engage in a joint effort to work out a mutually satisfactory solution.

Respectfully submitted,

NIXON PEABODY, LLP

Carlos R. Villamar
Reg. No. 43,224

NIXON PEABODY LLP
CUSTOMER NO.: 22204
401 9th Street, N.W., Suite 900
Washington, DC 20004
Tel: 202-585-8000
Fax: 202-585-8080

3621

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin WANG, *et al.*
Group Art Unit	3621
Examiner Name	Evens J. Augustin
Total Number of Pages in This Submission	
Attorney Docket Number	111325-104 (230300)

ENCLOSURES *(check all that apply)*

- ☒ Fee Transmittal Form
  - ☐ 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)
- ☐ Response to Missing Parts/ Incomplete Application
  - ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Assignment Papers *(for an Application)*
- ☐ Drawing(s)
- ☐ 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
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to Group (Appeal Notice, Brief, Reply Brief)
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Application Data Sheet
- ☐ Request for Corrected Filing Receipt with Enclosures
- ☐ A self-addressed prepaid postcard for acknowledging receipt
- ☐ Other Enclosure(s) (*please identify below*):

Remarks

☒ 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	Carlos R. Villamar Registration No. 43,224 Nixon Peabody LLP 401 9th Street, N.W., Suite 900 Washington, D.C. 20004-2128
Signature	
Date	August 17, 2005

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(a)]

I hereby certify that this correspondence is being:

- ☐ deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop _____, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450
- ☐ transmitted by facsimile on the date shown below to the United States Patent and Trademark Office at (703) _____.

_____ Date

_____ Signature

_____ Typed or printed name

FEE TRANSMITTAL FOR FY 2005

Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT | **$1,170.00**

Complete if Known	
Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin WANG, *et al.*
Examiner Name	Evens J. Augustin
Art Unit	3621
Attorney Docket No.	111325-104 (230300)

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☐ Credit Card ☐ Money Order ☐ Other ☐ None

☒ Deposit Account:

Deposit Account Number: 19-2380

Deposit Account Name: Nixon Peabody LLP

The Commissioner is authorized to: *(check all that apply)*

☒ Charge fee(s) indicated below ☒ Credit any overpayments

☒ Charge any additional fee(s)

☐ Charge fee(s) indicated below, **except for the filing fee** to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	300	2001	150	Utility filing fee	
1002	200	2002	100	Design filing fee	
1003	200	2003	100	Plant filing fee	
1004	300	2004	150	Reissue filing fee	
1005	200	2005	100	Provisional filing fee	

SUBTOTAL (1) ($) 0

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

			Extra Claims		Fee from below		Fee Paid
Total Claims	28	-31** =	3	X	50.00	=	150.00
Independent Claims	3	-3** =		X		=	0
Multiple Dependent				X		=	0

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	50	2202	25	Claims in excess of 20
1201	200	2201	100	Independent claims in excess of 3
1203	360	2203	180	Multiple dependent claim, if not paid
1204	200	2204	100	** Reissue independent claims over original patent
1205	50	2205	25	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) $150.00

**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	
1051	130	2051	65	Surcharge – late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	120	2251	60	Extension for reply within first month	
1252	450	2252	225	Extension for reply within second month	
1253	1,020	2253	510	Extension for reply within third month	1,020.00
1254	1,590	2254	795	Extension for reply within fourth month	
1255	2,160	2255	1,080	Extension for reply within fifth month	
1401	500	2401	250	Notice of Appeal	
1402	500	2402	250	Filing a brief in support of an appeal	
1403	1,000	2403	500	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	500	2452	250	Petition to revive – unavoidable	
1453	1,500	2453	750	Petition to revive – unintentional	
1501	1,400	2501	700	Utility issue fee (or reissue)	
1502	800	2502	400	Design issue fee	
1503	1,100	2503	550	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	790	2809	395	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	790	2810	395	For each additional invention to be examined (37 CFR 1.129(b))	
1801	790	2801	395	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) $1,020.00

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(a)]

I hereby certify that this correspondence is being:

☐ deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop _____, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450

☐ transmitted by facsimile on the date shown below to the United States Patent and Trademark Office at (703) _____.

Date _____ Signature _____

Typed or printed name _____

SUBMITTED BY

				Complete *(if applicable)*	
Name *(Print/Type)*	Carlos R. Villamar	Registration No. *(Attorney/Agent)*	43,224	Telephone	(202) 585-8204
Signature				Date	August 17, 2005

SEND TO: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

3621

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin WANG, *et al.*
Group Art Unit	3621
Examiner Name	Evens J. Augustin
Total Number of Pages in This Submission	**Attorney Docket Number** 111325-104 (230300)

ENCLOSURES *(check all that apply)*

- ☒ Fee Transmittal Form
  - ☐ 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)
- ☐ Response to Missing Parts/ Incomplete Application
  - ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Assignment Papers *(for an Application)*
- ☐ Drawing(s)
- ☐ 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
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to Group (Appeal Notice, Brief, Reply Brief)
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Application Data Sheet
- ☐ Request for Corrected Filing Receipt with Enclosures
- ☐ A self-addressed prepaid postcard for acknowledging receipt
- ☐ Other Enclosure(s) (*please identify below*):

Remarks

☒ 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	Carlos R. Villamar Registration No. 43,224 Nixon Peabody LLP 401 9th Street, N.W., Suite 900 Washington, D.C. 20004-2128
Signature	
Date	August 17, 2005

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(a)]

I hereby certify that this correspondence is being:

- ☐ deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop _____, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450
- ☐ transmitted by facsimile on the date shown below to the United States Patent and Trademark Office at (703) _____.

_____ Date

_____ Signature

_____ Typed or printed name

FEE TRANSMITTAL FOR FY 2005

Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT | **$1,170.00**

Complete if Known	
Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin WANG, *et al.*
Examiner Name	Evens J. Augustin
Art Unit	3621
Attorney Docket No.	111325-104 (230300)

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☐ Credit Card ☐ Money Order ☐ Other ☐ None

☒ Deposit Account:

Deposit Account Number: 19-2380

Deposit Account Name: Nixon Peabody LLP

The Commissioner is authorized to: *(check all that apply)*

☒ Charge fee(s) indicated below ☒ Credit any overpayments

☒ Charge any additional fee(s)

☐ Charge fee(s) indicated below, **except for the filing fee** to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	300	2001	150	Utility filing fee	
1002	200	2002	100	Design filing fee	
1003	200	2003	100	Plant filing fee	
1004	300	2004	150	Reissue filing fee	
1005	200	2005	100	Provisional filing fee	

SUBTOTAL (1) ($) 0

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

			Extra Claims		Fee from below		Fee Paid
Total Claims	28	-31** =	3	X	50.00	=	150.00
Independent Claims	3	-3** =		X		=	0
Multiple Dependent				X		=	0

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	50	2202	25	Claims in excess of 20
1201	200	2201	100	Independent claims in excess of 3
1203	360	2203	180	Multiple dependent claim, if not paid
1204	200	2204	100	** Reissue independent claims over original patent
1205	50	2205	25	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) $150.00

**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	
1051	130	2051	65	Surcharge – late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	120	2251	60	Extension for reply within first month	
1252	450	2252	225	Extension for reply within second month	
1253	1,020	2253	510	Extension for reply within third month	1,020.00
1254	1,590	2254	795	Extension for reply within fourth month	
1255	2,160	2255	1,080	Extension for reply within fifth month	
1401	500	2401	250	Notice of Appeal	
1402	500	2402	250	Filing a brief in support of an appeal	
1403	1,000	2403	500	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	500	2452	250	Petition to revive – unavoidable	
1453	1,500	2453	750	Petition to revive – unintentional	
1501	1,400	2501	700	Utility issue fee (or reissue)	
1502	800	2502	400	Design issue fee	
1503	1,100	2503	550	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	790	2809	395	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	790	2810	395	For each additional invention to be examined (37 CFR 1.129(b))	
1801	790	2801	395	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) **$1,020.00**

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(a)]

I hereby certify that this correspondence is being:

☐ deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop _____, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450

☐ transmitted by facsimile on the date shown below to the United States Patent and Trademark Office at (703) _____.

Date _____ Signature _____

Typed or printed name _____

SUBMITTED BY				Complete *(if applicable)*	
Name *(Print/Type)*	Carlos R. Villamar	Registration No. *(Attorney/Agent)*	43,224	Telephone	(202) 585-8204
Signature				Date	August 17, 2005

SEND TO: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a)	Docket Number (Optional) 111325-104 (230300)

CERTIFICATE OF MAILING OR TRANSMISSION
[37 CFR 1.8(a)]

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to Mail Stop _____, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, or being facsimile transmitted to the USPTO at _____, on _____.

Signature: _____

Name: _____

In re Application of
Xin **WANG**, *et al.*

Application Number: 10/162,212 | Filed: June 5, 2002

For **RIGHTS OFFERING AND GRANTING**

Group Art Unit: 3621 | Examiner: Evens J. Augustin

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above identified application.

The requested extension and appropriate entity fee are as follows (check time period desired):

- ☐ One month (37 CFR 1.17(a)(1)) - ($60/$120) $_____
- ☐ Two months (37 CFR 1.17(a)(2)) - ($225/$450) $_____
- ☒ Three months (37 CFR 1.17(a)(3)) - ($510/$1020) $ 1,020.00
- ☐ Four months (37 CFR 1.17(a)(4)) - ($795/$1590) $_____
- ☐ Five months (37 CFR 1.17(a)(5)) - ($1080/$2160) $_____

☐ Applicant claims small entity status.

☐ A check to cover the fee is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☒ The Commissioner has already been authorized to charge fees in this application to a Deposit Account.

☒ The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 19-2380. I have enclosed a duplicate copy of this sheet.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

I am the ☐ applicant/inventor

☐ assignee of record of the entire interest. See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96).

☐ attorney or agent of record.

☒ attorney or agent under 37 CFR 1.34(a). Registration number if acting under 37 CFR 1.34(a) 43,224.

_____ August 17, 2005
Signature | Date

Carlos R. Villamar | (202) 585-8204
Typed or printed name | Telephone Number

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☐ Total of _____ forms are submitted.

SEND TO: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

W671550.1

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2001

Application or Docket Number: 10/162,212

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	21	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	21 minus 20=	* 1
INDEPENDENT CLAIMS	3 minus 3 =	*
MULTIPLE DEPENDENT CLAIM PRESENT		☐

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
BASIC FEE	370.00	OR	BASIC FEE	740.00
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL		OR	TOTAL	

CLAIMS AS AMENDED - PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 31	Minus	** 21	= 10
Independent	* 3	Minus	*** 3	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
~~25~~ X$ 9=		OR	~~50~~ X$18=	500 00
~~100~~ X42=		OR	~~200~~ X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

FEE TRANSMITTAL FOR FY 2005

Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT | **$1,170.00**

Complete if Known	
Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin WANG, *et al.*
Examiner Name	Evens J. Augustin
Art Unit	3621
Attorney Docket No.	111325-104 (230300)

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☐ Credit Card ☐ Money Order ☐ Other ☐ None

☒ Deposit Account:

Deposit Account Number: 19-2380

Deposit Account Name: Nixon Peabody LLP

The Commissioner is authorized to: *(check all that apply)*

☒ Charge fee(s) indicated below ☒ Credit any overpayments

☒ Charge any additional fee(s)

☐ Charge fee(s) indicated below, except for the filing fee to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	300	2001	150	Utility filing fee	
1002	200	2002	100	Design filing fee	
1003	200	2003	100	Plant filing fee	
1004	300	2004	150	Reissue filing fee	
1005	200	2005	100	Provisional filing fee	

SUBTOTAL (1) ($) 0

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

		Extra Claims	Fee from below	Fee Paid
Total Claims	28 -31** =	3	X 50.00 =	150.00
Independent Claims	1 -3** =		X =	0
Multiple Dependent	350.00 DA		X =	0

08/24/2005 SMORELAN 00000004 192380 10162212

01 FC:1202

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	50	2202	25	Claims in excess of 20
1201	200	2201	100	Independent claims in excess of 3
1203	360	2203	180	Multiple dependent claim, if not paid
1204	200	2204	100	** Reissue independent claims over original patent
1205	50	2205	25	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) $150.00

**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	
1051	130	2051	65	Surcharge – late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	120	2251	60	Extension for reply within first month	
1252	450	2252	225	Extension for reply within second month	
1253	1,020	2253	510	Extension for reply within third month	1,020.00
1254	1,590	2254	795	Extension for reply within fourth month	
1255	2,160	2255	1,080	Extension for reply within fifth month	
1401	500	2401	250	Notice of Appeal	
1402	500	2402	250	Filing a brief in support of an appeal	
1403	1,000	2403	500	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	500	2452	250	Petition to revive – unavoidable	
1453	1,500	2453	750	Petition to revive – unintentional	
1501	1,400	2501	700	Utility issue fee (or reissue)	
1502	800	2502	400	Design issue fee	
1503	1,100	2503	550	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	790	2809	395	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	790	2810	395	For each additional invention to be examined (37 CFR 1.129(b))	
1801	790	2801	395	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid **SUBTOTAL (3)** $1,020.00

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(a)]

I hereby certify that this correspondence is being:

☐ deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop _____, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450

☐ transmitted by facsimile on the date shown below to the United States Patent and Trademark Office at (703) _____.

Date _____ Signature _____

Typed or printed name _____

SUBMITTED BY

				Complete *(if applicable)*	
Name *(Print/Type)*	Carlos R. Villamar	Registration No. *(Attorney/Agent)*	43,224	Telephone	(202) 585-8204
Signature				Date	August 17, 2005

SEND TO: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

W671547.1

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/162,212	06/05/2002	Xin Wang	111325-104	3700

22204 7590 08/31/2005

NIXON PEABODY, LLP
401 9TH STREET, NW
SUITE 900
WASHINGTON, DC 20004-2128

EXAMINER	
AUGUSTIN, EVENS J	
ART UNIT	**PAPER NUMBER**
3621	

DATE MAILED: 08/31/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Interview Summary	Application No.	Applicant(s)	
	10/162,212	WANG ET AL.	
	Examiner	Art Unit	
	Evens Augustin	3621	

All participants (applicant, applicant's representative, PTO personnel):

(1) *Evens Augustin*. (3) *Bijan Tadayon*.

(2) *Carlos Villamar*. (4)_____.

Date of Interview: *05 August 2005*.

Type: a)☐ Telephonic b)☐ Video Conference
c)☒ Personal [copy given to: 1)☒ applicant 2)☒ applicant's representative]

Exhibit shown or demonstration conducted: d)☐ Yes e)☐ No.
If Yes, brief description: _____.

Claim(s) discussed: *1-28*.

Identification of prior art discussed: *Downs (US. 6226618*.

Agreement with respect to the claims f)☐ was reached. g)☒ was not reached. h)☐ N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: *Applicant was exploring ways to reconstruct claim language to overcome prior art language*.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN ONE MONTH FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.

Examiner Note: You must sign this form unless it is an Attachment to a signed Office action.

PRIMARY EXAMINER
Examiner's signature, if required

Summary of Record of Interview Requirements

Manual of Patent Examining Procedure (MPEP), Section 713.04, Substance of Interview Must be Made of Record
A complete written statement as to the substance of any face-to-face, video conference, or telephone interview with regard to an application must be made of record in the application whether or not an agreement with the examiner was reached at the interview.

Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews
Paragraph (b)
In every instance where reconsideration is requested in view of an interview with an examiner, a complete written statement of the reasons presented at the interview as warranting favorable action must be filed by the applicant. An interview does not remove the necessity for reply to Office action as specified in §§ 1.111, 1.135. (35 U.S.C. 132)

37 CFR §1.2 Business to be transacted in writing.
All business with the Patent or Trademark Office should be transacted in writing. The personal attendance of applicants or their attorneys or agents at the Patent and Trademark Office is unnecessary. The action of the Patent and Trademark Office will be based exclusively on the written record in the Office. No attention will be paid to any alleged oral promise, stipulation, or understanding in relation to which there is disagreement or doubt.

The action of the Patent and Trademark Office cannot be based exclusively on the written record in the Office if that record is itself incomplete through the failure to record the substance of interviews.

It is the responsibility of the applicant or the attorney or agent to make the substance of an interview of record in the application file, unless the examiner indicates he or she will do so. It is the examiner's responsibility to see that such a record is made and to correct material inaccuracies which bear directly on the question of patentability.

Examiners must complete an Interview Summary Form for each interview held where a matter of substance has been discussed during the interview by checking the appropriate boxes and filling in the blanks. Discussions regarding only procedural matters, directed solely to restriction requirements for which interview recordation is otherwise provided for in Section 812.01 of the Manual of Patent Examining Procedure, or pointing out typographical errors or unreadable script in Office actions or the like, are excluded from the interview recordation procedures below. Where the substance of an interview is completely recorded in an Examiners Amendment, no separate Interview Summary Record is required.

The Interview Summary Form shall be given an appropriate Paper No., placed in the right hand portion of the file, and listed on the "Contents" section of the file wrapper. In a personal interview, a duplicate of the Form is given to the applicant (or attorney or agent) at the conclusion of the interview. In the case of a telephone or video-conference interview, the copy is mailed to the applicant's correspondence address either with or prior to the next official communication. If additional correspondence from the examiner is not likely before an allowance or if other circumstances dictate, the Form should be mailed promptly after the interview rather than with the next official communication.

The Form provides for recordation of the following information:
- Application Number (Series Code and Serial Number)
- Name of applicant
- Name of examiner
- Date of interview
- Type of interview (telephonic, video-conference, or personal)
- Name of participant(s) (applicant, attorney or agent, examiner, other PTO personnel, etc.)
- An indication whether or not an exhibit was shown or a demonstration conducted
- An identification of the specific prior art discussed
- An indication whether an agreement was reached and if so, a description of the general nature of the agreement (may be by attachment of a copy of amendments or claims agreed as being allowable). Note: Agreement as to allowability is tentative and does not restrict further action by the examiner to the contrary.
- The signature of the examiner who conducted the interview (if Form is not an attachment to a signed Office action)

It is desirable that the examiner orally remind the applicant of his or her obligation to record the substance of the interview of each case. It should be noted, however, that the Interview Summary Form will not normally be considered a complete and proper recordation of the interview unless it includes, or is supplemented by the applicant or the examiner to include, all of the applicable items required below concerning the substance of the interview.

A complete and proper recordation of the substance of any interview should include at least the following applicable items:
1) A brief description of the nature of any exhibit shown or any demonstration conducted,
2) an identification of the claims discussed,
3) an identification of the specific prior art discussed,
4) an identification of the principal proposed amendments of a substantive nature discussed, unless these are already described on the Interview Summary Form completed by the Examiner,
5) a brief identification of the general thrust of the principal arguments presented to the examiner,
   (The identification of arguments need not be lengthy or elaborate. A verbatim or highly detailed description of the arguments is not required. The identification of the arguments is sufficient if the general nature or thrust of the principal arguments made to the examiner can be understood in the context of the application file. Of course, the applicant may desire to emphasize and fully describe those arguments which he or she feels were or might be persuasive to the examiner.)
6) a general indication of any other pertinent matters discussed, and
7) if appropriate, the general results or outcome of the interview unless already described in the Interview Summary Form completed by the examiner.

Examiners are expected to carefully review the applicant's record of the substance of an interview. If the record is not complete and accurate, the examiner will give the applicant an extendable one month time period to correct the record.

Examiner to Check for Accuracy

If the claims are allowable for other reasons of record, the examiner should send a letter setting forth the examiner's version of the statement attributed to him or her. If the record is complete and accurate, the examiner should place the indication, "Interview Record OK" on the paper recording the substance of the interview along with the date and the examiner's initials.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/162,212	06/05/2002	Xin Wang	111325-104	3700

22204 7590 10/21/2005

NIXON PEABODY, LLP
401 9TH STREET, NW
SUITE 900
WASHINGTON, DC 20004-2128

EXAMINER	
AUGUSTIN, EVENS J	
ART UNIT	**PAPER NUMBER**
3621	

DATE MAILED: 10/21/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/162,212	WANG ET AL.	
	Examiner	Art Unit	
	Evens Augustin	3621	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 August 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-19 and 29-40* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-19 and 29-40* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

Response to Amendment

This is in response to an amendment file on August 17th, 2005 for letter for patent filed on July 15th, 2004. In the amendment, claims 1, 4, 13, 15 and 16 have been amended. Claims 20-28 have been cancelled. Claims 29-40 have been added. Claims 1-19 and 29-40 are pending in the letter.

Response to Arguments

1. The United States Patent and Trademark Office (USPTO) has considered the applicant's arguments filed on August 17th, 2005, but the arguments are not persuasive.

Applicant argues that the prior art fails to disclose, teach or suggest the noted features, e.g., generating or receiving of an offer, selecting or receiving a selection of desired usage rights and meta-rights from the offer, generating a license based on the selected right. The USPTO respectfully disagrees with applicant's characterization of the prior arts' inventive.

In particular, Downs et al. disclose an invention that broadly relates to the field of electronic commerce and more particularly to a system and related tools for the secure delivery and rights management of digital assets, such as print media, films, games, and music over global communications networks such as the Internet and the World Wide Web. Downs et al.'s invention generates usage rights and derivation of those rights (meta rights). For example, a usage right is the ability to distribute the content or making copies, or the ability to compress the content, or type purchase that can be made. Meta rights are the users' rights vis-à-vis the content in question and may include the number of copies that can be made or different compression speed or the owning versus renting the content (columns 59-60). The actual numbers of copies,

compression speed or owning versus rental are state variable as they define the rights (column 59, lines 15-30). The system then generates licensing with those usage/meta rights (column 7, lines 1-10).

Downs et al. invention presents the digital content offering to consumers (column 48, lines 32-36). The digital content owners receive requests from distributors to sale digital content (column 42, lines 65-67, column 43, lines 1-2). The content owners then license the original content to the distributors (Electronic Digital Content Store) (column 9, lines 5-13). The supplier of digital content has an interface for multi users (column 49, lines 13-17). The supplier also generates the content for distribution (column 9, lines 15-20), with usage conditions (column 9, line 33), kept in a database (figure 1A, item 60). The consumer device also has a user interface (column 20, line 120), a license database (figure 1D, item 197) and is interconnected to a public network such as the internet (column 23, lines 5-15)

Status of Claims

2. Claims 1-19 and 29-40 have been examined.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) The invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

4. Claims 1-13, 15-18 and 29-40 are rejected under 35 U.S.C. 102(b) as being anticipated by Downs et al. (U.S 6,226,618).

As per claims 1-13, 15-18 and 20-28, Downs et al. discloses a system for Electronic Content Delivery, comprising of:

- Usage rights and other downstream rights (column 9, lines 33-35, column 10, lines 15-18). The system also presents the digital content offering to consumers (column 48, lines 32-36). The digital content owners receive requests from distributors to sale digital content (column 42, lines 65-67, column 43, lines 1-2). The content owners then license the original content to the distributors (Electronic Digital Content Store) (column 9, lines 5-13)
- The content suppliers specify usage rights. They also specify that the distributors can customized those original usage rights and provide secondary usage rights (column 9, lines 33-36 - column 10, lines 13-18)
- Receiving a request for usage rights of digital content from a second consumer or end-user (column 18, step 136). The content stores can offer contents with their own customized usage conditions, as long as the customized conditions don't invalidate the original conditions set by the content provider (column 21, lines 23-36). Those secondary usage rights are then validated (column 21, lines 36-51)
- The Secure Digital Content Electronic Distribution system uses multiple formats of digital content such as pictures, movies, videos, music, programs, multimedia and games (column 6, lines 45-48)

- The digital content owners receive requests from distributors to sale digital content (column 42, lines 65-67, column 43, lines 1-2). The content owners then license the original content to the distributors (Electronic Digital Content Store) (column 9, lines 5-13). The content stores offer their own customized usage conditions to end-users, as long as the customized conditions don't invalidate the original conditions set by the content provider (column 21, lines 23-36). The end-users don't get a license until the conditions are validated/authenticated throughout the supply chain (column 22, lines 26-52)
- The license comprises of transaction data (column 24, lines 37-38). The transaction data includes unique transaction ID (column 23, line 62). The license also comprises of digital signature (column 24, lines 45-47) and usage conditions (column 24, line 10-11)
- The first supplier as the content proprietor (column 9, lines 5-15). The first consumers are distributors such as electronic content stores (column 9, lines 63-65)
- Usage rights attached to contents offered to consumers (column 21, lines 30-33). The system verifies that the supplier has the authority to distribute the content (column 22, lines 45-50), and the supplier has the right to receive content (column 42, lines 52-55). The system also verifies that the requested usage conditions are in agreement with the allowable conditions (column 26, lines 20-23). If the conditions are not valid, the system terminates the request (column 10, lines 65-67). Otherwise, the system validates and sends license authorization to consumer (column 10, lines 60-64)

- If license is not validated or approved, the system determines if the user is entitled to the content, then authenticates and retransmits the content the user(negotiation) (column 48, lines 1-25)
- The supplier of digital content has an interface for multi users (column 49, lines 13-17). The supplier also generates the content for distribution (column 9, lines 15-20), with usage conditions (column 9, line 33), kept in a database (figure 1A, item 60). The consumer device also has a user interface (column 20, line 120), a license database (figure 1D, item 197) and is interconnected to a public network such as the internet (column 23, lines 5-15)
- Offer templates containing predetermined usage and meta-rights (column 26, line 62). The system also keeps the identity of the purchaser (column 20, lines 36-37)
- The consumer device has the ability to provide data/digital content originated from the content provider (column 79, lines 35-41
- The end user devices such personal computers (column 79, line 16-17) and the packaged application provide means for the user to accept digital content (column 80, lines 20-25). The system also provides means to specify and apply usage rights and to authenticate those rights (column 42, lines 35-56)
- The invention includes the means and devices to (hardware and software combination) (columns 53, lines 65-67, column 54, lines 1-3) implement the above steps
- The invention has the ability to revoke licenses and create a revocation of licenses that have been revoked (column 37, lines 65-67, column 38, lines 10-20)

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Downs et al. (U.S 6,226,618).

As per claims 14, Downs et al. discloses a system comprising of:

- The system currently uses audio data as an example and specifies usage rights accordingly (column 59, Lines 37-67).
- The system also supports other types of digital content such as pictures, movies, videos, music, programs, multimedia and games (column 6, lines 45-48).

Downs et al. did not explicitly describe a system that wherein the usage rights are associated with copy, transfer, loan, play, print, back-up, restore, delete, extract embed, edit, authorize, install/un-install. However, Downs et al. discloses a system that supports digital content such as pictures, movies, videos, music, programs, multimedia and games (column 6, lines 47-48). Therefore, it would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to specify usage conditions for a particular digital content in order to include the rights of as many digital content formats as possible.

7. Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over Downs et al. (U.S 6,226,618) in view of Hitson et al. (US 20020010759)

As per claim 19, Downs et al. discloses a system comprising of:

- Usage rights and other downstream rights (column 9, lines 33-35, column 10, lines 15-18). The system also presents the digital content offering to consumers (column 48, lines 32-36). The digital content owners receive requests from distributors to sale digital content (column 42, lines 65-67, column 43, lines 1-2). The content owners then license the original content to the distributors (Electronic Digital Content Store) (column 9, lines 5-13)
- The content suppliers specify usage rights. They also specify that the distributors can customized those original usage rights and provide secondary usage rights (column 9, lines 33-36 - column 10, lines 13-18)
- Receiving a request for usage rights of digital content from a second consumer or end-user (c18, step 136). The content stores can then can offer content with their own customized usage conditions, as long as the customized conditions don't invalidate the original conditions set by the content provider (column 21, lines 23-36). The stores send those usage conditions to the user and a clearinghouse for validation (column 21, lines 36-51)

Downs et al. did not explicitly describe a system in which conditions are filtered and applied, based on user preferences. However, Hitson et al. discloses a system and method for content distribution in which content is selected based on user preferences (page 1, paragraph 11). Therefore, it would have been obvious for one of ordinary skill in the art at the time of the

applicant's invention to design a system in which contents are filtered and applied, based on user preferences. It would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to design such system because it would provide personalized content delivery, and would provide economic incentives to content providers by supplying a means of target marketing to users based upon user content preferences page 1, paragraph 11).

Conclusion

8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Evens Augustin whose telephone number is 571-272-6860. The examiner can normally be reached on Monday thru Friday 8 to 5 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jim Trammel can be reached on 571-272-6712.

Any response to this action should be mailed to:

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is 571-272-6584.

Evens J. Augustin
October 17, 2005
Art Unit 3621

PRIMARY EXAMINER

Index of Claims

Application/Control No.	Applicant(s)/Patent under Reexamination
10/162,212	WANG ET AL.
Examiner	**Art Unit**
Evens Augustin	3621

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim		Date
Final	Original	10/17/05
	1	√
	2	√
	3	√
	4	√
	5	√
	6	√
	7	√
	8	√
	9	√
	10	√
	11	√
	12	√
	13	√
	14	√
	15	√
	16	√
	17	√
	18	√
	19	√
	20	
	21	
	22	
	23	
	24	
	25	
	26	
	27	
	28	
	29	√
	30	√
	31	√
	32	√
	33	√
	34	√
	35	√
	36	√
	37	√
	38	√
	39	√
	40	√
	41	
	42	
	43	
	44	
	45	
	46	
	47	
	48	
	49	
	50	

Claim		Date
Final	Original	
	51	
	52	
	53	
	54	
	55	
	56	
	57	
	58	
	59	
	60	
	61	
	62	
	63	
	64	
	65	
	66	
	67	
	68	
	69	
	70	
	71	
	72	
	73	
	74	
	75	
	76	
	77	
	78	
	79	
	80	
	81	
	82	
	83	
	84	
	85	
	86	
	87	
	88	
	89	
	90	
	91	
	92	
	93	
	94	
	95	
	96	
	97	
	98	
	99	
	100	

Claim		Date
Final	Original	
	101	
	102	
	103	
	104	
	105	
	106	
	107	
	108	
	109	
	110	
	111	
	112	
	113	
	114	
	115	
	116	
	117	
	118	
	119	
	120	
	121	
	122	
	123	
	124	
	125	
	126	
	127	
	128	
	129	
	130	
	131	
	132	
	133	
	134	
	135	
	136	
	137	
	138	
	139	
	140	
	141	
	142	
	143	
	144	
	145	
	146	
	147	
	148	
	149	
	150	

NOTICE OF APPEAL FROM THE EXAMINER TO THE BOARD OF PATENT APPEALS AND INTERFERENCES	Docket Number (Optional) 111325-104 (230300)

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.89(a)]	In re Application of Xin WANG, *et al.*	
I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, or being facsimile transmitted to the USPTO at _____, on _____.	Application Number: 10/162,212	Filed: June 5, 2002
Signature: _____	For: **RIGHTS OFFERING AND GRANTING**	
Name: _____	Group Art Unit: 3621	Examiner : Evens J. Augustin

Applicant hereby **appeals** to the Board of Patent Appeals and Interferences from the decision of the examiner.

The fee for this Notice of Appeal is (37 CFR 1.17(b)) $ **500.00**

☐ Applicant claims small entity status. See 37 CFR 1.27. Therefore, the fee shown above is reduced by half, and the resulting fee is: $_____

☐ A check in the amount of the fee is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☒ The Commissioner has already been authorized to charge fees in this application to a Deposit Account. I have enclosed a duplicate copy of this sheet.

☒ The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. 19-2380. I have enclosed a duplicate copy of this sheet.

☐ A petition for an extension of time under 37 CFR 1.136(a) (PTO/SB/22) is enclosed.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

I am the

☐ applicant/inventor.

☐ assignee of record of the entire interest. See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)

☐ attorney or agent of record

☒ attorney or agent acting under 37 CFR 1.34(a).
Registration number if acting under 37 CFR 1.34(a) **43,224.**

Signature

Carlos R. Villamar
Typed or printed name

1/19/06
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

☐ *Total of _____ forms are submitted.

01/20/2006 SZEWDIE1 00000100 192380 10162212

01 FC:1401 500.00 DA

W699070.1

AFR JFW

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin **WANG**, *et al.*
Group Art Unit	3621
Examiner Name	Evens J. Augustin
Attorney Docket Number	111325-104 (230300)
Confirmation Number	3700

Total Number of Pages in This Submission

ENCLOSURES *(check all that apply)*

- ☒ Fee Transmittal Form
  - ☐ 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)
- ☐ Response to Missing Parts/ Incomplete Application
  - ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Assignment Papers *(for an Application)*
- ☐ Drawing(s)
- ☐ 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
- ☒ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to Group (Appeal Notice, Brief, Reply Brief)
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Application Data Sheet
- ☐ Request for Corrected Filing Receipt with Enclosures
- ☐ A self-addressed prepaid postcard for acknowledging receipt
- ☐ Other Enclosure(s) (*please identify below*):

Remarks

☒ 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	Carlos R. Villamar Registration No. 43,224 **Nixon Peabody LLP** 401 9th Street, N.W., Suite 900 Washington, D.C. 20004-2128
Signature	
Date	1/19/06

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(a)]

I hereby certify that this correspondence is being:

- ☐ deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop _____, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450
- ☐ transmitted by facsimile on the date shown below to the United States Patent and Trademark Office at (703) _____.

_____ Date

_____ Signature

_____ Typed or printed name

FEE TRANSMITTAL FOR FY 2005

Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT | **$500.00**

Complete if Known	
Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin WANG, *et al.*
Examiner Name	Evens J. Augustin
Art Unit	3621
Attorney Docket No.	111325-104 (230300)

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☐ Credit Card ☐ Money Order ☐ Other ☐ None

☒ Deposit Account:

Deposit Account Number: 19-2380

Deposit Account Name: Nixon Peabody LLP

The Commissioner is authorized to: *(check all that apply)*

☒ Charge fee(s) indicated below ☒ Credit any overpayments

☒ Charge any additional fee(s)

☐ Charge fee(s) indicated below, **except for the filing fee** to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	300	2001	150	Utility filing fee	
1002	200	2002	100	Design filing fee	
1003	200	2003	100	Plant filing fee	
1004	300	2004	150	Reissue filing fee	
1005	200	2005	100	Provisional filing fee	

SUBTOTAL (1) ($) 0

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

		Extra Claims		Fee from below		Fee Paid
Total Claims		-20** = 0	X		=	0
Independent Claims		-3** = 0	X		=	0
Multiple Dependent			X		=	0

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	50	2202	25	Claims in excess of 20
1201	200	2201	100	Independent claims in excess of 3
1203	360	2203	180	Multiple dependent claim, if not paid
1204	200	2204	100	** Reissue independent claims over original patent
1205	50	2205	25	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 0

**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	
1051	130	2051	65	Surcharge – late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	120	2251	60	Extension for reply within first month	
1252	450	2252	225	Extension for reply within second month	
1253	1,020	2253	510	Extension for reply within third month	
1254	1,590	2254	795	Extension for reply within fourth month	
1255	2,160	2255	1,080	Extension for reply within fifth month	
1401	500	2401	250	Notice of Appeal	500.00
1402	500	2402	250	Filing a brief in support of an appeal	
1403	1,000	2403	500	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	500	2452	250	Petition to revive – unavoidable	
1453	1,500	2453	750	Petition to revive – unintentional	
1501	1,400	2501	700	Utility issue fee (or reissue)	
1502	800	2502	400	Design issue fee	
1503	1,100	2503	550	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	790	2809	395	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	790	2810	395	For each additional invention to be examined (37 CFR 1.129(b))	
1801	790	2801	395	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) **$500.00**

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(a)]

I hereby certify that this correspondence is being:

☐ deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop _____, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450

☐ transmitted by facsimile on the date shown below to the United States Patent and Trademark Office at (703) _____.

Date _____ Signature _____

Typed or printed name _____

SUBMITTED BY				Complete *(if applicable)*	
Name *(Print/Type)*	Carlos R. Villamar	Registration No. *(Attorney/Agent)*	43,224	Telephone	(202) 585-8204
Signature				Date	11/19/06

SEND TO: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

W699072.1

Docket No. 111325-104 (230300)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:	)	
Xin **WANG,** *et al.*	)	Examiner: Augustin, Evens J.
Serial No. 10/162,212	)	Group Art Unit: 3621
Filed: June 5, 2002	)	Confirmation No. 3700
For: **RIGHTS OFFERING AND GRANTING**	)	

U.S. Patent and Trademark Office
Customer Services Window, Mail Stop AF
Randolph Building
401 Dulany Street
Alexandria, VA 22314

RESPONSE TO AFTER FINAL REJECTION

In response to the final Office Action mailed **October 21, 2005**, reconsideration of the above identified application is respectfully requested:

REMARKS

The following remarks are submitted to be fully responsive to the final Office Action of **October 21, 2005**. Reconsideration and allowance of this application are respectfully requested. Claims 1-19, and 29-40 are pending in the present application.

Referring now to the present Office Action, claims 1-13, 15-18 and 20-40 were rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 6,226,618 to *Downs et al.*, claim 14 was rejected under 35 U.S.C. §103(a) as being unpatentable over *Downs et al.*, and claim 19 was rejected under 35 U.S.C. §103(a) as being unpatentable over *Downs et al.* in view of U.S. Patent Application No. 20020010759 to *Hitson et al.*

First, Applicants wish to note that the present Office Action fails show how claims 29-40 are anticipated by *Downs et al.*. Accordingly, no *prima facie* case has been made with respect to claims 29-40 and Applicants submit that claims 29-40 are patentable over *Downs et al.* and *Hitson et al.*, alone or in combination.

In addition, the present Office Action appears to be repeating the previous rejection and as such has failed to address the features recited in the previously amended independent claims 1, 15 and in added independent claim 38. Accordingly, no *prima facie* case has been made with respect to independent claims 1, 15, and 38 and Applicants submit that independent claims 1, 15, and 38 are patentable over *Downs et al.* and *Hitson et al.*, alone or in combination, and as further set forth below.

Claims 1-19, and 29-40 are patentably distinguishable over *Downs et al.* and *Hitson et al.*, because *Downs et al.* and *Hitson et al.*, alone or in combination, fail to disclose, teach or suggest all of the features recited in the present claims. For example, independent claim 1 (emphasis added) recites:

> A method for transferring usage rights adapted to be associated with items within a digital rights management system, said method comprising:
>
> **generating, by a supplier, at least one first offer including usage rights and meta-rights for the items, said usage rights defining a manner of use for the items, said meta-rights specifying rights to derive usage rights or other meta-rights for the items;**
>
> presenting, by the supplier, said offer to a first consumer in the system,
>
> wherein the offer expresses what rights a consumer can acquire for the items;
>
> **receiving, by the supplier, a selection of from the first consumer indicating desired usage rights and meta-rights**; and
>
> generating, by the supplier, a first license granting to the first consumer the usage rights and meta-rights for the items,

wherein said first license grants the usage rights and meta-rights that are selected by the first consumer during said receiving step; and

independent claim 15 (emphasis added) recites:

A system for transferring usage rights adapted to be associated with an item, to be licensed in multi-tier channels of distribution with downstream rights and conditions assigned at least at one level, said system comprising:

a supplier component, comprising:

a supplier user interface module;

an offer generator module for generating an offer including at least usage rights and meta-rights for the item, the usage rights defining a manner of use for the item, the meta-rights specifying rights to derive usage rights or other meta-rights for the item;

a rights composer module for composing a draft license;

a repository for supplier's rights;

a supplier management database; and

a consumer component comprising:

a consumer user interface module;

an offer-consideration module configured to analyze the offers generated by the supplier component and select offers based on the analysis;

a repository for consumer's rights;

a consumer management database; and

a communication link coupling said supplier component and said consumer component,

wherein the rights composer module is configured to compose a license granting the usage rights and meta-rights that are selected by the offer-consideration module of the consumer component; and

independent claim 38 (emphasis added) recites:

A method for transferring usage rights adapted to be associated with an item within a digital rights management system, the method being performed by a consumer device within the system, the method comprising:

receiving, by the consumer device, from a supplier device within the system at least one offer including usage rights and meta-rights for the item,

wherein the usage rights define a manner of use for the item, and **the meta-rights specify rights to derive usage rights or other meta-rights for the item**, and

the offer expresses what rights the consumer device can acquire for the item;

selecting, by the consumer device, desired usage rights and meta-rights from the received offer,

wherein the selected rights express what rights the consumer device desires to acquire for the item; and

receiving, by the consumer device, a license from the supplier device,

wherein the received license grants the usage rights and meta-rights that are selected and provided by the consumer device.

Contrary to the assertion in the present Office Action, *Downs et al.* fails to disclose, teach or suggest the noted features recited in independent claims 1, and 15. *Downs et al.* also fails to disclose, teach or suggest the noted features recited in independent claim 38. The present Office Action takes a position at page 2 that "[m]eta rights are the user's rights vis-à-vis the content in question and may include the number of copies that can actually be made or different compression speed or the owning versus renting the content," citing cols. 59-60 of *Downs et al.* However, cols. 59-60 of Downs et al. merely disclose usage conditions and meta-data, but otherwise do not disclose, teach or suggest "meta-rights," which govern the exercising of other usage rights or meta-rights, as claimed in independent claims 1, 15, and 38.

For example, although the cited portions of *Downs et al.* in the present Office Action may generally pertain to usage conditions, such as wholesale price, retail price, copy restrictions, etc., *Downs et al.* fails to disclose, teach or suggest usage rights in the manner claimed, much less meta-rights which govern the exercising of other usage rights or meta-rights, as recited in independent claims 1, 15, and 38. In addition, *Downs et al.* fails to disclose, teach or suggest the noted features, e.g., generating or receiving of an offer, selecting or receiving a selection of desired usage rights and meta-rights from the offer, generating a license based on the selected right, etc., as recited independent claims 1, 15, and 38.

Accordingly, the portions of *Downs et al.* cited in the present Office Action do not support the rejection of independent claims 1, 15 and 38. Specifically, it appears that the concept of meta-rights is misunderstood, as page 2 of the Office Action states that "[m]eta rights are the user's rights vis-à-vis the content in question ...," whereas meta-rights specify "rights to derive usage rights or other meta-rights," as defined independent claims 1, 15 and 38. For example, a right to play a movie is a usage right, whereas a right to issue or grant such a play right to users is a meta right.

Contrary to the present Office Action, *Downs et al.* cols 59-60 do not disclose meta-rights, but rather are directed to rights to End-User(s), and any restrictions on the End-User(s) with regard to the use of the Content. For example, col. 59, lines 15-30 provides a list of usage conditions, most of which are not rights, and none of which are meta rights, as defined in independent claims 1, 15 and 38.

Similarly, *Downs et al.* (1) col. 7, lines 1-10 of are directed to licensing usage conditions, but are silent with respect to any licensing of meta rights, (2) col 48, lines 32-36 are directed to offering content (e.g., packaging, transferring and securing content), but are silent with respect to offering rights and meta-rights, (3) col 42, lines 65-67, and col 43, lines 1-2 are directed to a store making a request to content owners to be a content seller, but fail to disclose, teach or suggest what rights the store requests and what rights the store wants to issue to others (e.g., end-users), (4) col 9, lines 5-13 describe a content distribution value chain, but fail to disclose, teach or suggest conveying rights and meta-rights from one to another in the chain, (5) col 49, lines 13-17 are directed to a user interface to access content, but fail to disclose, teach or suggest how to manage rights and meta-rights, (6) col 9, lines 15-20 are directed to packaging and tracking content, but are silent with respect to any rights or meta-rights, and (7) col 9 line 33, and Fig 1A, item 60 describe an interface to enter and a database to store usage conditions, but fail to disclose, teach or suggest meta-rights in the manner claimed.

Hitson et al. is directed to a system and method for allowing multimedia content to be delivered to a computer, personal desktop assistant, portable media player, or other electronic device, wherein the content may include advertisements, and distribution of content may emulate current television and/or radio broadcasts, the content may be encrypted or otherwise secured, and such security may restrict use of some content, including limiting content to specific devices, specific users, or a predefined number of playbacks. The users may also indicate a preference for a particular content type or types, and content may be chosen based on user preferences. Users may further refine content preferences as content is experienced, and may indicate a desire to purchase content or find out more about specific content, and such desires may be recorded for later review or action. However, *Hitson et al.* fails to cure the noted deficiencies in *Downs et al.* and fails to disclose, teach or suggest rights and meta-rights in the manner claimed. Accordingly, *Downs et al.* and *Hitson et al.*, alone or in combination, fail to disclose, teach or suggest the noted features recited in independent claims 1, 15 and 38.

The invention recited in independent claims 1, 15 and 38 and claims dependent therefrom recognizes and solves the following problems:

> **[0010]** However, there are limitations associated with the above-mentioned paradigms wherein only usage rights and conditions associated with content

> are specified by content owners or other grantors of rights. Once purchased by an end user, a consumer, or a distributor, of content along with its associated usage rights and conditions has no means to be legally passed on to a next recipient in a distribution chain. Further the associated usage rights have no provision for specifying rights to derive other rights, i.e. rights to modify, transfer, offer, grant, obtain, transfer, delegate, track, surrender, exchange, transport, exercise, revoke, or the like. Common content distribution models often include a multi-tier distribution and usage chain. Known DRM systems do not facilitate the ability to prescribe rights and conditions for all participants along a content distribution and usage chain. Therefore, it is difficult for a content owner to commercially exploit content unless the owner has a relationship with each party in the distribution chain.

The invention recited in independent claims 1, 15 and 38 and claims dependent therefrom provides the following advantages:

> **[0063]** Overall, as can be seen in the description of FIGS. 6, 7, and 8 above, the consumer sends a request, and then a license is constructed. Either the supplier or the consumer could draft the content of the license, but in the example above the supplier does so. The request is a subset of an offer and the offer has one or more options. The supplier makes the offer available to the consumer sending the request (and to other consumers if that is the desire), and the consumer (including other consumers, if applicable) makes choices. Then, the supplier analyzes the choices, and constructs the license (i.e. a grant of rights). Note that the request can also be rejected, or a counter proposal could be made and the same process could then repeat for the counter proposal.

By contrast, *Downs et al.* and *Hitson et al.*, alone or in combination, fail disclose, teach or suggest the noted features, fail to recognize or solve the noted problems, and fail to provide the advantages of the inventions recited in independent claims 1, 15 and 38.

The dependent claims are allowable on their on merits and for at least the reasons as argued above with respect to independent claims 1, 15 and 38.

In view of the foregoing, it is submitted that the present application is in condition for allowance and a notice to that effect is respectfully requested. However, if the Examiner deems that any issue remains after considering this response, the Examiner is invited to contact the undersigned attorney to expedite the prosecution and engage in a joint effort to work out a mutually satisfactory solution.

Respectfully submitted,

NIXON PEABODY, LLP

Carlos R. Villamar
Reg. No. 43,224

NIXON PEABODY LLP
CUSTOMER NO.: 22204
401 9th Street, N.W., Suite 900
Washington, DC 20004
Tel: 202-585-8000
Fax: 202-585-8080

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2001

Application or Docket Number: 10/162,212

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	21	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	21 minus 20=	* 1
INDEPENDENT CLAIMS	3 minus 3 =	* —
MULTIPLE DEPENDENT CLAIM PRESENT		☐

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
BASIC FEE	370.00	OR	BASIC FEE	740.00
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL		OR	TOTAL	

CLAIMS AS AMENDED - PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐				

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 31	Minus	** 21	= 10
Independent	* 3	Minus	*** 3	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐				

1-FEE

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
~~25~~ X$ 9=		OR	~~50~~ X$18=	500.00
~~100~~ X42=		OR	~~200~~ X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* NOCLAIMS LISTED	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐				

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/162,212	06/05/2002	Xin Wang	111325-104	3700

22204 7590 02/14/2006

NIXON PEABODY, LLP
401 9TH STREET, NW
SUITE 900
WASHINGTON, DC 20004-2128

EXAMINER	
AUGUSTIN, EVENS J	

ART UNIT	PAPER NUMBER
3621	

DATE MAILED: 02/14/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Advisory Action Before the Filing of an Appeal Brief	Application No. 10/162,212	Applicant(s) WANG ET AL.	
	Examiner Evens Augustin	Art Unit 3621	

--The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

THE REPLY FILED 19 January 2005 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE.

1. ☒ The reply was filed after a final rejection, but prior to or on the same day as filing a Notice of Appeal. To avoid abandonment of this application, applicant must timely file one of the following replies: (1) an amendment, affidavit, or other evidence, which places the application in condition for allowance; (2) a Notice of Appeal (with appeal fee) in compliance with 37 CFR 41.31; or (3) a Request for Continued Examination (RCE) in compliance with 37 CFR 1.114. The reply must be filed within one of the following time periods:
   a) ☐ The period for reply expires *3* months from the mailing date of the final rejection.
   b) ☐ The period for reply expires on: (1) the mailing date of this Advisory Action, or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection.

   Examiner Note: If box 1 is checked, check either box (a) or (b). ONLY CHECK BOX (b) WHEN THE FIRST REPLY WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. See MPEP 706.07(f).

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

NOTICE OF APPEAL

2. ☒ The Notice of Appeal was filed on *1/19/2005*. A brief in compliance with 37 CFR 41.37 must be filed within two months of the date of filing the Notice of Appeal (37 CFR 41.37(a)), or any extension thereof (37 CFR 41.37(e)), to avoid dismissal of the appeal. Since a Notice of Appeal has been filed, any reply must be filed within the time period set forth in 37 CFR 41.37(a).

AMENDMENTS

3. ☐ The proposed amendment(s) filed after a final rejection, but prior to the date of filing a brief, will not be entered because
   (a) ☐ They raise new issues that would require further consideration and/or search (see NOTE below);
   (b) ☐ They raise the issue of new matter (see NOTE below);
   (c) ☐ They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or
   (d) ☐ They present additional claims without canceling a corresponding number of finally rejected claims.

   NOTE: _____. (See 37 CFR 1.116 and 41.33(a)).
4. ☐ The amendments are not in compliance with 37 CFR 1.121. See attached Notice of Non-Compliant Amendment (PTOL-324).
5. ☐ Applicant's reply has overcome the following rejection(s): _____.
6. ☐ Newly proposed or amended claim(s) _____ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).
7. ☐ For purposes of appeal, the proposed amendment(s): a) ☐ will not be entered, or b) ☐ will be entered and an explanation of how the new or amended claims would be rejected is provided below or appended.
   The status of the claim(s) is (or will be) as follows:
   Claim(s) allowed: _____.
   Claim(s) objected to: _____.
   Claim(s) rejected: _____.
   Claim(s) withdrawn from consideration: _____.

AFFIDAVIT OR OTHER EVIDENCE

8. ☐ The affidavit or other evidence filed after a final action, but before or on the date of filing a Notice of Appeal will not be entered because applicant failed to provide a showing of good and sufficient reasons why the affidavit or other evidence is necessary and was not earlier presented. See 37 CFR 1.116(e).
9. ☐ The affidavit or other evidence filed after the date of filing a Notice of Appeal, but prior to the date of filing a brief, will not be entered because the affidavit or other evidence failed to overcome all rejections under appeal and/or appellant fails to provide a showing a good and sufficient reasons why it is necessary and was not earlier presented. See 37 CFR 41.33(d)(1).
10. ☐ The affidavit or other evidence is entered. An explanation of the status of the claims after entry is below or attached.

REQUEST FOR RECONSIDERATION/OTHER

11. ☒ The request for reconsideration has been considered but does NOT place the application in condition for allowance because: See Continuation Sheet.
12. ☐ Note the attached Information Disclosure Statement(s). (PTO/SB/08 or PTO-1449) Paper No(s). _____
13. ☐ Other: _____.

PRIMARY EXAMINER

Continuation of 11. does NOT place the application in condition for allowance because: The claims are anticipated/made obvious by Downs et al. which discloses a system in which a supplier provides content to first customer (digital store). The content are provided with the suppliers usage rights. The first customer has the ability to offer the content to a second customer (end-user), with its own customized usage, different from the usage rights of the supplier.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/162,212	06/05/2002	Xin Wang	111325-104	3700

22204 7590 02/16/2006

NIXON PEABODY, LLP
401 9TH STREET, NW
SUITE 900
WASHINGTON, DC 20004-2128

EXAMINER	
AUGUSTIN, EVENS J	
ART UNIT	**PAPER NUMBER**
3621	

DATE MAILED: 02/16/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Advisory Action Before the Filing of an Appeal Brief	**Application No.** 10/162,212	**Applicant(s)** WANG ET AL.	
	Examiner Evens Augustin	**Art Unit** 3621	

--The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

THE REPLY FILED 19 January 2006 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE.

1. ☒ The reply was filed after a final rejection, but prior to or on the same day as filing a Notice of Appeal. To avoid abandonment of this application, applicant must timely file one of the following replies: (1) an amendment, affidavit, or other evidence, which places the application in condition for allowance; (2) a Notice of Appeal (with appeal fee) in compliance with 37 CFR 41.31; or (3) a Request for Continued Examination (RCE) in compliance with 37 CFR 1.114. The reply must be filed within one of the following time periods:
   a) ☒ The period for reply expires *3* months from the mailing date of the final rejection.
   b) ☐ The period for reply expires on: (1) the mailing date of this Advisory Action, or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection.

   Examiner Note: If box 1 is checked, check either box (a) or (b). ONLY CHECK BOX (b) WHEN THE FIRST REPLY WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. See MPEP 706.07(f).

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

NOTICE OF APPEAL

2. ☒ The Notice of Appeal was filed on *19 January 2006*. A brief in compliance with 37 CFR 41.37 must be filed within two months of the date of filing the Notice of Appeal (37 CFR 41.37(a)), or any extension thereof (37 CFR 41.37(e)), to avoid dismissal of the appeal. Since a Notice of Appeal has been filed, any reply must be filed within the time period set forth in 37 CFR 41.37(a).

AMENDMENTS

3. ☐ The proposed amendment(s) filed after a final rejection, but prior to the date of filing a brief, will not be entered because
   (a) ☐ They raise new issues that would require further consideration and/or search (see NOTE below);
   (b) ☐ They raise the issue of new matter (see NOTE below);
   (c) ☐ They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or
   (d) ☐ They present additional claims without canceling a corresponding number of finally rejected claims.

   NOTE: _____. (See 37 CFR 1.116 and 41.33(a)).
4. ☐ The amendments are not in compliance with 37 CFR 1.121. See attached Notice of Non-Compliant Amendment (PTOL-324).
5. ☐ Applicant's reply has overcome the following rejection(s): _____.
6. ☐ Newly proposed or amended claim(s) _____ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).
7. ☐ For purposes of appeal, the proposed amendment(s): a) ☐ will not be entered, or b) ☐ will be entered and an explanation of how the new or amended claims would be rejected is provided below or appended.
   The status of the claim(s) is (or will be) as follows:
   Claim(s) allowed: _____.
   Claim(s) objected to: _____.
   Claim(s) rejected: _____.
   Claim(s) withdrawn from consideration: _____.

AFFIDAVIT OR OTHER EVIDENCE

8. ☐ The affidavit or other evidence filed after a final action, but before or on the date of filing a Notice of Appeal will not be entered because applicant failed to provide a showing of good and sufficient reasons why the affidavit or other evidence is necessary and was not earlier presented. See 37 CFR 1.116(e).
9. ☐ The affidavit or other evidence filed after the date of filing a Notice of Appeal, but prior to the date of filing a brief, will not be entered because the affidavit or other evidence failed to overcome all rejections under appeal and/or appellant fails to provide a showing a good and sufficient reasons why it is necessary and was not earlier presented. See 37 CFR 41.33(d)(1).
10. ☐ The affidavit or other evidence is entered. An explanation of the status of the claims after entry is below or attached.

REQUEST FOR RECONSIDERATION/OTHER

11. ☒ The request for reconsideration has been considered but does NOT place the application in condition for allowance because: See Continuation Sheet.
12. ☐ Note the attached Information Disclosure Statement(s). (PTO/SB/08 or PTO-1449) Paper No(s).
13. ☐ Other: _____.

FIRMIN BACKER
PRIMARY EXAMINER

Continuation of 11. does NOT place the application in condition for allowance because: Downs et al. invention presents the digital content offering to consumers (column 48, lines 32-36). The digital content owners receive requests from distributors to sale digital content (column 42, lines 65-67, column 43, lines 1-2). The content owners then license the original content to the distributors (Electronic Digital Content Store) (column 9, lines 5-13). The supplier of digital content has an interface for multi users (column 49, lines 13-17). The supplier also generates the content for distribution (column 9, lines 15-20), with usage conditions (column 9, line 33), kept in a database (figure 1A, item 60). The consumer device also has a user interface (column 20, line 120), a license database (figure 1D, item 197) and is interconnected to a public network such as the internet (column 23, lines 5-15).

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Patent Application of:	)	Confirmation No.: 3700
Xin WANG, *et al.*	)	Group Art Unit: 3621
Serial No. 10/162,212	)	Examiner: Evens J. Augustin
Filed: June 5, 2002	)	
For: **RIGHTS OFFERING AND GRANTING**	)	

United States Patent and Trademark Office
Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

APPEAL BRIEF

As set forth in the Notice of Appeal filed January 19, 2006, Appellants hereby appeal the Examiner's final rejection of claims 1-19 and 29-40 of the above-identified application. Appellants respectfully request that the Board of Patent Appeals and Interferences reverse the final rejection of these claims.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

ContentGuard Holdings, Inc., is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-19 and 29-40 have been finally rejected and are the subject matter of this appeal. Claims 20-28 have been canceled.

IV. STATUS OF AMENDMENTS

No amendment has been filed or submitted after the final rejection mailed October 21, 2005.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed to a method for transferring usage rights adapted to be associated with items. The method includes generating, by a supplier, at least one first offer containing usage rights and meta-rights for the item. The usage rights define a manner of use for the items, and the meta-rights specify rights to derive usage rights or other meta-rights including, presenting an offer to a first consumer, receiving a selection from the first consumer indicating desired usage rights and meta-rights, and generating, by the supplier, a first license granting the desired usage rights and meta-rights to the first consumer. Independent claim 1 is at least supported by paragraph 33 of the Appellants' published patent application.

Claims 2-14, 29-32, 34, and 35 ultimately depend from independent claim 1. Claims 2-14, 29-32, 34, and 35 describe additional limitations of the method of independent claims 1, including, for example, "receiving a request generated by a second consumer for a license including at least one of usage rights and meta-rights for the items." Claims 2 and 3 are at least supported by paragraphs 35-37; claims 6-10 are at least supported by paragraphs 40-43; claims 11-14 are at least supported by paragraphs 51-53; claims 29 and 30 are at least supported by paragraph 54; claims 31 and 32 are at least supported by paragraph 47; and claims 34 and 35 are at least supported by paragraph 33 of Appellants' published patent application.

Independent claim 15 is directed to a system for transferring usage rights adapted to be associated with an item to be licensed in multi-tier channels of distribution with downstream rights and conditions assigned at least one level. The system includes a supplier component, that further includes a supplier user interface module, an offer generator module for generating an offer containing at least usage rights and of meta-rights, a rights composer module for composing a draft license, and a repository for supplier's rights, a supplier management database. The system further includes a consumer component that further includes a consumer user interface module, an offer-consideration module configured to analyze the offers generated by the supplier component and select offers based on the analysis, and a repository for consumer's rights, a consumer management database. The rights composer module is configured to compose a license granting the usage rights and meta-rights that are selected by the offer-consideration module of the consumer component. Independent claim 15 is at least supported by paragraph 41 of the Appellants' published patent application.

Claims 16-19, 33, 36, and 37 ultimately depend from independent claim 15. Claims 16-19, 33, 36, and 37 describe additional limitations of the method of independent claims 15, including, for example, a "means for filtering offers based on supplier preferences." Claim 16 is at least supported by paragraphs 32; claim 17 is at least supported by paragraph 51; claims 18-19 are at least supported by paragraph 57; claim 33 is at least supported by paragraph 47; and claims 36 and 37 are at least supported by paragraph 33 of Appellants' published patent application.

Independent claim 38 is directed to a method for transferring usage rights adapted to be associated with an item within a digital rights management system. The method is performed by a consumer device within the system and includes receiving, by the consumer device, from a supplier device within the system at least one offer including usage rights and meta-rights for the item. The usage rights define a manner of use for the item, the meta-rights specify rights to derive usage rights or other meta-rights for the item, and the offer expresses what rights the consumer device can acquire for the item. The method further includes selecting, by the consumer device, desired usage rights and meta-rights from the received offer. The selected rights express what rights the consumer device desires to acquire for the item. The method further includes receiving, by the consumer device, a license from the supplier device, wherein the received license grants the usage rights and meta-rights that

are selected and provided by the consumer device. Independent claim 54 is at least supported by paragraph 33 of the Appellants' published patent application.

Claims 39 and 40 ultimately depend from independent claim 15. Claims 39 and 40 additional limitations of the method of independent claims 38, including, for example, the method "is implemented with one or more hardware and/or software components configured to perform the steps of the method." Claims 39 and 40 are at least supported by paragraph 47 of Appellants' published patent application.

VI. GROUNDS OF REJECTION

Appellant respectfully requests the Board to reverse the following grounds of rejection: rejection of claims 1-13, 15-18 and 29-40 under 35 U.S.C. § 102(b) as being anticipated by Downs et al. (hereinafter "Downs"), U.S. Patent Number 6,226,618; rejection of claim 14 as being unpatentable under 35 U.S.C. § 103(a) over Downs; and rejection of claim 19 as being unpatentable under 35 U.S.C. § 103(a) over Downs in view of Hitson et al. (hereinafter "Hitson"), U.S. Application Number 20020010759.

VII. ARGUMENTS

Rejection Under 35 U.S.C. § 102

If all claimed elements/steps are disclosed, expressly or inherently, in a single prior art reference, that reference is said to "anticipate" the claimed invention, thereby invalidating the claim(s) under 35 U.S.C. §102. *Scripps Clinic & Research Foundation v. Genentech, Inc.*, 927 F.2d 1565, 1576, 18 USPQ2d 1001, 1010 (Fed. Cir. 1991).

For a novelty rejection, "[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). *M.P.E.P.* §2131. Appellant respectfully submits that Downs does not teach, disclose, or suggest each and every element of claims 1-13, 15-18 and 29-40.

1. Claims 1-13, 15-18 and 29-40 are not anticipated by the Downs reference.

Claims 1-13, 15-18 and 29-40 have been rejected under 35 U.S.C. § 102(b) as being anticipated by Downs. This rejection is respectfully traversed and reversal of the Examiner's position with respect thereto is earnestly solicited in that the Downs reference cited by the Examiner neither discloses nor suggests that which is presently set forth by Appellants' claimed invention.

Initially, Appellants note that the Examiner's final rejection fails to show how claims 29-40 are anticipated by Downs. The Examiner's final rejection does not specifically refer to claims 29-40 in the Response to Arguments and accordingly, no *prima facie* case has been made with respect to claims 29-40.

Further, Appellants contend that the Final Office Action of October 21, 2005 fails to address the features recited in amended independent claims 1 and 15. The Final Office Action repeats the previous rejection and thus does not make a *prima facie* case under 35 U.S.C. § 102(b) for the rejection.

A. Claims 1, 15, and 38 are not anticipated by Downs.

According to the present invention, one of the novel features of independent claims 1, 15, and 38 is the use and transferability of meta-rights. For example, independent claim 1 (emphasis added) recites:

> A method for transferring usage rights adapted to be associated with items within a digital rights management system, said method comprising:
> **generating, by a supplier, at least one first offer including usage rights and meta-rights for the items, said usage rights defining a manner of use for the items, said meta-rights specifying rights to derive usage rights or other meta-rights for the items;**
> presenting, by the supplier, said offer to a first consumer in the system,
> wherein the offer expresses what rights a consumer can acquire for the items;
> **receiving, by the supplier, a selection of from the first consumer indicating desired usage rights and meta-rights**; and
> generating, by the supplier, a first license granting to the first consumer the usage rights and meta-rights for the items,
> **wherein said first license grants the usage rights and meta-rights that are selected by the first consumer during said receiving step.**

Independent claim 15 (emphasis added) recites:

> A system for transferring usage rights adapted to be associated with an item, to be licensed in multi-tier channels of distribution with downstream rights and conditions assigned at least at one level, said system comprising:
> a supplier component, comprising:
> a supplier user interface module;
> **an offer generator module for generating an offer including at least usage rights and meta-rights for the item, the usage rights defining a manner of use for the item, the meta-rights specifying rights to derive usage rights or other meta-rights for the item;**
> a rights composer module for composing a draft license;
> a repository for supplier's rights;
> a supplier management database; and
> a consumer component comprising:

> a consumer user interface module;
> **an offer-consideration module configured to analyze the offers generated by the supplier component and select offers based on the analysis;**
> a repository for consumer's rights;
> a consumer management database; and
> a communication link coupling said supplier component and said consumer component,
> **wherein the rights composer module is configured to compose a license granting the usage rights and meta-rights that are selected by the offer-consideration module of the consumer component.**

Independent claim 38 (emphasis added) recites:

> A method for transferring usage rights adapted to be associated with an item within a digital rights management system, the method being performed by a consumer device within the system, the method comprising:
> **receiving, by the consumer device, from a supplier device within the system at least one offer including usage rights and meta-rights for the item,**
> wherein the usage rights define a manner of use for the item, and **the meta-rights specify rights to derive usage rights or other meta-rights for the item**, and
> the offer expresses what rights the consumer device can acquire for the item;
> **selecting, by the consumer device, desired usage rights and meta-rights from the received offer,**
> wherein the selected rights express what rights the consumer device desires to acquire for the item; and
> receiving, by the consumer device, a license from the supplier device,
> **wherein the received license grants the usage rights and meta-rights that are selected and provided by the consumer device.**

The portions of Downs cited by the Examiner generally pertain to usage conditions, such as wholesale price, retail price, copy restrictions, etc. However, Downs fails to disclose the use of meta-rights as recited in Appellants' claims 1, 15, and 38.

Specifically, it appears that the concept of meta-rights is misunderstood, as page 2 of the Final Office Action states that "[m]eta rights are the user's rights vis-à-vis the content in question." As defined in independent claims 1, 15 and 38, meta-rights specify "rights to derive usage rights or other meta-rights." They can also specify transfer rights and can permit granting of rights to others or derivation of rights. (Paragraph 0033 of Appellants' published patent application). For example, a right to play a movie is a usage right, whereas a right to issue or grant such a play right to users is a meta-right.

Downs discloses usage rights that are directed to end-users and are contained in metadata that is transmitted to the user. (Col. 26, lines 60-62). The Examiner has misunderstood metadata to be equivalent with meta-rights as recited in Appellants'

independent claims 1, 15, and 38. Metadata, as disclosed by Downs, includes descriptive information about the content items such as the artist name and CD cover art, and copy restrictions, wholesale price, and business rules. (Col. 26 lines 43-46). However, Downs fails to disclose meta-rights that include the rights to offer, grant, obtain, transfer, delegate, track, surrender, exchange, and revoke usage rights to/from others as is disclosed in the present invention. (Appellants' Paragraph 0033, See also Claims 1, 15, 38). The Examiner has failed to note this distinction between meta-rights and metadata, as meta-rights are rights about rights, and not rights about content directly (i.e., usage rights) nor descriptive information about the content items (i.e., metadata).

Additionally, Appellants contend that the Examiner's assertion of end-user's rights to modify the existing usage rights as disclosed by Downs is incorrect. Downs discloses that content stores, and not end users or other consumers, can offer content with their own customized usage conditions, as long as the customized conditions don't invalidate the original conditions set by the content provider. (Col. 21, lines 23-36). This capability is apparently only built into the behavior of the content stores, rather than being specified explicitly in a license as rights that the content stores can have and are later enforced. Accordingly, Downs fails to disclose or suggest that the second consumer, or end-user, has the ability to obtain the rights to modify the existing conditions or restrictions from the content owner or distributor, defined as meta-rights in Appellants' claims 1, 15, and 38. Thus, Downs cannot be said to teach meta-rights when cited to refer to end-user's usage rights. These limitations of previous inventions are highlighted in Appellants specification:

> **[0010]** However, there are limitations associated with the above-mentioned paradigms wherein only usage rights and conditions associated with content are specified by content owners or other grantors of rights. Once purchased by an end user, a consumer, or a distributor, of content along with its associated usage rights and conditions has no means to be legally passed on to a next recipient in a distribution chain. Further the associated usage rights have no provision for specifying rights to derive other rights, i.e. rights to modify, transfer, offer, grant, obtain, transfer, delegate, track, surrender, exchange, transport, exercise, revoke, or the like. Common content distribution models often include a multi-tier distribution and usage chain. Known DRM systems do not facilitate the ability to prescribe rights and conditions for all participants along a content distribution and usage chain. Therefore, it is difficult for a content owner to commercially exploit content unless the owner has a relationship with each party in the distribution chain.

Accordingly, the present invention solves the deficiencies of the prior art, most notably the Downs reference. Further, Downs fails to disclose each and every limitation of each independent claim 1, 15, and 38. Therefore, the allowance of independent claims 1, 15, and 38 is respectfully requested.

Appellants further contend that each of the dependent claims 2, 3, 5, 8, 9, 12-14, 17, 18, 29, 31-37, 39 and 40, all recite similar novel features of meta-rights as described above in reference to independent claims 1, 15, and 38. In light of Downs failing to disclose the concept of meta-rights, Appellants contend that each dependent claim listed above is appropriate for allowance. Therefore, the allowance of dependent claims 2, 3, 5, 8, 9, 12-14, 17, 18, 29, 31-37, 39 and 40 is respectfully requested.

B. Claim 10 is not anticipated by Downs.

Appellants contend that Downs fails to disclose "generating at least one offer...collecting usage rights and meta-rights available to be offered." as recited in Appellants' claim 10. Downs discloses an end-user submitting a purchase request to the Electronic Digital Content Store, data transmitted to an end-user's device, and a Clearinghouse to validate usage conditions purchased by the end-user. (Col. 18, steps 136-143). However, as stated above Downs does not disclose or teach the use of meta-rights when transferring data to an end-user, nor does Downs disclose or teach specifically the delivery of a license including user requested usage rights and meta-rights when transferring data to an end-user. Thus, Down cannot be said to disclose or teach making an offer of meta-rights to an end-user as is recited in Appellants' claim 10. Therefore, the allowance of claim 10 is respectfully requested.

C. Claims 4 and 30 are not anticipated by Downs.

Appellants contend that Downs fails to disclose "generating, by a second supplier, a second offer including rights derived from said meta-rights included in the first license," as recited in Appellants' claim 4 and "at least one derived right in the license is for a second consumer, the license includes...meta-rights permitting derived rights to be offered to a third consumer," as is recited in Appellants' claim 30. Downs discloses a license that is encrypted by the Content Provider and is used by the end-user device to be decrypted before content is released. (Col.23, lines 29-34, Col. 24, lines 47-62). However, Downs fails to disclose a second license generated by the second supplier as recited in Appellants' claims 4 and 30. Furthermore, as recited above, Downs does not disclose the use of meta-rights. Thus, it cannot be said that Downs discloses the distribution of meta-rights to subsequent end users as recited in Appellants' claims 4 and 30. Therefore, the allowance of claims 4 and 30 are respectfully requested.

D. Claim 6 is not anticipated by Downs.

Appellants contend that the Examiner has not met the burden of establishing a *prima facie* case for the rejection of dependent claim 6. The Examiner does not specifically address the novelties found in claim 6, such as "providing said first license as a customized draft license," and likewise has not addressed any relevant prior art that discloses the invention. In the Final Office Action of October 21, 2005 the Examiner refers to Downs, Col. 48, lines 1-25, and states that "If license is not validated or approved, the system determines if the user is entitled to the content, then authenticates and retransmits the content the user (negotiation)." (Final Office Action, page 6). It appears that the Examiner has taken the position that when content cannot be downloaded by the end-user and is subsequently retransmitted after a determination by the Electronic Digital Content Store (Downs, Col. 48, lines 1-25), that it is a negotiation. However, a simple retransmission of data upon a failed attempt does not constitute a negotiation as recited in Appellants' claim 6. Appellants' claim 6 recites a customized draft license provided to the consumer, which is an assemblage of rights and conditions based on a choice by the consumer. (Appellants' Paragraph 41). That draft license can then be accepted by the consumer and subsequently authenticated to become an authenticated license. (Appellants' Paragraphs 41-42). The choice and possible bargaining of rights (Appellants' Paragraph 41) is what constitutes the negotiation of the draft license as recited in Appellants' claim 6. Thus, Downs fails to disclose each and every limitation of Appellants' claim 6. Therefore, the allowance of claim 6 is respectfully requested.

E. Claim 7 is not anticipated by Downs.

Appellants contend that the Examiner has not met the burden of establishing a *prima facie* case for the rejection of dependent claim 7. The Examiner has not shown that each and every limitation of claim 7, such as "at least one grant including usage rights, meta-rights," is disclosed by the Downs reference. Specifically, Downs discloses a license that includes a Symmetric Key and Transaction Data, and in a further embodiment, a digital signature. (Col. 24, lines 36-47). However, Appellants claim 7 includes the grant of meta-rights with the license. Downs, for the reasons stated above, does not disclose the granting of meta-rights with a license. Therefore, the allowance of claim 7 is respectfully requested.

F. Claims 11 and 16 are not anticipated by Downs.

Appellants disagree with the Examiner's rejection of dependent claims 11 and 16 and contend that Downs does not disclose each and every element of the claims. Claim 11 discloses "applying the specific usage rights and meta-rights to the offer as a filter," and claim 16 discloses "the supplier component further comprises offer-templates...wherein said

offer-template includes one or more predetermined usage rights and meta-rights." Specifically, Downs discloses the use of SC(s) templates, defining parts and records as well as encryption methods, to make sure that anything that the Electronic Digital Content Store overrides is within the scope of its authorization. (Col. 31, lines 47-53, Col. 40, lines 3-8). However, as the Examiner concedes on page 8 of the Final Office Action, "Downs et al. did not explicitly describe a system in which conditions are filtered and applied based on user preferences," as is recited in Appellants' claim 11. Further, Downs also fails to disclose a supplier component that includes an offer template and consumer profile information that includes usage rights and meta-rights as is recited in Appellants' claim 16. Clearly, it cannot be said that Downs discloses the claimed features of claims 11 and 16 when it is conceded by the Examiner in another section of the Final Office Action, that in fact it does not.

Additionally, Appellants' claims 11 and 16 include the use of meta-rights. Downs, for the reasons stated above, does not disclose the use of meta-rights in a template. Therefore, the allowance of claims 11 and 16 is respectfully requested.

Conclusion

In light of Downs failing to disclose each and every limitation of the Appellants' present invention for the reasons set forth above, the Examiner's final rejection under 35 U.S.C. § 102(b) is inappropriate and Appellants respectfully request the Board to reverse each ground of rejection.

Rejection Under 35 U.S.C. § 103

1. Claim 19 is not unpatentable over Downs in view of Hitson.

A patent may not be obtained if the subject matter sought to be patented would be obvious to a person having ordinary skill in the art to which the subject matter pertains. 35 U.S.C. § 103. A determination of obviousness is a legal conclusion based on underlying findings of fact. *Velander v. Garner,* 348 F.3d 1359, 1363 (Fed. Cir. 2003). The Supreme Court in *Graham v. John Deere*, 383 U.S. 1 at 18, 148 USPQ 459 at 167 (1996), set forth the basic test for patentability under 35 U.S.C. §103:

> Under §103, the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background, the obviousness or non-obviousness of the subject matter is determined. Such secondary considerations as commercial success, long felt but unresolved need, failure of others, etc., might be utilized to give light to the circumstances surrounding the origin of the subject matter to be patented.

Moreover, in *In re Ehrreich* and *Avery*, 200 USPQ 504, 509-510 (CCPA 1979), the Court of Customs and Patent Appeals further clarified the basic test set forth in *Graham v. John Deere*:

> We must not here consider a reference in a vacuum, but against the background of the other references of record which may disprove theories and speculations in the reference or reveal previously undiscovered or unappreciated problems. The question in a §103 case is what the references would collectively suggest to one of ordinary skill in the art. *In re Simon*, 461 F.2d 1387, 174 USPQ 114 (CCPA 1972). It is only by proceeding in this manner that we may fairly determine the scope and content of the prior art according to the mandate of *Graham v. John Deere*, 383 US 1, 17, 148 USPQ 459, 467 (1966)(Emphasis in original.)

Thus, "[t]he mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination," *In re Mills*, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990). Further, analyzing the claimed invention as a whole in view of the prior art as a whole, one indicium of non-obviousness is a "teaching away" from the claimed invention by the prior art at the time the invention was made. *See U.S. v. Adams*, 148 USPQ 479 (1966). Essentially, teaching away from a claimed invention is a per se demonstration of lack of *prima facie* obviousness.

Where the prior art provides "only general guidance and is not specific as to the particular form of the invention or how to achieve it, [such a suggestion] may make an approach 'obvious to try,' but it does not make the invention obvious." *Ex parte Obukowicz*, 27 USPQ2d, 1063, 1065 (U.S. Patent and Trademark Office Board of Appeals and Interferences, 1992) and *In re O'Farrell*, 7 USPQ2d 1673, 1681 (Fed. Cir. 1988).

Factors including unexpected results, new features, solution of a different problem, novel properties are all considerations in the determination of obviousness. These secondary considerations (objective evidence of non-obviousness), as outlined in *Graham v. John Deere*, must be evaluated before reaching an ultimate decision under 35 U.S.C. §103. Accordingly, the recognition and solution of a problem is considered indicia of non-obviousness. For example, as the Court of Appeals stated in *In re Sponnable*, "[A] patentable invention may lie in the discovery of a source of a problem even though the remedy may be obvious once the source of the problem is identified. This is *part* of the 'subject matter as a whole' which should always be considered in determining the obviousness of an invention under 35 U.S.C. §103." Donald S. Chisum, *Chisum on Patents* § 5.04[7][c][ii], at 5-506 (Rel.

51, 1994) (quoting In re Sponnable 405 F.2d at 578, 585-86, 160 USPQ 237, 243-244 (CCPA 1969)(emphasis in original).

It should be noted that three criteria must be met to establish a *prima facie* case of obviousness. *M.P.E.P. §2143*. First, there must be some teaching, suggestion or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988). Second, there must be a reasonable expectation of success. *In re Rhinehart*, 531 F.2d 1048, 189 USPQ 143 (CCPA 1976). Last, the prior art must teach or suggest all the claim limitations. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974).

The Appellants respectfully contend that the Examiner has failed to set forth a *prima facie* case of obviousness, since the applied references, taken alone or in combination, fail to teach, disclose or suggest all limitations recited in the claimed invention. Specifically, the applied references at least fail to teach, suggest or disclose the use of meta-rights, as claimed. Accordingly, since the applied references fail to teach, suggest or disclose each and every claimed feature, the applied references cannot anticipate nor render obvious the claimed invention.

Specifically, Appellants contend that the combination of Downs and Hitson do not disclose, teach, or otherwise suggest the invention claimed in dependent claim 19 as asserted by the Examiner. Initially, it should be noted that the Downs reference, as described above, does not disclose or suggest the concept of meta-rights as recited in Appellants application. Nonetheless, the Examiner has again attempted to used Downs to apply the concept of rights transferring, when it clearly does not teach the concept.

Moreover, the Hitson reference, (U.S. Application 20020010759), discloses that a user may initially indicate content preferences, and the present invention may select content based on user preferences. (Hitson, Paragraph 11). However, the Hitson reference fails to disclose, teach, or otherwise suggest a means for "selecting preferred usage rights and meta-rights in the offer," as is recited in Appellants claim 19. The Examiner has again confused the concept of meta-rights with content rights and the Hitson reference does not mention or even suggest the selection of meta-rights during the selection process.

The cited references of Downs and Hitson fail to disclose, teach, or otherwise suggest in combination, the present invention. One of ordinary skill in the art would not look to combine two references that clearly do not teach or suggest the concept of meta-rights in an offer-selection process. Thus, the Examiner has not met a *prima facie* case for a showing of obviousness under U.S.C. § 103 and the Appellants respectfully request the allowance of claim 19.

Conclusion

In light of the combination of Downs and Hitson failing to disclose, teach, or suggest Appellants' present invention for the reasons set forth above, the Examiner's final rejection under 35 U.S.C. § 103(a) is inappropriate and Appellants respectfully request the Board to reverse this ground of rejection.

Respectfully submitted,
NIXON PEABODY, LLP

Date: **July 7, 2006**

/Carlos R. Villamar, Reg. # 43,224/
Carlos R. Villamar
Reg. No. 43, 224

CUSTOMER NO. 22204
NIXON PEABODY, LLP
401 9th Street, Suite 900
Washington D.C. 2004-2128
Tel: (202) 585-8250
Fax: (202) 585-8080

VIII. CLAIM APPENDIX

Claims Involved in the Appeal

1. A method for transferring usage rights adapted to be associated with items within a digital rights management system, said method comprising:

generating, by a supplier, at least one first offer including usage rights and meta-rights for the items, said usage rights defining a manner of use for the items, said meta-rights specifying rights to derive usage rights or other meta-rights for the items;

presenting, by the supplier, said offer to a first consumer in said system,

wherein the offer expresses what rights the consumer can acquire for the items;

receiving, by the supplier, a selection from the first consumer indicating desired usage rights and meta-rights; and

generating, by the supplier, a first license granting to the first consumer the usage rights and meta-rights for the items,

wherein the first license grants the usage rights and meta-rights that are selected by the first consumer during the receiving step.

2. The method of claim 1, wherein said license specifies one or more conditions which must be satisfied in order for said usage right to be exercised and one or more conditions which must be satisfied in order for said meta-rights to be exercised.

3. The method of claim 1, further comprising the step of receiving a request for a license from the first consumer.

4. The method of claim 1, further comprising:

receiving a request generated by a second consumer for a license including at least one of usage rights and meta-rights for the items;

generating, by a second supplier, a second offer including rights derived from said meta-rights included in the first license, wherein the second supplier is the first consumer; and

generating, by the second supplier, a second license including rights derived from said meta-rights included in the second offer.

5. The method of claim 1, wherein the item comprises digital content.

6. The method of claim 1, further comprising the steps of:
providing said first license as a customized draft license to the first consumer;
accepting a confirmation of said customized draft license from the first consumer; and
authenticating said draft license to create an authenticated license.

7. The method of claim 1, wherein said first license comprises a license identification, a digital signature, and at least one grant, said at least one grant including usage rights, meta-rights, a named principal designating the first consumer to whom rights are granted, and a condition list.

8. The method of claim 1, wherein the first supplier is at least one of a provider, distributor, retailer, consumer, and a user.

9. The method of claim 1, wherein the first consumer is at least one of a provider, distributor, retailer, consumer, and a user.

10. The method of claim 1, wherein the step of generating at least one offer comprises the steps of:

collecting usage rights and meta-rights available to be offered;

determining if the supplier has a right to offer the available usage rights and meta-rights;

terminating the generating of a set of offers, if a right to offer other usage and meta rights does not exist;

composing an offer based on available rights if the supplier has the right to offer other usage and meta rights; and

authenticating said offer.

11. The method of claim 10, wherein said composing step comprises:

determining if a consumer has requested an offer including specific usage rights and meta-rights;

applying the specific usage rights and meta-rights to the offer as a filter; and

determining if an offer template corresponds to the filtered offer and if so applying said offer template as an offer,

12. The method of claim 6, wherein said step of generating a first license further comprises the steps of:

determining if the suppler has the right to grant the rights;

terminating the step of customizing a draft license, if the supplier does not have the right to grant the rights;

analyzing one or more choices received from the consumer;

determining if the choices are acceptable; and

creating a draft license based on the choices if the choices are acceptable.

13. The method of claim 12, wherein said step of generating a first license further comprises:

presenting the draft license to the consumer;

re-negotiating a license if the first license is not approved by the consumer; and

authenticating the draft license if the first consumer approves the draft license.

14. The method of claim 1, wherein said usage rights specify rights to copy, transfer, loan, play, print, back-up, restore, delete, extract, embed, edit, authorize, install, or un-install the items.

15. A system for transferring usage rights adapted to be associated with an item, to be licensed in multi-tier channels of distribution with downstream rights and conditions assigned at least at one level, said system comprising:

a supplier component, comprising:

a supplier user interface module;

an offer generator module for generating an offer including at least usage rights and meta-rights for the item, the usage rights defining a manner of use for the item, the meta-rights specifying rights to derive usage rights or other meta-rights for the item;

a rights composer module for composing a draft license;

a repository for supplier's rights;

a supplier management database; and

a consumer component comprising:

a consumer user interface module;

an offer-consideration module configured to analyze the offers generated by the supplier component and select offers based on the analysis;

a repository for consumer's rights;

a consumer management database; and

a communication link coupling said supplier component and said consumer component,

wherein the rights composer module is configured to compose a license granting the usage rights and meta-rights that are selected by the offer-consideration module of the consumer component.

16. A system as recited in claim 15, wherein the supplier component further comprises offer-templates and consumer profile information, wherein said offer-template includes one or more predetermined usage rights and meta-rights, and wherein said consumer profile information comprises at least one of consumer identity information, account information, purchase history information, consumer preferences information, and credit rating information.

17. A system as recited in claim 15, wherein said consumer component further comprises a supplier-preference module for providing supplier information.

18. The system of claim 15, wherein said offer-consideration module comprises:

means for determining if the consumer can accept an offer;

means for applying selection logic to the offer;

means for specifying contingencies; and

means for authenticating choices and providing the choices to said supplier component.

19. The system of claim 18, wherein said means for applying comprises:

means for parsing the offer and selecting preferred usage rights and meta-rights in the offer;

means for filtering offers based on supplier preferences;

means for applying consumer preferences; and

means for selecting options based on the output of said means for parsing, said means for filtering, and said means for applying consumer preferences.

20-28. (Canceled)

29. The method of claim 1, the method being for generating a license to digital content to be used within the system for at least one of managing use and distribution of the digital content, wherein the license permits the first consumer to exercise the at least one meta-right and permits the first consumer to offer at least one derived right from the at least one meta-right and generate a license including the at least one derived right.

30. The method of claim 29, wherein the at least one derived right in the license is for a second consumer, the license includes usage rights to be exercised by the second consumer and meta-rights permitting derived rights to be offered to a third consumer.

31. The method of claim 1, wherein said method is implemented with one or more hardware and/or software components configured to perform the steps of the method.

32. The method of claim 1, wherein said method is implemented with one or more computer readable instructions embedded on a computer readable medium and configured to cause one or more computer processors to perform the steps of the method.

33. The system of claim 15, wherein said system is implemented with one or more hardware and/or software components.

34. The method of claim 1, wherein rights derived from said meta-rights include rights that revoke at least one of a usage right, and a meta-right.

35. The method of claim 1, wherein rights derived from said meta-rights include rights that reduce or expand at least one of a usage right, and a meta-right.

36. The system of claim 15, wherein rights derived from said meta-rights include rights that revoke at least one of a usage right, and a meta-right.

37. The system of claim 15, wherein rights derived from said meta-rights include rights that reduce or expand at least one of a usage right, and a meta-right.

38. A method for transferring usage rights adapted to be associated with an item within a digital rights management system, the method being performed by a consumer device within the system, the method comprising:

receiving, by the consumer device, from a supplier device within the system at least one offer including usage rights and meta-rights for the item,

wherein the usage rights define a manner of use for the item, and the meta-rights specify rights to derive usage rights or other meta-rights for the item, and

the offer expresses what rights the consumer device can acquire for the item;

selecting, by the consumer device, desired usage rights and meta-rights from the received offer,

wherein the selected rights express what rights the consumer device desires to acquire for the item; and

receiving, by the consumer device, a license from the supplier device,

wherein the received license grants the usage rights and meta-rights that are selected and provided by the consumer device.

39. The method of claim 38, wherein said method is implemented with one or more hardware and/or software components configured to perform the steps of the method.

40. The method of claim 38, wherein said method is implemented with one or more computer readable instructions embedded on a computer readable medium and configured to cause one or more computer processors to perform the steps of the method.

XI. EVIDENCE APPENDIX

There is no additional evidence relied upon in this brief.

X. RELATED PROCEEDINGS APPENDIX

There are no related appeals or interferences.

REQUEST FOR ORAL HEARING BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES	Docket Number (Optional) 111325-230300

I hereby certify that is correspondence is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on _____. Signature _____. Typed or printed name _____.	In re Application of Xin **WANG**, *et al.*	
	Application Number 10/162,212	Filed June 5, 2002
	For: **RIGHTS OFFERING AND GRANTING**	
	Group Art Unit 3621	Examiner Evens J. Augustin

Applicant hereby requests an oral hearing before the Board of Patent Appeals and Interferences from in the appeal of the above-identified application.

The fee for this Request for Oral Hearing is (37 CFR 1.17(d)) $**1,000.00**.

☐ Applicant claims small entity status. See 37 CFR 1.27. Therefore, the fee shown above is reduced by half, and the resulting fee is: $_____.

☐ A check in the amount of the fee is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☐ The Commissioner has already been authorized to charge fees in this application to a Deposit Account. I have enclosed a duplicate copy of this sheet.

☒ The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. 19-2380 (111325-230300).

☒ A petition for an extension of time under 37 CFR 1.136(b) (PTO/SB/23) is enclosed. For extensions of time in reexamination proceedings, see 37 CFR 1.550(c).

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

I am the

☐ applicant/inventor.

/Carlos R. Villamar, Reg. # 43,224/
Carlos R. Villamar
Signature

☐ assignee of record of the entire interest.
See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO-SB/96)

☒ attorney or agent of record.

Carlos R. Villamar , Reg. No. 43,224
Typed or printed name

☐ attorney or agent acting under 37 CFR 1.34(a).
Registration number if acting under 37 CFR 1.34(a). _____.

July 7, 2006
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

07/10/2006 IADDI 00000090 192380 10162212
01 FC:1463 1000.00 DA

☐ *Total of _____ forms are submitted.

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a)	Docket Number (Optional) 111325-230300

CERTIFICATE OF MAILING OR TRANSMISSION
[37 CFR 1.8(a)]

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to Mail Stop _____, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, or being facsimile transmitted to the USPTO at _____, on _____.

Signature: _____

Name: _____

In re Application of Xin **WANG**, *et al.*	
Application Number: 10/162,212	Filed: June 5, 2002
For **RIGHTS OFFERING AND GRANTING**	
Group Art Unit: 3621	Examiner: Evens J. Augustin

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above identified application.

The requested extension and appropriate entity fee are as follows (check time period desired):

- ☐ One month (37 CFR 1.17(a)(1)) - ($60/$120) $ _____
- ☐ Two months (37 CFR 1.17(a)(2)) - ($225/$450) $ _____
- ☐ Three months (37 CFR 1.17(a)(3)) - ($510/$1020) $ _____
- ☒ Four months (37 CFR 1.17(a)(4)) - ($795/$1590) $ 1,590.00
- ☐ Five months (37 CFR 1.17(a)(5)) - ($1080/$2160) $ _____

☐ Applicant claims small entity status.

☐ A check to cover the fee is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☒ The Commissioner has already been authorized to charge fees in this application to a Deposit Account.

☒ The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 19-2380. I have enclosed a duplicate copy of this sheet.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

I am the ☐ applicant/inventor

- ☐ assignee of record of the entire interest. See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96).
- ☐ attorney or agent of record.
- ☒ attorney or agent under 37 CFR 1.34(a). Registration number if acting under 37 CFR 1.34(a) **43,224**.

/Carlos R. Villamar, Reg. # 43,224/ Carlos R. Villamar
Signature

July 7, 2006
Date

Carlos R. Villamar
Typed or printed name

(202) 585-8204
Telephone Number

07/10/2006 JADD01 00000098 192380 10162212
01 FC:1254 1590.00 DA

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☐ Total of _____ forms are submitted.

SEND TO: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

10040013.1

FEE TRANSMITTAL FOR FY 2005

Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT | **$3,090.00**

Complete if Known	
Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin WANG, *et al.*
Examiner Name	Evens J. Augustin
Art Unit	3621
Attorney Docket No.	111325-230300

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☐ Credit Card ☐ Money Order ☐ Other ☐ None

☒ Deposit Account:

Deposit Account Number: 19-2380

Deposit Account Name: Nixon Peabody LLP

The Commissioner is authorized to: *(check all that apply)*

☒ Charge fee(s) indicated below ☒ Credit any overpayments

☒ Charge any additional fee(s)

☐ Charge fee(s) indicated below, **except for the filing fee** to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	300	2001	150	Utility filing fee	
1002	200	2002	100	Design filing fee	
1003	200	2003	100	Plant filing fee	
1004	300	2004	150	Reissue filing fee	
1005	200	2005	100	Provisional filing fee	

SUBTOTAL (1) ($) 0

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

		Extra Claims		Fee from below		Fee Paid
Total Claims	-20** =		X		=	0
Independent Claims	-3** =		X		=	0
Multiple Dependent			X		=	0

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	50	2202	25	Claims in excess of 20
1201	200	2201	100	Independent claims in excess of 3
1203	360	2203	180	Multiple dependent claim, if not paid
1204	200	2204	100	** Reissue independent claims over original patent
1205	50	2205	25	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 0

**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	
1051	130	2051	65	Surcharge – late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	120	2251	60	Extension for reply within first month	
1252	450	2252	225	Extension for reply within second month	
1253	1,020	2253	510	Extension for reply within third month	
1254	1,590	2254	795	Extension for reply within fourth month	1,590.00
1255	2,160	2255	1,080	Extension for reply within fifth month	
1401	500	2401	250	Notice of Appeal	
1402	500	2402	250	Filing a brief in support of an appeal	500.00
1403	1,000	2403	500	Request for oral hearing	1,000.00
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	500	2452	250	Petition to revive – unavoidable	
1453	1,500	2453	750	Petition to revive – unintentional	
1501	1,400	2501	700	Utility issue fee (or reissue)	
1502	800	2502	400	Design issue fee	
1503	1,100	2503	550	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	790	2809	395	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	790	2810	395	For each additional invention to be examined (37 CFR 1.129(b))	
1801	790	2801	395	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) **$3,090.00**

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(a)]

I hereby certify that this correspondence is being:

☐ deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop _____, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450

☐ transmitted by facsimile on the date shown below to the United States Patent and Trademark Office at (703) _____.

Date _____ Signature _____

Typed or printed name _____

SUBMITTED BY

Name *(Print/Type)*	Carlos R. Villamar	Registration No. *(Attorney/Agent)*	43,224	Complete *(if applicable)* Telephone	(202) 585-8204
Signature	/Carlos R. Villamar, Reg. # 43,224/ Carlos R. Villamar			Date	July 7, 2006

SEND TO: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

10039693.1

AF
TFW

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/162,212
Filing Date	June 5, 2002
First Named Inventor	Xin **WANG**, *et al.*
Group Art Unit	3621
Examiner Name	Evens J. Augustin
Total Number of Pages in This Submission	
Attorney Docket Number	111325-230300

ENCLOSURES *(check all that apply)*

- ☒ Fee Transmittal Form
  - ☐ 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)
- ☐ Response to Missing Parts/ Incomplete Application
  - ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Assignment Papers *(for an Application)*
- ☐ Drawing(s)
- ☐ 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
- ☒ Appeal Communication to Board of Appeals and Interferences
  - ☒ Request for Oral Hearing
- ☒ Appeal Communication to Group (Appeal Notice, Brief, Reply Brief)
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Application Data Sheet
- ☐ Request for Corrected Filing Receipt with Enclosures
- ☐ A self-addressed prepaid postcard for acknowledging receipt
- ☐ Other Enclosure(s) (*please identify below*):

Remarks

☒ 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	Carlos R. Villamar, Reg. 43,224 **Nixon Peabody LLP** 401 9th Street, N.W., Suite 900 Washington, D.C. 20004-2128
Signature	/Carlos R. Villamar, Reg. # 43,224/ Carlos R. Villamar
Date	July 7, 2006

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(a)]

I hereby certify that this correspondence is being:

- ☐ deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop _____, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450
- ☐ transmitted by facsimile on the date shown below to the United States Patent and Trademark Office at (703) _____.

_____ Date

_____ Signature

_____ Typed or printed name

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/162,212	06/05/2002	Xin Wang	111325-104	3700

22204 7590 12/12/2006

NIXON PEABODY, LLP
401 9TH STREET, NW
SUITE 900
WASHINGTON, DC 20004-2128

EXAMINER	
AUGUSTIN, EVENS J	
ART UNIT	**PAPER NUMBER**
3621	

DATE MAILED: 12/12/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/162,212
Filing Date: June 05, 2002
Appellant(s): WANG ET AL.

Carlos Villamar
Nixon Peabody LLP
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 07/07/06 appealing from the Office action mailed on 2/16/06.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6226618	Downs et al.	5-2001
20020010759	Hitson et al.	2-2002

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Status of Claims

1. Claims 1-19 and 29-40 have been examined.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) The invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

3. Claims 1-13, 15-18 and 29-40 are rejected under 35 U.S.C. 102(b) as being anticipated by Downs et al. (U.S 6,226,618).

As per claims 1-13, 15-18 and 20-28, Downs et al. discloses a system for Electronic Content Delivery, comprising of:

- Usage rights and other downstream rights (column 9, lines 33-35, column 10, lines 15-18). The system also presents the digital content offering to consumers (column 48, lines 32-36). The digital content owners receive requests from distributors to sale digital content (column 42, lines 65-67, column 43, lines 1-2). The content owners then license the original content to the distributors (Electronic Digital Content Store) (column 9, lines 5-13)

- The content suppliers specify usage rights. They also specify that the distributors can customized those original usage rights and provide secondary usage rights (column 9, lines 33-36 - column 10, lines 13-18)
- Receiving a request for usage rights of digital content from a second consumer or end-user (column 18, step 136). The content stores can offer contents with their own customized usage conditions, as long as the customized conditions don't invalidate the original conditions set by the content provider (column 21, lines 23-36). Those secondary usage rights are then validated (column 21, lines 36-51)
- The Secure Digital Content Electronic Distribution system uses multiple formats of digital content such as pictures, movies, videos, music, programs, multimedia and games (column 6, lines 45-48)
- The digital content owners receive requests from distributors to sale digital content (column 42, lines 65-67, column 43, lines 1-2). The content owners then license the original content to the distributors (Electronic Digital Content Store) (column 9, lines 5-13). The content stores offer their own customized usage conditions to end-users, as long as the customized conditions don't invalidate the original conditions set by the content provider (column 21, lines 23-36). The end-users don't get a license until the conditions are validated/authenticated throughout the supply chain (column 22, lines 26-52)
- The license comprises of transaction data (column 24, lines 37-38). The transaction data includes unique transaction ID (column 23, line 62). The license also comprises of digital signature (column 24, lines 45-47) and usage conditions (column 24, line 10-11)

- The first supplier as the content proprietor (column 9, lines 5-15). The first consumers are distributors such as electronic content stores (column 9, lines 63-65)
- Usage rights attached to contents offered to consumers (column 21, lines 30-33). The system verifies that the supplier has the authority to distribute the content (column 22, lines 45-50), and the supplier has the right to receive content (column 42, lines 52-55). The system also verifies that the requested usage conditions are in agreement with the allowable conditions (column 26, lines 20-23). If the conditions are not valid, the system terminates the request (column 10, lines 65-67). Otherwise, the system validates and sends license authorization to consumer (column 10, lines 60-64)
- If license is not validated or approved, the system determines if the user is entitled to the content, then authenticates and retransmits the content the user(negotiation) (column 48, lines 1-25)
- The supplier of digital content has an interface for multi users (column 49, lines 13-17). The supplier also generates the content for distribution (column 9, lines 15-20), with usage conditions (column 9, line 33), kept in a database (figure 1A, item 60). The consumer device also has a user interface (column 20, line 120), a license database (figure 1D, item 197) and is interconnected to a public network such as the internet (column 23, lines 5-15)
- Offer templates containing predetermined usage and meta-rights (column 26, line 62). The system also keeps the identity of the purchaser (column 20, lines 36-37)
- The consumer device has the ability to provide data/digital content originated from the content provider (column 79, lines 35-41

- The end user devices such personal computers (column 79, line 16-17) and the packaged application provide means for the user to accept digital content (column 80, lines 20-25). The system also provides means to specify and apply usage rights and to authenticate those rights (column 42, lines 35-56)
- The invention includes the means and devices to (hardware and software combination) (columns 53, lines 65-67, column 54, lines 1-3) implement the above steps
- The invention has the ability to revoke licenses and create a revocation of licenses that have been revoked (column 37, lines 65-67, column 38, lines 10-20)

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Downs et al. (U.S. 6,226,618).

As per claims 14, Downs et al. discloses a system comprising of:

- The system currently uses audio data as an example and specifies usage rights accordingly (column 59, Lines 37-67).
- The system also supports other types of digital content such as pictures, movies, videos, music, programs, multimedia and games (column 6, lines 45-48).

Downs et al. did not explicitly describe a system that wherein the usage rights are associated with copy, transfer, loan, play, print, back-up, restore, delete, extract embed, edit, authorize, install/un-install. However, Downs et al. discloses a system that supports digital content such as pictures, movies, videos, music, programs, multimedia and games (column 6, lines 47-48). Therefore, it would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to specify usage conditions for a particular digital content in order to include the rights of as many digital content formats as possible.

6. Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over Downs et al. (U.S 6,226,618) in view of Hitson et al. (US 20020010759)

As per claim 19, Downs et al. discloses a system comprising of:

- Usage rights and other downstream rights (column 9, lines 33-35, column 10, lines 15-18). The system also presents the digital content offering to consumers (column 48, lines 32-36). The digital content owners receive requests from distributors to sale digital content (column 42, lines 65-67, column 43, lines 1-2). The content owners then license the original content to the distributors (Electronic Digital Content Store) (column 9, lines 5-13)
- The content suppliers specify usage rights. They also specify that the distributors can customized those original usage rights and provide secondary usage rights (column 9, lines 33-36 - column 10, lines 13-18)
- Receiving a request for usage rights of digital content from a second consumer or end-user (c18, step 136). The content stores can then can offer content with their own

customized usage conditions, as long as the customized conditions don't invalidate the original conditions set by the content provider (column 21, lines 23-36). The stores send those usage conditions to the user and a clearinghouse for validation (column 21, lines 36-51)

Downs et al. did not explicitly describe a system in which conditions are filtered and applied, based on user preferences. However, Hitson et al. discloses a system and method for content distribution in which content is selected based on user preferences (page 1, paragraph 11). Therefore, it would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to design a system in which contents are filtered and applied, based on user preferences. It would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to design such system because it would provide personalized content delivery, and would provide economic incentives to content providers by supplying a means of target marketing to users based upon user content preferences page 1, paragraph 11).

(10) Response to Argument

Argument 1: Claims 1-13, 15-18 and 29-40 not anticipated by downs.

Response 1: The USPTO respectfully disagrees with appellant's argument.

For example, with regard to claim 1 (**see appendix A**):

Supplier = Content provider
First Consumer = Digital content store or distributor
Usage Rights = Usage conditions such as copy protection
First license = Digital certificate given to distributor
Meta-rights = Sub-rights, or additional usage conditions derived from the usage rights

The limitation of "*generating, by a supplier, at least one first offer, including usage rights and meta-rights for the items*", the prior art by Downs et al. teaches that content providers

(entity that supplies the content), providing (equivalent to generating) **usage conditions (equivalent to usage rights)**. The content providers also stipulate that the content stores or distributors can add or narrow the original usage rights (**meta-rights or rights derived from usage rights or sub-rights**) (column 21, lines 30-36).

The limitation of "*said usage rights defining a manner of use for the items*", the prior art by Downs et al. teaches that **usage rights are restrictions on how many copies can be made for a particular content, which is manner in which the content can be used** (column 9, 32-34, col. 26, lines 10-12).

The limitation of "*said meta-rights specifying rights to derive usage rights or other meta-rights for the items*", the prior art by Downs et al. teaches that **content stores or distributors can add or narrow the original usage rights (sub-rights)** (column 21, lines 30-36).

The limitation of "*Presenting, by the supplier, said offer to a first consumer in said system*", the prior art by Downs et al. teaches that content providers set and **transmit (equivalent to presenting) the usage conditions to the content stores** (column 21, 30-32), which are the **first customers or distributors** of the content providers.

The limitation of "*wherein the offer expresses what rights the consumer can acquire for the items;*", the prior art by Downs et al. teaches that the offer expresses the rights that consumer can acquire (column 59-60, lines 17-30).

The limitation of "*Receiving, by the supplier, a selection from the first consumer indicating desired usage rights and meta-rights*", the prior art by Downs et al. teaches that distributors **(first customer) making a request to digital content owners to sale digital content** (column 42, lines 65-67, column 43, lines 1-2). The two parties then come to an

agreement (column 43, lines 4-5). Inherently, the content provider **receives the request**, and the distributors **agree to sell content usage rights and meta-rights**.

The limitation of "*Generating, by the supplier, a first license granting to the first consumer the usage rights and meta rights for the items*", the prior art by Downs et al. teaches that after the agreement between the content provider and the distributor (first customer), **a digital certificate is created (same as generated)** and sent to the distributor (column 43, lines 14-18). This mechanism provides **licensing authorization** by **enabling intermediate (in this case the distributors)** or End-User(s) to unlock content after verification of a successful completion of a licensing transaction (column 7, lines 13-16).

The limitation of "*Wherein the first license grants the usage rights and meta-rights that are selected by the first consumer during the receiving step*", the prior art by Downs et al. teaches that After the agreement between the content provider and the distributor (first customer), **a digital certificate is created and sent to the distributor** (column 43, lines 14-18). **Inherently the agreement and certificate is for the content/usage rights requested by the distributor.**

Claim 38 is the same as claim 1except that it is written from the consumer's point of point. Therefore, all of the above responses apply to claim 35 as well.

With regard to claim 15, the prior by Downs et al. teaches that the supplier of digital content has **an interface for multi users** (column 49, lines 13-17). The supplier also **generates the content for distribution (equivalent to generating offer)** (column 9, lines 15-20), with **usage conditions** or rights (column 9, line 33), **kept in a database** (figure 1A, item 60). The

consumer device also has a user interface (column 20, line 120), a **license database** (figure 1D, item 197) and is interconnected to a public network such as the internet (column 23, lines 5-15). The content suppliers also specify that the **distributors can customized those original usage rights and provide secondary usage rights (sub-rights or meta rights)** (column 9, lines 33-36 - column 10, lines 13-18). The Distributors (first customer) make a request to digital content owners to sale digital content (column 42, lines 65-67, column 43, lines 1-2). The two parties then come to an agreement (column 43, lines 4-5). Inherently, the content provider receives the request, and the distributors agree to sell content usage rights and meta-rights. After the agreement between the content provider and the distributor (first customer), a digital certificate is created (**same as generated or drafted**) and sent to the distributor (column 43, lines 14-18)

According to applicant's specification (par. 45), "**rights repository** 412 stores rights granted to the user of supplier component 402 and may include functions for indexing, searching and updating the rights stored within". Therefore, the **rights repository is interpreted as** a **license database** (figure 1D, item 197, column 80, line 30) inside the user device, that houses the license or rights to the content. **A database inherently has the aspects of indexing, searching and updating.**

Consumer and supplier are interconnected to a public network such as the internet (**communication link**) (column 23, lines 5-15). The prior art by Downs et al. also contains an order analysis to analyze and validate orders (column 43, lines 59-67, and column 44, lines 1-5). Before transmitting the Content SC(s) to the End-User Device(s), **analysis and verifications are performed on the End-User's request** (column 70, lines 4-6).

Argument # 2: Claim 10, not anticipated by Downs et al.

Response #2:

Claim 10	**Downs et al.**
Collecting rights and meta-rights to be offered to supplier	Content provider has usage conditions or rights processing mechanism that allows it to specify product uses and restrictions (column 52, lines 9-22). This is equivalent to collecting rights and meta-rights to be offered to supplier
Determining if content store has right to content	The system verifies that content has the proper authorization to sell the content (column 22, lines 42-48)
Terminating the offer if rights are not valid	Downs provides licensing authorization by enabling **intermediate (distributors or content stores or vendors) or End-User(s)** to unlock content after verification of a successful completion of a licensing transaction (column 7, lines 13-16, column 10, lines 50-67). Terminate the request if not verifiable (column 10, lines 65-67) **(request used synonymously with offer, since the request is being made for a content offering)**
Compose offer if rights are valid	Once verifications are satisfied allow consumers to use content column 10, lines 61-65)
Authenticate offer	Validates the integrity and **authenticity** of the information in the request **(request used synonymously with offer, since the request is being made for a content offering)** (column 10, lines 55-56)

Argument # 3: Claims 4 and 30, not anticipated by Downs et al.

Response #3: With regard to claim 4, the prior art by Downs et al. teaches the aspect of receiving a request for usage rights of digital content from a second consumer **or end-user** (column 18, step 136). The content stores can offer contents with their own customized usage conditions, as long as the customized conditions don't invalidate the original conditions set by the content provider (column 21, lines 23-36). Those secondary usage rights or **meta-rights** are then validated (column 21, lines 36-51).

With regard claim 30, the system specifies whether or not the content can be reproduced to **another end user external device** (column 20, 45-48). In this case the other end user external device is considered **the third consumer**.

Argument # 4: Claims 6, 7, 11 and 16, not anticipated by Downs et al.

Response #4: Claim 6- Downs et al. teaches that after the agreement between the content provider and the distributor (first customer), **a digital certificate is created (same as generated)** and sent to the distributor (column 43, lines 14-18). This mechanism provides **licensing authorization** by **enabling intermediate (in this case the distributors)** or End-User(s) to unlock content after verification of a successful completion of a licensing transaction (column 7, lines 13-16).

Claim 7-The license comprises of transaction data (column 24, lines 37-38). The transaction data includes unique transaction ID (column 23, line 62). The license also comprises of digital signature (column 24, lines 45-47) and usage conditions (column 24, line 10-11)

Claim 11 – Downs teaches that the system verifies that the supplier has the authority to distribute the content (column 22, lines 45-50), and the supplier has the right to receive content (column 42, lines 52-55). The system also verifies that the requested usage conditions are in agreement with the allowable conditions (column 26, lines 20-23). In this case u**sage conditions are considered filtered, as they restrict the usage of content, based on those conditions.** If the conditions are not valid, the system terminates the request (column 10, lines 65-67). Otherwise, the system validates and sends license authorization to consumer (column 10, lines 60-64).

Claim 16 - Downs teaches the aspects Offer templates containing predetermined usage and meta-rights (column 26, line 62). The system also keeps the identity of the purchaser (column 20, lines 36-37)

Argument # 5: Claim 19, not unpatentable over Downs et al., in view of Hitson et al.

Response #5: To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations

With regard to motivation, Downs et al. did not explicitly describe a system in which conditions are filtered and applied, **based on user preferences**. However, Hitson et al. discloses a system and method for content distribution in which content is selected based on

user preferences (page 1, paragraph 11). Therefore, it would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to design a system in which contents are filtered and applied, based on user preferences. It would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to design such system because **(Motivation)** it would provide personalized content delivery, and would provide economic incentives to content providers by supplying a means of target marketing to users based upon user content preferences page 1, paragraph 11).

There is reasonable expectation of success. The prior art by Downs et al. deals with method and apparatus of securely providing data to a user's system, the Hitson et al. deals with system and method for composing and delivering music and other audio content via the Internet. Since both references deal with content distribution, there is reasonable expectation of success.

The references by Downs et al. and Hitson et al., combined implicitly or explicitly teach all of the limitations of claimed invention. Therefore, the three basic criteria to establish a prima facie case of obviousness are met.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Evens J. Augustin EA
October 28, 2006
Art Unit 3621

Conferees:

Vincent Millin – Appeal Conference Specialist

Andrew Fisher – Supervisor Art Unit 3621 AF

Evens Augustin – Examiner Art Unit 3621 EA

Appendix A

Limitation #	Claims 1 and 38	Prior Art (Downs, US 6226618)
1	Generating, by a supplier, at least one first offer, including **usage rights** and **meta-rights** for the items, ,	Content providers (entity that supplies the content), providing (equivalent to generating) **usage conditions** (equivalent to **usage rights**), The content providers also stipulate that the content stores or distributors can add or narrow the original usage rights (**meta-rights or rights derived from usage rights**) (column 21, lines 30-36).
2	said usage rights defining a manner of use for the items	**Usage rights being copy restrictions, which is manner in which the content can be used** (column 9, 32-34, col. 26, lines 10-12).
3	said meta-rights specifying rights to derive usage rights or other meta-rights for the items	**Content stores or distributors can add or narrow the original usage rights (sub-rights)** (column 21, lines 30-36)
4	Presenting, by the supplier, said offer to a first consumer in said system,	Content providers set and **transmit (equivalent to presenting) the usage conditions to the content stores** (column 21, 30-32), which are the **first customers** or distributors of the content providers
5	wherein the offer expresses what rights the consumer can acquire for the items;	Offer expresses the rights that consumer can acquire (column 59-60, lines 17-30)
6	Receiving, by the supplier, a selection from the first consumer indicating desired usage rights and meta-rights	Distributors **(first customer) making a request to digital content owners to sale digital content** (column 42, lines 65-67, column 43, lines 1-2). The two parties then come to an agreement (column 43, lines 4-5). Inherently, the content provider **receives the request**, and the distributors **agree to sell content usage rights and meta-rights.**
7	Generating, by the supplier, a first license granting to the first consumer the usage rights and meta rights for the items	After the agreement between the content provider and the distributor (first customer), a **digital certificate is created (same as generated)** and sent to the distributor (column 43, lines 14-18). This mechanism provides **licensing authorization** by enabling **intermediate** (in this case the distributors) or End-User(s) to unlock content after verification of a successful completion of a licensing transaction (column 7, lines 13-16)
8	Wherein the first license grants the usage rights and meta-rights that are selected by the first consumer during the receiving step	After the agreement between the content provider and the distributor (first customer), **a digital certificate is created and sent to the distributor** (column 43, lines 14-18). **Inherently the agreement and certificate is for the content/usage rights requested by the distributor**

Supplier = Content provider
First Consumer = digital content store or distributor
Usage Rights = Usage conditions such as copy protection
First license = Digital certificate given to distributor
Meta-rights = Subrights, or addtionnal usage conditions derived from the usage rights

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/162,212	06/05/2002	Xin Wang	111325-104	3700

22204 7590 02/13/2007

NIXON PEABODY, LLP
401 9TH STREET, NW
SUITE 900
WASHINGTON, DC 20004-2128

EXAMINER	
AUGUSTIN, EVENS J	
ART UNIT	**PAPER NUMBER**
3621	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
2 MONTHS	02/13/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/162,212
Filing Date: June 05, 2002
Appellant(s): WANG ET AL.

Carlos Villamar
Nixon Peabody LLP
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 07/07/06 appealing from the Office action mailed on 2/16/06.

(1) Real Party in Interest

The real party in interest is ContentGuard Holdings, Inc.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6226618	Downs et al.	5-2001
20020010759	Hitson et al.	2-2002

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Status of Claims

1. Claims 1-19 and 29-40 have been examined.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) The invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

3. Claims 1-13, 15-18 and 29-40 are rejected under 35 U.S.C. 102(b) as being anticipated by Downs et al. (U.S 6,226,618).

As per claims 1-13, 15-18 and 20-28, Downs et al. discloses a system for Electronic Content Delivery, comprising of:

- Usage rights and other downstream rights (column 9, lines 33-35, column 10, lines 15-18). The system also presents the digital content offering to consumers (column 48, lines 32-36). The digital content owners receive requests from distributors to sale digital content (column 42, lines 65-67, column 43, lines 1-2). The content owners then license the original content to the distributors (Electronic Digital Content Store) (column 9, lines 5-13)

- The content suppliers specify usage rights. They also specify that the distributors can customized those original usage rights and provide secondary usage rights (column 9, lines 33-36 - column 10, lines 13-18)
- Receiving a request for usage rights of digital content from a second consumer or end-user (column 18, step 136). The content stores can offer contents with their own customized usage conditions, as long as the customized conditions don't invalidate the original conditions set by the content provider (column 21, lines 23-36). Those secondary usage rights are then validated (column 21, lines 36-51)
- The Secure Digital Content Electronic Distribution system uses multiple formats of digital content such as pictures, movies, videos, music, programs, multimedia and games (column 6, lines 45-48)
- The digital content owners receive requests from distributors to sale digital content (column 42, lines 65-67, column 43, lines 1-2). The content owners then license the original content to the distributors (Electronic Digital Content Store) (column 9, lines 5-13). The content stores offer their own customized usage conditions to end-users, as long as the customized conditions don't invalidate the original conditions set by the content provider (column 21, lines 23-36). The end-users don't get a license until the conditions are validated/authenticated throughout the supply chain (column 22, lines 26-52)
- The license comprises of transaction data (column 24, lines 37-38). The transaction data includes unique transaction ID (column 23, line 62). The license also comprises of digital signature (column 24, lines 45-47) and usage conditions (column 24, line 10-11)

- The first supplier as the content proprietor (column 9, lines 5-15). The first consumers are distributors such as electronic content stores (column 9, lines 63-65)
- Usage rights attached to contents offered to consumers (column 21, lines 30-33). The system verifies that the supplier has the authority to distribute the content (column 22, lines 45-50), and the supplier has the right to receive content (column 42, lines 52-55). The system also verifies that the requested usage conditions are in agreement with the allowable conditions (column 26, lines 20-23). If the conditions are not valid, the system terminates the request (column 10, lines 65-67). Otherwise, the system validates and sends license authorization to consumer (column 10, lines 60-64)
- If license is not validated or approved, the system determines if the user is entitled to the content, then authenticates and retransmits the content the user(negotiation) (column 48, lines 1-25)
- The supplier of digital content has an interface for multi users (column 49, lines 13-17). The supplier also generates the content for distribution (column 9, lines 15-20), with usage conditions (column 9, line 33), kept in a database (figure 1A, item 60). The consumer device also has a user interface (column 20, line 120), a license database (figure 1D, item 197) and is interconnected to a public network such as the internet (column 23, lines 5-15)
- Offer templates containing predetermined usage and meta-rights (column 26, line 62). The system also keeps the identity of the purchaser (column 20, lines 36-37)
- The consumer device has the ability to provide data/digital content originated from the content provider (column 79, lines 35-41

- The end user devices such personal computers (column 79, line 16-17) and the packaged application provide means for the user to accept digital content (column 80, lines 20-25). The system also provides means to specify and apply usage rights and to authenticate those rights (column 42, lines 35-56)
- The invention includes the means and devices to (hardware and software combination) (columns 53, lines 65-67, column 54, lines 1-3) implement the above steps
- The invention has the ability to revoke licenses and create a revocation of licenses that have been revoked (column 37, lines 65-67, column 38, lines 10-20)

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Downs et al. (U.S 6,226,618).

As per claims 14, Downs et al. discloses a system comprising of:

- The system currently uses audio data as an example and specifies usage rights accordingly (column 59, Lines 37-67).
- The system also supports other types of digital content such as pictures, movies, videos, music, programs, multimedia and games (column 6, lines 45-48).

Downs et al. did not explicitly describe a system that wherein the usage rights are associated with copy, transfer, loan, play, print, back-up, restore, delete, extract embed, edit, authorize, install/un-install. However, Downs et al. discloses a system that supports digital content such as pictures, movies, videos, music, programs, multimedia and games (column 6, lines 47-48). Therefore, it would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to specify usage conditions for a particular digital content in order to include the rights of as many digital content formats as possible.

6. Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over Downs et al. (U.S 6,226,618) in view of Hitson et al. (US 20020010759)

As per claim 19, Downs et al. discloses a system comprising of:

- Usage rights and other downstream rights (column 9, lines 33-35, column 10, lines 15-18). The system also presents the digital content offering to consumers (column 48, lines 32-36). The digital content owners receive requests from distributors to sale digital content (column 42, lines 65-67, column 43, lines 1-2). The content owners then license the original content to the distributors (Electronic Digital Content Store) (column 9, lines 5-13)
- The content suppliers specify usage rights. They also specify that the distributors can customized those original usage rights and provide secondary usage rights (column 9, lines 33-36 - column 10, lines 13-18)
- Receiving a request for usage rights of digital content from a second consumer or end-user (c18, step 136). The content stores can then can offer content with their own

customized usage conditions, as long as the customized conditions don't invalidate the original conditions set by the content provider (column 21, lines 23-36). The stores send those usage conditions to the user and a clearinghouse for validation (column 21, lines 36-51)

Downs et al. did not explicitly describe a system in which conditions are filtered and applied, based on user preferences. However, Hitson et al. discloses a system and method for content distribution in which content is selected based on user preferences (page 1, paragraph 11). Therefore, it would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to design a system in which contents are filtered and applied, based on user preferences. It would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to design such system because it would provide personalized content delivery, and would provide economic incentives to content providers by supplying a means of target marketing to users based upon user content preferences page 1, paragraph 11).

(10) Response to Argument

Argument 1: Claims 1-13, 15-18 and 29-40 not anticipated by downs.

Response 1: The USPTO respectfully disagrees with appellant's argument.

For example, with regard to claim 1 (**see appendix A**):

Supplier = Content provider
First Consumer = Digital content store or distributor
Usage Rights = Usage conditions such as copy protection
First license = Digital certificate given to distributor
Meta-rights = Sub-rights, or additional usage conditions derived from the usage rights

The limitation of "*generating, by a supplier, at least one first offer, including usage rights and meta-rights for the items*", the prior art by Downs et al. teaches that content providers

(entity that supplies the content), providing (equivalent to generating) **usage conditions (equivalent to usage rights)**. The content providers also stipulate that the content stores or distributors can add or narrow the original usage rights (**meta-rights or rights derived from usage rights or sub-rights**) (column 21, lines 30-36).

The limitation of "*said usage rights defining a manner of use for the items*", the prior art by Downs et al. teaches that **usage rights are restrictions on how many copies can be made for a particular content, which is manner in which the content can be used** (column 9, 32-34, col. 26, lines 10-12).

The limitation of "*said meta-rights specifying rights to derive usage rights or other meta-rights for the items*", the prior art by Downs et al. teaches that **content stores or distributors can add or narrow the original usage rights (sub-rights)** (column 21, lines 30-36).

The limitation of "*Presenting, by the supplier, said offer to a first consumer in said system*", the prior art by Downs et al. teaches that content providers set and **transmit (equivalent to presenting) the usage conditions to the content stores** (column 21, 30-32), which are the **first customers or distributors** of the content providers.

The limitation of "*wherein the offer expresses what rights the consumer can acquire for the items;*", the prior art by Downs et al. teaches that the offer expresses the rights that consumer can acquire (column 59-60, lines 17-30).

The limitation of "*Receiving, by the supplier, a selection from the first consumer indicating desired usage rights and meta-rights*", the prior art by Downs et al. teaches that distributors **(first customer) making a request to digital content owners to sale digital content** (column 42, lines 65-67, column 43, lines 1-2). The two parties then come to an

agreement (column 43, lines 4-5). Inherently, the content provider **receives the request**, and the distributors **agree to sell content usage rights and meta-rights**.

The limitation of "*Generating, by the supplier, a first license granting to the first consumer the usage rights and meta rights for the items*", the prior art by Downs et al. teaches that after the agreement between the content provider and the distributor (first customer), **a digital certificate is created (same as generated)** and sent to the distributor (column 43, lines 14-18). This mechanism provides **licensing authorization** by **enabling intermediate (in this case the distributors)** or End-User(s) to unlock content after verification of a successful completion of a licensing transaction (column 7, lines 13-16).

The limitation of "*Wherein the first license grants the usage rights and meta-rights that are selected by the first consumer during the receiving step*", the prior art by Downs et al. teaches that After the agreement between the content provider and the distributor (first customer), **a digital certificate is created and sent to the distributor** (column 43, lines 14-18). **Inherently the agreement and certificate is for the content/usage rights requested by the distributor.**

Claim 38 is the same as claim 1except that it is written from the consumer's point of point. Therefore, all of the above responses apply to claim 35 as well.

With regard to claim 15, the prior by Downs et al. teaches that the supplier of digital content has **an interface for multi users** (column 49, lines 13-17). The supplier also **generates the content for distribution (equivalent to generating offer)** (column 9, lines 15-20), with **usage conditions** or rights (column 9, line 33), **kept in a database** (figure 1A, item 60). The

consumer device also has a user interface (column 20, line 120), a **license database** (figure 1D, item 197) and is interconnected to a public network such as the internet (column 23, lines 5-15). The content suppliers also specify that the **distributors can customized those original usage rights and provide secondary usage rights (sub-rights or meta rights)** (column 9, lines 33-36 - column 10, lines 13-18). The Distributors (first customer) make a request to digital content owners to sale digital content (column 42, lines 65-67, column 43, lines 1-2). The two parties then come to an agreement (column 43, lines 4-5). Inherently, the content provider receives the request, and the distributors agree to sell content usage rights and meta-rights. After the agreement between the content provider and the distributor (first customer), a digital certificate is created (**same as generated or drafted**) and sent to the distributor (column 43, lines 14-18)

According to applicant's specification (par. 45), "**rights repository** 412 stores rights granted to the user of supplier component 402 and may include functions for indexing, searching and updating the rights stored within". Therefore, the **rights repository is interpreted as** a **license database** (figure 1D, item 197, column 80, line 30) inside the user device, that houses the license or rights to the content. **A database inherently has the aspects of indexing, searching and updating**.

Consumer and supplier are interconnected to a public network such as the internet (**communication link**) (column 23, lines 5-15). The prior art by Downs et al. also contains an order analysis to analyze and validate orders (column 43, lines 59-67, and column 44, lines 1-5). Before transmitting the Content SC(s) to the End-User Device(s), **analysis and verifications are performed on the End-User's request** (column 70, lines 4-6).

Argument # 2: Claim 10, not anticipated by Downs et al.

Response #2:

Claim 10	**Downs et al.**
Collecting rights and meta-rights to be offered to supplier	Content provider has usage conditions or rights processing mechanism that allows it to specify product uses and restrictions (column 52, lines 9-22). This is equivalent to collecting rights and meta-rights to be offered to supplier
Determining if content store has right to content	The system verifies that content has the proper authorization to sell the content (column 22, lines 42-48)
Terminating the offer if rights are not valid	Downs provides licensing authorization by enabling **intermediate (distributors or content stores or vendors) or End-User(s)** to unlock content after verification of a successful completion of a licensing transaction (column 7, lines 13-16, column 10, lines 50-67). Terminate the request if not verifiable (column 10, lines 65-67) **(request used synonymously with offer, since the request is being made for a content offering)**
Compose offer if rights are valid	Once verifications are satisfied allow consumers to use content column 10, lines 61-65)
Authenticate offer	Validates the integrity and **authenticity** of the information in the request **(request used synonymously with offer, since the request is being made for a content offering)** (column 10, lines 55-56)

Argument # 3: Claims 4 and 30, not anticipated by Downs et al.

Response #3: With regard to claim 4, the prior art by Downs et al. teaches the aspect of receiving a request for usage rights of digital content from a second consumer **or end-user** (column 18, step 136). The content stores can offer contents with their own customized usage conditions, as long as the customized conditions don't invalidate the original conditions set by the content provider (column 21, lines 23-36). Those secondary usage rights or **meta-rights** are then validated (column 21, lines 36-51).

With regard claim 30, the system specifies whether or not the content can be reproduced to **another end user external device** (column 20, 45-48). In this case the other end user external device is considered **the third consumer**.

Argument # 4: Claims 6, 7, 11 and 16, not anticipated by Downs et al.

Response #4: Claim 6- Downs et al. teaches that after the agreement between the content provider and the distributor (first customer), **a digital certificate is created (same as generated)** and sent to the distributor (column 43, lines 14-18). This mechanism provides **licensing authorization** by **enabling intermediate (in this case the distributors)** or End-User(s) to unlock content after verification of a successful completion of a licensing transaction (column 7, lines 13-16).

Claim 7-The license comprises of transaction data (column 24, lines 37-38). The transaction data includes unique transaction ID (column 23, line 62). The license also comprises of digital signature (column 24, lines 45-47) and usage conditions (column 24, line 10-11)

Claim 11 – Downs teaches that the system verifies that the supplier has the authority to distribute the content (column 22, lines 45-50), and the supplier has the right to receive content (column 42, lines 52-55). The system also verifies that the requested usage conditions are in agreement with the allowable conditions (column 26, lines 20-23). In this case u**sage conditions are considered filtered, as they restrict the usage of content, based on those conditions.** If the conditions are not valid, the system terminates the request (column 10, lines 65-67). Otherwise, the system validates and sends license authorization to consumer (column 10, lines 60-64).

Claim 16 - Downs teaches the aspects Offer templates containing predetermined usage and meta-rights (column 26, line 62). The system also keeps the identity of the purchaser (column 20, lines 36-37)

Argument # 5: Claim 19, not unpatentable over Downs et al., in view of Hitson et al.

Response #5: To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations

With regard to motivation, Downs et al. did not explicitly describe a system in which conditions are filtered and applied, **based on user preferences**. However, Hitson et al. discloses a system and method for content distribution in which content is selected based on

user preferences (page 1, paragraph 11). Therefore, it would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to design a system in which contents are filtered and applied, based on user preferences. It would have been obvious for one of ordinary skill in the art at the time of the applicant's invention to design such system because **(Motivation)** it would provide personalized content delivery, and would provide economic incentives to content providers by supplying a means of target marketing to users based upon user content preferences page 1, paragraph 11).

There is reasonable expectation of success. The prior art by Downs et al. deals with method and apparatus of securely providing data to a user's system, the Hitson et al. deals with system and method for composing and delivering music and other audio content via the Internet. Since both references deal with content distribution, there is reasonable expectation of success.

The references by Downs et al. and Hitson et al., combined implicitly or explicitly teach all of the limitations of claimed invention. Therefore, the three basic criteria to establish a prima facie case of obviousness are met.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Evens J. Augustin
October 28, 2006
Art Unit 3621

Conferees:

Vincent Millin – Appeal Conference Specialist

Andrew Fisher – Supervisor Art Unit 3621

Evens Augustin – Examiner Art Unit 3621

Appendix A

Limitation #	Claims 1 and 38	Prior Art (Downs, US 6226618)
1	Generating, by a supplier, at least one first offer, including **usage rights** and **meta-rights** for the items, ,	Content providers (entity that supplies the content), providing (equivalent to generating) **usage conditions** (equivalent to **usage rights**), The content providers also stipulate that the content stores or distributors can add or narrow the original usage rights (**meta-rights or rights derived from usage rights**) (column 21, lines 30-36).
2	said usage rights defining a manner of use for the items	**Usage rights being copy restrictions, which is manner in which the content can be used** (column 9, 32-34, col. 26, lines 10-12).
3	said meta-rights specifying rights to derive usage rights or other meta-rights for the items	**Content stores or distributors can add or narrow the original usage rights (sub-rights)** (column 21, lines 30-36)
4	Presenting, by the supplier, said offer to a first consumer in said system,	Content providers set and **transmit (equivalent to presenting) the usage conditions to the content stores** (column 21, 30-32), which are the **first customers** or distributors of the content providers
5	wherein the offer expresses what rights the consumer can acquire for the items;	Offer expresses the rights that consumer can acquire (column 59-60, lines 17-30)
6	Receiving, by the supplier, a selection from the first consumer indicating desired usage rights and meta-rights	Distributors **(first customer) making a request to digital content owners to sale digital content** (column 42, lines 65-67, column 43, lines 1-2). The two parties then come to an agreement (column 43, lines 4-5). Inherently, the content provider **receives the request**, and the distributors **agree to sell content usage rights and meta-rights.**
7	Generating, by the supplier, a first license granting to the first consumer the usage rights and meta rights for the items	After the agreement between the content provider and the distributor (first customer), **a digital certificate is created (same as generated)** and sent to the distributor (column 43, lines 14-18). This mechanism provides **licensing authorization** by enabling **intermediate** (in this case the distributors) or End-User(s) to unlock content after verification of a successful completion of a licensing transaction (column 7, lines 13-16)
8	Wherein the first license grants the usage rights and meta-rights that are selected by the first consumer during the receiving step	After the agreement between the content provider and the distributor (first customer), **a digital certificate is created and sent to the distributor** (column 43, lines 14-18). **Inherently the agreement and certificate is for the content/usage rights requested by the distributor**

Supplier = **Content provider**
First Consumer = **digital content store or distributor**
Usage Rights = **Usage conditions such as copy protection**
First license = **Digital certificate given to distributor**
Meta-rights = **Subrights, or addtionnal usage conditions derived from the usage rights**

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 1/00

A1

(11) International Publication Number: WO 98/09209

(43) International Publication Date: 5 March 1998 (05.03.98)

(21) International Application Number: PCT/US97/15243

(22) International Filing Date: 29 August 1997 (29.08.97)

(30) Priority Data:
08/706,206 30 August 1996 (30.08.96) US

(71) Applicant: INTERTRUST TECHNOLOGIES CORP. [US/US]; 460 Oakmead Parkway, Sunnyvale, CA 94086 (US).

(72) Inventors: GINTER, Karl, L.; 10404 43rd Avenue, Beltsville, MD 20705 (US). SHEAR, Victor, H.; 5203 Battery Lane, Bethesda, MD 20814 (US). SIBERT, W., Olin; 30 Ingleside Road, Lexington, MA 02173-2522 (US). SPAHN, Francis, J.; 2410 Edwards Avenue, El Cerrito, CA 94530 (US). VAN WIE, David, M.; 1250 Lakeside Drive, Sunnyvale, CA 94086 (US).

(74) Agent: FARIS, Robert, W.; Nixon & Vanderhye P.C., 8th floor, 1100 North Glebe Road, Arlington, VA 22201-4714 (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

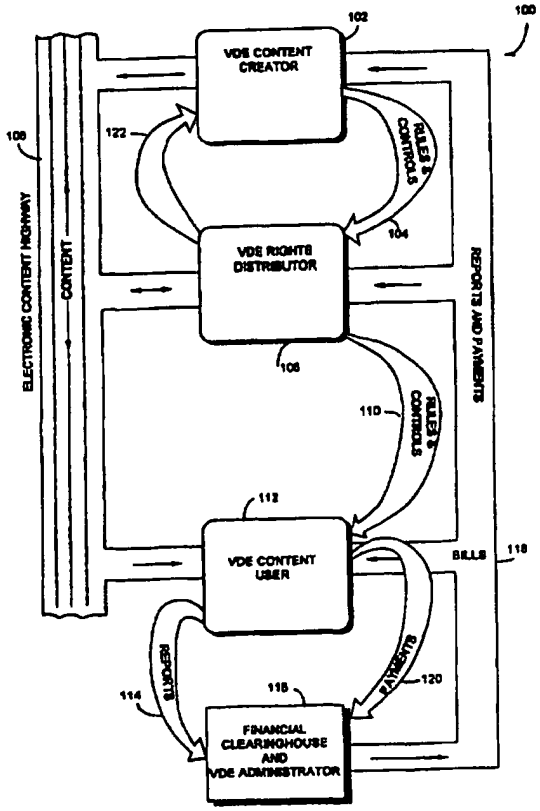
With international search report.

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: SYSTEMS AND METHODS FOR SECURE TRANSACTION MANAGEMENT AND ELECTRONIC RIGHTS PROTECTION

(57) Abstract

The present invention provides systems and methods for electronic commerce including secure transaction management and electronic rights protection. Electronic appliances such as computers employed in accordance with the present invention help to ensure that information is accessed and used only in authorized ways, and maintain the integrity, availability, and/or confidentiality of the information. Secure subsystems used with such electronic appliances provide a distributed virtual distribution environment (VDE) that may enforce a secure chain of handling and control, for example, to control and/or meter or otherwise monitor use of electronically stored or disseminated information. Such a virtual distribution environment may be used to protect rights of various participants in electronic commerce and other electronic or electronic-facilitated transactions. Secure distributed and other operating system environments and architectures, employing, for example, secure semiconductor processing arrangements that may establish secure, protected environments at each node. These techniques may be used to support an end-to-end electronic information distribution capability that may be used, for example, utilizing the "electronic highway".



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		

SYSTEMS AND METHODS FOR SECURE TRANSACTION MANAGEMENT AND ELECTRONIC RIGHTS PROTECTION

Field(s) of the Invention(s)

This invention generally relates to computer and/or electronic security.

More particularly, this invention relates to systems and techniques for secure transaction management. This invention also relates to computer-based and other electronic appliance-based technologies that help to ensure that information is accessed and/or otherwise used only in authorized ways, and maintains the integrity, availability, and/or confidentiality of such information and processes related to such use.

The invention also relates to systems and methods for protecting rights of various participants in electronic commerce and other electronic or electronically-facilitated transactions.

The invention also relates to secure chains of handling and control for both information content and information employed to regulate the use of such content and consequences of such use. It also relates to systems and techniques that manage, including meter and/or limit and/or otherwise monitor use of electronically stored and/or disseminated information. The invention

particularly relates to transactions, conduct and arrangements that make use of, including consequences of use of, such systems and/or techniques.

The invention also relates to distributed and other operating systems, environments and architectures. It also generally relates to secure architectures, including, for example, tamper-resistant hardware-based processors, that can be used to establish security at each node of a distributed system.

Background and Summary of the Invention(s)

Telecommunications, financial transactions, government processes, business operations, entertainment, and personal business productivity all now depend on electronic appliances. Millions of these electronic appliances have been electronically connected together. These interconnected electronic appliances comprise what is increasingly called the "information highway." Many businesses, academicians, and government leaders are concerned about how to protect the rights of citizens and organizations who use this information (also "electronic" or "digital") highway.

Electronic Content

Today, virtually anything that can be represented by words, numbers, graphics, or system of commands and instructions can be formatted into electronic digital information. Television, cable, satellite transmissions, and on-line services transmitted over telephone lines, compete to distribute digital information and entertainment to homes and businesses. The owners and marketers of this content include software developers, motion picture and recording companies, publishers of books, magazines, and newspapers, and information database providers. The popularization of on-line services has also enabled the individual personal computer user to participate as a content provider. It is estimated that the worldwide market for electronic information in 1992 was approximately $40 billion and is expected to grow to $200 billion by 1997, according to Microsoft Corporation. The present invention can materially enhance the revenue of content providers, lower the distribution costs and the costs for content, better support advertising and usage information gathering, and better satisfy the needs of electronic information users. These improvements can lead to a significant increase in the amount and variety of electronic information and the methods by which such information is distributed.

The inability of conventional products to be shaped to the needs of electronic information providers and users is sharply in contrast to the present invention. Despite the attention devoted by a cross-section of America's largest telecommunications, computer, entertainment and information provider companies to some of the problems addressed by the present invention, only the present invention provides commercially secure, effective solutions for configurable, general purpose electronic commerce transaction/distribution control systems.

Controlling Electronic Content

The present invention provides a new kind of "virtual distribution environment" (called "VDE" in this document) that secures, administers, and audits electronic information use. VDE also features fundamentally important capabilities for managing content that travels "across" the "information highway." These capabilities comprise a rights protection solution that serves all electronic community members. These members include content creators and distributors, financial service providers, end-users, and others. VDE is the first general purpose, configurable, transaction control/rights protection solution for users of computers, other electronic appliances, networks, and the information highway.

A fundamental problem for electronic content providers is extending their ability to control the use of proprietary information. Content providers often need to limit use to authorized activities and amounts. Participants in a business model involving, for example, provision of movies and advertising on optical discs may include actors, directors, script and other writers, musicians, studios, publishers, distributors, retailers, advertisers, credit card services, and content end-users. These participants need the ability to embody their range of agreements and requirements, including use limitations, into an "extended" agreement comprising an overall electronic business model. This extended agreement is represented by electronic content control information that can automatically enforce agreed upon rights and obligations. Under VDE, such an extended agreement may comprise an electronic contract involving all business model participants. Such an agreement may alternatively, or in addition, be made up of electronic agreements between subsets of the business model participants. Through the use of VDE, electronic commerce can function in the same way as traditional commerce—that is commercial relationships regarding products and services can be shaped through the negotiation of one or more agreements between a variety of parties.

Commercial content providers are concerned with ensuring proper compensation for the use of their electronic information. Electronic digital information, for example a CD recording, can today be copied relatively easily and inexpensively. Similarly, unauthorized copying and use of software programs deprives rightful owners of billions of dollars in annual revenue according to the International Intellectual Property Alliance. Content providers and distributors have devised a number of limited function rights protection mechanisms to protect their rights. Authorization passwords and protocols, license servers, "lock/unlock" distribution methods, and non-electronic contractual limitations imposed on users of shrink-wrapped software are a few of the more prevalent content protection schemes. In a commercial context, these efforts are inefficient and limited solutions.

Providers of "electronic currency" have also created protections for their type of content. These systems are not sufficiently adaptable, efficient, nor flexible enough to support the generalized use of electronic currency. Furthermore, they do not provide sophisticated auditing and control configuration capabilities. This means that current electronic currency tools lack the sophistication needed for many real-world financial business models. VDE provides means for anonymous currency

and for "conditionally" anonymous currency, wherein currency related activities remain anonymous except under special circumstances.

VDE Control Capabilities

VDE allows the owners and distributors of electronic digital information to reliably bill for, and securely control, audit, and budget the use of, electronic information. It can reliably detect and monitor the use of commercial information products. VDE uses a wide variety of different electronic information delivery means: including, for example, digital networks, digital broadcast, and physical storage media such as optical and magnetic disks. VDE can be used by major network providers, hardware manufacturers, owners of electronic information, providers of such information, and clearinghouses that gather usage information regarding, and bill for the use of, electronic information.

VDE provides comprehensive and configurable transaction management, metering and monitoring technology. It can change how electronic information products are protected, marketed, packaged, and distributed. When used, VDE should result in higher revenues for information providers and greater

user satisfaction and value. Use of VDE will normally result in lower usage costs, decreased transaction costs, more efficient access to electronic information, re-usability of rights protection and other transaction management implementations, greatly improved flexibility in the use of secured information, and greater standardization of tools and processes for electronic transaction management. VDE can be used to create an adaptable environment that fulfills the needs of electronic information owners, distributors, and users; financial clearinghouses; and usage information analyzers and resellers.

Rights and Control Information

In general, the present invention can be used to protect the rights of parties who have:

(a) proprietary or confidentiality interests in electronic information. It can, for example, help ensure that information is used only in authorized ways;

(b) financial interests resulting from the use of electronically distributed information. It can help ensure that content providers will be paid for use of distributed information; and

(c) interests in electronic credit and electronic currency storage, communication, and/or use including electronic cash, banking, and purchasing.

Protecting the rights of electronic community members involves a broad range of technologies. VDE combines these technologies in a way that creates a "distributed" electronic rights protection "environment." This environment secures and protects transactions and other processes important for rights protection. VDE, for example, provides the ability to prevent, or impede, interference with and/or observation of, important rights related transactions and processes. VDE, in its preferred embodiment, uses special purpose tamper resistant Secure Processing Units (SPUs) to help provide a high level of security for VDE processes and information storage and communication.

The rights protection problems solved by the present invention are electronic versions of basic societal issues. These issues include protecting property rights, protecting privacy rights, properly compensating people and organizations for their work and risk, protecting money and credit, and generally protecting the security of information. VDE employs a system that uses a common set of processes to manage rights issues in an efficient, trusted, and cost-effective way.

VDE can be used to protect the rights of parties who create electronic content such as, for example: records, games, movies, newspapers, electronic books and reference materials, personal electronic mail, and confidential records and communications. The invention can also be used to protect the rights of parties who provide electronic products, such as publishers and distributors; the rights of parties who provide electronic credit and currency to pay for use of products, for example, credit clearinghouses and banks; the rights to privacy of parties who use electronic content (such as consumers, business people, governments); and the privacy rights of parties *described* by electronic information, such as privacy rights related to information contained in a medical record, tax record, or personnel record.

In general, the present invention can protect the rights of parties who have:

(a) commercial interests in electronically distributed information -- the present invention can help ensure, for example, that parties, will be paid for use of distributed information in a manner consistent with their agreement;

(b) proprietary and/or confidentiality interests in electronic information -- the present invention can, for example, help ensure that data is used only in authorized ways;

(c) interests in electronic credit and electronic currency storage, communication, and/or use -- this can include electronic cash, banking, and purchasing; and

(d) interests in electronic information derived, at least in part, from use of other electronic information.

VDE Functional Properties

VDE is a cost-effective and efficient rights protection solution that provides a unified, consistent system for securing and managing transaction processing. VDE can:

(a) audit and analyze the use of content,

(b) ensure that content is used only in authorized ways, and

(c) allow information regarding content usage to be used only in ways approved by content users.

In addition, VDE:

(a) is very configurable, modifiable, and re-usable;

(b) supports a wide range of useful capabilities that may be combined in different ways to accommodate most potential applications;

(c) operates on a wide variety of electronic appliances ranging from hand-held inexpensive devices to large mainframe computers;

(d) is able to ensure the various rights of a number of different parties, and a number of different rights protection schemes, simultaneously;

(e) is able to preserve the rights of parties through a series of transactions that may occur at different times and different locations;

(f) is able to flexibly accommodate different ways of securely delivering information and reporting usage; and

(g) provides for electronic analogues to "real" money and credit, including anonymous electronic cash, to pay for products and services and to support personal (including home) banking and other financial activities.

VDE economically and efficiently fulfills the rights protection needs of electronic community members. Users of VDE will not require additional rights protection systems for different information highway products and rights problems—nor will they be required to install and learn a new system for each new information highway application.

VDE provides a unified solution that allows all content creators, providers, and users to employ the same electronic rights protection solution. Under authorized circumstances, the participants can freely exchange content and associated content control sets. This means that a user of VDE may, if allowed, use the same electronic system to work with different kinds of content having different sets of content control information. The

content and control information supplied by one group can be used by people who normally use content and control information supplied by a different group. VDE can allow content to be exchanged "universally" and users of an implementation of the present invention can interact electronically without fear of incompatibilities in content control, violation of rights, or the need to get, install, or learn a new content control system.

The VDE securely administers transactions that specify protection of rights. It can protect electronic rights including, for example:

(a) the property rights of authors of electronic content,

(b) the commercial rights of distributors of content,

(c) the rights of any parties who facilitated the distribution of content,

(d) the privacy rights of users of content,

(e) the privacy rights of parties portrayed by stored and/or distributed content, and

(f) any other rights regarding enforcement of electronic agreements.

VDE can enable a very broad variety of electronically enforced commercial and societal agreements. These agreements can include electronically implemented contracts, licenses, laws, regulations, and tax collection.

Contrast With Traditional Solutions

Traditional content control mechanisms often require users to purchase more electronic information than the user needs or desires. For example, infrequent users of shrink-wrapped software are required to purchase a program at the same price as frequent users, even though they may receive much less value from their less frequent use. Traditional systems do not scale cost according to the extent or character of usage and traditional systems can not attract potential customers who find that a fixed price is too high. Systems using traditional mechanisms are also not normally particularly secure. For example, shrink-wrapping does not prevent the constant illegal pirating of software once removed from either its physical or electronic package.

Traditional electronic information rights protection systems are often inflexible and inefficient and may cause a content provider to choose costly distribution channels that increase a product's price. In general these mechanisms restrict product pricing, configuration, and marketing flexibility. These compromises are the result of techniques for controlling information which cannot accommodate both different content models and content models which reflect the many, varied requirements, such as content delivery strategies, of the model participants. This can limit a provider's ability to deliver sufficient overall value to justify a given product's cost in the eyes of many potential users. VDE allows content providers and distributors to create applications and distribution networks that reflect content providers' and users' preferred business models. It offers users a uniquely cost effective and feature rich system that supports the ways providers *want* to distribute information and the ways users *want* to use such information. VDE supports content control models that ensure rights and allow content delivery strategies to be shaped for maximum commercial results.

Chain of Handling and Control

VDE can protect a collection of rights belonging to various parties having in rights in, or to, electronic information. This

information may be at one location or dispersed across (and/or moving between) multiple locations. The information may pass through a "chain" of distributors and a "chain" of users. Usage information may also be reported through one or more "chains" of parties. In general, VDE enables parties that (a) have rights in electronic information, and/or (b) act as direct or indirect agents for parties who have rights in electronic information, to ensure that the moving, accessing, modifying, or otherwise using of information can be securely controlled by rules regarding how, when, where, and by whom such activities can be performed.

VDE Applications and Software

VDE is a secure system for regulating electronic conduct and commerce. Regulation is ensured by control information put in place by one or more parties. These parties may include content providers, electronic hardware manufacturers, financial service providers, or electronic "infrastructure" companies such as cable or telecommunications companies. The control information implements "Rights Applications." Rights applications "run on" the "base software" of the preferred embodiment. This base software serves as a secure, flexible, general purpose foundation that can accommodate many different rights applications, that is, many different business models and their respective participant requirements.

A rights application under VDE is made up of special purpose pieces, each of which can correspond to one or more basic electronic processes needed for a rights protection environment. These processes can be combined together like building blocks to create electronic agreements that can protect the rights, and may enforce fulfillment of the obligations, of electronic information users and providers. One or more providers of electronic information can easily combine selected building blocks to create a rights application that is unique to a specific content distribution model. A group of these pieces can represent the capabilities needed to fulfill the agreement(s) between users and providers. These pieces accommodate many requirements of electronic commerce including:

- the distribution of permissions to use electronic information;

- the persistence of the control information and sets of control information managing these permissions;

- configurable control set information that can be selected by users for use with such information;

- data security and usage auditing of electronic information; and

- a secure system for currency, compensation and debit management.

For electronic commerce, a rights application, under the preferred embodiment of the present invention, can provide electronic enforcement of the business agreements between all participants. Since different groups of components can be put together for different applications, the present invention can provide electronic control information for a wide variety of different products and markets. This means the present invention can provide a "unified," efficient, secure, and cost-effective system for electronic commerce and data security. This allows VDE to serve as a single standard for electronic rights protection, data security, and electronic currency and banking.

In a VDE, the separation between a rights application and its foundation permits the efficient selection of sets of control information that are appropriate for each of many different types of applications and uses. These control sets can reflect both rights of electronic community members, as well as obligations

(such as providing a history of one's use of a product or paying taxes on one's electronic purchases). VDE flexibility allows its users to electronically implement and enforce common social and commercial ethics and practices. By providing a unified control system, the present invention supports a vast range of possible transaction related interests and concerns of individuals, communities, businesses, and governments. Due to its open design, VDE allows (normally under securely controlled circumstances) applications using technology independently created by users to be "added" to the system and used in conjunction with the foundation of the invention. In sum, VDE provides a system that can fairly reflect and enforce agreements among parties. It is a broad ranging and systematic solution that answers the pressing need for a secure, cost-effective, and fair electronic environment.

VDE Implementation

The preferred embodiment of the present invention includes various tools that enable system designers to directly insert VDE capabilities into their products. These tools include an Application Programmer's Interface ("API") and a Rights Permissioning and Management Language ("RPML"). The RPML provides comprehensive and detailed control over the use of the invention's features. VDE also includes certain user

interface subsystems for satisfying the needs of content providers, distributors, and users.

Information distributed using VDE may take many forms. It may, for example, be "distributed" for use on an individual's own computer, that is the present invention can be used to provide security for locally stored data. Alternatively, VDE may be used with information that is dispersed by authors and/or publishers to one or more recipients. This information may take many forms including: movies, audio recordings, games, electronic catalog shopping, multimedia, training materials, E-mail and personal documents, object oriented libraries, software programming resources, and reference/record keeping information resources (such as business, medical, legal, scientific, governmental, and consumer databases).

Electronic rights protection provided by the present invention will also provide an important foundation for trusted and efficient home and commercial banking, electronic credit processes, electronic purchasing, true or conditionally anonymous electronic cash, and EDI (Electronic Data Interchange). VDE provides important enhancements for improving data security in organizations by providing "smart"

transaction management features that can be far more effective than key and password based "go/no go" technology.

VDE normally employs an integration of cryptographic and other security technologies (e.g. encryption, digital signatures, etc.), with other technologies including: component, distributed, and event driven operating system technology, and related communications, object container, database, smart agent, smart card, and semiconductor design technologies.

I. Overview

A. VDE Solves Important Problems and Fills Critical Needs

The world is moving towards an integration of electronic information appliances. This interconnection of appliances provides a foundation for much greater electronic interaction and the evolution of electronic commerce. A variety of capabilities are required to implement an electronic commerce environment. VDE is the first system that provides many of these capabilities and therefore solves fundamental problems related to electronic dissemination of information.

Electronic Content

VDE allows electronic arrangements to be created involving two or more parties. These agreements can themselves comprise a collection of agreements between participants in a commercial value chain and/or a data security chain model for handling, auditing, reporting, and payment. It can provide efficient, reusable, modifiable, and consistent means for secure electronic content: distribution, usage control, usage payment, usage auditing, and usage reporting. Content may, for example, include:

- financial information such as electronic currency and credit;
- commercially distributed electronic information such as reference databases, movies, games, and advertising; and
- electronic properties produced by persons and organizations, such as documents, e-mail, and proprietary database information.

VDE enables an electronic commerce marketplace that supports differing, competitive business partnerships, agreements, and evolving overall business models.

The features of VDE allow it to function as the first trusted electronic information control environment that can conform to, and support, the bulk of conventional electronic commerce and data security requirements. In particular, VDE enables the participants in a business value chain model to create an electronic version of traditional business agreement terms and conditions and further enables these participants to shape and evolve their electronic commerce models as they believe appropriate to their business requirements.

VDE offers an architecture that avoids reflecting specific distribution biases, administrative and control perspectives, and content types. Instead, VDE provides a broad-spectrum, fundamentally configurable and portable, electronic transaction control, distributing, usage, auditing, reporting, and payment operating environment. VDE is not limited to being an application or application specific toolset that covers only a limited subset of electronic interaction activities and participants. Rather, VDE supports systems by which such applications can be created, modified, and/or reused. As a result,

the present invention answers pressing, unsolved needs by offering a system that supports a standardized control environment which facilitates interoperability of electronic appliances, interoperability of content containers, and efficient creation of electronic commerce applications and models through the use of a programmable, secure electronic transactions management foundation and reusable and extensible executable components. VDE can support a single electronic "world" within which most forms of electronic transaction activities can be managed.

To answer the developing needs of rights owners and content providers and to provide a system that can accommodate the requirements and agreements of all parties that may be involved in electronic business models (creators, distributors, administrators, users, credit providers, etc.), VDE supplies an efficient, largely transparent, low cost and sufficiently secure system (supporting both hardware/ software and software only models). VDE provides the widely varying secure control and administration capabilities required for:

1. Different types of electronic content,

2. Differing electronic content delivery schemes,

3. Differing electronic content usage schemes,

4. Different content usage platforms, and

5. Differing content marketing and model strategies.

VDE may be combined with, or integrated into, many separate computers and/or other electronic appliances. These appliances typically include a secure subsystem that can enable control of content use such as displaying, encrypting, decrypting, printing, copying, saving, extracting, embedding, distributing, auditing usage, etc. The secure subsystem in the preferred embodiment comprises one or more "protected processing environments", one or more secure databases, and secure "component assemblies" and other items and processes that need to be kept secured. VDE can, for example, securely control electronic currency, payments, and/or credit management (including electronic credit and/or currency receipt, disbursement, encumbering, and/or allocation) using such a "secure subsystem."

VDE provides a secure, distributed electronic transaction management system for controlling the distribution and/or other usage of electronically provided and/or stored information. VDE

controls auditing and reporting of electronic content and/or appliance usage. Users of VDE may include content creators who apply content usage, usage reporting, and/or usage payment related control information to electronic content and/or appliances for users such as end-user organizations, individuals, and content and/or appliance distributors. VDE also securely supports the payment of money owed (including money owed for content and/or appliance usage) by one or more parties to one or more other parties, in the form of electronic credit and/or currency.

Electronic appliances under control of VDE represent VDE 'nodes' that securely process and control; distributed electronic information and/or appliance usage, control information formulation, and related transactions. VDE can securely manage the integration of control information provided by two or more parties. As a result, VDE can construct an electronic agreement between VDE participants that represent a "negotiation" between, the control requirements of, two or more parties and enacts terms and conditions of a resulting agreement. VDE ensures the rights of each party to an electronic agreement regarding a wide range of electronic activities related to electronic information and/or appliance usage.

Through use of VDE's control system, traditional content providers and users can create electronic relationships that reflect traditional, non-electronic relationships. They can shape and modify commercial relationships to accommodate the evolving needs of, and agreements among, themselves. VDE does not require electronic content providers and users to modify their business practices and personal preferences to conform to a metering and control application program that supports limited, largely fixed functionality. Furthermore, VDE permits participants to develop business models not feasible with non-electronic commerce, for example, involving detailed reporting of content usage information, large numbers of distinct transactions at hitherto infeasibly low price points, "pass-along" control information that is enforced without involvement or advance knowledge of the participants, etc.

The present invention allows content providers and users to formulate their transaction environment to accommodate:

(1) desired content models, content control models, and content usage information pathways,

(2) a complete range of electronic media and distribution means,

(3) a broad range of pricing, payment, and auditing strategies,

(4) very flexible privacy and/or reporting models,

(5) practical and effective security architectures, and

(6) other administrative procedures that together with steps (1) through (5) can enable most "real world" electronic commerce and data security models, including models unique to the electronic world.

VDE's transaction management capabilities can enforce:

(1) privacy rights of users related to information regarding their usage of electronic information and/or appliances,

(2) societal policy such as laws that protect rights of content users or require the collection of taxes derived from electronic transaction revenue, and

(3) the proprietary and/or other rights of parties related to ownership of, distribution of, and/or other commercial rights related to, electronic information.

VDE can support "real" commerce in an electronic form, that is the progressive creation of commercial relationships that form, over time, a network of interrelated agreements representing a value chain business model. This is achieved in part by enabling content control information to develop through the interaction of (negotiation between) securely created and independently submitted sets of content and/or appliance control information. Different sets of content and/or appliance control information can be submitted by different parties in an electronic business value chain enabled by the present invention. These parties create control information sets through the use of their respective VDE installations. Independently, securely deliverable, component based control information allows efficient interaction among control information sets supplied by different parties.

VDE permits multiple, separate electronic arrangements to be formed between subsets of parties in a VDE supported electronic value chain model. These multiple agreements together comprise a VDE value chain "extended" agreement.

VDE allows such constituent electronic agreements, and therefore overall VDE extended agreements, to evolve and reshape over time as additional VDE participants become involved in VDE content and/or appliance control information handling. VDE electronic agreements may also be extended as new control information is submitted by existing participants. With VDE, electronic commerce participants are free to structure and restructure their electronic commerce business activities and relationships. As a result, the present invention allows a competitive electronic commerce marketplace to develop since the use of VDE enables different, widely varying business models using the same or shared content.

A significant facet of the present invention's ability to broadly support electronic commerce is its ability to securely manage independently delivered VDE component objects containing control information (normally in the form of VDE objects containing one or more methods, data, or load module VDE components). This independently delivered control information can be integrated with senior and other pre-existing content control information to securely form derived control information using the negotiation mechanisms of the present invention. All requirements specified by this derived control information must be satisfied before VDE controlled content can

be accessed or otherwise used. This means that, for example, all load modules and any mediating data which are listed by the derived control information as required must be available and securely perform their required function. In combination with other aspects of the present invention, securely, independently delivered control components allow electronic commerce participants to freely stipulate their business requirements and trade offs. As a result, much as with traditional, non-electronic commerce, the present invention allows electronic commerce (through a progressive stipulation of various control requirements by VDE participants) to evolve into forms of business that are the most efficient, competitive and useful.

VDE provides capabilities that rationalize the support of electronic commerce and electronic transaction management. This rationalization stems from the reusability of control structures and user interfaces for a wide variety of transaction management related activities. As a result, content usage control, data security, information auditing, and electronic financial activities, can be supported with tools that are reusable, convenient, consistent, and familiar. In addition, a rational approach—a transaction/distribution control standard—allows all participants in VDE the same foundation set of hardware control and security, authoring, administration,

and management tools to support widely varying types of information, business market model, and/or personal objectives.

Employing VDE as a general purpose electronic transaction/distribution control system allows users to maintain a single transaction management control arrangement on each of their computers, networks, communication nodes, and/or other electronic appliances. Such a general purpose system can serve the needs of many electronic transaction management applications without requiring distinct, different installations for different purposes. As a result, users of VDE can avoid the confusion and expense and other inefficiencies of different, limited purpose transaction control applications for each different content and/or business model. For example, VDE allows content creators to use the same VDE foundation control arrangement for both content authoring and for licensing content from other content creators for inclusion into their products or for other use. Clearinghouses, distributors, content creators, and other VDE users can all interact, both with the applications running on their VDE installations, and with each other, in an entirely consistent manner, using and reusing (largely transparently) the same distributed tools, mechanisms, and consistent user interfaces, regardless of the type of VDE activity.

VDE prevents many forms of unauthorized use of electronic information, by controlling and auditing (and other administration of use) electronically stored and/or disseminated information. This includes, for example, commercially distributed content, electronic currency, electronic credit, business transactions (such as EDI), confidential communications, and the like. VDE can further be used to enable commercially provided electronic content to be made available to users in user defined portions, rather than constraining the user to use portions of content that were "predetermined" by a content creator and/or other provider for billing purposes.

VDE, for example, can employ:

(1) Secure metering means for budgeting and/or auditing electronic content and/or appliance usage;

(2) Secure flexible means for enabling compensation and/or billing rates for content and/or appliance usage, including electronic credit and/or currency mechanisms for payment means;

(3) Secure distributed database means for storing control and usage related information (and employing validated compartmentalization and tagging schemes);

(4) Secure electronic appliance control means;

(5) A distributed, secure, "virtual black box" comprised of nodes located at every user (including VDE content container creators, other content providers, client users, and recipients of secure VDE content usage information) site. The nodes of said virtual black box normally include a secure subsystem having at least one secure hardware element (a semiconductor element or other hardware module for securely executing VDE control processes), said secure subsystems being distributed at nodes along a pathway of information storage, distribution, payment, usage, and/or auditing. In some embodiments, the functions of said hardware element, for certain or all nodes, may be performed by software, for example, in host processing environments of electronic appliances;

(6) Encryption and decryption means;

(7) Secure communications means employing authentication, digital signaturing, and encrypted transmissions. The secure subsystems at said user nodes utilize a protocol that establishes and authenticates each node's and/or participant's identity, and establishes one or more secure host-to-host encryption keys for communications between the secure subsystems; and

(8) Secure control means that can allow each VDE installation to perform VDE content authoring (placing content into VDE containers with associated control information), content distribution, and content usage; as well as clearinghouse and other administrative and analysis activities employing content usage information.

VDE may be used to migrate most non-electronic, traditional information delivery models (including entertainment, reference materials, catalog shopping, etc.) into an adequately secure digital distribution and usage management

and payment context. The distribution and financial pathways managed by a VDE arrangement may include:

- content creator(s),
- distributor(s),
- redistributor(s),
- client administrator(s),
- client user(s),
- financial and/or other clearinghouse(s),
- and/or government agencies.

These distribution and financial pathways may also include:

- advertisers,
- market survey organizations, and/or
- other parties interested in the user usage of information securely delivered and/or stored using VDE.

Normally, participants in a VDE arrangement will employ the same secure VDE foundation. Alternate embodiments support VDE arrangements employing differing VDE foundations. Such alternate embodiments may employ procedures to ensure certain interoperability requirements are met.

Secure VDE hardware (also known as SPUs for Secure Processing Units), or VDE installations that use software to substitute for, or complement, said hardware (provided by Host Processing Environments (HPEs)), operate in conjunction with secure communications, systems integration software, and distributed software control information and support structures, to achieve the electronic contract/rights protection environment of the present invention. Together, these VDE components comprise a secure, virtual, distributed content and/or appliance control, auditing (and other administration), reporting, and payment environment. In some embodiments and where commercially acceptable, certain VDE participants, such as clearinghouses that normally maintain sufficiently physically secure non-VDE processing environments, may be allowed to employ HPEs rather VDE hardware elements and interoperate, for example, with VDE end-users and content providers. VDE components together comprise a configurable, consistent, secure and "trusted" architecture for distributed, asynchronous control of electronic content and/or appliance usage. VDE supports a "universe wide" environment for electronic content delivery, broad dissemination, usage reporting, and usage related payment activities.

VDE provides generalized configurability. This results, in part, from decomposition of generalized requirements for supporting electronic commerce and data security into a broad range of constituent "atomic" and higher level components (such as load modules, data elements, and methods) that may be variously aggregated together to form control methods for electronic commerce applications, commercial electronic agreements, and data security arrangements. VDE provides a secure operating environment employing VDE foundation elements along with secure independently deliverable VDE components that enable electronic commerce models and relationships to develop. VDE specifically supports the unfolding of distribution models in which content providers, over time, can expressly agree to, or allow, subsequent content providers and/or users to participate in shaping the control information for, and consequences of, use of electronic content and/or appliances. A very broad range of the functional attributes important for supporting simple to very complex electronic commerce and data security activities are supported by capabilities of the present invention. As a result, VDE supports most types of electronic information and/or appliance: usage control (including distribution), security, usage auditing, reporting, other administration, and payment arrangements.

VDE, in its preferred embodiment, employs object software technology and uses object technology to form "containers" for delivery of information that is (at least in part) encrypted or otherwise secured. These containers may contain electronic content products or other electronic information and some or all of their associated permissions (control) information. These container objects may be distributed along pathways involving content providers and/or content users. They may be securely moved among nodes of a Virtual Distribution Environment (VDE) arrangement, which nodes operate VDE foundation software and execute control methods to enact electronic information usage control and/or administration models. The containers delivered through use of the preferred embodiment of the present invention may be employed both for distributing VDE control instructions (information) and/or to encapsulate and electronically distribute content that has been at least partially secured.

Content providers who employ the present invention may include, for example, software application and game publishers, database publishers, cable, television, and radio broadcasters, electronic shopping vendors, and distributors of information in electronic document, book, periodical, e-mail and/or other forms. Corporations, government agencies, and/or individual

"end-users" who act as storers of, and/or distributors of, electronic information, may also be VDE content providers (in a restricted model, a user provides content only to himself and employs VDE to secure his own confidential information against unauthorized use by other parties). Electronic information may include proprietary and/or confidential information for personal or internal organization use, as well as information, such as software applications, documents, entertainment materials, and/or reference information, which may be provided to other parties. Distribution may be by, for example, physical media delivery, broadcast and/or telecommunication means, and in the form of "static" files and/or streams of data. VDE may also be used, for example, for multi-site "real-time" interaction such as teleconferencing, interactive games, or on-line bulletin boards, where restrictions on, and/or auditing of, the use of all or portions of communicated information is enforced.

VDE provides important mechanisms for both enforcing commercial agreements and enabling the protection of privacy rights. VDE can securely deliver information from one party to another concerning the use of commercially distributed electronic content. Even if parties are separated by several "steps" in a chain (pathway) of handling for such content usage information, such information is protected by VDE through encryption and/or

other secure processing. Because of that protection, the accuracy of such information is guaranteed by VDE, and the information can be trusted by all parties to whom it is delivered. Furthermore, VDE guarantees that all parties can trust that such information cannot be received by anyone other than the intended, authorized, party(ies) because it is encrypted such that only an authorized party, or her agents, can decrypt it. Such information may also be derived through a secure VDE process at a previous pathway-of-handling location to produce secure VDE reporting information that is then communicated securely to its intended recipient's VDE secure subsystem. Because VDE can deliver such information securely, parties to an electronic agreement need not trust the accuracy of commercial usage and/or other information delivered through means other than those under control of VDE.

VDE participants in a commercial value chain can be "commercially" confident (that is, sufficiently confident for commercial purposes) that the direct (constituent) and/or "extended" electronic agreements they entered into through the use of VDE can be enforced reliably. These agreements may have both "dynamic" transaction management related aspects, such as content usage control information enforced through budgeting, metering, and/or reporting of electronic information

and/or appliance use, and/or they may include "static" electronic assertions, such as an end-user using the system to assert his or her agreement to pay for services, not to pass to unauthorized parties electronic information derived from usage of content or systems, and/or agreeing to observe copyright laws. Not only can electronically reported transaction related information be trusted under the present invention, but payment may be automated by the passing of payment tokens through a pathway of payment (which may or may not be the same as a pathway for reporting). Such payment can be contained within a VDE container created automatically by a VDE installation in response to control information (located, in the preferred embodiment, in one or more permissions records) stipulating the "withdrawal" of credit or electronic currency (such as tokens) from an electronic account (for example, an account securely maintained by a user's VDE installation secure subsystem) based upon usage of VDE controlled electronic content and/or appliances (such as governments, financial credit providers, and users).

VDE allows the needs of electronic commerce participants to be served and it can bind such participants together in a universe wide, trusted commercial network that can be secure enough to support very large amounts of commerce. VDE's security and metering secure subsystem core will be present at

all physical locations where VDE related content is (a) assigned usage related control information (rules and mediating data), and/or (b) used. This core can perform security and auditing functions (including metering) that operate within a "virtual black box," a collection of distributed, very secure VDE related hardware instances that are interconnected by secured information exchange (for example, telecommunication) processes and distributed database means. VDE further includes highly configurable transaction operating system technology, one or more associated libraries of load modules along with affiliated data, VDE related administration, data preparation, and analysis applications, as well as system software designed to enable VDE integration into host environments and applications. VDE's usage control information, for example, provide for property content and/or appliance related: usage authorization, usage auditing (which may include audit reduction), usage billing, usage payment, privacy filtering, reporting, and security related communication and encryption techniques.

VDE extensively employs methods in the form of software objects to augment configurability, portability, and security of the VDE environment. It also employs a software object architecture for VDE content containers that carries protected

content and may also carry both freely available information (e.g, summary, table of contents) and secured content control information which ensures the performance of control information. Content control information governs content usage according to criteria set by holders of rights to an object's contents and/or according to parties who otherwise have rights associated with distributing such content (such as governments, financial credit providers, and users).

In part, security is enhanced by object methods employed by the present invention because the encryption schemes used to protect an object can efficiently be further used to protect the associated content control information (software control information and relevant data) from modification. Said object techniques also enhance portability between various computer and/or other appliance environments because electronic information in the form of content can be inserted along with (for example, in the same object container as) content control information (for said content) to produce a "published" object. As a result, various portions of said control information may be specifically adapted for different environments, such as for diverse computer platforms and operating systems, and said various porticns may all be carried by a VDE container.

An objective of VDE is supporting a transaction/distribution control standard. Development of such a standard has many obstacles, given the security requirements and related hardware and communications issues, widely differing environments, information types, types of information usage, business and/or data security goals, varieties of participants, and properties of delivered information. A significant feature of VDE accommodates the many, varying distribution and other transaction variables by, in part, decomposing electronic commerce and data security functions into generalized capability modules executable within a secure hardware SPU and/or corresponding software subsystem and further allowing extensive flexibility in assembling, modifying, and/or replacing, such modules (e.g. load modules and/or methods) in applications run on a VDE installation foundation. This configurability and reconfigurability allows electronic commerce and data security participants to reflect their priorities and requirements through a process of iteratively shaping an evolving extended electronic agreement (electronic control model). This shaping can occur as content control information passes from one VDE participant to another and to the extent allowed by "in place" content control information. This process allows users of VDE to recast existing control

information and/or add new control information as necessary (including the elimination of no longer required elements).

VDE supports trusted (sufficiently secure) electronic information distribution and usage control models for both commercial electronic content distribution and data security applications. It can be configured to meet the diverse requirements of a network of interrelated participants that may include content creators, content distributors, client administrators, end users, and/or clearinghouses and/or other content usage information users. These parties may constitute a network of participants involved in simple to complex electronic content dissemination, usage control, usage reporting, and/or usage payment. Disseminated content may include both originally provided and VDE generated information (such as content usage information) and content control information may persist through both chains (one or more pathways) of content and content control information handling, as well as the direct usage of content. The configurability provided by the present invention is particularly critical for supporting electronic commerce, that is enabling businesses to create relationships and evolve strategies that offer competitive value. Electronic commerce tools that are not inherently configurable and interoperable will ultimately fail to produce products (and

services) that meet both basic requirements and evolving needs of most commerce applications.

VDE's fundamental configurability will allow a broad range of competitive electronic commerce business models to flourish. It allows business models to be shaped to maximize revenues sources, end-user product value, and operating efficiencies. VDE can be employed to support multiple, differing models, take advantage of new revenue opportunities, and deliver product configurations most desired by users. Electronic commerce technologies that do not, as the present invention does:

- support a broad range of possible, complementary revenue activities,
- offer a flexible array of content usage features most desired by customers, and
- exploit opportunities for operating efficiencies,

will result in products that are often intrinsically more costly and less appealing and therefore less competitive in the marketplace.

Some of the key factors contributing to the configurability intrinsic to the present invention include:

(a) integration into the fundamental control environment of a broad range of electronic appliances through portable API and programming language tools that efficiently support merging of control and auditing capabilities in nearly any electronic appliance environment while maintaining overall system security;

(b) modular data structures;

(c) generic content model;

(d) general modularity and independence of foundation architectural components;

(e) modular security structures;

(f) variable length and multiple branching chains of control; and

(g) independent, modular control structures in the form of executable load modules that can be maintained in one or more libraries, and assembled into control methods and models, and where such model control

schemes can "evolve" as control information passes through the VDE installations of participants of a pathway of VDE content control information handling.

Because of the breadth of issues resolved by the present invention, it can provide the emerging "electronic highway" with a single transaction/distribution control system that can, for a very broad range of commercial and data security models, ensure against unauthorized use of confidential and/or proprietary information and commercial electronic transactions. VDE's electronic transaction management mechanisms can enforce the electronic rights and agreements of all parties participating in widely varying business and data security models, and this can be efficiently achieved through a single VDE implementation within each VDE participant's electronic appliance. VDE supports widely varying business and/or data security models that can involve a broad range of participants at various "levels" of VDE content and/or content control information pathways of handling. Different content control and/or auditing models and agreements may be available on the same VDE installation. These models and agreements may control content in relationship to, for example, VDE installations and/or users in general; certain specific users, installations, classes and/or other

groupings of installations and/or users; as well as to electronic content generally on a given installation, to specific properties, property portions, classes and/or other groupings of content.

Distribution using VDE may package both the electronic content and control information into the same VDE container, and/or may involve the delivery to an end-user site of different pieces of the same VDE managed property from plural separate remote locations and/or in plural separate VDE content containers and/or employing plural different delivery means. Content control information may be partially or fully delivered separately from its associated content to a user VDE installation in one or more VDE administrative objects. Portions of said control information may be delivered from one or more sources. Control information may also be available for use by access from a user's VDE installation secure sub-system to one or more remote VDE secure sub-systems and/or VDE compatible, certified secure remote locations. VDE control processes such as metering, budgeting, decrypting and/or fingerprinting, may as relates to a certain user content usage activity, be performed in a user's local VDE installation secure subsystem, or said processes may be divided amongst plural secure subsystems which may be located in the same user VDE installations and/or in a network server and in the user installation. For example, a local VDE

installation may perform decryption and save any, or all of, usage metering information related to content and/or electronic appliance usage at such user installation could be performed at the server employing secure (e.g., encrypted) communications between said secure subsystems. Said server location may also be used for near real time, frequent, or more periodic secure receipt of content usage information from said user installation, with, for example, metered information being maintained only temporarily at a local user installation.

Delivery means for VDE managed content may include electronic data storage means such as optical disks for delivering one portion of said information and broadcasting and/or telecommunicating means for other portions of said information. Electronic data storage means may include magnetic media, optical media, combined magneto-optical systems, flash RAM memory, bubble memory, and/or other memory storage means such as huge capacity optical storage systems employing holographic, frequency, and/or polarity data storage techniques. Data storage means may also employ layered disc techniques, such as the use of generally transparent and/or translucent materials that pass light through layers of data carrying discs which themselves are physically packaged together as one

thicker disc. Data carrying locations on such discs may be, at least in part, opaque.

VDE supports a general purpose foundation for secure transaction management, including usage control, auditing, reporting, and/or payment. This general purpose foundation is called "VDE Functions" ("VDEFs"). VDE also supports a collection of "atomic" application elements (e.g., load modules) that can be selectively aggregated together to form various VDEF capabilities called control methods and which serve as VDEF applications and operating system functions. When a host operating environment of an electronic appliance includes VDEF capabilities, it is called a "Rights Operating System" (ROS). VDEF load modules, associated data, and methods form a body of information that for the purposes of the present invention are called "control information." VDEF control information may be specifically associated with one or more pieces of electronic content and/or it may be employed as a general component of the operating system capabilities of a VDE installation.

VDEF transaction control elements reflect and enact content specific and/or more generalized administrative (for example, general operating system) control information. VDEF capabilities which can generally take the form of applications