(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 October 2001 (18.10.2001)

PCT

(10) International Publication Number
WO 01/77811 A1

(51) International Patent Classification[7]: G06F 7/00, 11/30, 12/14, 15/16, 15/173, 17/00, 17/30, H04L 9/00, 9/32

(21) International Application Number: PCT/US01/10441

(22) International Filing Date: 29 March 2001 (29.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/545,286 7 April 2000 (07.04.2000) US

(71) Applicant: **EVERDREAM, INC.** [US/US]; 6591 Dumbarton Circle, Fremont, CA 94555 (US).

(72) Inventor: **JOOSTE, Sarel, Kobus**; 1271 Lawson Avenue, Waverley, 0186 Pretoria (ZA).

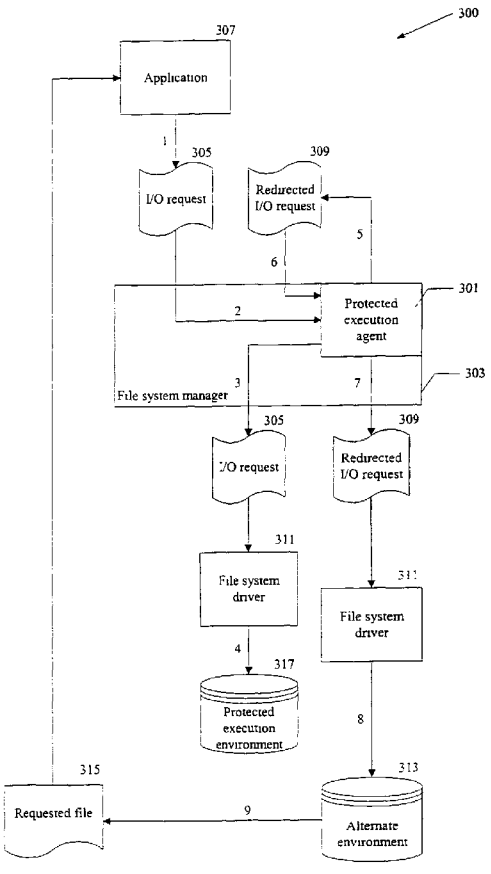
(74) Agents: **MALLIE, Michael, J.** et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 7th Floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, 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,

[Continued on next page]

(54) Title: PROTECTED EXECUTION ENVIRONMENTS WITHIN A COMPUTER SYSTEM



(57) Abstract: A protected execution agent (310) installs itself within a file system manager (303) on the computer (300) to control modifications to a protected execution environment (317) by intercepting (2) I/O requests (305) from applications (307). If an unauthorized application (307) attempts to modify the protected execution environment (317), the protected execution agent (301) terminates the original I/O request (305) and creates a redirected I/O request (309) that specifies a corresponding directory path within an alternate environment. A configuration utility is responsible for determining which installed applications (307) are authorized to change the protected execution environment (317). The configuration utility also establishes a parent-child relationship between an unauthorized application that invokes or "spawns" an authorized application, with the authorized child application being considered unauthorized when performing processes on behalf of the unauthorized parent application.

WO 01/77811 A1

IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

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.

PROTECTED EXECUTION ENVIRONMENTS WITHIN A COMPUTER SYSTEM

FIELD OF THE INVENTION

This invention relates generally to computer operating environments, and more particularly to protecting such an operating environment from modification.

COPYRIGHT NOTICE/PERMISSION

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever. The following notice applies to the software and data as described below and in the drawings hereto: Copyright © 1999, Everdream Corporation, All Rights Reserved.

BACKGROUND OF THE INVENTION

Technical support for personal computer systems, whether owned by a business or by an individual, is complicated by the installation of new application programs that frequently overwrite components or settings required by already-installed applications, causing the older applications to fail. Such failures can sometimes be remedied using diagnostic software that removes the newly installed application and the reinstates the original version of the components or settings. However, the user then is left without the features offered by the new application.

One solution to the problem of conflicting applications is to prohibit the user from installing new applications or new versions of installed applications into a stable operating environment, but that solution is unacceptable to most users. An alternate solution is disclosed in U.S. Patent Application Serial Number 09/418,697 titled METHOD AND APPARATUS FOR CONFIGURING A HARD DISK AND FOR PROVIDING SUPPORT FOR A COMPUTER SYSTEM filed on October 15, 1999 and assigned to the same assignee as the present application. The invention in the

Nintendo Exhibit 1003
Part 2 of 2
Page 00629

09/418,697 application partitions the hard disk on a computer with a stable operating environment into separate logical environments, one of which represents the stable operating environment and is protected from subsequent user modifications. Applications installed after the partitioning are written to one of the logical environments that permits user modifications. Only one of the environments is active on the computer at a time. Thus, when the user wants to execute an application that is resident in an inactive environment, the computer terminates the active environment and switches to the environment that contains the requested application.

Because some users may find the switch confusing, there exists a need to protect a stable operating environment from modifications while allowing the user to install and execute new applications without the necessity of switching active environments.

SUMMARY OF THE INVENTION

The above-mentioned shortcomings, disadvantages and problems are addressed by the present invention, which will be understood by reading and studying the following specification.

Each application installed on the computer is classified as authorized or unauthorized. The directory paths for authorized applications form a protected execution environment. Only input/output (I/O) requests from authorized applications are allowed to change the protected execution environment. A protected execution agent installs itself within a file system manager on the computer to control modifications to the protected execution environment by intercepting I/O requests from applications. If an unauthorized application attempts to modify the protected execution environment, the protected execution agent terminates the original I/O request and creates a redirected I/O request that specifies a corresponding directory path for the file within an alternate environment. The file path in the alternate environment corresponds to the file path in the original I/O request. The protected execution agent then submits the redirected I/O request to the file system manager. When the protected execution agent intercepts the redirected request, it recognizes the request as its own and allows the request to continue onto the appropriate file system driver for processing. The file system driver then performs the requested I/O operation against the corresponding directory. Consequently,

Nintendo Exhibit 1003
Part 2 of 2
Page 00630

when an unauthorized application attempts to store a file in a directory within the protected execution environment, the file is instead stored in the corresponding directory. Because the application is unaware of the change, subsequent I/O requests for that file continue to specify the path in the protected execution environment. The protected execution agent intercepts each such subsequent request and submits a redirected I/O request that specifies the corresponding directory. Thus, the present invention "virtualizes" I/O requests from unauthorized applications that are directed to the protected environment.

In one aspect, a configuration utility scans the file system and creates a list of all installed applications. The list is then evaluated to categorize the installed applications as authorized or unauthorized. A special "mirror" directory structure is created that mimics the directory paths of the authorized applications, with the executable file for each authorized application being represented as a null file entry within the mirror directory structure. The protected execution agent queries the file system manager for the corresponding executable in the mirror directory structure when it needs to determine if an application is authorized. If the file system manager indicates that the executable exists, the application is authorized. The configuration utility can be run when a new application is installed or at any point to change the category of an application.

In another aspect, the protected execution agent establishes a parent-child relationship between an unauthorized application that invokes or "spawns" an authorized application to perform some process on behalf of the unauthorized application. All subsequent I/O requests from the authorized child application are treated as if they were submitted by the unauthorized parent application until the parent-child relationship is terminated.

The present invention describes systems, clients, servers, methods, and computer-readable media of varying scope. In addition to the aspects and advantages of the present invention described in this summary, further aspects and advantages of the invention will become apparent by reference to the drawings and by reading the detailed description that follows.

Nintendo Exhibit 1003
Part 2 of 2
Page 00631

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagram of one embodiment of an operating environment suitable for practicing the present invention;

FIG. 2 is a diagram of one embodiment of a computer system suitable for use in the operating environment of FIG. 1;

FIG. 3 is a diagram illustrating a system-level overview of an embodiment of the invention;

FIG. 4 is a flowchart of a configuration utility method to be performed by a computer to according to an embodiment of the invention; and

FIGs. 5A and 5B are flowcharts of a protected execution agent method to be performed by a computer to according to an embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

In the following detailed description of embodiments of the invention, reference is made to the accompanying drawings in which like references indicate similar elements, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that logical, mechanical, electrical and other changes may be made without departing from the scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.

The detailed description is divided into four sections. In the first section, the hardware and the operating environment in conjunction with which embodiments of the invention may be practiced are described. In the second section, a system level overview of the invention is presented. In the third section, methods for an embodiment of the invention are described. Finally, in the fourth section, a conclusion of the detailed description is provided.

Operating Environment

The following description of FIGs. 1 and 2 is intended to provide an overview of

Nintendo Exhibit 1003
Part 2 of 2
Page 00632

computer hardware and other operating components suitable for implementing the invention, but is not intended to limit the applicable environments. One of skill in the art will immediately appreciate that the invention can be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention can also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network.

FIG. 1 shows several computer systems that are coupled together through a network 103, such as the Internet. The term "Internet" as used herein refers to a network of networks which uses certain protocols, such as the TCP/IP protocol, and possibly other protocols such as the hypertext transfer protocol (HTTP) for hypertext markup language (HTML) documents that make up the World Wide Web (web). The physical connections of the Internet and the protocols and communication procedures of the Internet are well known to those of skill in the art. Access to the Internet 103 is typically provided by Internet service providers (ISP), such as the ISPs 105 and 107. Users on client systems, such as client computer systems 121, 125, 135, and 137 obtain access to the Internet through the Internet service providers, such as ISPs 105 and 107. Access to the Internet allows users of the client computer systems to exchange information, receive and send e-mails, and view documents, such as documents which have been prepared in the HTML format. These documents are often provided by web servers, such as web server 109 which is considered to be "on" the Internet. Often these web servers are provided by the ISPs, such as ISP 105, although a computer system can be set up and connected to the Internet without that system being also an ISP as is well known in the art.

The web server 109 is typically at least one computer system which operates as a server computer system and is configured to operate with the protocols of the World Wide Web and is coupled to the Internet. Optionally, the web server 109 can be part of an ISP which provides access to the Internet for client systems. The web server 109 is shown coupled to the server computer system 111 which itself is coupled to web content 110, which can be considered a form of a media database. It will be appreciated that

Nintendo Exhibit 1003
Part 2 of 2
Page 00633

while two computer systems 109 and 111 are shown in FIG. 2, the web server system 109 and the server computer system 111 can be one computer system having different software components providing the web server functionality and the server functionality provided by the server computer system 111 which will be described further below.

Client computer systems 121, 125, 135, and 137 can each, with the appropriate web browsing software, view HTML pages provided by the web server 109. The ISP 105 provides Internet connectivity to the client computer system 121 through the modem interface 123 which can be considered part of the client computer system 121. The client computer system can be a personal computer system, a network computer, a WebTV system, or other such computer system. Similarly, the ISP 107 provides Internet connectivity for client systems 125, 135, and 137, although as shown in FIG. 1, the connections are not the same for these three computer systems. Client computer system 125 is coupled through a modem interface 127 while client computer systems 135 and 137 are part of a LAN. While FIG. 1 shows the interfaces 123 and 127 as generically as a "modem," it will be appreciated that each of these interfaces can be an analog modem, ISDN modem, cable modem, satellite transmission interface (e.g. "Direct PC"), or other interfaces for coupling a computer system to other computer systems. Client computer systems 135 and 137 are coupled to a LAN bus 133 through network interfaces 139 and 141, which can be Ethernet network or other network interfaces. The LAN bus 133 is also coupled to a gateway computer system 131 which can provide firewall and other Internet related services for the local area network. This gateway computer system 131 is coupled to the ISP 107 to provide Internet connectivity to the client computer systems 135 and 137. The gateway computer system 131 can be a conventional server computer system. Also, the web server system 109 can be a conventional server computer system.

Alternatively, as well-known, a server computer system 143 can be directly coupled to the LAN bus 133 through a network interface 145 to provide files 147 and other services to the clients 135, 137, without the need to connect to the Internet through the gateway system 131.

FIG. 2 shows one example of a conventional computer system that can be used as a client computer system or a server computer system or as a web server system. It will also be appreciated that such a computer system can be used to perform many of the

Nintendo Exhibit 1003
Part 2 of 2
Page 00634

functions of an Internet service provider, such as ISP 105. The computer system 201 interfaces to external systems through the modem or network interface 203. It will be appreciated that the modem or network interface 203 can be considered to be part of the computer system 201. This interface 203 can be an analog modem, ISDN modem, cable modem, token ring interface, satellite transmission interface (e.g. "Direct PC"), or other interfaces for coupling a computer system to other computer systems. The computer system 201 includes a processor 205, which can be a conventional microprocessor such as an Intel Pentium microprocessor or Motorola Power PC microprocessor. Memory 209 is coupled to the processor 205 by a bus 207. Memory 209 can be dynamic random access memory (DRAM) and can also include static RAM (SRAM). The bus 207 couples the processor 205 to the memory 209 and also to non-volatile storage 215 and to display controller 211 and to the input/output (I/O) controller 217. The display controller 211 controls in the conventional manner a display on a display device 213 which can be a cathode ray tube (CRT) or liquid crystal display. The input/output devices 219 can include a keyboard, disk drives, printers, a scanner, and other input and output devices, including a mouse or other pointing device. The display controller 211 and the I/O controller 217 can be implemented with conventional well known technology. A digital image input device 221 can be a digital camera which is coupled to an I/O controller 217 in order to allow images from the digital camera to be input into the computer system 201. The non-volatile storage 215 is often a magnetic hard disk, an optical disk, or another form of storage for large amounts of data. Some of this data is often written, by a direct memory access process, into memory 209 during execution of software in the computer system 201. One of skill in the art will immediately recognize that the term "computer-readable medium" includes any type of storage device that is accessible by the processor 205 and also encompasses a carrier wave that encodes a data signal.

It will be appreciated that the computer system 201 is one example of many possible computer systems which have different architectures. For example, personal computers based on an Intel microprocessor often have multiple buses, one of which can be considered to be a peripheral bus. Network computers are another type of computer system that can be used with the present invention. Network computers do not usually

Nintendo Exhibit 1003
Part 2 of 2
Page 00635

include a hard disk or other mass storage, and the executable programs are loaded from a network connection into the memory 209 for execution by the processor 205. A WebTV system, which is known in the art, is also considered to be a computer system according to the present invention, but it may lack some of the features shown in FIG. 2, such as certain input or output devices. A typical computer system will usually include at least a processor, memory, and a bus coupling the memory to the processor.

It will also be appreciated that the computer system 201 is controlled by operating system software which includes a file management system, such as a disk operating system, which is part of the operating system software. One example of an operating system software with its associated file management system software is the family of operating systems known as Windows from Microsoft Corporation of Redmond, Washington, and their associated file management systems. The file management system is typically stored in the non-volatile storage 215 and causes the processor 205 to execute the various acts required by the operating system to input and output data and to store data in memory, including storing files on the non-volatile storage 215.

System Level Overview

A system level overview of the operation of an embodiment of the invention is described by reference to FIG. 3. When application software is installed on a computer system, the executables and supporting files are stored on non-volatile storage in a directory or directories arranged in a hierarchical tree structure. Because multiple applications can share executables, the installation of one application may overwrite a shared executable previously stored by a previously installed application, causing the earlier installed application to fail. Additionally, many operating systems maintain a configuration file or "registry" in which various settings for the installed applications are recorded and the settings for a later installed application may conflict with those of an earlier installed application.

FIG. 3 illustrates a computer system 300 in which the present invention protects executables and files for a pre-determined set of applications from being modified by later installed applications. The pre-determined set of applications are classified as

Nintendo Exhibit 1003
Part 2 of 2
Page 00636

authorized; all other applications are classified as unauthorized. The directory paths for the authorized applications form a protected execution environment 317 (shown generically as hard disk 317) that only authorized applications are allowed to change. A protected execution agent 301 is installed into a file system manager 303 to control modifications to the protected execution environment 317 by intercepting input/output (I/O) requests from all applications 307.

When an application 307 issues 1 a file I/O request 305, the protected execution agent 301 intercepts 2 the request 305 and determines if the application 307 is authorized. If so, the protected execution agent 301 allows the I/O request 305 to continue 3 onto the appropriate file system driver 311 without taking any action. The file system driver 311 then performs 4 the requested I/O operation against the specified directory in the protected execution environment 317.

On the other hand, if the application 307 is unauthorized and is attempting to modify the protected execution environment 317, the protected execution agent 301 does not allow the original I/O request 305 to continue on, essentially terminating the original I/O request 305. The protected execution agent 301 creates a redirected I/O request 309 that specifies a directory path in an alternate environment (illustrated generically as hard disk 313) as a substitute for the original I/O request 305. The protected execution agent 301 then submits 5 the redirected I/O request 309 to the file system manager 303 and registers itself with the file system manager 303 as responsible for the original directory path. When the protected execution agent 301 intercepts 6 the redirected I/O request 309, it recognizes the request as its own and allows the redirected I/O request 309 to continue 7 onto the appropriate file system driver 311 for processing. The file system driver 311 consequently performs 8 the requested I/O operation against the alternate environment 313.

Thus, when an unauthorized application 307 attempts to store a file 315 in a directory in the protected execution environment 317, the file 315 is instead stored in the corresponding directory in the alternate environment 313. However, subsequent I/O requests 305 for the file 315 continue to specify the original directory path in the protected execution environment 317 since the application 307 is unaware of the change. Because the protected execution agent 301 has registered for the original directory path,

Nintendo Exhibit 1003
Part 2 of 2
Page 00637

each subsequent I/O request 305 is delivered directly to it and it submits the corresponding redirected I/O request 309 to the file system manager 303 as described above. When the redirected I/O request 309 is processed 8 by the file system driver 311, the file system driver 311 retrieves 9 the file 315 from the corresponding directory in the alternate environment 313 for subsequent processing by the application 307.

The system level overview of the operation of an embodiment of the invention has been described in this section of the detailed description. A protected execution agent intercepts I/O requests from applications and permits only authorized applications to modify a protected execution environment. Attempts to modify the protected execution environment by unauthorized applications are directed to an alternate environment. The alternate environment serves as a "virtualization" environment to hold files for unauthorized applications that cannot be stored in the protected environment because of their potential for conflict with files for authorized applications. The protected execution agent acts as a "virtualization" agent, causing I/O requests from the unauthorized applications to be directed to the virtualization environment without having to change the programming of the unauthorized applications.

While the invention is not limited to any particular number of environments, for sake of clarity a simplified arrangement of one protected execution environment and one alternate environment has been described. It will be appreciated also that the environments have been represented in FIG. 3 as separate hard disks but that the invention does not require the environments to be physically separate.

The present invention is particularly useful to organizations that support multiple computer users. The support organization would configure the protected execution environment to include only directories for supported (authorized) applications before delivering the computer to the user. Whenever the computer is booted, the protected execution agent is installed in the file system and prevents the user from installing unsupported (unauthorized) applications into the directories for supported (authorized) applications, thereby reducing the likelihood of failure of the supported applications.

Furthermore, by establishing a global set of supported applications, the support organization can maintain a master version of the protected execution environment for multiple computers and use the master version to replace a corrupted protected execution

Nintendo Exhibit 1003
Part 2 of 2
Page 00638

environment on a computer without having to perform lengthy diagnostic analysis to determine the cause of the failure. In situations in which the computers are remote from the support organization, a copy of the master version of the protected execution environment may be stored in a secured area on each computer and the remote support organization can use a network connection to copy the master version over a corrupted version. Alternatively, a network connection can be used to download the master version to the problem computer to overwrite the corrupted version.

Methods of Embodiments of the Invention

In the previous section, a system level overview of the operations of embodiments of the invention was described. In this section, the particular methods of the invention are described in terms of computer software with reference to a series of flowcharts. The methods to be performed by a computer constitute computer programs made up of computer-executable instructions. Describing the methods by reference to a flowchart enables one skilled in the art to develop such programs including such instructions to carry out the methods on suitably configured computers (the processor of the computer executing the instructions from computer-readable media). If written in a programming language conforming to a recognized standard, such instructions can be executed on a variety of hardware platforms and for interface to a variety of operating systems. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein. Furthermore, it is common in the art to speak of software, in one form or another (e.g., program, procedure, process, application, module, logic...), as taking an action or causing a result. Such expressions are merely a shorthand way of saying that execution of the software by a computer causes the processor of the computer to perform an action or a produce a result.

Referring first to FIG. 4, the acts to be performed by a computer executing a configuration utility method 400 are shown. The utility method 400 is performed whenever a support provider, such as a system administrator or a remote support organization, wants to classify the applications installed on a computer as authorized

Nintendo Exhibit 1003
Part 2 of 2
Page 00639

(supported) or unauthorized (unsupported) to configure the protected execution environment. In such situations, the user of the computer is prevented from executing the utility method 400 through conventional security techniques. It will be apparent that such restrictions are not necessary when the computer is a stand-alone system and the user is acting in the capacity of a systems administrator.

If this is the first execution of the method 400 (block 401), the method 400 scans the file system of the computer to determine all installed applications (block 403) and creates a list (block 405). The installed applications are categorized as authorized or unauthorized (block 407). In one embodiment, a user manually categorizes the applications. In another embodiment, the categories are automatically assigned based on a master list of authorized applications.

The protected execution environment is defined by the directory paths for the authorized applications. In one embodiment, the utility user can manually alter the directory structure of the protected execution environment as part of the processing of block 407. In an alternate embodiment, alternations to the directory structure for the protected execution environment are input as parameters to the utility method 400. One of skill in the art will immediately perceive that the alternate environment can also be created by the configuration utility from the directory paths of the unauthorized applications although such an embodiment is not shown.

A special "mirror" directory structure is created that mimics the directory structure of the protected execution environment (block 409). Each executable file for an authorized application appears as a null entry in the appropriate directory in the mirror directory structure. In an alternate embodiment, non-executable files in the protected execution environment may also have corresponding null entries in the mirror directory structure. The mirror directory structure is distinguished from the directory structure of the protected execution environment by prefixing each directory path in the system with a unique top level directory name. For example in a Microsoft Windows environment, the mirror directory structure that corresponds to the standard Windows directory structure of "c:\windows" may be "c:\everdreamagentprocess\windows" so that the null entry for the Windows main executable is found at "c:\everdreamagentprocess\windows\win.exe."

Nintendo Exhibit 1003
Part 2 of 2
Page 00640

On each subsequent execution, the utility method 400 retrieves the previous list of installed applications (block 411), scans the file system (block 413), and determines if a new application has been installed since the last execution of the utility (block 415). If so, the new application(s) is categorized as authorized or unauthorized (block 407). If no new applications have been installed, one or more of the previously installed applications are re-categorized (block 417), either manually or automatically as described above. In either case, the directory structure of the protected execution environment changes and mirror directory structure is updated to reflect the change (block 409). It will be appreciated that the utility method 400 can also be executed with parameters that indicate changes in the category of a previously installed application so that a scan of the file system is unnecessary.

FIGs. 5A-B illustrate methods that performs the processes for one embodiment of the protected execution agent 301 after it has been installed in a file system manager. It will be appreciated that the particulars of installation depend on the operating system of the computer. For example, in a computer running Windows 98, the protected execution agent 301 is a virtual device driver (VxD) and is coupled into a "hook chain" for the Windows installable file system so that the protected execution agent receives all I/O requests ahead of the other VxDs in the hook chain.

In the embodiment illustrated in FIG. 5A, when the protected execution agent method 500 intercepts an I/O request, it determines if the request is a redirected I/O request that it created (block 501). If so, the method 500 passes the request onto the next component for processing (block 513) and then waits for another I/O request (block 515). The next component is determined by the file and operating systems of the computer so that, for example, if the method 500 is for a protected execution agent executing under Windows 98, the appropriate component is the next VxD in the installable file system hook chain. In this fashion, other utilities that install themselves in the file system manager, such as virus scanners, operate against a file in the alternate environment without having to be aware of the configuration of the computer created by the invention.

If the I/O request is not a redirected request, the method 500 determines if the application is authorized (block 503). In one embodiment, the load path of the

Nintendo Exhibit 1003
Part 2 of 2
Page 00641

application, i.e., the directory path for the application executable, is used to determine if the application is authorized as described in detail below with reference to FIG. 5B. It will be appreciated that other techniques can be used to designate the category of an application and are contemplated as within the scope of the invention.

If the application is authorized, the method 500 passes the original I/O request onto the next component (block 513) and waits for another I/O request (block 515). On the other hand, if the application is unauthorized, the method 500 determines if the I/O request was issued against the protected execution environment (block 505). If so, it terminates the original I/O request (block 507), creates a redirected I/O request directed at the corresponding directory and submits the redirected I/O request to the file system (block 509). An alternate embodiment redirects each I/O request issued by an unauthorized application, even if the directory path specified is not part of the protected execution environment. If the corresponding directory does not exist, at block 507 the original directory path is used to create a corresponding directory path in the alternate environment.

The method 500 also claims responsibility for the original directory path so that all subsequent I/O requests will be passed immediately to the protected execution agent for processing (block 511). In an alternate embodiment, the agent method 500 claims responsibility for all directory paths in the protected execution environment when first installed in the file system.

In most operating environments, each instance of an executing application is assigned a process identifier and each I/O request issued by an application contains its process identifier. However the operating system may not maintain a relationship between the process identifier for an application and its load path. Therefore, additional processing may be necessary to associate the load path and the process identifier when the load path is used to determine if the application is authorized or unauthorized. FIG. 5B illustrates one embodiment of a method 520 that creates and maintains the relationship between a process identifier and the load path for an application and uses that relationship to determine the category of the application. The embodiment in FIG. 5B assumes a particular ordering of I/O requests will occur when an application is executed. One of skill in the art will immediately conceive of alternate embodiments

Nintendo Exhibit 1003
Part 2 of 2
Page 00642

that rely on a different order of I/O requests upon reading the following description of the processing represented by FIG. 5B. The method 520 illustrated in FIG. 5B also creates and maintains an active process data structure that associates a process identifier for an application with its load path and a parent-child data structure that tracks the relationship between an unauthorized application that spawns an authorized application. These data structures are created when an application is executed as described in detail further below.

When the protected execution agent method 500 intercepts an original I/O request, it calls the method 520 to determine if the application issuing the request is authorized. Assuming for the moment that the originating application is not spawning another application, control falls through to block 533 which retrieves the load path from an active process data structure using the process identifier of the originating application. The load path is used to create the corresponding path in the mirror directory (block 535). The method 520 submits a query on the mirror directory path to the file system (block 537). Recalling that only executables for authorized applications have entries in the mirror directory data structure, an successful query on the load path (block 539) means the application that originated the I/O request is authorized (block 543). However, the originating application may have been spawned by an unauthorized application and is thus, unauthorized. The parent-child data structure is used at block 541 to determine if an originating authorized application is a child of an unauthorized parent application. If so, then this instance of the originating application is designated as unauthorized (block 545). For purposes of this example, processing falls through block 547 and control is returned to the protected execution agent method 500. An unauthorized application spawned by an authorized application is still considered unauthorized, so no check of the parent-child data structure is required when the query at block 539 is unsuccessful.

The creation of the parent-child and the active process data structures is now described. One of skill in the art will immediately recognize that many well-known types of data structures, such as indexed files, data bases, hash tables, etc., are suitable for storing the information required to track the active processes and the parent-child relationships. The underlying support services for the data structures of the present

Nintendo Exhibit 1003
Part 2 of 2
Page 00643

invention may be provided by the file system or by third-party applications as is conventional and the technicalities are therefore not included in the following description.

Returning to the beginning blocks of FIG. 5B, when one application invokes the execution of a second application, such as when Windows 95/98 responds to a double-click on an executable file, the first I/O request issued is an "open" on the executable file for the second application (block 521). A file handle for the executable is extracted from the I/O open request and saved (block 523). The method 520 then continues processing at block 533 as described above.

After the executable file is opened, it must be initialized. When an initialization I/O request is intercepted (block 525), the saved file handle corresponding to the file handle in the request is flagged (block 527) because the next I/O request that contains the same file handle will also contain the directory path for the executable file. Thus, when an I/O request is intercepted that contains a file handle corresponding to a flagged file handle (block 529), the process identifier and directory path are extracted from the I/O request and saved as an entry in the active process data structure (block 531). Processing then continues at block 533 and progresses as described previously until block 547 is reached. Because the I/O request is on a file handle corresponding to a flagged file handle, the test at block 547 is true and the saved file handle is unflagged at block 549. Assuming that the originating application is unauthorized, an entry in the parent-child data structure is created to track the relationship between the originating parent application and the newly-executed child application (block 553). An alternate embodiment that maintains all relationships between parent and children applications in the parent-child data structure can be implemented by eliminating the test at block 551.

The entry in the parent-child data structure can contain the process identifiers for the parent and child applications or any other identifier that uniquely identifies the applications. Additionally, the active process data structure can be keyed on any other identifier that is uniquely associated with the directory path of the executable. Furthermore, it will be readily apparent to one of skill in the art that entries in the active process and parent-child data structures can be deleted or marked as unused using techniques specific to the type of data structure, although such actions are not illustrated

Nintendo Exhibit 1003
Part 2 of 2
Page 00644

in FIG. 5B.

The particular methods performed by a computer to execute one embodiment of the protected execution agent and supporting configuration utility of the invention have been described. The method performed to implement the configuration utility has been shown by reference to a flowchart in FIG. 4, including all the acts from 401 until 417. The methods performed to implement the protected execution agent have been shown by reference to flowcharts in FIGs. 5A-B, including all acts from 501 until 515 and from 521 until 553.

Conclusion

A protected execution agent and supporting configuration utility have been described that create and prevent unauthorized modifications to a protected execution environment on a computer, thus preventing the installation of one application from overwriting components and settings required by another application. Thus, the present invention is particularly applicable for a remote support organization in isolating supported applications on a computer from unsupported ones without adversely impacting the user's access to either type of application.

Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiments shown. This application is intended to cover any adaptations or variations of the present invention. For example, those of ordinary skill within the art will appreciate that the protected execution agent could intercept only certain I/O requests, such as open, initialize, find, close, to increase the speed of processing the I/O requests. Furthermore, those of ordinary skill within the art will appreciate that the protected execution agent can be extended to provide backup capabilities by saving copies of files that are being deleted from the protected execution environment.

The terminology used in this application with respect to the operating and file systems is meant to include all such environments that do not automatically isolate user modifications to pre-determined areas. Therefore, it is manifestly intended that this invention be limited only by the following claims and equivalents thereof.

Nintendo Exhibit 1003
Part 2 of 2
Page 00645

CLAIMS

What is claimed is:

1. A method of providing a protected execution environment on a computer comprising:

intercepting an input/output request for a file from an application;

determining if the application is authorized to modify the protected execution environment;

creating a redirected input/output request to an alternate environment when the application is not authorized to modify the protected execution environment and the file is within the protected execution environment; and

submitting the redirected input/output request to a file system manager.

2. The method of claim 1 further comprising:

allowing the redirected input/output request to continue when it is intercepted.

3. The method of claim 1 further comprising:

creating the protected execution environment.

4. The method of claim 1 wherein the protected execution environment comprises a directory for each of the applications that is authorized to modify the protected execution environment.

5. The method of claim 1 further comprising:

categorizing each application installed on the computer as authorized or not authorized to modify the protected execution environment.

6. The method of claim 1 wherein the alternate environment comprises a directory associated with an application that is not authorized to modify the protected execution environment.

Nintendo Exhibit 1003
Part 2 of 2
Page 00646

7. The method of claim 1 wherein the redirected input/output request specifies a directory in the alternate environment that corresponds to a directory in the protected execution environment specified in the input/output request.

8. The method of claim 1, wherein a parent-child relationship is maintained between an application that invokes another application.

9. The method of claim 1, wherein determining if the application is authorized to modify the protected execution environment comprises:

designating the application as not authorized to modify the protected execution environment if the application was invoked by another application that is not authorized to modify the protected execution environment.

10. The method of claim 1, further comprising:

creating a null entry in a mirror directory structure for an executable for each application authorized to modify the protected execution environment,

wherein determining if the application is authorized to modify the protected execution environment comprises:

querying the existence of the executable for the application in the mirror directory structure.

11. The method of claim 10, further comprising:

maintaining an association between an executing application and a directory path for the executable for the executing application,

wherein querying for the existence of the executable in the mirror data structure comprises:

specifying the directory path for the executable associated with the executing application.

12. A method for operating a computer system with a protected execution

Nintendo Exhibit 1003
Part 2 of 2
Page 00647

environment comprising:

executing a configuration utility to categorize a plurality of applications installed on the computer system as authorized or not authorized to modify the protected execution environment;

defining the protected execution based on the authorized applications; and

installing a protected execution agent in a file system to intercept input/output requests submitted by the applications, wherein the protected execution agent directs an input/output request to an alternate environment if the application that submitted the request is not authorized and the request is directed to the protected execution environment.

13. The method of claim 12 wherein the configuration utility defines the protected execution environment when categorizing the plurality of applications.

14. The method of claim 12 wherein the alternate environment is defined based on at least one application that is not authorized.

15. The method of claim 12, wherein the alternate environment is defined by the configuration utility when categorizing the plurality of applications.

16. The method of claim 12, wherein the configuration utility further creates a null entry in a mirror directory structure for an executable for each authorized application and the protected execution agent further queries the existence of the executable for an executing application in the mirror directory structure to determine if the application is authorized.

17. The method of claim 16, wherein the protected execution agent further maintains an association between the executing application and a directory path for the executable for the executing application.

18. The method of claim 12, wherein the protected execution agent designates a

Nintendo Exhibit 1003
Part 2 of 2
Page 00648

second application as not authorized if it was invoked by a first application that is not authorized.

19. The method of claim 18, wherein the protected execution agent maintains a parent-child relationship between the first and second applications.

20. The method of claim 12, wherein the protected execution agent is installed in a hook chain in a file system manager to intercept the input/output requests before the requests are processed by any other agent installed in the hook chain.

21. The method of claim 12, wherein the configuration utility is executed prior to providing the computer system to a user and the protected execution agent is installed each time the computer system is booted.

22. The method of claim 12, further comprising:

saving a copy of the protected execution environment; and

recovering from a failure of the computer system by replacing the protected execution environment with the copy.

23. The method of claim 22, wherein the copy is saved on the computer system in a secure location.

24. The method of claim 22, wherein the copy is saved on a remote computer server and downloaded to the computer system.

25. A method of determining a category for an application on a computer comprising:

categorizing the application as a first type;

creating a directory in a second directory structure for the application when it is a first type, wherein the second directory structure mirrors a first directory structure that contains an executable for the application;

Nintendo Exhibit 1003
Part 2 of 2
Page 00649

creating a null entry for the executable for the application in the directory in the second directory structure when the application is the first type;

querying the existence of the executable for the application in the second directory structure, wherein the application is determined to be the first type when the executable exists.

26. A computer-readable medium having stored thereon computer-executable instructions for performing a method comprising:

intercepting an input/output request for a file from an application;

determining if the application is authorized to modify the protected execution environment;

creating a redirected input/output request to an alternate environment when the application is not authorized to modify the protected execution environment and the file is within the protected execution environment; and

submitting the redirected input/output request to a file system manager.

27. The computer-readable medium of claim 26 having further computer-readable instructions comprising:

allowing the redirected input/output request to continue when it is intercepted.

28. The computer-readable medium of claim 26 having further computer-readable instructions comprising:

categorizing each application installed on the computer as authorized or not authorized to modify the protected execution environment.

29. The computer-readable medium of claim 26 having further computer-readable instructions comprising:

creating the protected execution environment from a directory for each of the applications that is authorized to modify the protected execution environment.

30. The computer-readable medium of claim 26 having further computer-readable

Nintendo Exhibit 1003
Part 2 of 2
Page 00650

instructions comprising:

creating the alternate environment from a directory associated with an application that is not authorized to modify the protected execution environment.

31. The computer-readable medium of claim 26 having further computer-readable instructions comprising:

storing a directory path specified in the input/output request in the redirected input/output request to direct the request to a corresponding directory path in the alternate environment.

32. The computer-readable medium of claim 26 having further computer-readable instructions comprising:

maintaining a parent-child data structure to track between relationships between applications that invoke other applications.

33. The computer-readable medium of claim 26 having further computer-readable instructions comprising:

designating the application as not authorized to modify the protected execution environment if the application was invoked by another application that is not authorized to modify the protected execution environment.

34. The computer-readable medium of claim 26 having further computer-readable instructions comprising:

creating a null entry in a mirror directory structure for an executable for each application authorized to modify the protected execution environment; and

querying the existence of the executable for the application in the mirror directory structure when determining if the application is authorized to modify the protected execution environment.

35. The computer-readable medium of claim 34 having further computer-readable instructions comprising:

Nintendo Exhibit 1003
Part 2 of 2
Page 00651

maintaining an association between an executing application and a directory path for the executable for the executing application; and

specifying the directory path for the executable associated with the executing application when querying for the existence of the executable in the mirror data structure.

36. A computer system comprising:

a processing unit;

a memory coupled to the processing unit through a system bus;

a computer-readable medium coupled to the processing through the system bus; and

a protected environment agent executing from the computer-readable medium, wherein the protected environment agent causes the processing unit to intercept input/output requests submitted by applications executing on the computer system and further causes the processing unit to redirect each input/output request to an alternate environment if the application that submitted the request is not authorized to modify a protected execution environment and the request is directed to the protected execution environment.

37. The computer system of claim 36 further comprising:

a configuration utility executing from the computer-readable medium, wherein the configuration utility causes the processing unit to categorize each application installed on the computer system as authorized or not authorized to modify the protected execution environment and further to cause the processing unit to define the protected execution environment to contain directories associated with the authorized applications.

Nintendo Exhibit 1003
Part 2 of 2
Page 00652

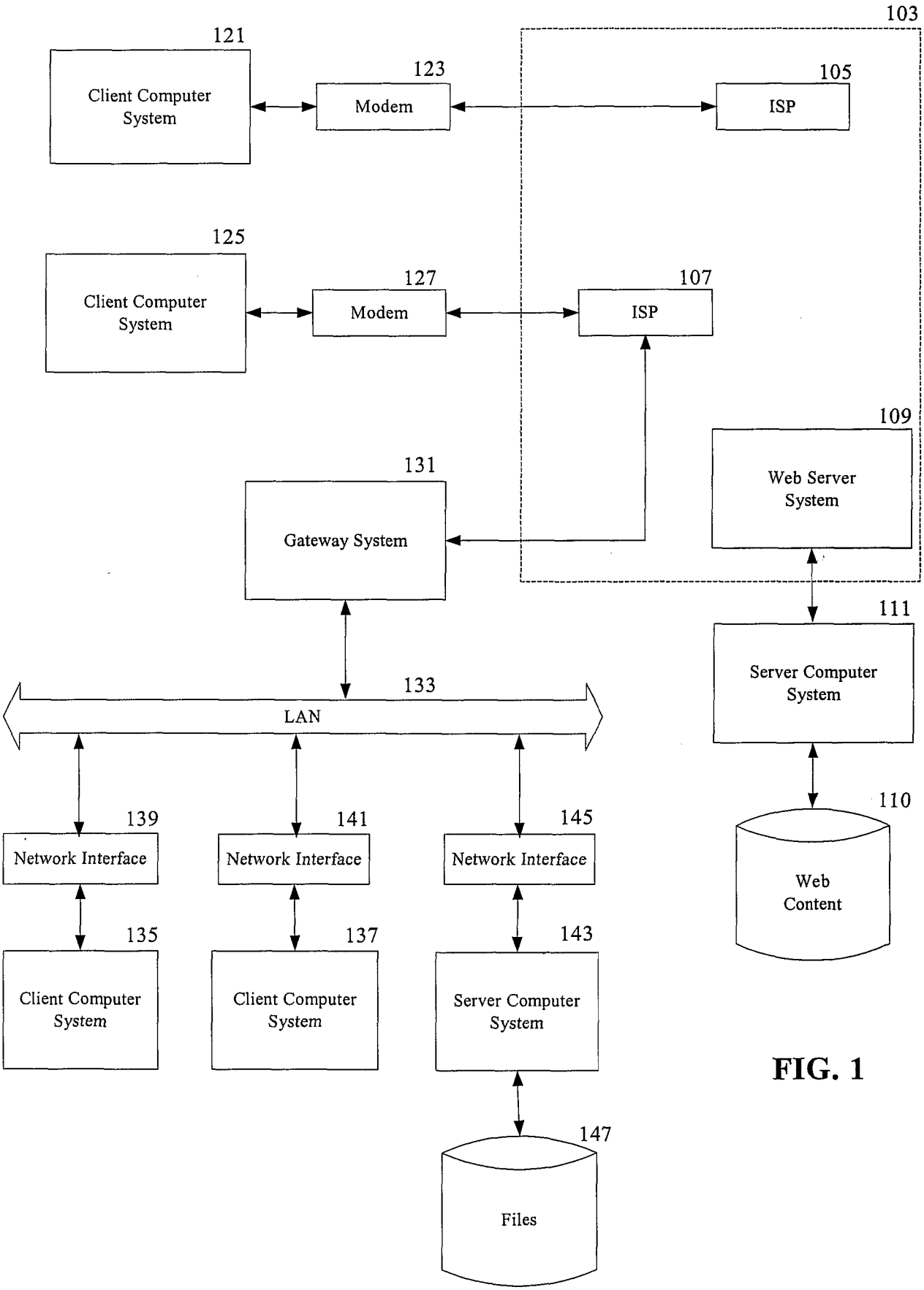

101
103
121
Client Computer System
123
Modem
105
ISP
125
Client Computer System
127
Modem
107
ISP
109
Web Server System
131
Gateway System
111
Server Computer System
133
LAN
110
Web Content
139
Network Interface
141
Network Interface
145
Network Interface
135
Client Computer System
137
Client Computer System
143
Server Computer System
147
Files

FIG. 1

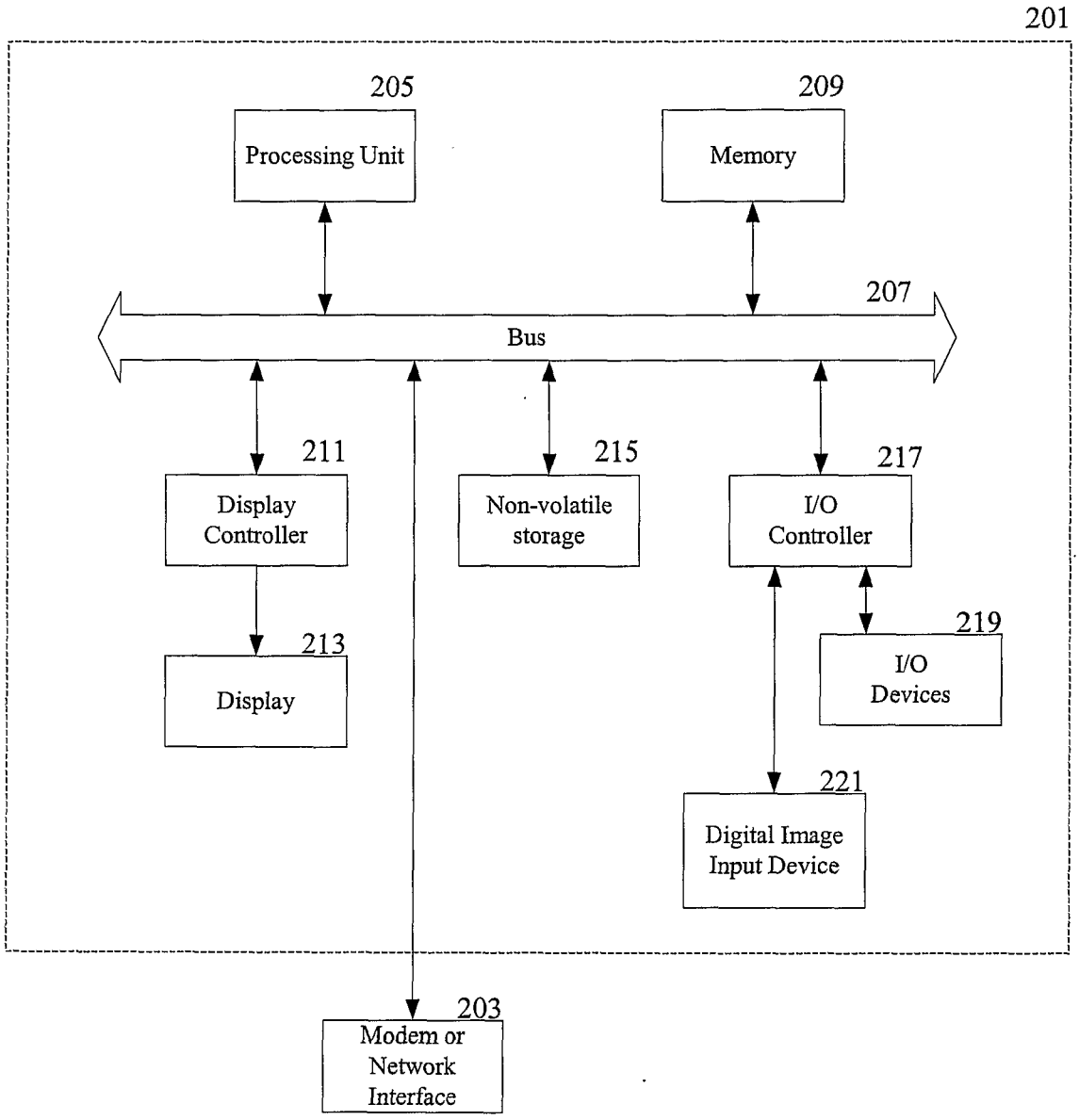

201
205
Processing Unit
209
Memory
207
Bus
211
Display Controller
215
Non-volatile storage
217
I/O Controller
213
Display
219
I/O Devices
221
Digital Image Input Device
203
Modem or Network Interface

FIG. 2

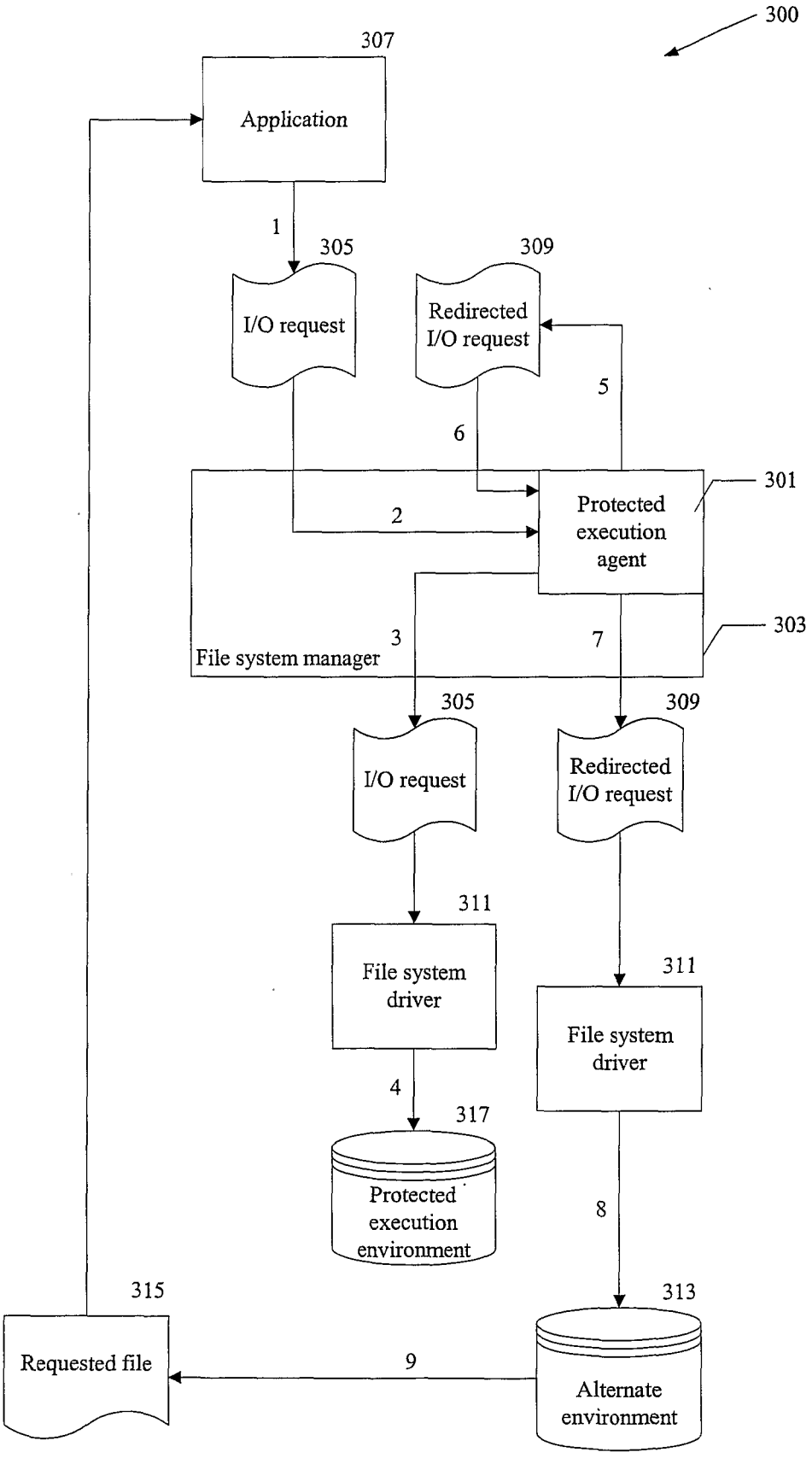

300
307
Application
1
305
I/O request
309
Redirected I/O request
5
6
301
Protected execution agent
2
303
File system manager
3
7
305
I/O request
309
Redirected I/O request
311
File system driver
311
File system driver
4
317
Protected execution environment
8
315
Requested file
9
313
Alternate environment

FIG. 3

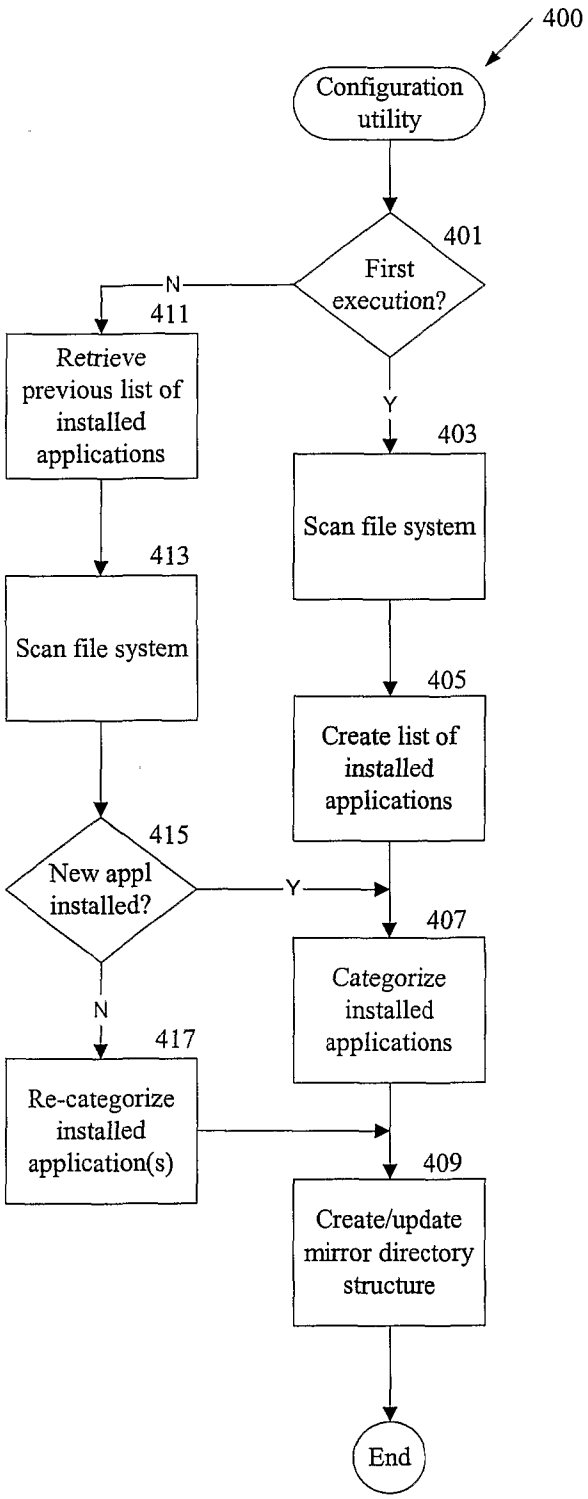

400
Configuration utility
401
First execution?
N
411
Retrieve previous list of installed applications
413
Scan file system
415
New appl installed?
N
417
Re-categorize installed application(s)
Y
403
Scan file system
405
Create list of installed applications
407
Categorize installed applications
409
Create/update mirror directory structure
End

FIG. 4

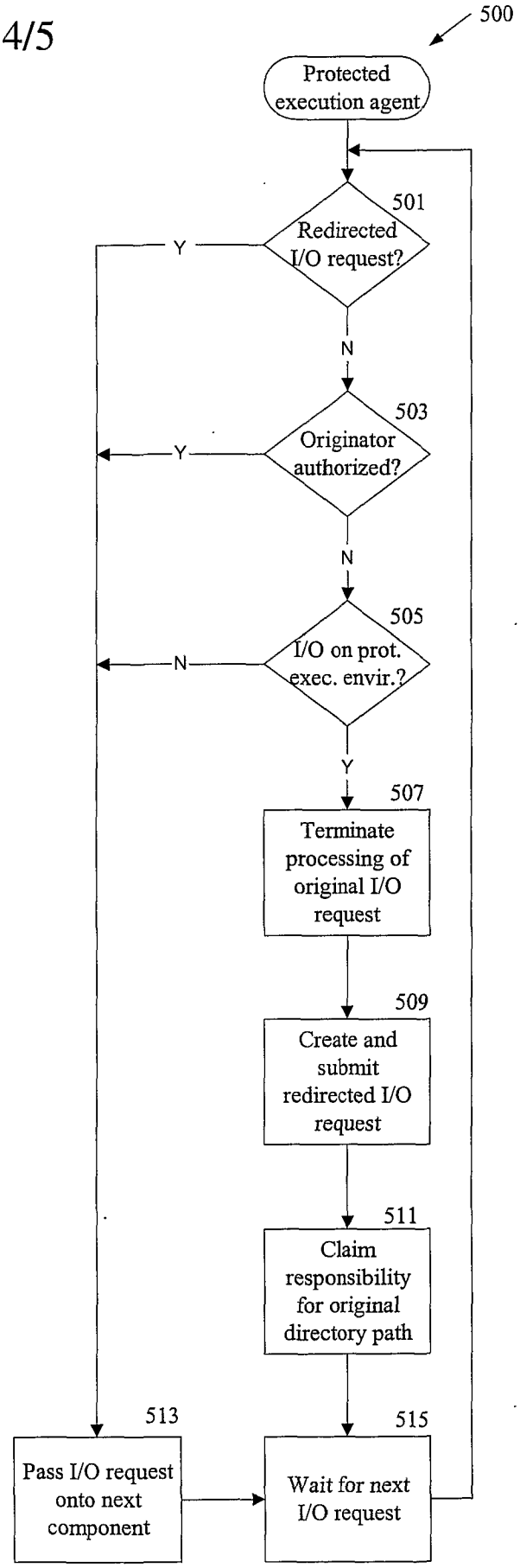

500
Protected execution agent
501
Redirected I/O request?
Y
N
503
Originator authorized?
Y
N
505
I/O on prot. exec. envir.?
N
Y
507
Terminate processing of original I/O request
509
Create and submit redirected I/O request
511
Claim responsibility for original directory path
513
Pass I/O request onto next component
515
Wait for next I/O request

FIG. 5A

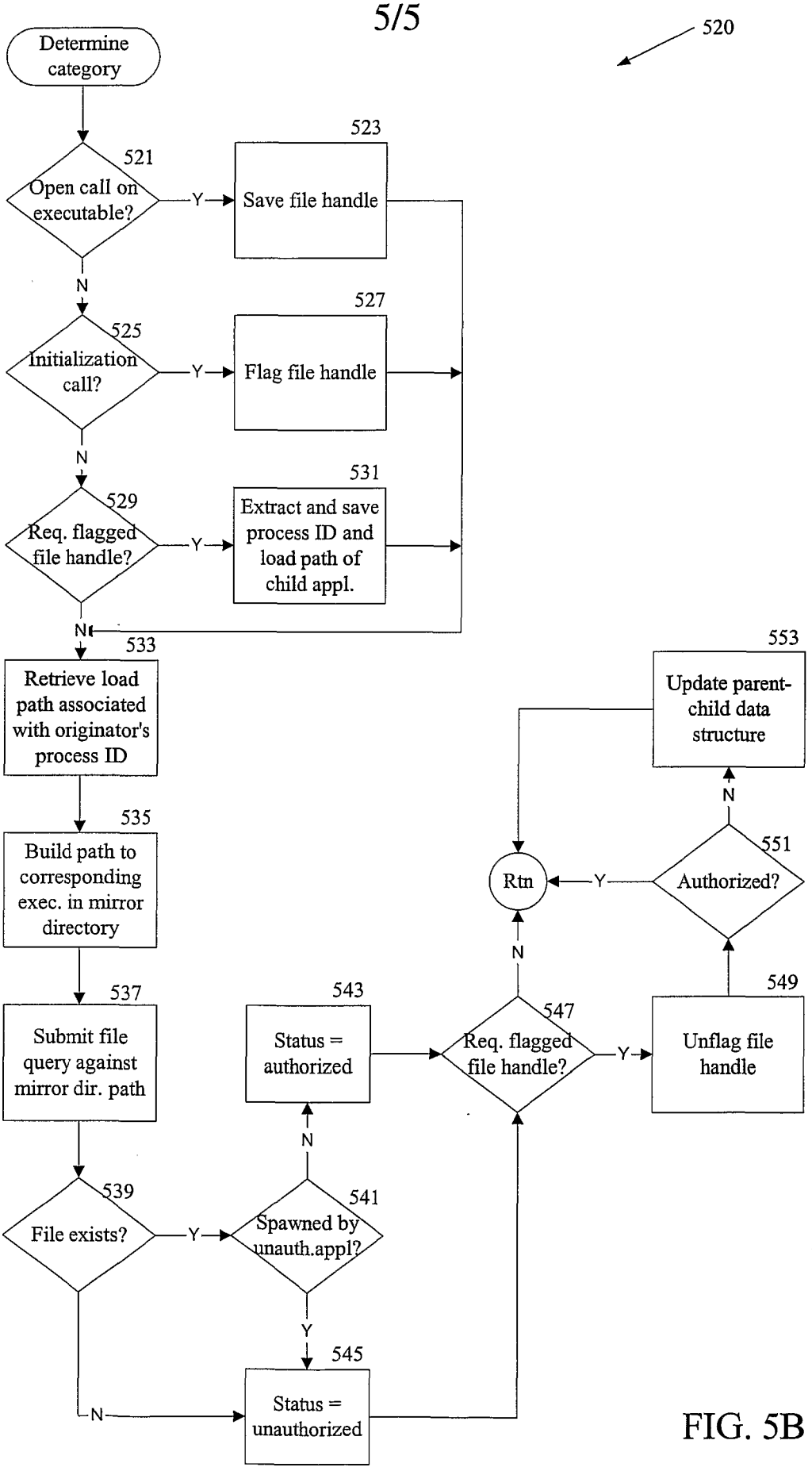

520
Determine category
521
Open call on executable?
Y
523
Save file handle
N
525
Initialization call?
Y
527
Flag file handle
N
529
Req. flagged file handle?
Y
531
Extract and save process ID and load path of child appl.
N
533
Retrieve load path associated with originator's process ID
535
Build path to corresponding exec. in mirror directory
537
Submit file query against mirror dir. path
539
File exists?
Y
541
Spawned by unauth.appl?
N
543
Status = authorized
N
545
Status = unauthorized
547
Req. flagged file handle?
Y
549
Unflag file handle
N
551
Authorized?
Y
Rtn
N
553
Update parent-child data structure

FIG. 5B

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/10441

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :Please See Extra Sheet.
US CL :Please See Extra Sheet.
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. : 713/200, 201; 709/226, 229; 707/1, 9, 10, 100, 103X, 103Y, 103Z, 103R

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

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

Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6,026,402 A (VOSSEN et al) 15 February 2000, see entire document	1,3-26,28-37
A,E	US 6,236,996 B1 (BAPAT et al) 22 May 2001, see entire document	1,12,25,26,36
A,P	US 6,085,191 A (FISCHER et al) 04 July 2000, see entire document	1,12,25,26,36
A	US 6,038,563 A (BAPAT et al) 14 March 2000, see entire document	1,12,25,26,36
A	US 5,999,978 A (ANGAL et al) 07 December 1999, see entire document	1,12,25,26,36

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

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search	Date of mailing of the international search report
17 JUNE 2001	26 JUL 2001

Name and mailing address of the ISA/US	Authorized officer
Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	CHRISTOPHER A. REVAK
Facsimile No. (703) 305-3230	Telephone No. (703) 305-9618

Nintendo Exhibit 1003
Part 2 of 2
Page 00658

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/10441

A. CLASSIFICATION OF SUBJECT MATTER:
IPC (7):

G06F 7/00, 11/30, 12/14, 15/16, 15/173, 17/00, 17/30; H04L 9/00, 9/32

A. CLASSIFICATION OF SUBJECT MATTER:
US CL :

713/200, 201; 709/226, 229; 707/1, 9, 10, 100, 103X, 103Y, 103Z, 103R

B. FIELDS SEARCHED
Electronic data bases consulted (Name of data base and where practicable terms used):

BRS (FILES: USPAT, JPO, EPO, DERWENT, IBM TDB'S), DIALOG (COMPSCI, ELECTRON, SOFTWARE)

search terms: request, requests, requesting, requested, command, commands, files, file, object, objects, application, applications, modify, modification, modified, modifying, change, changed, changing, alter, altered, altering, alteration, authorized, authorized, authorizing, authorization, permit, permitting, permitted, allow, allowed, allowing, grant, granting, granted, redirect, redirected, redirecting, unauthorize, unauthorized, hacker, malicious, illicit, illegal, compromise, compromising, compromised

Nintendo Exhibit 1003
Part 2 of 2
Page 00659

EP1427166

Publication Title:

Access management

Abstract:

Abstract of EP1427166

12a2 Access control to a networked peripheral device by a walk-up user, wherein the networked peripheral device is accessible by both the walk-up user and a remote user, based on centralized access management information. Access control comprises receiving authenticated information for the walk-up user from the networked peripheral device at a centralized location, determining at a server connected to the network a first level of access to the networked peripheral device, determining at the networked peripheral device a second level of access to the networked peripheral device by the walk-up user based on received access management information for the walk-up user, and allowing the walk-up user to access determined user-available features of the networked peripheral device based on the determined first and second levels of access.

Data supplied from the esp@cenet database - Worldwide

Courtesy of http://v3.espacenet.com

This Patent PDF Generated by Patent Fetcher(TM), a service of Stroke of Color, Inc.

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

(11) EP 1 427 166 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
09.06.2004 Bulletin 2004/24

(51) Int Cl.[7]: **H04L 29/06**

(21) Application number: **03257583.9**

(22) Date of filing: **02.12.2003**

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

(30) Priority: **05.12.2002 US 309884**

(71) Applicant: **CANON KABUSHIKI KAISHA**
Tokyo (JP)

(72) Inventors:
- **Iwamoto, Neil Y.**
  Irvine CA 92618-3731 (US)
- **Seikh, Attaullah**
  Irvine CA 92618-3731 (US)
- **Paek, Jeanette Yang**
  Irvine CA 92618-3731 (US)
- **Martinez, Martin**
  Irvine CA 92618-3731 (US)
- **Slick, Royce E.**
  Irvine CA 92618-3731 (US)
- **Chern, Wei-Jhy**
  Irvine CA 92618-3731 (US)
- **Khosrova, Eliza**
  Irvine CA 92618-3731 (US)
- **Yang, Joseph**
  Irvine CA 92618-3731 (US)

(74) Representative:
Beresford, Keith Denis Lewis et al
BERESFORD & Co.
16 High Holborn
London WC1V 6BX (GB)

(54) Access management

(57) Access control to a networked peripheral device by a walk-up user, wherein the networked peripheral device is accessible by both the walk-up user and a remote user, based on centralized access management information. Access control comprises receiving authenticated information for the walk-up user from the networked peripheral device at a centralized location, determining at a server connected to the network a first level of access to the networked peripheral device, determining at the networked peripheral device a second level of access to the networked peripheral device by the walk-up user based on received access management information for the walk-up user, and allowing the walk-up user to access determined user-available features of the networked peripheral device based on the determined first and second levels of access.

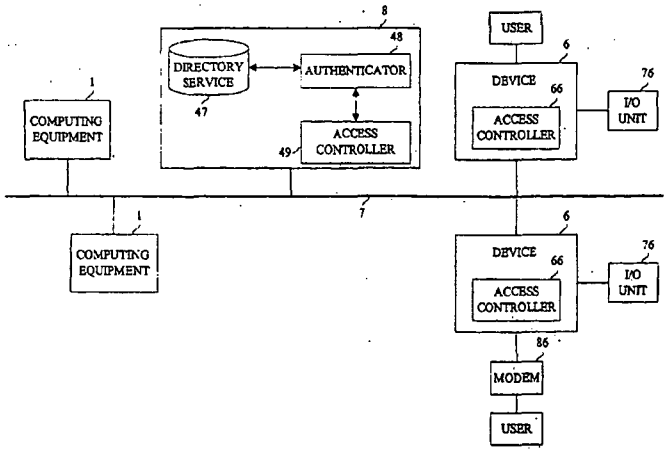


Fig. 3

Printed by Jouve, 75001 PARIS (FR)

EP 1 427 166 A2

Description

[0001] The invention relates to a two-pass, or two-level, access management for network peripheral devices, and more particularly, managing access to a network peripheral device based on user privileges information at both a centralized access manager and at the network peripheral device.

[0002] Networked peripheral devices are typically multifunction devices that handle functions such as printing, scanning, copying or faxing and are often relied upon in performing enterprise level tasks. Controlling usage of a networked peripheral device means that a walk-up user of the device, i.e., one that gains access to the device locally at the device, or a remote user of the device, i.e., a non walk-up user, is allowed access to only those services and/or features available on the device that are authorized. For example, a user may have access to copying but not a printing service of a networked peripheral device that offers faxing, printing, copying and scanning services. Likewise, a user may only be allowed access to a black-and-white, but not a color printing feature of the printing service offered by the device.

[0003] Controlling usage of services and/or features offered by a networked peripheral device is often necessary for economic as well as other reasons. For example, color ink cartridges containing ink for color printing typically cost significantly more than black ink cartridges that are used for black and white printing. It may therefore be desirable to limit access to color printing features of a printer in order to save costs.

[0004] One conventional approach to controlling access to a networked peripheral device by a non-enterprise user is to program each device separately as to the access privileges of the user. This approach requires presence of an input mechanism such as a keyboard or a magnetic card reader on the device through which the user logs into the device. Following login by the user, the device examines the user's access privileges, which are maintained by the device, and allows the user access to the device based on the programmed information concerning the user's access privileges. Adoption of this approach is cumbersome since it requires separate programming of each networked peripheral device and reprogramming of each device in response to changes in access policy.

[0005] Another conventional approach to controlling access to a networked peripheral device by a non-enterprise user is to program user access privileges on a keycard that is carried by the user. Upon swiping the keycard on a card reader installed on the device, the user access policy is transmitted from the card to the device. The user is then allowed to use the device in accordance with the limitations contained in the user access policy. This approach also suffers from the drawback suffered by the previous alternative in that each individual card needs to be programmed in order to encode user access policy and reprogrammed each time the policy is revised as to that user.

[0006] It would therefore be beneficial to be able to control and customize access to services and/or features of a networked peripheral device by a non-enterprise user using a centralized mechanism, which provides both enterprise user authentication and non-enterprise user authentication and access privilege information to manage device/resource usage, at the enterprise level.

[0007] The present invention addresses the above by providing a centralized level of access management to networked peripheral devices for both walk-up and remote users in order to prevent unauthorized use by a non-enterprise user of services and/or features that are available on a device based on authentication of the user at the device.

[0008] In one aspect, the invention concerns creating a context-sensitive user interface for a networked peripheral device. The user supplies authentication information to a device via any input means including, but not limited to, a keypad, a smart card, or any other input method that is supported by the device. The device communicates the authentication information provided by the user to an authentication server, which provides authentication and authorization services for both walk-up and remote for users of the device, and where networked peripheral device access policy information for users is stored. The server authenticates a user and performs a first-level check to determine the level of access of a user based on access policy information for the user. Based on the outcome of the authentication and authorization performed at the server, access management information is in turn transmitted to the device, which performs its own authorization check to determine the level of access of a user based on the received access management information.

[0009] The access policy information may concern access to a service offered by the networked peripheral device itself or to certain features of services available on the device. Upon authentication of the user by the authentication server and upon a successful first-level user access check, the information (e.g., privilege information or other access policy information) about the user's level of access to the device is passed back to the device. The device determines the user's access to services/features of the device based on the access policy information supplied by the authentication server. The device can create a customized user interface such as a customized service menu for the user that incorporates access policy information for the user. The customized service menu is then displayed on the device, allowing the user access to the determined features.

[0010] Providing a centralized location for access management information for use by a networked peripheral device in order to prevent unauthorized use of device services/features based on authentication of the user addresses the current problems associated with

Nintendo Exhibit 1003
Part 2 of 2
Page 00662

controlling access to a networked peripheral device by a walk-up user and eliminates the need for individual programming of each device or keycard in response to changes in access policy.

[0011] In addition, providing two levels of access management (i.e., two-pass access management) helps to conserve computing resources and allows the network peripheral to deny access to a user. A network-level authorization portion of the two-pass authorization aids in avoiding needless expenditures of network processing resources (e.g., a network client or server computer) in the preparation of data in response to a user's request where the user-requested service/feature is not authorized, and can provide early feedback to the user concerning whether or not the user can access the peripheral device. The device-level authorization portion of the two-pass authorization helps to ensure that prepared data received from the network by the peripheral does not contain instructions for performing an unauthorized service/feature. For example, the second-level check can avoid an attempt to spoof (e.g., by altering the prepared data) to gain unauthorized access rights to services/features of the peripheral.

[0012] This brief summary has been provided so that the nature of the invention may be understood quickly. A more complete understanding of the invention can be obtained by reference to the following detailed description of the preferred embodiment(s) thereof in connection with the attached drawings.

[0013] Figure 1 is a view showing the outward appearance of a representative hardware embodying the present invention.

[0014] Figure 2 is a detailed block diagram showing the internal architecture of the computer shown in Figure 1 in accordance with an embodiment of the present invention.

[0015] Figure 3 is a block diagram showing an overview of components for use in managing and/or controlling access to network peripheral devices according to an embodiment of the present invention.

[0016] Figure 4, which comprises Figures 4A and 4B, illustrates a flow diagram of process steps to manage and/or control access to a networked peripheral device by a walk-up user according to an embodiment of the present invention.

[0017] Figure 5 is a screen shot of the display screen showing a UI at a stage prior to walk-up user login to the device.

[0018] Figure 6 is a screen shot of the display screen showing a UI through which a walk-up user would log into the device.

[0019] Figure 7 is a screen shot of the display screen showing a customized service menu corresponding to services/features available to a walk-up user.

[0020] Figure 8 is a screen shot of the display screen showing another customized service menu corresponding to services/features available to a walk-up user.

[0021] Figure 9 is a screen shot of the display screen showing a UI through which walk-up user would log out of the device.

[0022] Figure 10 illustrates a flow diagram of process steps of a networked peripheral device for customizing a user interface based on access control information to manage and/or control access to the device by a walk-up user according to an embodiment of the present invention.

[0023] Figure 11, which comprises Figures 11A and 11B, illustrates a flow diagram of process steps to manage and/or control access to a networked peripheral device by a remote user according to an embodiment of the present invention.

[0024] Figure 12 illustrates a flow diagram of process steps of a networked peripheral device to manage and/or control access to the device by a remote user according to an embodiment of the present invention.

[0025] Figure 13, which comprises Figures 13A and 13B, illustrates a flow diagram of process steps of a server to manage and/or control access to the device by a remote user or a walk-up user according to an embodiment of the present invention.

[0026] Figure 1 is a view showing the outward appearance of a representative hardware embodying the present invention. Computing equipment 1 includes host processor 9 comprising a personal computer (hereinafter "PC") preferably having windowing operating system such as Microsoft Windows 2000®, Windows ME®, etc. operating system. Provided with the computing equipment 1 are color monitor 2 including display screen 10, keyboard 4 for entering text data and user commands, and pointing device 16. Pointing device 16 preferably comprises a mouse, for pointing, selecting and manipulating objects displayed on display screen 10.

[0027] Computing equipment 1 includes a computer-readable memory medium such as a fixed disk 17 and/or floppy disk drive 20 and/or CD-ROM drive 19. Such computer readable memory media allow computing equipment 1 to access information such as user-related data, computer executable process steps, application programs, and the like, stored on removable and non-removable memory media. In addition, computing equipment 1 is connected to server 8 through an enterprise network 7 and can acquire information and application programs from the server 8 through network 7. Enterprise network is preferably an Intranet but can also be a Local Area Network (LAN), a Wide Area Network (WAN) or the Internet, for example. The computing equipment 1 is connected to networked peripheral device 6 through the network 7. Device 6 includes one or more buttons 11, which may be programmable. As is discussed in more detail below, buttons 11 may be enabled or disabled by device 6 depending on the received access policy.

[0028] Like computing equipment, server 8 is a computer preferably having a windowing operating system. The server 8 has a storage device 41, which is prefera-

Nintendo Exhibit 1003
Part 2 of 2
Page 00663

bly a large fixed disk for storing files. While device 41 is shown to be external to server 8, it need not be. Other devices on the network 7 can therefore use the server 8 as a file server and for storing applications such as an access controller configured to control user access to services/features available on the device 6 and an authenticator configured to authenticate a user and for storing user and device configuration information on a directory service, which are described in more detail with reference to Figure 3, and which directory service contains such information as user account information and access policy information. The directory service is preferably a Microsoft Active Directory, which is a component of the Windows 2000® that provides directory services to a Windows environment. In addition to providing for central management and sharing of information on network resources and users, Microsoft Active Directory® acts as the central authority for network security that will be discussed below with reference to Figure 4. It should be apparent that while the storage of authentication and access control information is discussed with reference to a directory service, any means of storing this information can be used with the present invention, including but not limited to a domain controller.

[0029] The interface between the directory service, which contains authentication and access policy information, and other components is provided by the authenticator and/or by the authenticator, components of server 8, which are responsible for authenticating users and providing access management information stored on the directory service. The authenticator and/or access controller are preferably located on server 8 with the directory service, but can also be implemented on a remote system, or server.

[0030] Figure 2 is a detailed block diagram showing the internal architecture of computing equipment 1. As shown in Figure 2, computing equipment 1 includes central processing unit ("CPU") 20 that interfaces with computer bus 25. Also interfacing with computer bus 25 are fixed disk 3, network interface 21 for accessing network 7, random access memory ("RAM") 30 for use as main memory, read only memory ("ROM") 29, floppy disk interface 28, CD-ROM interface 24, display interface 26 to monitor 10, keyboard interface 22 to keyboard 4, mouse interface 27 to pointing device 16, and peripheral device interface 23 to a stand alone, non-networked device 6.

[0031] Main memory 30 interfaces with computer bus 25 so as to provide RAM storage to CPU 20 during execution of software programs such as the operating system, application programs, and device drivers. More specifically, CPU 20 loads computer-executable process steps from disk 3 or other memory media into a region of main memory 30, and thereafter executes the stored process steps from main memory 30 in order to execute software programs. Data can be stored in main memory 29, where the data can be accessed by CPU 20 during execution. As also shown in Figure 2, fixed disk 3 contains a windowing operating system 51, application programs 52 such as application word processing, spreadsheet, database, imaging, graphics, etc. applications, and device drivers 53 such as networked peripheral device driver 54.

[0032] Figure 3 is a block diagram showing an overview of components for use in managing and/or controlling access to network peripheral devices according to an embodiment of the present invention. Briefly, server 8 includes a host processor (not shown) that has a windowing operating system. The server 8 uses storage device 41, which is a preferably a large fixed disk for storing numerous files, to store directory service 47. Directory service 47 contains user access policy information and other information such as user authorization information. Access policy information refers to access control information (e.g., rules) that has been defined at an enterprise level concerning user access to services/features available on the networked peripheral device 6. For example, a user may have access to a copying but not a printing service of a multifunction networked peripheral device 6 that offers faxing, printing, copying and scanning services. Likewise, a user may only be allowed access to a black-and-white printing feature, but not a color printing feature, of a printing service available on the device 6.

[0033] The device 6 includes an access controller 66, which allows the user to access the device consistent with the determined level of access. Access controller is preferably an embedded computer system with an internal architecture similar to that shown in Figure 2 including some or all of the interfaces shown. The access controller 66 controls user access to services/features available on the device 6 based on the access policy information provided to it by server 8. Access controller 66 may enforce the access policy associated with a user through creation of a user interface that is customized according to the user's level of access. Alternatively, access controller 66 may disable/enable buttons (e.g., button 11) on or displayed by, device 6. Of course, a combination of a user interface and buttons disabling/enablings may also be used. Device 6 further comprises the components needed to perform the services/features of the device 6. In a case of a multifunction device, for example, device 6 further comprising scanning, printing, faxing and copying components.

[0034] According to Figure 3, the server 8 is connected to the computing equipment 1 and to networked peripheral device 6 through an enterprise network 7. The networked peripheral device 6 is preferably a multifunction device that offers faxing, copying, printing and scanning services but may be any type of networked peripheral device. Each of services offered by networked peripheral devices may include associated features. For example, printing may be available in color and black-and-white; scanning may be available in color, black-and-white and be available at various resolution levels.

Nintendo Exhibit 1003
Part 2 of 2
Page 00664

The following is an example of a structure of an enterprise access policy for use with a multi-functional networked peripheral device, which includes access/privilege information at both the service and feature levels:

Services	Features	Policy
Print	B/W	Y/N
	Color	Y/N
	Daily Quota (Pages)	0-2000
Scan	B/W	Y/N
	Color	Y/N
	Resolution	L, H/L, H
	Daily Quota (Pages)	0-100
Fax	Local	Y/N
	Long Distance	Y/N
	Resolution	H, H/L, L
	Daily Quota (Pages)	0-200
Copy	B/W	Y/N
	Color	Y/N
	Resolution	H, H/L, L
	Daily Quota (Pages)	0-1500

[0035] In the above example, the print service includes black-and-white (i.e., B/W), color and daily quota features. The information under the policy column identifies whether or not the feature is available and/or a number from zero to two thousand that represents a daily quota (e.g., a number of pages) the number of pages the user is allowed. For example, a user may be limited to printing 200 pages in a single day.

[0036] In addition to features that are similar to those of the print service, the scan service is available at high (H), medium (H/L), and low (L) resolutions as indicated under the policy column (i.e., H, H/L, L). The daily quota feature for the scan service is between zero and one hundred pages. The Fax service includes local, long distance, and daily quota features. The information under the policy column identifies whether or not the feature is available and/or a number from zero to two hundred that represents the number of pages the user can fax. In addition to the features that are similar to those of the print service, the copy service is available at various H, H/L and L resolutions that are indicated under the policy column. The information under the policy column identifies whether or not the feature is available and/or a number from zero to two hundred that represents the number of pages the user can copy.

[0037] After the authenticator 48 has authenticated the user based on information stored in the directory service 47, access controller 49 determines whether the user is authorized to access the device services/features available on the device 6 based on the access policy information stored in directory service 47. While authenticator 48 and access controller 49 are shown as separate components in Fig. 3, it should be apparent that they may be the same component or may both be part of another component. In a case that authentication by authenticator 48 and access control determination by access controller 49 are successful on server 8, access policy information is transmitted to device 6. If either authentication by authenticator 48 fails or access controller 49 determines that there is a need to deny access based on the access policy associated with a user, a message indicating that authentication failed or access is denied for the particular user and/or requested services/features is sent to device 6.

[0038] Once the access policy information is received by device 6, the access controller 66 determines the user's level of access to services/features available on the device 6 based on the received access policy information. The access controller 66 preferably enforces the enterprise access policy for the user by creating a customized user experience (e.g., customized UI, customized service menu) for the user. The customized service menu is then displayed on the device 6. Input/Output (I/O) unit 76 on the networked peripheral device 6 provides the customized service menu. I/O unit 76 may be an external unit that is attached to the device 6 but may also be built into the device 6, and may provide a display unit as well as input mechanism (e.g., keyboard and/or media reader).

[0039] The customized service menu allows the user to use the determined services and/or features available on the networked peripheral device 6 in accordance with enterprise access policy information for the user. The customized service menu is preferably displayed on a touch-screen that allows the user to activate the keys by touching virtual keys that are displayed on the screen on which the menu is displayed. In such a case, the user may gain access to the device 6 by activating the virtual keys that are displayed on the I/O unit 76 corresponding to available services/features on the device 6. However, keys can also be activated through other means such as use of a pointing device 16, where I/O unit 76 comprises computing equipment 1.

[0040] Generally, the I/O unit that is used at the device 6 can be non-integrated where the input and output functions are performed by separate units. For example, I/O unit 76 may comprise a separate keypad. I/O unit can also be integrated where the same unit performs both input and output functions. For example, I/O unit may be a touch screen that displays output including virtual keys that are activated in response to the user's touch.

[0041] A user of the networked peripheral device 6 can be a walk-up user or a remote user. A walk-up user

Nintendo Exhibit 1003
Part 2 of 2
Page 00665

is defined as one who gains access to the device 6 locally at the device. A remote user is a non-walk-up user. In the case of the walk-up user, as discussed in more detail with respect to Figure 4, the authentication information received by device 6 is transmitted from the device 6 to the authenticator 48 and device 6 receives access information policy from the authenticator 48. In the case of the remote user who may have already logged on to the network, device driver 54 of Figure 2 requests access policy information that corresponds to the user and device 6 and provides authenticator 48 with user login and device information.

[0042] The authenticator 48 transmits access policy information to device driver 54. Authenticator 48 notifies device 6 of access policy associated with the authenticated user preferably along with the job, which was submitted by the user via device driver 54. If authenticator 48 is unable to authenticate a user based on the authentication information sent by device 6, it may send an "authentication failed" message or a "no services/features available," message, or both. In a case that the user is authenticated, the authenticator 48 forwards the access policy information to access controller 49, which determines based on the job request and the access policy information whether access should be granted. If so, the access policy information is transmitted along with the job request to the device 6. The device 6 then processes the job request to the extent it conforms to the access policy information. If access controller 49 determines that access should be denied, it may send an "access denied", or a "requested services/features unavailable", message, or both.

[0043] Figure 4 illustrates a flow diagram of process steps to manage and/or control access to a networked peripheral device by a walk-up user according to the present invention. Before allowing the user to access the device 6, the request must be vouched for by a trusted application such as the authenticator 48, which is stored on the server 8. All authentication information is kept in a directory service 47 that exists on the server 8. A user initiates a job by providing authentication information to the networked peripheral device 6. The user can use any of the services that are available on the networked peripheral device 6, for which the user is authorized, and any feature corresponding to any of those services such as black-and-white or color features of a printing service, for which the user is authorized.

[0044] Since a walk-up user accesses the device directly, in step S301 a walk-up user provides authentication information to the device 6. Preferably, a single, universal sign-on functionality is in effect according to which the authentication information is the user's username and password. In any case, the same authentication information may be used to authenticate a user for other purposes (e.g., access to server 8, or files stored thereon). Advantageously, a universal sign-on avoids entry of separate, unique user names/passwords for login at the device 6 and for any other purposes for which authentication is a prerequisite.

[0045] Step S302 causes the device 6 to communicate the authentication information provided by the user to the authenticator 48, which in turn determines if the user is an authorized user. The authenticator 48 accomplished this in step S303 by comparing or testing the authentication information provided by the user to user authentication information (e.g., user name and/or password) stored in directory service 47.

[0046] Step S304 determines whether the user has been successful or unsuccessful in obtaining authentication from the authenticator 48. If the user is unsuccessful in obtaining authentication for the job request, the authenticator 48 preferably communicates this failure through an "authentication failed" message that is transmitted to the device 6 per step S305. If the user is successful in obtaining authentication, then in step S306 the authenticator 48 retrieves access policy information from directory services 47. The access policy information retrieved for a walk-up user may be the same as or different from the user's enterprise access policy information, which may also be stored directory services 47.

[0047] In step S307, access controller 49 examines the walk-up user's level of access to services/features using the retrieved access policy information. In step S308, a determination is made whether to allow the user to access device 6, and in particular the services/features needed to perform the job requested by the user. If access is not to be allowed, processing continues in step S309 to transmit an "access denied" message to device 6. If server 8 determines that the user is allowed to have access, processing continues in step S310 to transmit the access policy to device 6.

[0048] In other words, the server 8 performs an authorization check in a first pass, and if the check is successful (i.e., the server allows the user access), the server 8 forwards the access policy information to device 6 so that it can perform the second, and final, check to determine whether the user is able to access features/services of device 6.

[0049] The communication between the device 6 and the server 8 is preferably conducted via a secure communication channel that uses a secure transport protocol such as secure socket layer (SSL), transport layer security (TLS) to minimize chances of unauthorized access to the device by hackers. In addition, the access control information is preferably stored in an encrypted form using a cryptographic algorithm. A cryptographic hashing and signing operation can be used to detect unauthorized modification, or tampering, to the access information. Encrypted access information is decrypted when received by device 6. If a cryptographic hashing and signing operation is used, device 6 can verify the hash in an effort to determine whether or not the received access control information has been tampered with. In addition, device 6 may encrypt authentication information and/or use a cryptographic hashing and signing operation for transmitting the authentication in-

Nintendo Exhibit 1003
Part 2 of 2
Page 00666

formation to server 8.

[0050] Although the encryption is preferably performed such that the information stored at the directory service 47 is encrypted, it can alternatively be stored in non-encrypted form at the directory service 47. Accordingly, access policy information is stored in a non-encrypted form at the directory service 47 and is encrypted at the authenticator 48. Propriety protocols for communications between the server 8 and the device 6, and use of propriety secure transport mechanisms in connection with standard protocols is also contemplated for use with the present invention.

[0051] Once the device 6 obtains the user access policy information from the authentication server 8, the access controller 66 proceeds in step S311 to determine user access to services and/or features offered by the device 6 based on the access policy information received from the authenticator 48. The following is an example of user access policy information for a multi-functional networked peripheral device:

Services	Features	Policy
Print	B/W	Y
	Color	Y
	Daily Quota	150
Scan	B/W	Y
	Color	N
	Resolution	L
	Daily Quota	50
Fax	Local	Y
	Long Distance	N
	Resolution	H/L, L
	Daily Quota	25
Copy	B/W	Y
	Color	N
	Resolution	H, H/L, L
	Daily Quota	500

[0052] In the above example, user is allowed to use both black-and-white and color features of the print service up to a daily maximum of 150 pages. The user can use the scan service for scanning up to 50 pages of black-and-white pages per day at low resolution, fax up to 25 pages a day as long as the faxes are not transmitted over long distance telephone line, and are transmitted at medium or low resolution, and can use the copying service to make up to 500 black-and-white copies a day at all resolutions.

[0053] In step S312 the user is allowed access to services/features of the device 6 consistent with the determined level of access. This may be implemented by the access controller 66 through creation of a user interface, which includes selections based on the access policy information obtained for the particular user. Accordingly, the device 6 creates a customized user interface such as a customized service menu for the user that incorporates the access policy for the user. The customized service menu is then displayed on the device 6 with services/features appearing as virtual keys on the I/O unit 76.

[0054] Figures 5-9 are views of the display screen 10 showing changes in a UI at various stages of a walk-up user's interaction with the device 6.

[0055] Figure 5 is a screen shot of the display screen showing a UI before user seeks access to the device 6. As indicated, the device is locked and requires the walk-up user to log in.

[0056] Figure 6 is a screen shot of the display screen showing a UI through which user would log in to the device 6. In the example of Figure 6, the user logs in by providing a username and password, which are transmitted in a secure manner to server 8.

[0057] Figure 7 is a screen shot of the display screen showing a customized service menu corresponding to services/features available to a walk-up user. Buttons 800 at the top of the screen are preferably virtual (i.e., non-physical) buttons. In this case, they represent the services of device 6 that are available to the user (i.e., scanftp, logout, scopy). The portion below buttons 800 is area 701, which is a display of a job corresponding to the "scopy" service. The scopy service allows the user to scan and copy using device 6. The virtual buttons 702 on the right hand side of the screen correspond to available features (copy, number of pages, paper selection). That is, the user can copy, set the number of pages and select paper using buttons 702. Scroll buttons 703 allow the user to scroll through the job listings displayed in area 701.

[0058] Figure 8 is a screen shot of the display screen showing another example of a customized service menu corresponding to services/features available to a walk-up user. In the example of Figure 8, area 701 corresponds to a "Hold&Print" service of the device 6 and provides a list of "Hold&Print" jobs. The screen includes buttons 800, which, in this case, correspond to the "Hold & Print", "Scan FTP", and "Logout" services of device 6. The "Hold&Print" service allows the user to store-up print jobs and to initiate printing of a stored job at device 6. The virtual buttons 802 on the right hand side of the screen correspond to available features (update list, print job and delete job). That is, the user can update/refresh list, print a job, or delete a job using buttons 802. Scroll buttons 703 allow the user to scroll through the job listing displayed in area 701.

[0059] Figure 9 is a screen shot of the display screen showing a UI though which walk-up user would log out of the device 6. Similar to Figure 8, the screen includes buttons 800, which, in this case, correspond to the "Scan FTP", "scopy", and "Logout" services of device 6.

[0060] In a case that the UI includes "unauthorized"

Nintendo Exhibit 1003
Part 2 of 2
Page 00667

selections, these selections may be disables such that the keys corresponding to unauthorized services/features are grayed out. The user then proceeds to use the device 6 in accordance with the determined level of access by activating non-grayed out keys that are displayed on the I/O unit 76 preferably through a touch screen.

[0061] Figure 10 illustrates a flow diagram of process steps of a networked peripheral device for customizing a user interface based on access control information to manage and/or control access to the device by a walk-up user according to the present invention. In step S1001 the device 6 inquires into whether user has provided authentication and any job request information to the device 6 (e.g., authentication information received via the screen depicted in Figure 6). A job is not initiated by the device until such information is provided by the user. After such information is provided, the user can request a job to be performed using services/features of device 6 (e.g., via the screens depicted in Figures 7 to 9). In order for device 6 to perform the user's job request, server 8 must successfully authenticate the user and successfully perform server-side authorization, and device 6 performs its authorization pass, which must also be successful.

[0062] Step S1002 causes the authentication and any job request information entered by user in step S1001 to be forwarded to authenticator 48. For example, once device 6 transmits the information to server 8 in step S1002, steps S303 to S310 of Figures 4A and 4B are performed at server 8. The authenticator 48, compares or tests the authentication information provided by the user against authentication information stored in the directory service 47, and access controller 49 determines whether or not to allow access, and the results are transmitted back to the device 6. With respect to authorization, access controller 49 may examine job request information if supplied to determine whether access for the particular job is to be allowed, and if no job request information is received at server 8, access controller 49 can determine whether or not the user has access to any of the services/features of device 6.

[0063] In step S1003, device 6 awaits the results from the server 8. Step S1004 inquires into the user's success in being authenticated and whether the user was given some level of access to services/features of device 6 by the authenticator 48 and access controller 49, respectively, based on the results received from server 8. For example, in a case that either authentication or authorization is unsuccessful, server 8 forwards a message indicating the error to device 6.

[0064] Should the user be unsuccessful in obtaining authentication/authorization, the error message received from server 8 is processed by device 6 in step S1005 by, for example, causing an "authorization failed" or "access denied" message to be displayed by the device 6, thus denying the user access to any service/features of the device 6. In case of successful authentication/authentication of the user by server 8, the device 6 determines the user's level of access to the services/features of the device 6 based on the received access policy information per step S1006. Step S1007 causes the device 6 to create a customized menu for the user based on the determined level of access.

[0065] A remote user may access device 6, for example, via a workstation such a computing equipment 1. However, the process in which a remote user accesses device 6, differs from that of a walk-up user. The following example concerns a remote user who seeks to print a job using device 6 from a workstation and an application that exists on the workstation:

1. The user initiates a print operation from within the application.

2. Device 6 determines the server 8 on which the authenticator 48 is running.

3. A secure pipe is created between the print driver and the authenticator 48, in which the authentication information is sent to server 8.

4. The device driver 54 on the workstation transmits a request to server 8 for access policy information. The request identifies the user and device 6. Driver 54 also provides authentication information and any job request information in conjunction with the request.

5. Authenticator 48 performs authentication access controller 49 performs authorization, and based on the outcome of authentication/authorization transmits a response (e.g., "authorization failed" or "access denied" message, or access policy information) to device 6.

6. Driver 54 forwards the received access policy information to device 6 along with the job submitted by the user.

[0066] Figure 11 illustrates a flow diagram of process steps to manage and/or control access to a networked peripheral device by a remote user according to the present invention. The user logs into the network 7 in step S1101. At step S1102 a determination is made as to whether the user has initiated a job request. Once the job request is initiated, device driver 54 requests access policy information that corresponds with the user and the device from server 8 and provides authentication and job request information to the server 8, per step S1103. Authentication information is preferably provided via a challenge and response mechanism, but can also be provided through other means such as user's username and password.

[0067] Prior to providing the requested access policy information, server 8 (e.g., authenticator 48 and access

Nintendo Exhibit 1003
Part 2 of 2
Page 00668

controller 49 of server 8) authenticates the user using the authentication information and determines whether the user has the level of access needed to use the service/features requested by the job request, per step S1104. In step S1105, server 8 determines whether either of the authentication and authorization was unsuccessful, and if so sends an appropriate message (e.g., an "authorization failed" or an "access denied" message). If both authentication and authorization were successful, processing continues with step S1108.

[0068] In step S1108, the server 8 sends access policy information back to the driver. Step S1109 causes the driver to forward the access policy information along with the job request to the device 6. In step S1110, the device 6 determines the user's level of access based on the received access policy information. In step S1111, the device 6 compares or tests the requested services/features against the user's level of access to determine whether the user's request conforms to the user's level of access.

[0069] A determination that the user's job request does not conform to the user's determined level of access, results in an "access denied" condition, per step S1112, thus denying the user's job request. Preferably, a message is sent to the user's workstation to alert the user of the "access denied" condition. Should the inquiry in step S1111 result in the determination that the user's job request does conform to the user's determined level of access, then step S1113 causes the device 6 to perform the requested job in accordance with the determined level of access.

[0070] Figure 11 provides a general overview of steps performed to control/manage access to device 6. Figure 12 illustrates the perspective of device 6. That is, Figure 12 illustrates a flow diagram of process steps for a networked peripheral device to manage and/or control access to the device by a remote user according to the present invention. In order to process a job that is remotely sent to a device 6, the device 6 needs to have both the requested services/features and the access policy information for the user. The job that is submitted by the user identifies the requested services/features. Device driver 54 provides the access policy information, which it received from server 8 as part of the job stream sent to device 6.

[0071] In step S1201, device driver 54 forwards the user's job to the device 6. The device 6, per step S1202, asks whether the user's access policy information is included along with the job request. If user's access policy information is not included along with the job request, then step S1203 causes an error message to be sent back to the driver 54, denying the user access to the requested services/features.

[0072] If the user's access policy information is included along with the job request, then step S1204 proceeds to determine the user's level of access based on the received access policy information. In step S1205, the device 6 compares or tests the requested services/features against the user's level of access to determine whether the user's request conforms to the user's level of access. A determination that the user's job request does not conform to the user's determined level of access, results in an error message, per step S1206, that is sent to driver 54, thus denying the user's job request. Should the inquiry in step S1205 result in the determination that the user's job request does conform to the user's determined level of access, then step S1207 causes the device 6 to perform the requested job.

[0073] Server 8 is configured to provide access policy information for a walk-up user and preferably for all enterprise users (i.e., walk-up and remote users) once the user has been authenticated by server 8.

[0074] Figure 13 illustrates a flow diagram of process steps of a server to manage and/or control access to the device by a remote user or walk-up user according to the present invention. In step S1301, server 8 awaits an access policy request along with authentication and job request information from the device 6 or driver 54. In step S1302 the authenticator 48, located on the server 8, retrieves authentication information from directory service 47. In step S1303, the retrieved information is compared or tested against the user authentication information received per step S1301. Step S1304 inquires into whether user has been successfully authenticated. If the user is unsuccessful in obtaining authentication then step S1305 causes an "authentication failed" message to be sent from the server 8. If the user is successfully authenticated, access, then the user's level of access to services/features is determined based on access policy information, per step S1306. In step S1308, a determination is made whether or not the user is allowed access to the services/features requested by virtue of the job request, and if the user access is insufficient, then step S1309 causes an "access denied" message to be sent from the server 8. If it is determined that the user has a sufficient level of access to the device 6, then step S1310 causes the user's access policy information to be sent from the server 8.

[0075] Server 8 may reside locally with respect to device 6, computing equipment 1, or both. In a case that the network 7 is the Internet, for example, server 8 may be remotely located with respect to device 6, computing equipment 1, or both. Even where server 8 is local, it may be preferable to use a trusted architecture in which access policy information, that is received from server 8 can be trusted.

[0076] While the invention is described above with respect to what is currently considered its preferred embodiment, it is to be understood that the invention is not limited to that described above. To the contrary, the invention is intended to cover various modifications and equivalent arrangements within the scope of the appended claims.

Nintendo Exhibit 1003
Part 2 of 2
Page 00669

Claims

1. A method for controlling access to a networked peripheral device by a walk-up user, wherein the networked peripheral device is accessible by both the walk-up user and a remote user based on centralized access management information accessible by a server, the method comprising:

   authenticating the walk-up user based on authentication information corresponding to the user;
   determining, at the server, a first level of access to the networked peripheral device available to the walk-up user based on the access management information;
   transmitting access management information for the walk-up user to the networked peripheral device a centralized location indicative of the determined first level of access;
   determining, at the networked peripheral device, a second level of access to the networked peripheral device available to the walk-up user based on the received access management information; and
   allowing the walk-up user to access to the networked peripheral device based on the first and second determined levels of access.

2. A method according to Claim 1, wherein the networked peripheral device is a multifunction peripheral device.

3. A method according to Claim 1, wherein the access management information is supplied by an authentication server once the authentication server authenticates the walk-up user based on authentication information received from the networked peripheral device.

4. A method according to Claim 1, wherein a user interface is devised by the networked peripheral device that is specific to the determined second access level.

5. A method according to Claim 1, wherein buttons on a keypad on the device are enabled and/or disabled according to the determined second access level.

6. A method according to Claim 1, wherein the access management information is supplied by an authentication server that authenticates both the walk-up user and the remote user.

7. A method according to Claim 3, wherein the authentication information is a username and/or password.

8. A method according to Claim 3, wherein the authentication information is entered by inserting a smart card at the networked peripheral device.

9. A method according to Claim 6, wherein the access management information is encrypted.

10. A method according to Claim 3, wherein the authentication information received from the networked peripheral device is encrypted.

11. A method according to Claim 1, wherein the determined first level of access indicates that the walk-up user should have no access to the networked peripheral device, and wherein access management information includes an "access denied" message.

12. A computer-readable memory medium in which computer-executable process steps are stored, the process steps for controlling access to a networked peripheral device by a walk-up user, wherein the networked peripheral device is accessible by both the walk-up user and a remote user based on centralized access management information accessible by a server, wherein the process steps comprise:

    an authenticating step of authenticating the walk-up user based on authentication information corresponding to the user;
    a first determining step of determining, at the server, a first level of access to the networked peripheral device available to the walk-up user based on the access management information;
    a transmitting step of transmitting access management information for the walk-up user to the networked peripheral device a centralized location based on the determined first level of access;
    a second determining step of determining, at the networked peripheral device, a second level of access to the networked peripheral device available to the walk-up user based on the received access management information; and
    an allowing step of allowing the walk-up user to access to the networked peripheral device based on the first and second determined levels of access.

13. A computer-executable program code stored on a computer readable medium, said computer-executable program code for controlling access to a networked peripheral device by a walk-up user, wherein the networked peripheral device is accessible by both the walk-up user and a remote user based on centralized access management information accessible by a server, said computer-executable program code comprising:

Nintendo Exhibit 1003
Part 2 of 2
Page 00670

code to authenticate the walk-up user based on authentication information corresponding to the user;
code to determine, at the server, a first level of access to the networked peripheral device available to the walk-up user based on the access management information;
code to transmit access management information for the walk-up user to the networked peripheral device a centralized location based on the determined first level of access;
code to determine, at the networked peripheral device, a second level of access to the networked peripheral device available to the walk-up user based on the received access management information; and
code to allow the walk-up user to access to the networked peripheral device based on the first and second determined levels of access determined by the network peripheral device.

14. An apparatus for controlling access to a networked peripheral device by a walk-up user, wherein the networked peripheral device is accessible by both the walk-up user and a remote user based on centralized access management information accessible by a server, said apparatus comprising means for performing the functions specified in any of Claims 1 to 11.

15. A server for use in controlling access to a networked peripheral device by a walk-up user, wherein the networked peripheral device is accessible by both the walk-up user and a remote user based on centralized access management information accessible by a server, the server comprising:

receiving a request for access policy information, the request including authentication information;
authenticating the user using the authentication information;
determining a level of access to features and/or services of the networked peripheral device available to the walk-up user based on centralized access management information; and
transmitting access management information for the user to the networked peripheral device, which comprises an access policy in a case that authentication of the user is successful and the user is authorized to use at least some portion of the features and/or services of the networked peripheral device.

16. A server according to Claim 15, further comprising:

transmitting an error message as the access management information in a case that either authentication of the user failed or the walk-up user has no level of access to features and/or services of the networked peripheral device.

17. A method of controlling access to a networked peripheral device by a walk-up user, wherein the networked peripheral device is accessible by both the walk-up user and a remote user based on centralized access management information accessible over a network, the method comprising:

transmitting authentication information over the network;
receiving, over the network, access management information for the walk-up user at the networked peripheral device from the centralized location;
examining the received access management information to determine whether it includes an access policy corresponding to the walk-up user;
determining, at the networked peripheral device, a level of access to the networked peripheral device available to the walk-up user based on received access policy in a case that an access policy is received by the networked peripheral device; and
allowing the walk-up user to access the networked peripheral device based on the determined level of access.

18. A method according to Claim 17, wherein the access policy is received by the network peripheral device in a case that the user is authenticated by a server connected to the network and a first-level authorization performed by the server based on the access management information of the walk-up user is successful.

Nintendo Exhibit 1003
Part 2 of 2
Page 00671

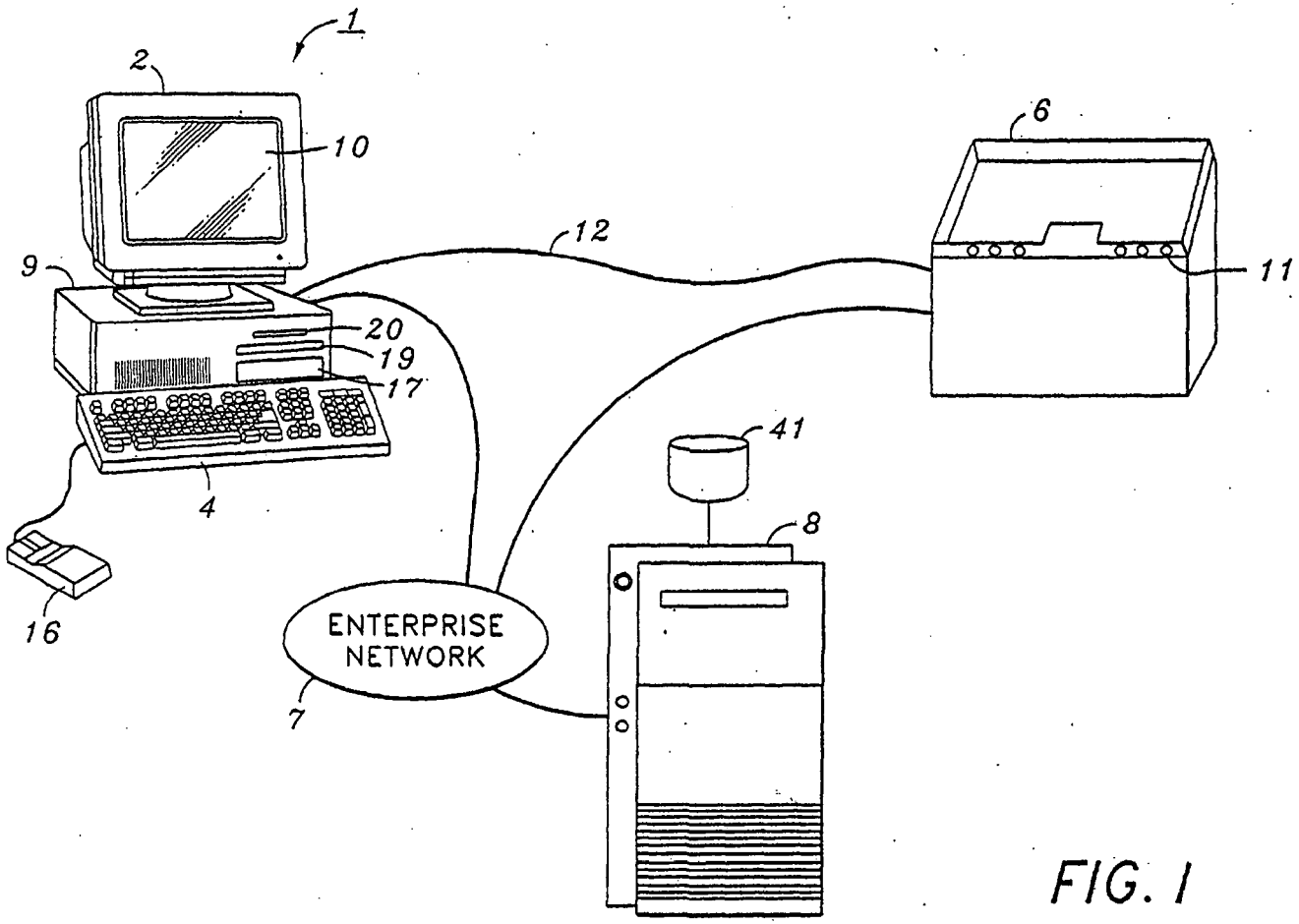


FIG. 1

Nintendo Exhibit 1003
Part 2 of 2
Page 00672

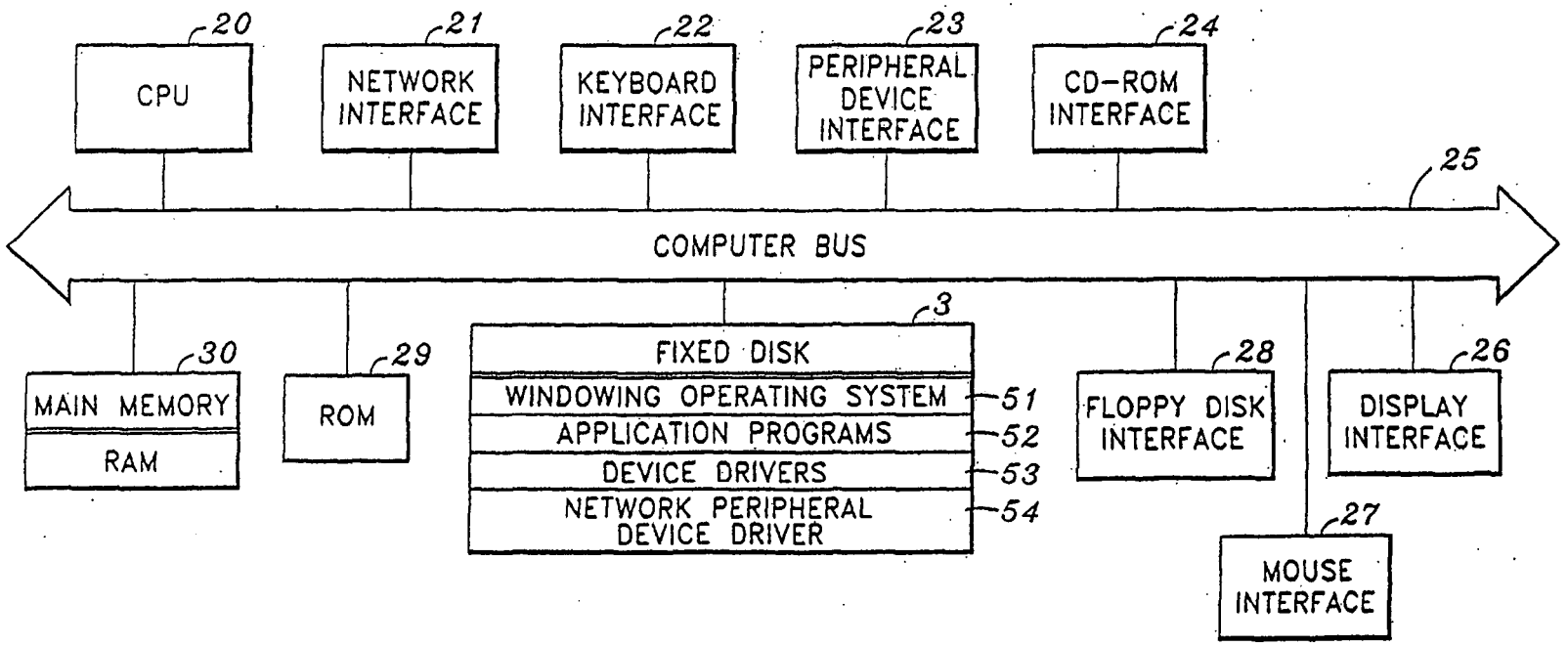


FIG. 2

EP 1 427 166 A2

Nintendo Exhibit 1003
Part 2 of 2
Page 00673

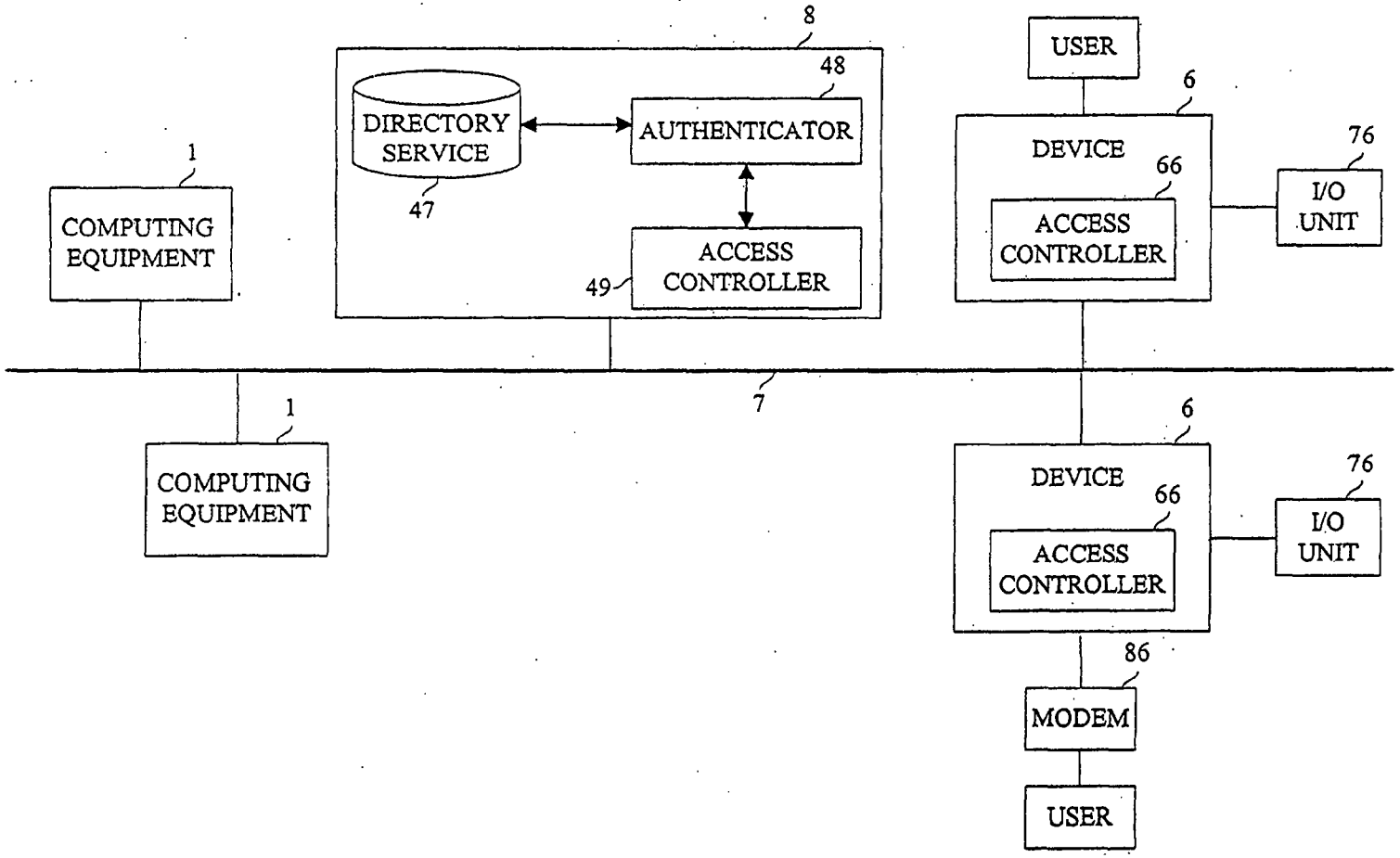
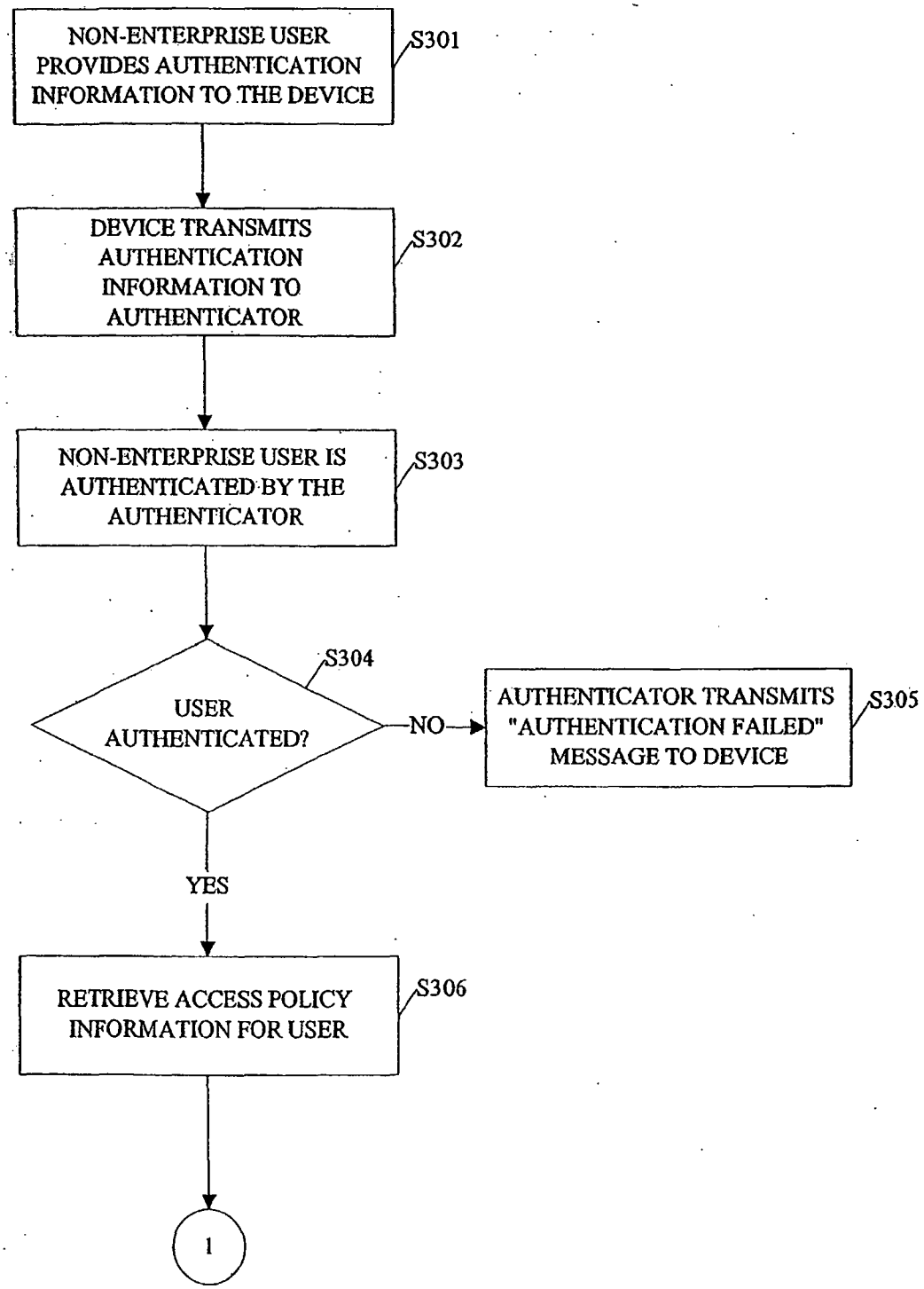


Fig. 3

EP 1 427 166 A2

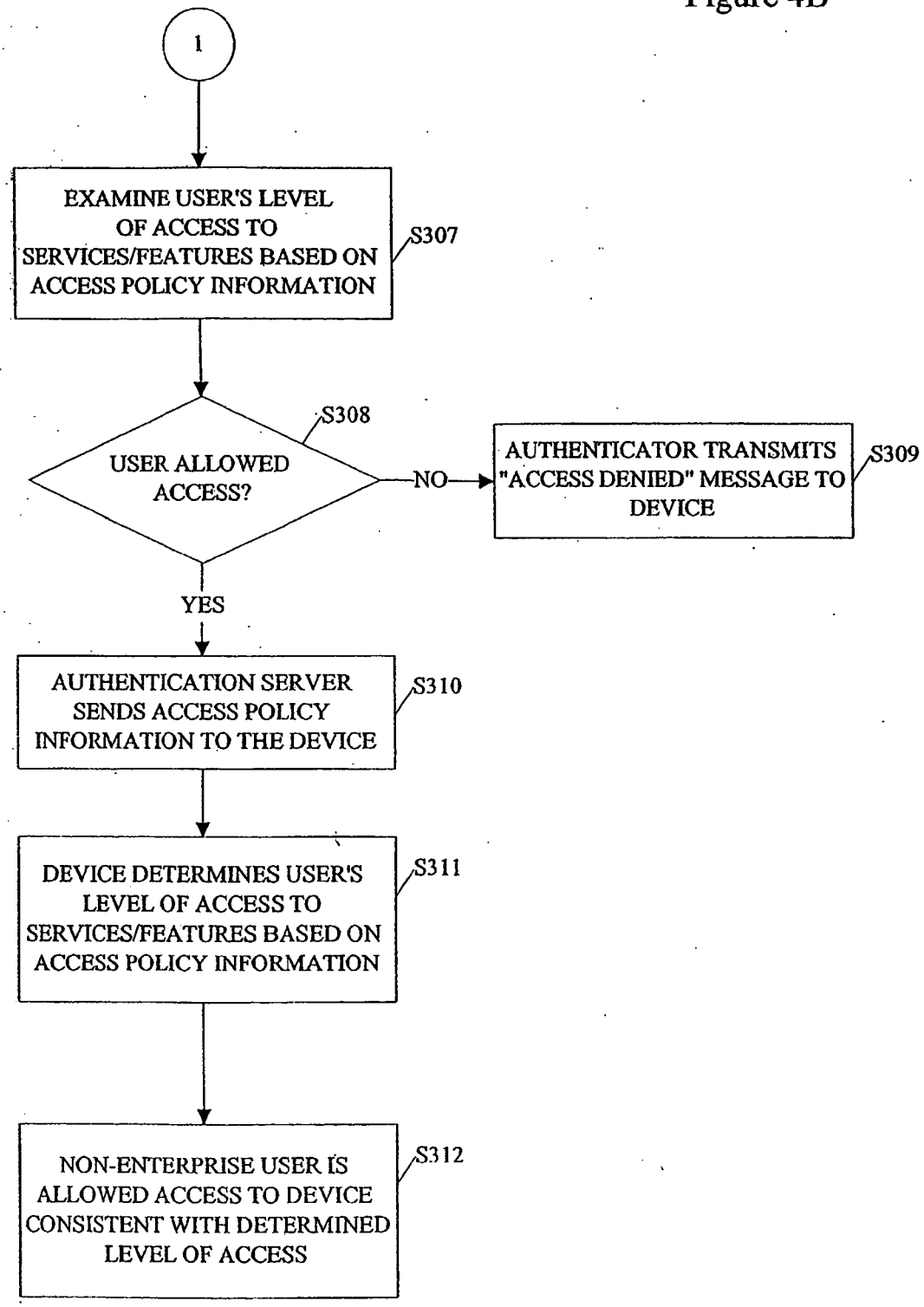
Nintendo Exhibit 1003
Part 2 of 2
Page 00674

Figure 4A



Nintendo Exhibit 1003
Part 2 of 2
Page 00675

Figure 4B



Nintendo Exhibit 1003
Part 2 of 2
Page 00676

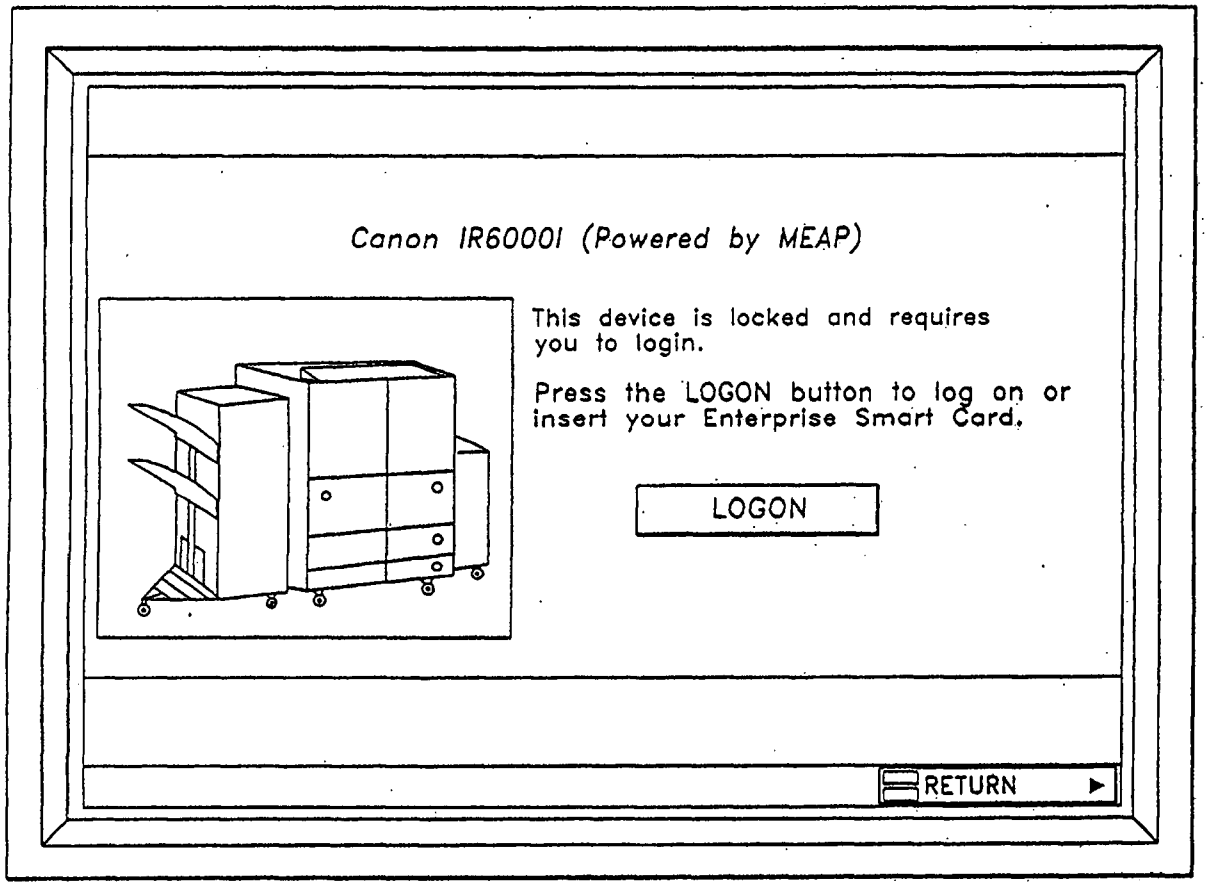


FIG. 5

Nintendo Exhibit 1003
Part 2 of 2
Page 00677

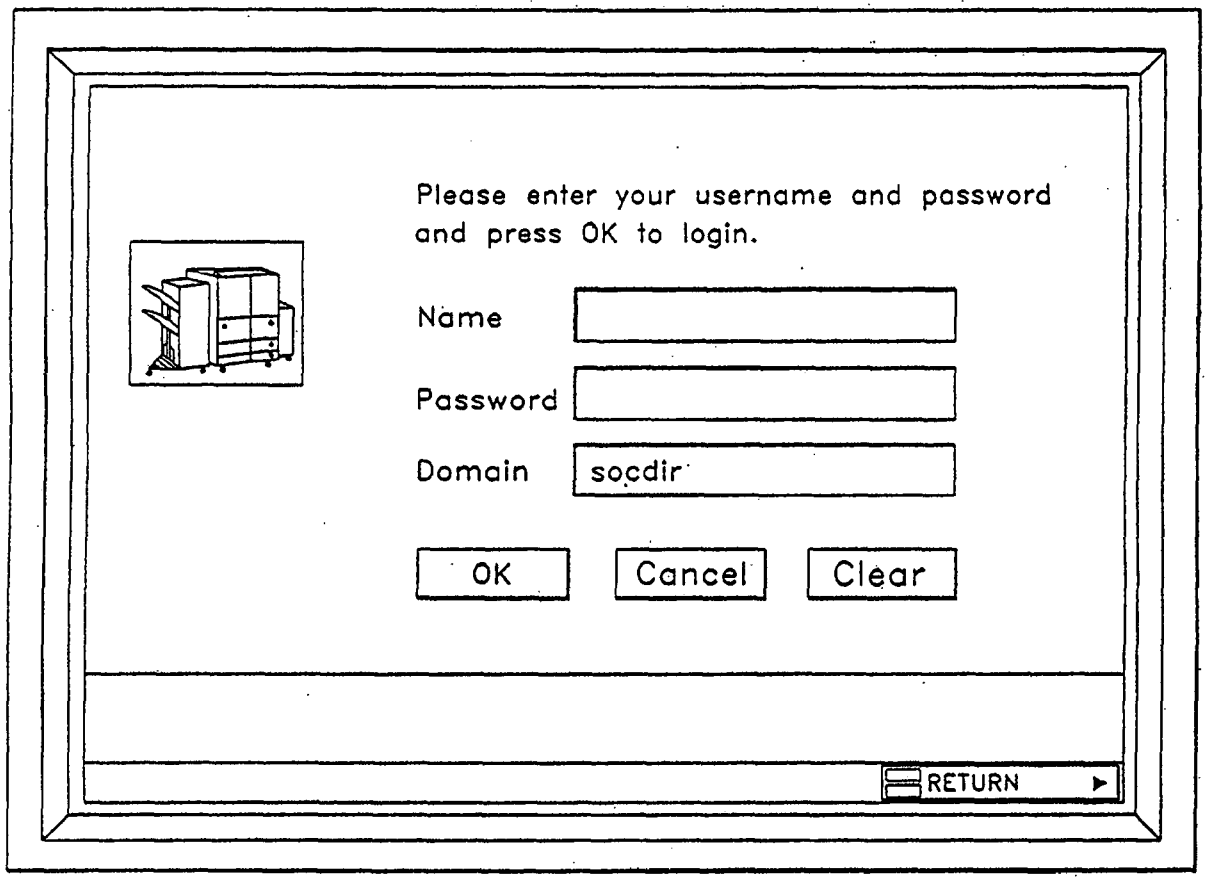


FIG. 6

Nintendo Exhibit 1003
Part 2 of 2
Page 00678

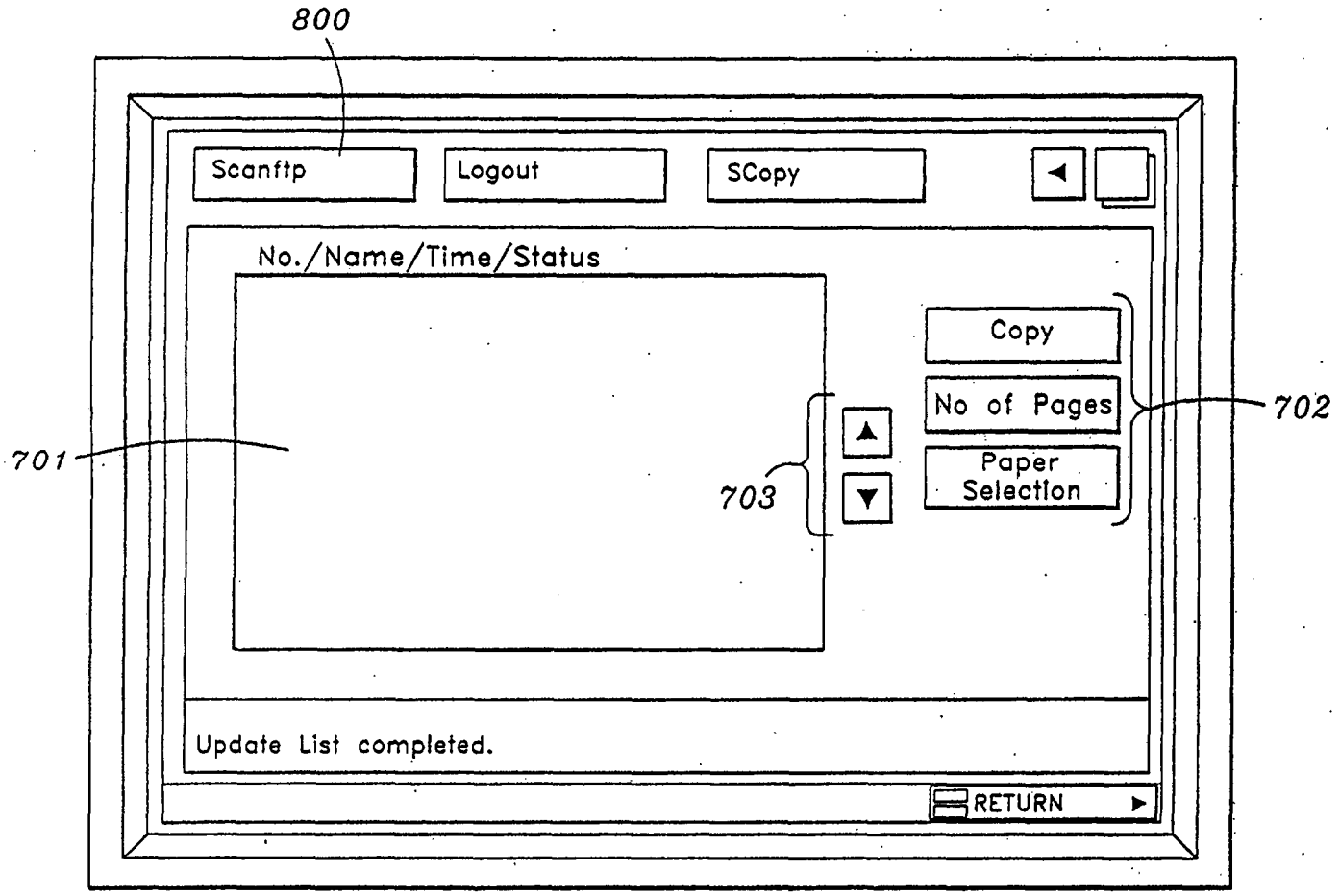


FIG. 7

Nintendo Exhibit 1003
Part 2 of 2
Page 00679

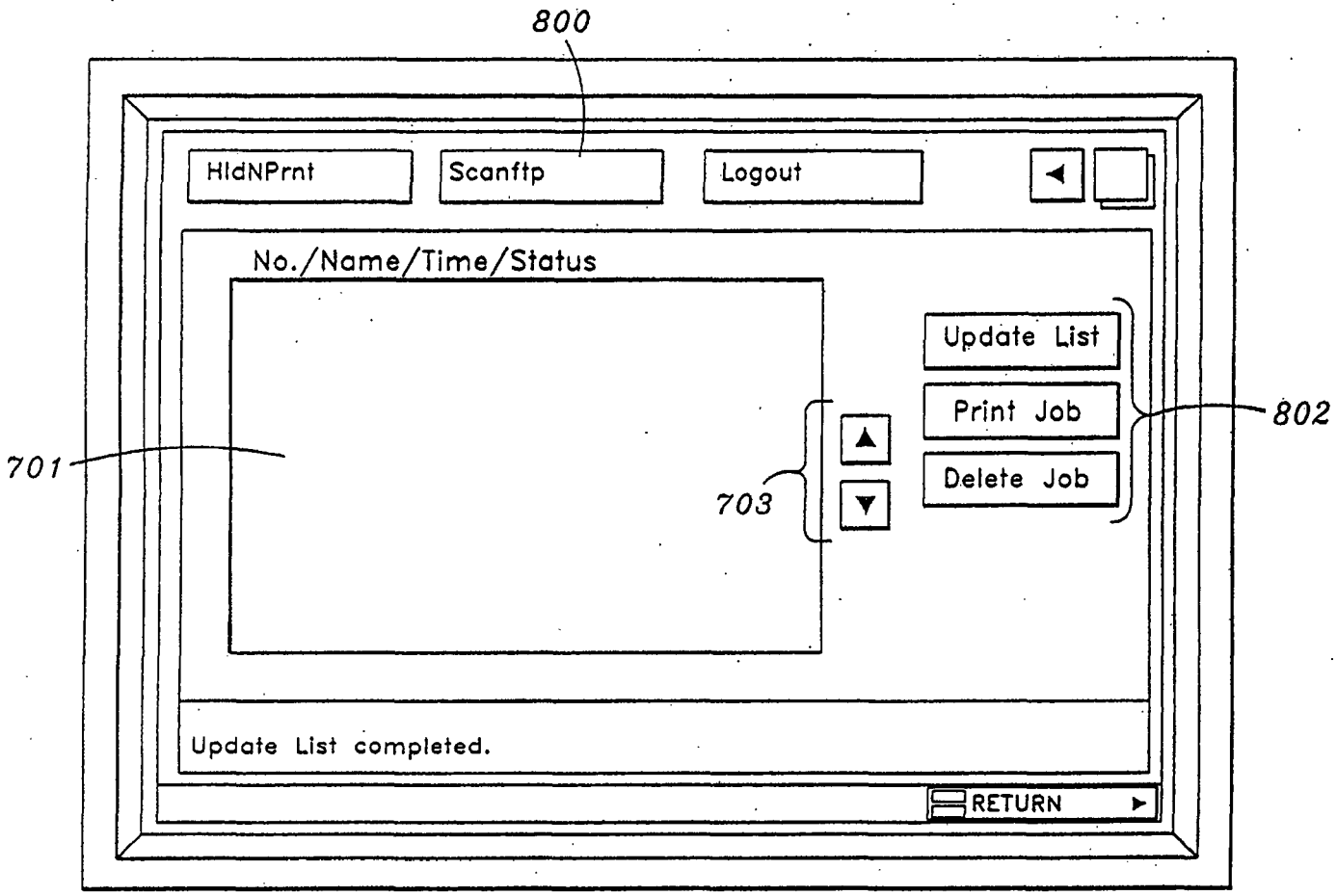


FIG. 8

Nintendo Exhibit 1003
Part 2 of 2
Page 00680

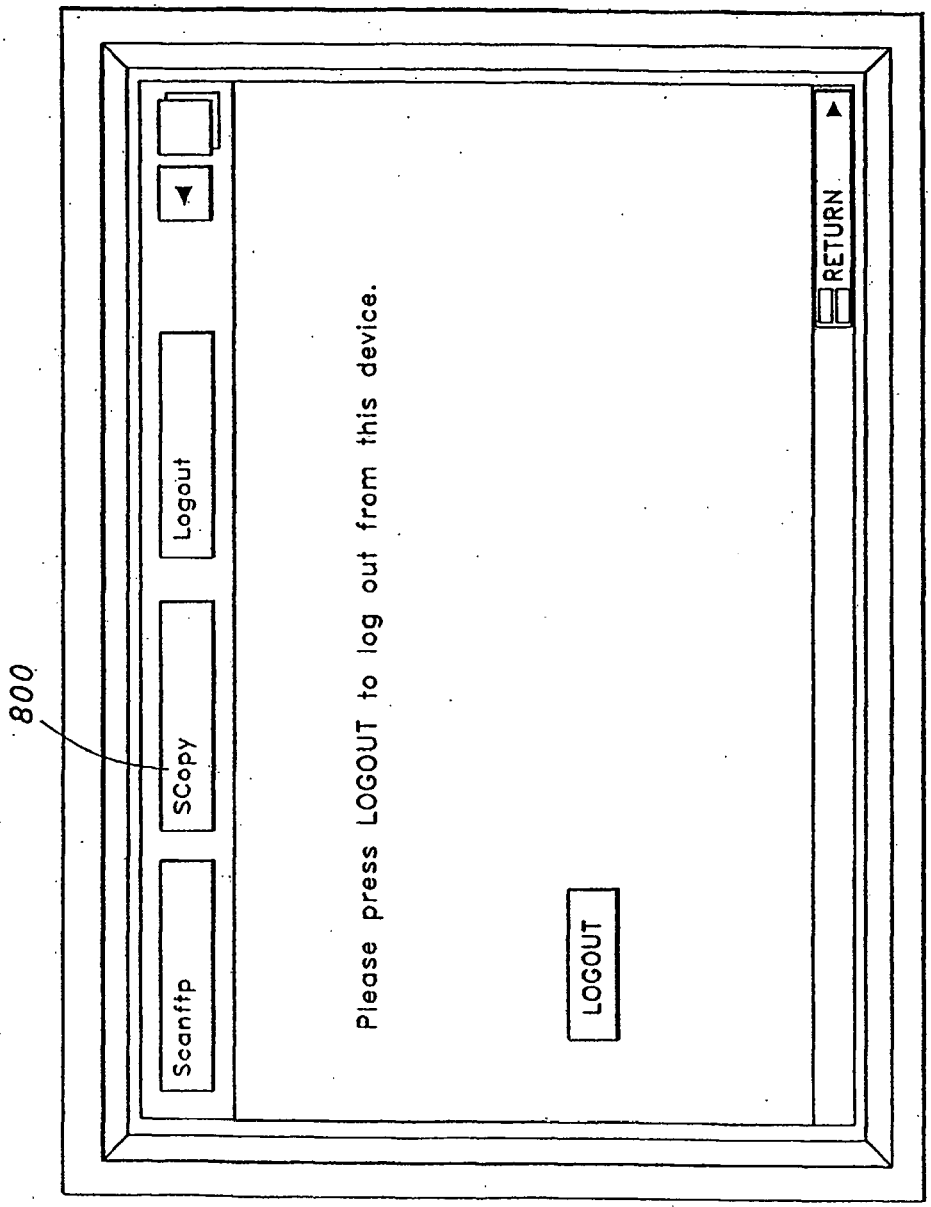
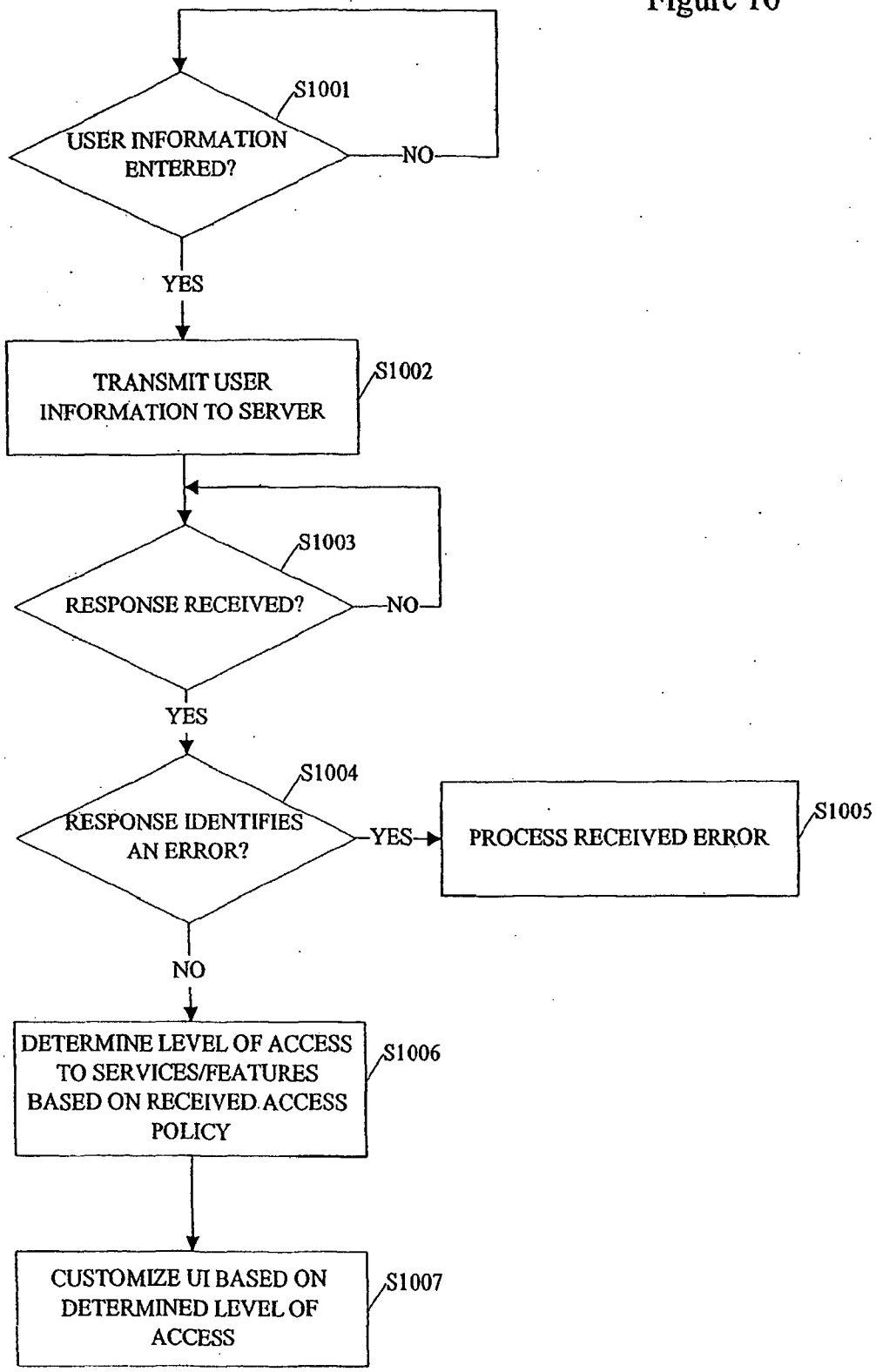


FIG. 9

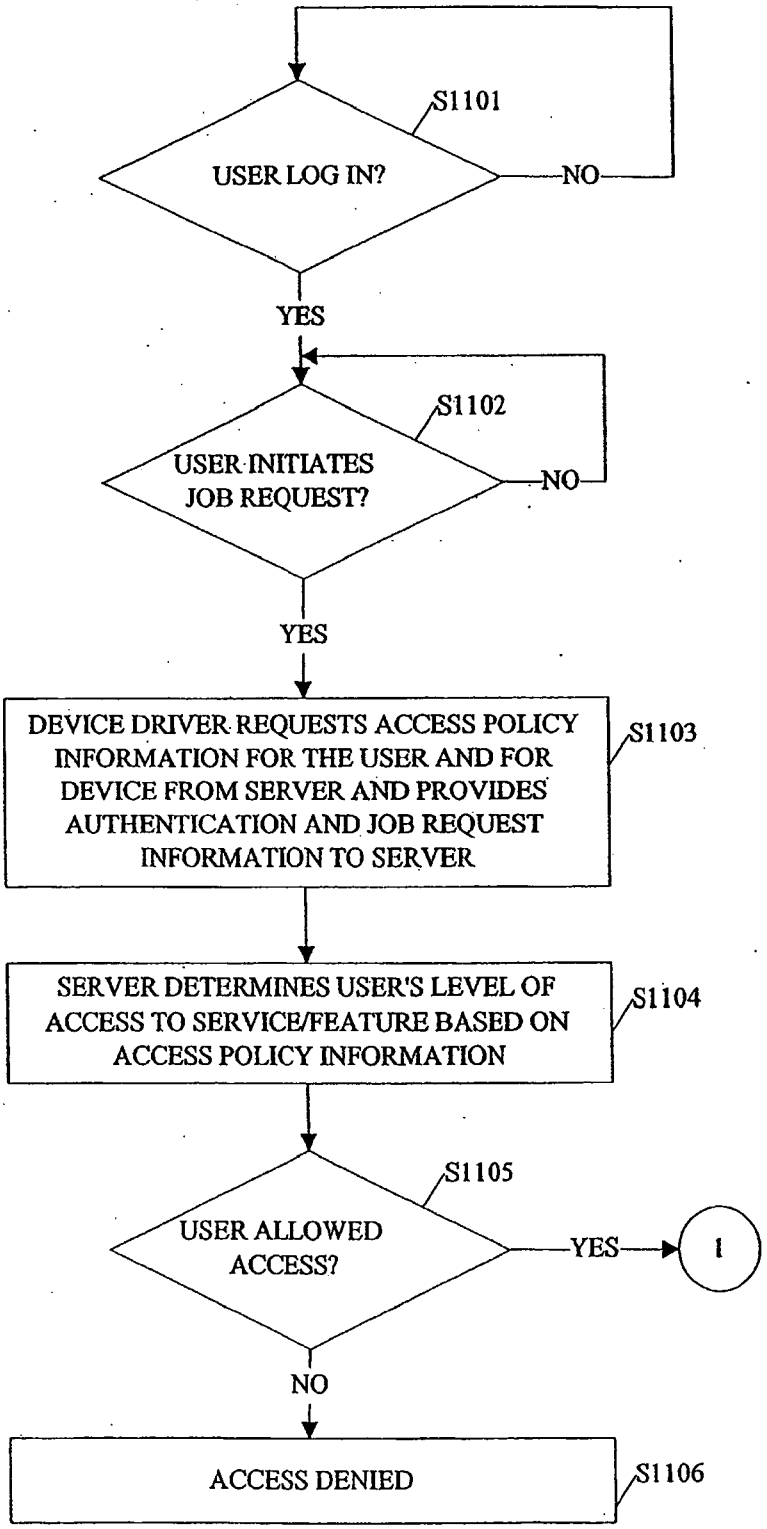
Nintendo Exhibit 1003
Part 2 of 2
Page 00681

Figure 10



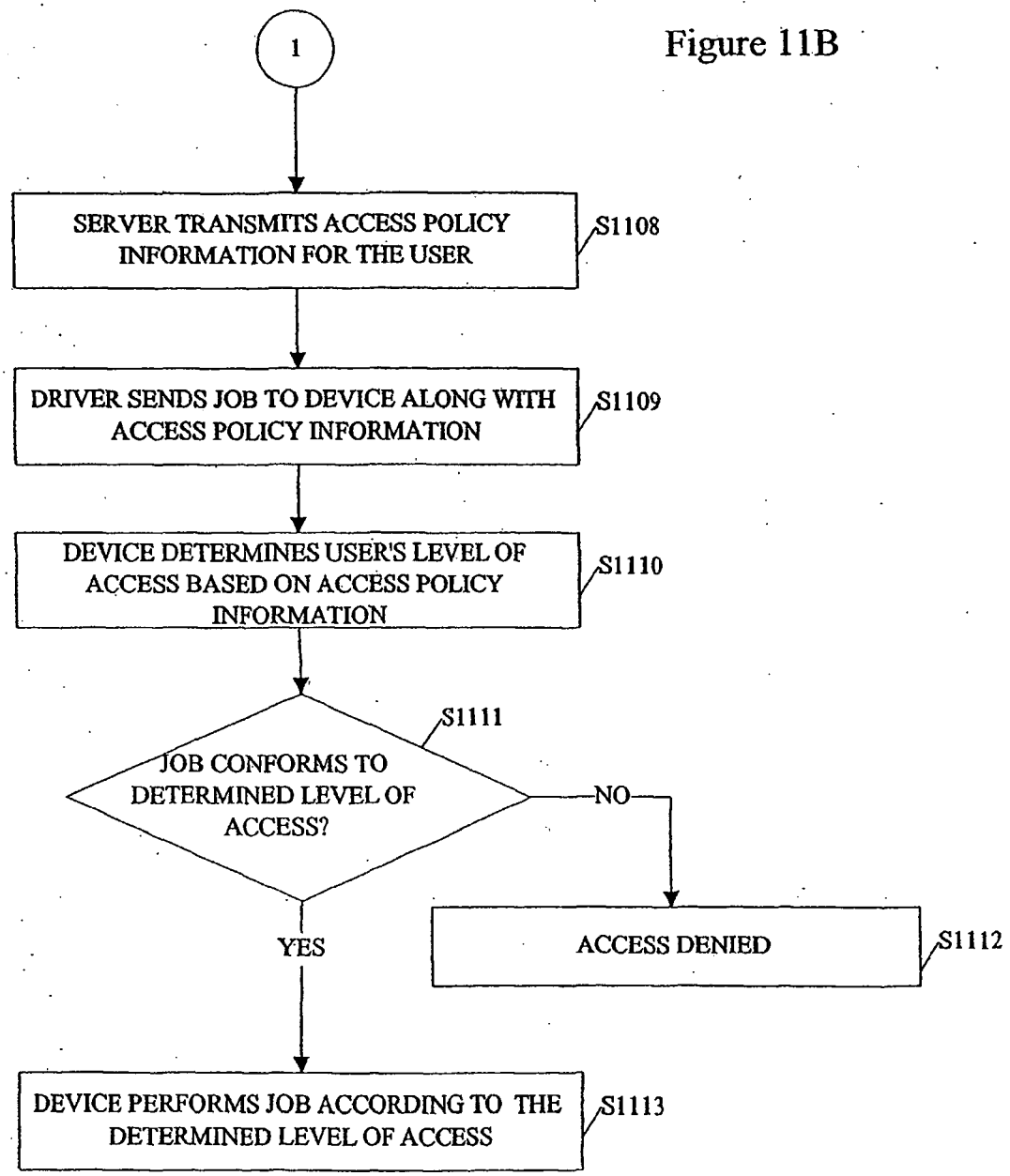
Nintendo Exhibit 1003
Part 2 of 2
Page 00682

Figure 11A



Nintendo Exhibit 1003
Part 2 of 2
Page 00683

Figure 11B



Nintendo Exhibit 1003
Part 2 of 2
Page 00684

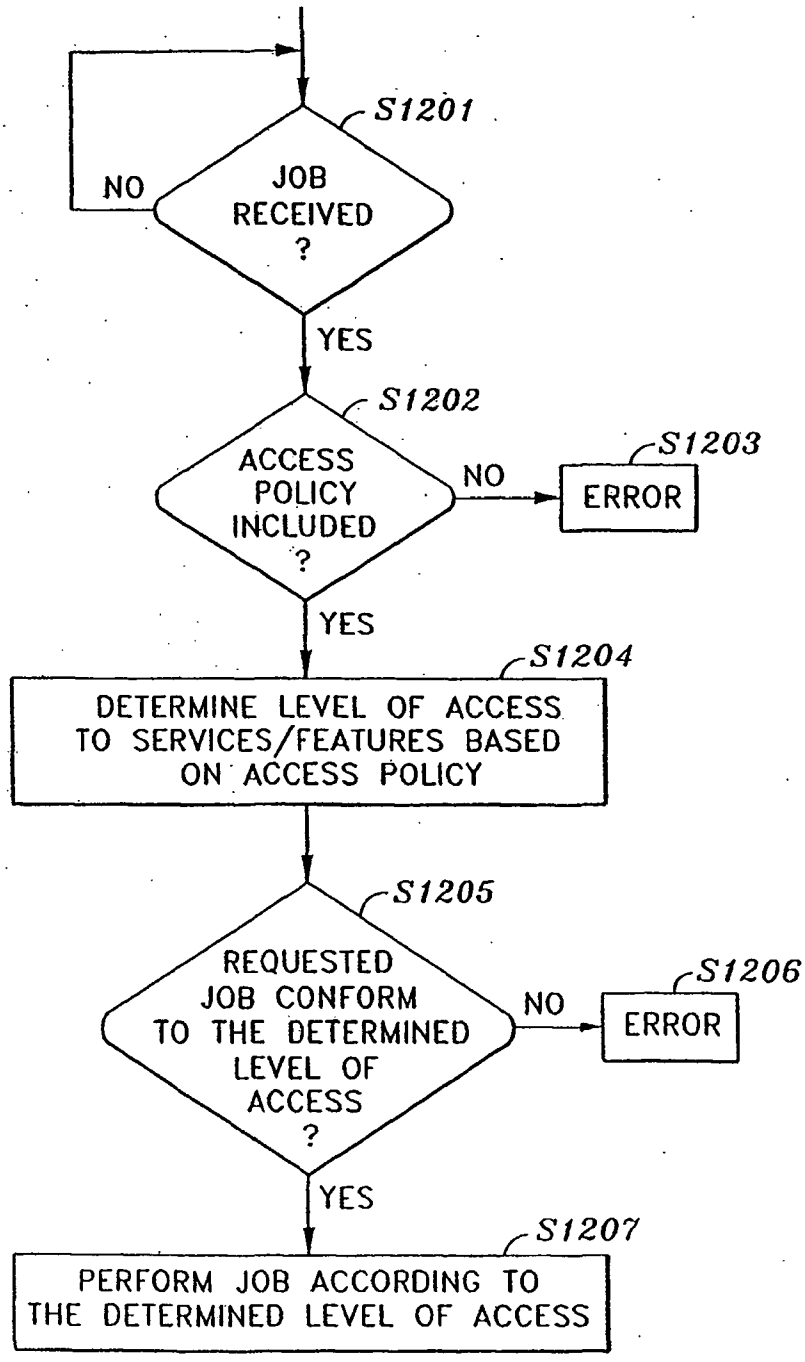


FIG. 12

Nintendo Exhibit 1003
Part 2 of 2
Page 00685

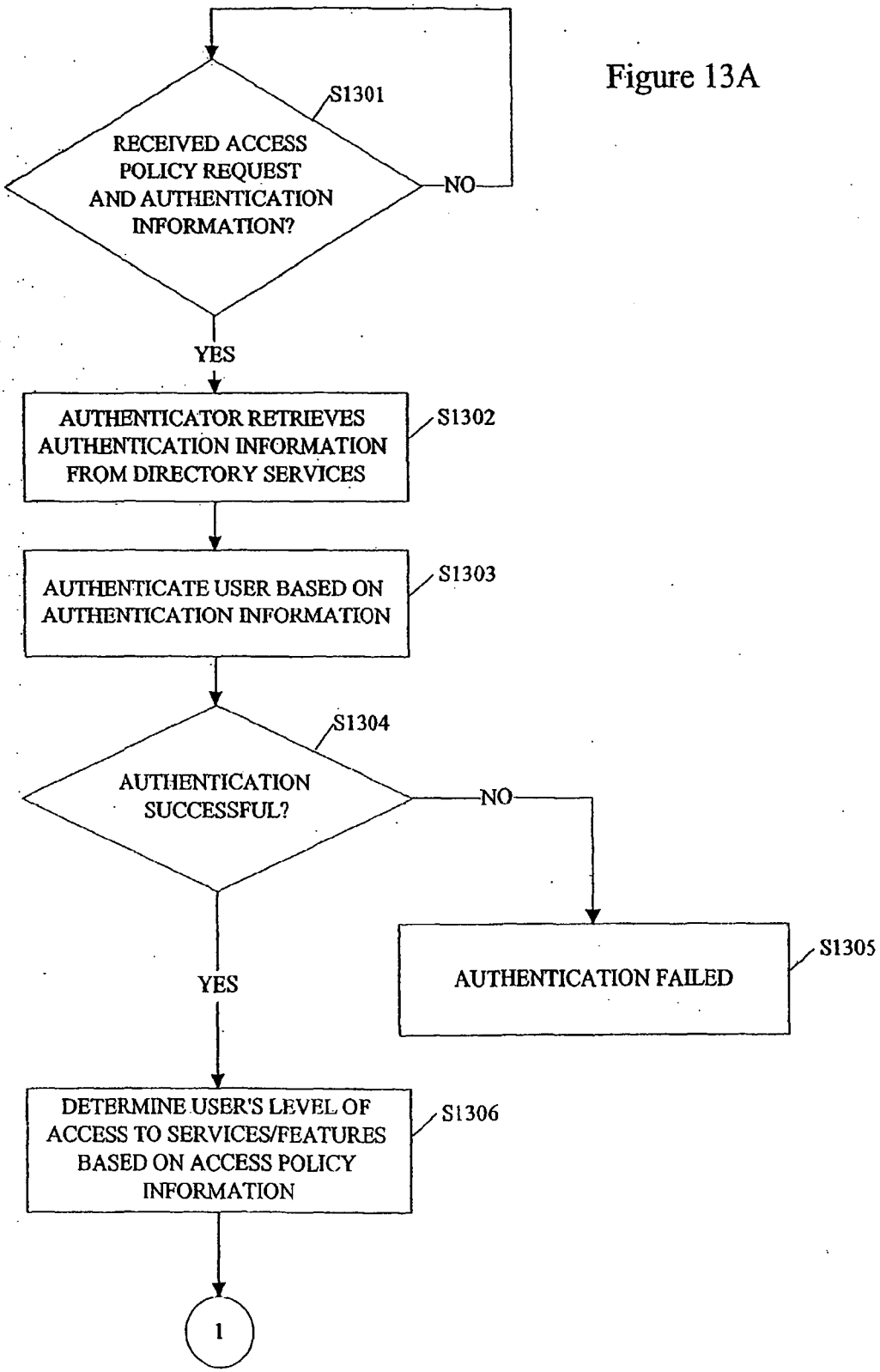
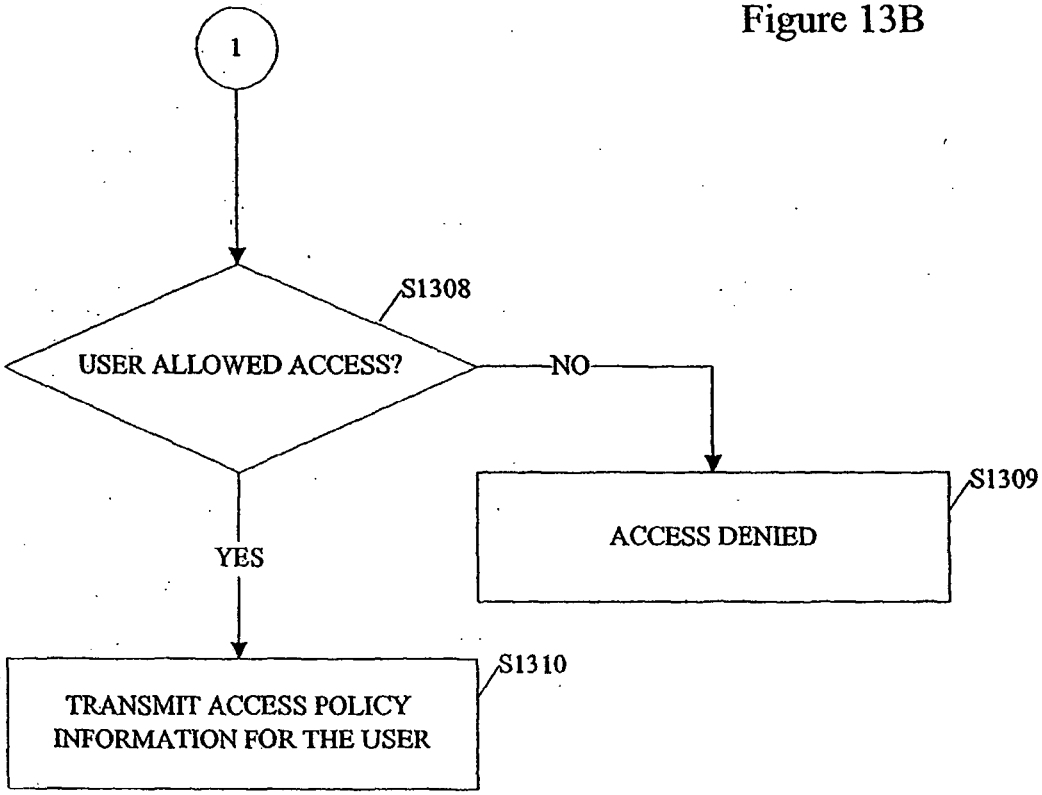


Figure 13A

Nintendo Exhibit 1003
Part 2 of 2
Page 00686

Figure 13B



Nintendo Exhibit 1003
Part 2 of 2
Page 00687

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-056360
(43)Date of publication of application : 20.02.2002

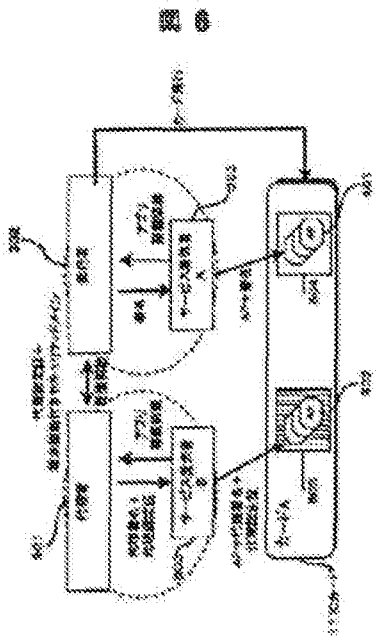
(51)Int.Cl. G06K 17/00
B42D 15/10
G06F 1/00
G06K 19/07
G09C 1/00

(21)Application number : 2000-249182
(22)Date of filing : 11.08.2000

(71)Applicant : HITACHI LTD
(72)Inventor : SATO AKIKO
MISHINA YUSUKE
OKI MASARU
BABA SATOMI
MATSUI TAKASHI

(54) IC CARD SYSTEM AND IC CARD

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a safety IC card and its operation method capable of conducting dynamic mounting of application after the publication of the card even if a service provider conducting application mounting on the IC card does not contract or communicate with a card publisher hoping for mounting.
SOLUTION: The publisher of the IC card previously gives a cipher key to a third party other than the publisher of the IC card, and entrusts an agency of a certification business to application to be mounted. A program or data having the function verifying agent certification is previously mounted on the card, and the validity of the mounting requested application is confirmed.



Nintendo Exhibit 1003
Part 2 of 2
Page 00688

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号
特開2002－56360
（P2002－56360A）
(43)公開日　平成14年２月20日(2002.2.20)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード（参考）
Ｇ０６Ｋ　17/00		Ｇ０６Ｋ　17/00	Ｄ	２Ｃ００５
			Ｂ	５Ｂ０３５
Ｂ４２Ｄ　15/10	５２１	Ｂ４２Ｄ　15/10	５２１	５Ｂ０５８
Ｇ０６Ｆ　1/00		Ｇ０９Ｃ　1/00	６４０Ｅ	５Ｂ０７６
Ｇ０６Ｋ　19/07			６６０Ａ	５Ｊ１０４

審査請求　未請求　請求項の数21　ＯＬ　（全 30 頁）　最終頁に続く

(21)出願番号　特願2000－249182(P2000－249182)

(22)出願日　平成12年８月11日(2000.8.11)

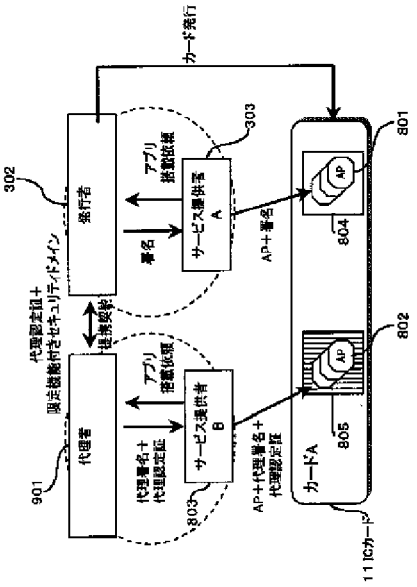
(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　佐藤　暁子
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内
(72)発明者　三科　雄介
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内
(74)代理人　100068504
弁理士　小川　勝男　（外２名）

最終頁に続く

(54)【発明の名称】　ＩＣカードシステム及びＩＣカード

(57)【要約】　（修正有）
【課題】ＩＣカード上へのアプリケーション搭載を行うサービス提供者が、搭載を希望するカード発行者と直接契約や通信を行わなくても、安全でかつカードの発行後に、アプリケーションの動的搭載が可能なＩＣカード、及び運用方法を提供する。
【解決手段】ＩＣカードの発行者は、ＩＣカードの発行者以外の第３者に、あらかじめ暗号鍵を渡しておき、搭載するアプリケーションへの認証業務の代理を委託する。また、代理認証を検証する機能を有するプログラムもしくはデータをカード上に搭載しておき、搭載要求のあったアプリケーションの正当性を確認する。

図 6



Nintendo Exhibit 1003
Part 2 of 2
Page 00689

1

【特許請求の範囲】

【請求項１】　第２の機関が、少なくとも、ＩＣカード発行に係わる第１の機関を特徴づけ且つサービス提供の為に持ちいる第１の特徴情報、及び当該第２の機関の第２の特殊情報とを、サービス情報提供に係わる第３の機関に提供が可能であり、

前記サービス情報提供に係わる第３の機関は、少なくとも、所望のアプリケーションと、前記ＩＣカード発行に係わる第１の機関を特徴づけ且つサービス提供の為に持ちいる第１の特徴情報と、当該第２の機関を特徴づけるところの第２の特殊情報と、をＩＣカードに提供が可能なことを特徴とするＩＣカードシステム。

【請求項２】　第２の機関が、少なくとも、ＩＣカード発行に係わる第１の機関を特徴づけ且つサービス提供の為に持ちいる第１の特徴情報、及び当該第２の機関の第２の特殊情報と、サービス情報のＩＣカードへの格納に関する条件を、サービス情報提供に係わる第３の機関に提供が可能であり、

前記サービス情報提供に係わる第３の機関は、少なくとも、アプリケーションと、前記第１の特徴情報と、前記第２の機関を特徴づけるところの第２の特徴情報と、前記アプリケーションのＩＣカードへの格納に関する条件と、前記格納条件に係わる第３の特徴情報と、をＩＣカードに提供が可能なことを特徴とするＩＣカードシステム。

【請求項３】　第３者機関が、当該第３者機関がＩＣカードの発行機関によってその代理であることを認定されたことを示す情報と、

当該ＩＣカードに搭載する所定の搭載情報を特徴づける第１の固有情報に当該第３者機関が代理の署名を行った情報と、をサービス提供に係わる第３の機関に提供し、

前記サービス提供に係わる第３の機関が、

少なくとも、

当該搭載情報と、

前記所定の搭載情報を特徴づける第１の固有情報に、当該第３者機関が代理の署名を行った情報と、

当該第３者機関がＩＣカードの発行機関によってその代理であることを認定されたことを示す情報と、をＩＣカードに送信することが可能なことを特徴とするＩＣカードシステム。

【請求項４】　少なくとも、所望のアプリケーションを搭載することが出来る領域と、オペレーション・システム領域とを有し、

第３者機関がＩＣカードの発行機関によってその代理であることを認定されたことを示す情報と、

アプリケーションと、

前記アプリケーションを特徴づける固有情報に、前記第３者機関が代理の署名を行った情報と、を受け取る手段と、

且つ前記第３者機関を特徴づける固有情報に付与された

2

前記ＩＣカードの発行機関の署名を検証することが可能な手段と、

前記アプリケーションを特徴づける固有情報に付与された第３者機関が代理の署名を検証することが可能な手段とを有すること、を特徴とするＩＣカード。

【請求項５】ＩＣカードの発行機関によって、第３者機関がその代理であることを認定されたことを示す情報を、前記第３者機関が受信するステップと、

前記第３者機関が、当該第３者機関を特徴付ける固有情報をブラインド化した情報を、ＩＣカードの発行機関に送信するステップと、

前記第３者機関が、ＩＣカードの発行機関から受領したをサービス情報の提供機関に送付するステップと、

前記サービス情報の提供機関が、サービス情報を、前記第１の固有情報と、当該第３者機関の秘密鍵と、前記第３者機関の公開鍵及び当該ＩＣカードの発行機関の秘密鍵と共に、格納されるサービス情報に所定の制限を与える固有情報が格納されたＩＣカードに、格納されるサービス情報に所定の制限を与えるこの固有情報が格納された領域を経由して、入力するステップと、

前記格納されるサービス情報に所定の制限を与えるこの固有情報に基づいて、前記サービス情報が格納されるべきもであることを確認するステップと、を有することを特徴とするＩＣカードシステム。

【請求項６】　第３者機関が、第３者機関の公開鍵に、当該第３者機関の公開鍵をＩＣカードの発行機関の秘密鍵を用いて暗号化した情報を付与した情報と、

当該ＩＣカードに搭載する所定の搭載情報を特徴づける第１の固有情報に、前記所定の搭載情報を特徴づける第１の固有情報を当該第３者機関の秘密鍵を用いて暗号化した情報を付与した情報と、

をサービス提供に係わる第３の機関に提供し、

前記サービス提供に係わる第３の機関が、

少なくとも、

当該搭載情報と、

前記所定の搭載情報を特徴づける第１の固有情報に、当該所定の搭載情報を特徴づける第１の固有情報を当該第３者機関の秘密鍵を用いて暗号化した情報を付与した前記情報と、

第３者機関の公開鍵に、当該第３者機関の公開鍵をＩＣカードの発行機関の秘密鍵を用いて暗号化した情報を付与した情報を付与した情報と、をＩＣカードに送信することが可能なことを特徴とするＩＣカードシステム。

【請求項７】　前記第１の固有情報が、サービス情報のハッシュ関数であることを特徴とする請求項６に記載のＩＣカードシステム。

【請求項８】　少なくとも、所望のアプリケーションを搭載することが出来る領域と、オペレーション・システム領域とを有し、

第３者機関の公開鍵に、第３者機関の公開鍵を前記カー

Nintendo Exhibit 1003
Part 2 of 2
Page 00690

ドの発行機関の秘密鍵を用いて暗号化した情報を付与した情報と、
アプリケーションと、
当該アプリケーションを特徴づける固有情報に、当該アプリケーションの固有情報を前記第３者機関の秘密鍵を用いて暗号化した情報を付与した情報と、を受け取る手段と、
前記カードの発行機関の秘密鍵を検証する手段と、且つ前記第３者機関の公開鍵に基づいて前記アプリケーションの固有情報を検証する手段を有することを特徴とするＩＣカード。
【請求項９】　第３者機関の公開鍵を受信し、
この第３者機関の公開鍵に、この第３者機関の公開鍵をＩＣカードの発行機関の秘密鍵を用いて暗号化した情報を付与した情報を、前記第３者機関に送信することが可能なことを特徴とするＩＣカード発行システム。
【請求項１０】　ＩＣカードへのサービス情報の提供機関よりの所定のサービス情報を特徴づける第１の固有情報を受領し、
前記第１の固有情報に、前記第１の固有情報を当該カードシステムの秘密鍵を用いて暗号化した情報を付与した情報と、
当該カードシステムの公開鍵に、当該カードシステムの公開鍵を当該ＩＣカードの発行機関の秘密鍵を用いて暗号化した情報を付与した情報とを、前記ＩＣカードへのサービス情報の提供機関に送信することが可能なカードシステム。
【請求項１１】　所定のサービス情報を特徴づける第１の固有情報を第３者機関に送信が可能であり、
且つ前記第１の固有情報に、前記第１の固有情報を当該第３者機関の秘密鍵を用いて暗号化した情報を付与した情報と、
前記第３者機関の公開鍵と、前記第３者機関の公開鍵をＩＣカードの発行機関の秘密鍵を用いて暗号化した情報を付与した情報とを受信し、　少なくとも、
当該搭載情報と、
前記所定の搭載情報を特徴づける第１の固有情報に、当該所定の搭載情報を特徴づける第１の固有情報を当該第３者機関の秘密鍵を用いて暗号化した情報を付与した前記情報と、
第３者機関の公開鍵に、当該第３者機関の公開鍵をＩＣカードの発行機関の秘密鍵を用いて暗号化した情報を付与した情報を付与した情報と、をＩＣカードに送信が可能なことを特徴とするサービス提供システム。
【請求項１２】　第３者機関が、ＩＣカードの発行機関の代理が可能なことを認定する情報を受信するステップと、
前記第３者機関が、前記第３者機関に対応するブラインド化した固有情報を、ＩＣカードの発行機関に送信するステップと、
前記第３者機関が、ＩＣカードの発行機関から受領したところの、格納されるサービス情報に所定の制限を与える固有情報を機能せしめるところの情報をサービス情報の提供機関に送付するステップと、
前記サービス情報の提供機関が、少なくとも、サービス情報を、前記第１の固有情報に前記第１の固有情報を、第３者機関の秘密鍵を用いて暗号化した情報を付与した情報と、前記第３者の公開鍵の公開鍵と、前記第３者機関の公開鍵をＩＣカードの発行機関の秘密鍵で暗号化した情報を付与した情報と共に、格納されるサービス情報に所定の制限を与える固有情報が格納されたＩＣカードに、格納されるサービス情報に所定の制限を与えるこの固有情報を機能せしめつつ、入力するステップと、
前記格納されるサービス情報に所定の制限を与えるこの固有情報に基づいて、前記サービス情報が格納されるべきもであることを確認するステップと、を有することを特徴とするＩＣカードシステム。
【請求項１３】　前記格納されるサービス情報に所定の制限を与えるこの固有情報が、搭載容量、搭載回数、搭載するサービス情報の個数、サービス情報の搭載可能な有効期間の群から選ばれた少なくとも１者であることを特徴とする請求項１２に記載のＩＣカードシステム。
【請求項１４】　アプリケションと、このアプリケションの格納に関する条件との格納が可能であり、前記アプリケションの格納に所定の条件に基づき、前記アプリケションの格納に関する条件に対する鍵情報を設定し、且つ前記アプリケションに付与された特徴情報を確認することが可能なことを特徴とするＩＣカード。
【請求項１５】　第３者機関の公開鍵を受信し、この第３者機関の公開鍵に、この第３者機関の公開鍵をＩＣカードの発行機関の秘密鍵を用いて暗号化した情報を付与した情報を前記第３者機関に送信が可能であり、
且つ前記第３者機関の要請に応じて、前記第３者機関のブラインド化した秘密鍵を受信し、この第３者機関のブラインド化した秘密鍵に、当該ＩＣカードの発行機関の秘密鍵を用いて暗号化した情報を付与した情報と、所望のアプリケーションの格納に関する条件とを前記第３者機関に送信が可能であることを特徴とするＩＣカードの発行システム。
【請求項１６】　サービス情報の格納に関する条件を受信し、
且つサービス情報の格納に関する条件と、前記第３者機関の秘密鍵と、前記第３者機関の秘密鍵にＩＣカードの発行機関の秘密鍵を用いて暗化した情報を付与した情報とを、前記サービス情報を提供する機関に送信が可能であり、
且つサービス情報に対する固有情報に、前記サービス情報に対する固有情報を当該カードシステムの秘密鍵によって暗号化した情報を付与した情報と、
当該カードシステムの公開鍵に、当該カード・システム

Nintendo Exhibit 1003
Part 2 of 2
Page 00691

の公開鍵を当該ＩＣカードの発行機関の秘密鍵を用いて暗号化した情報を付与した情報と、を前記サービス情報を提供する機関に送信することが可能なことを特徴とするカードシステム。

【請求項１７】　所定のサービス情報に固有に定められた第１の固有情報を第３者機関に送信が可能であり、且つ少なくとも前記第１の固有情報と、前記第３者機関の秘密鍵に、前記第３者機関の秘密鍵を当該ＩＣカードの発行システムの秘密鍵を用いて暗号化した情報を付与した情報と、前記第３者機関の公開鍵に前記第３者機関の公開鍵を当該ＩＣカードの発行システムの秘密鍵を用いて暗号化した情報を付与した情報とを受信し、
少なくとも前記第１の固有情報と、少なくとも当該第３者機関の秘密鍵に、前記第３者機関の秘密鍵を当該ＩＣカードの発行システムの秘密鍵を用いて暗号化した情報を付与した情報と、前記第３者機関の公開鍵に、前記第３者機関の公開鍵を前記ＩＣカードの発行機関の秘密鍵を用いて暗号化した情報を付与した情報と、を前記ＩＣカードに送信が可能なことを特徴とするサービス提供システム。

【請求項１８】　第１の機関が、当該第１の機関が発行するＩＣカードに対するアプリケーションの搭載を許可する情報の発信が可能であり、且つ少なくとも、前記第１の機関とは別異の第２の機関の代行が可能であることを示す情報と、前記第２の機関が発行したＩＣカードに対するアプリケーションの搭載を許可する情報との発信をも可能であることを特徴とするＩＣカードシステム。

【請求項１９】　少なくとも、第１の機関が、前記第１の機関とは別異の第２の機関に、当該第１の機関の代行が可能であることを示す情報の提供が可能であり、前記第２の機関が、前記第１の機関に、当該第２の機関の代行が可能であることを示す情報の提供が可能であり、
前記第１の機関が、当該第１の機関が発行するＩＣカードに対するアプリケーションの搭載を許可する情報の発信が可能であり、且つ少なくとも、前記第２の機関の代行が可能であることを示す情報と、前記第２の機関が発行したＩＣカードに対するアプリケーションの搭載を許可する情報との提供が可能であり、且つ前記第２の機関が、当該第２の機関が発行するＩＣカードに対するアプリケーションの搭載を許可する情報の提供が可能であり、且つ少なくとも、前記第１の機関の代行が可能であることを示す情報と、前記第１の機関が発行したＩＣカードに対するアプリケーションの搭載を許可する情報との提供が可能であることを特徴とするＩＣカードシステム。

【請求項２０】　ＩＣカードへのアプリケーション提供機関は、ＩＣカードが有するアプリケーションの格納に関する条件に規制されて当該アプリケーションのＩＣカードへの格納が可能であり、且つ当該ＩＣカードは複数の当該アプリケーションの格納に関する条件及びこれに対する複数のアプリケーションとを格納することが可能であり、且つ前記アプリケーションの格納に関する条件は前記ＩＣカードへのアプリケーションの提供システムとは別異の機関により提供された条件であることを特徴とするアプリケーションの搭載方法。

【請求項２１】　第３者機関が複数のＩＣカードの発行機関よりその代行が可能なことを示す情報の提出が可能であり、前記第３者機関は、複数サービス提供者の少なくとも１者からのアプリケーションの搭載の要求を前記第３者機関が受領し、前記ＩＣカードの発行機関の代行が可能なことを示す情報と要求を受けたアプリケーションの搭載を許可する情報とを前記要求を提出したサービス提供者に提供し、当該サービス提供者は前記ＩＣカードの発行機関の代行が可能なことを示す情報と要求を受けたアプリケーションの搭載を許可する情報とを受領し、所望のアプリケーションの提供が可能なことを特徴とするＩＣカードシステム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本願発明は、ＩＣカードの発行者以外の第三者による動的認証及びＩＣカードへのアプリケーション搭載方法及びその利用に関する諸技術に関するものである。

【０００２】

【従来の技術】ＩＣカードは、大量の情報が記録可能であり、かつ高度なセキュリティを有するため、磁気カードに代わる新しい情報記録媒体として近年普及しつつある。また、最近ではＩＣカードに複数のアプリケーションを搭載できるマルチアプリケーション機能や、カード発行後にアプリケーションを搭載可能なダイナミックローディング機能を備えたＩＣカード及びカードＯＳが主流となってきている。

【０００３】従来、これらのＩＣカードへのアプリケーション搭載方式では、該カードを発行した発行者による何らかの署名が付与されたアプリケーションのみを搭載可能としていた。カード発行後に当該カードにアプリケーションを搭載するいわゆる動的搭載も前提としているため、誰にでもアプリケーションが搭載可能であるシステムはセキュリティ上問題があるからである。

【０００４】図１にＩＣカード（１１）内に搭載されているＩＣ内部の基本的な領域の論理的な構成を示す。当該ＩＣは、ハードウェア層（１０１）とＯＳが搭載される領域、即ちＯＳ層（１０２）とアプリケーションが搭載される領域、即ちアプリケーション層（１０７）とを有する。ここで、マルチアプリケーションの搭載が可能とは、アプリケーション層（１０７）に複数のアプリケーション（１０６）が搭載できることである。又、ダイナミックローディングが可能であるとは、このアプリケーション（１０６）をカード発行後にアプリケーションが搭載あるいは削除が可能であることを示す。ＯＳ層

Nintendo Exhibit 1003
Part 2 of 2
Page 00692

（１０２）は通信処理部（１０３）、インタープリタ（１０４）、セキュリティ機構（１０５）などを有し、外部端末からのコマンドの受信やアプリケーションのコマンドの転送などを行っている。当然、アプリケーション層（１０７）とＯＳ層（１０２）との間にはアプリケーション・インターフェイス、ＯＳ層（１０２）とハードウェア層（１０１）との間にはハードウエア・インターフェイスが設けられる。

【０００５】図２はＩＣカードの発行やサービスの提供に関する従来のシステム構成図である。カード発行者（３０２）とは、ＩＣカードの発行と管理の業務を行う事業主体者であり、カードの責任を負う主体者である。また、サービス提供者（３０３）はＩＣカード上に搭載するアプリケーションの発行と管理の業務を行う事業主体者である。カード発行後に、アプリケーションをダイナミックに搭載する場合は、サービス提供者の端末からＩＣカードへ搭載することが多い。クライアント側（３０１）にはＩＣカードとコマンドのやり取りを行う外部端末（３０４）がある。カード発行者は発行者データベース（３０５）に発行管理データを保持しており、そのデータに従いカードを発行し利用者に配布する。発行管理データには、利用者からの申請情報や、カードの発行に必要な基本情報が記載されている。ＩＣカード搭載用アプリケーションは、サービス提供者が作成もしくは取得し、ＩＣカードへのアプリケーション搭載を行う。

【０００６】サービス提供者データベース（３０６）にはアプリケーション関連（以下、ＡＰ関連と称する）データが格納されている。前記のように、ＩＣカード上の全ての責任はカード発行者に帰属するため、アプリケーションの正当性をカード発行者が確認する手順が必要である。サービス提供者はカード発行者のＡＰの搭載を許可する処理部にＡＰ搭載の許可の要求を行う。カード発行者のデータベースには事前に登録されているＡＰに関するデータが格納されている。そして、要求のあったアプリケーションが不正なものではないか、要求をしているサービス提供者が不正ではないかなどをチェックした上で、ＡＰの搭載許可を発行する。サービス提供者はこのＡＰ搭載許可とアプリケーションを合わせてカードに送信し設定するのである。そのため、サービス提供者はあらかじめカード発行者と契約し、自身が正当な事業主であることや、搭載予定のアプリケーションが不正なものではないことを登録する必要がある。

【０００７】カード発行者とサービス提供者のサーバ間は、ネットワークでつながれている場合が多いと想定されるが、書類形式でやり取りすることも可能である。

【０００８】またこれに加え、アプリケーション搭載の際に、カード内でアプリケーション搭載を専用に行うアプリケーションもしくはデータを用いる場合がある。この方式は“セキュリティドメイン”というアプリケーションをサービス提供者毎に用意し、アプリケーション搭載・削除・管理の役割を担わせている。図３にセキュリティドメイン機構を持つＩＣカードのＩＣ内部の基本的な領域の論理的な構成を示す。セキュリティドメイン（１０８）はアプリケーション層に存在し自らが責任を持つアプリケーション（１０６）を管理する。アプリケーションはカードに搭載や削除される場合、このセキュリティドメインを通じて行われる。また鍵データなどのセキュリティ情報やＩＤ番号などのデータ管理をセキュリティドメインに委託する場合もある。

【０００９】

【発明が解決しようとする課題】上記のようなこれまでのＩＣカードへのアプリケーション搭載方式は、高セキュリティを保ちながらアプリケーションの動的な搭載や削除を可能としている。しかし、複数カード発行者やそのサービス提供者間での運用に際してはいくつかの難点がある。

【００１０】前述のように、サービス提供者は、アプリケーションの搭載を希望するカードのカード発行者に対して、あらかじめ契約を結ぶ必要がある。つまり、カード発行者は、発行するカードにアプリケーションの搭載を希望する全てのサービス提供者と契約関係にあり、そのサービス提供者からの要求を受け、アプリケーション搭載や削除を行う場合は上記アプリケーション搭載方法で良い。図５のように発行者（３０２）と契約関係にあるサービス提供者（３０３）がカードＡ（１１）にアプリケーションを搭載する場合は従来方法でいい。しかし、サービス提供者（８０３）のように、カード発行者とあらかじめ複雑な事前契約やアプリケーションの登録をいちいち行わずにアプリケーションを搭載する要求がでてくるだろう。そもそも、ＩＣカードは複数アプリケーションを搭載可能なことが大きな利点であり、多数のサービスが提供可能であることはカード運用側のカード発行者にとっても利用側のユーザにとっても多大なメリットであるため、このような要求は年々増加するものと予想される。しかし、現在のアプリケーション搭載方式に従い、事前契約などせずにサービスを供給することは不可能である。図５に示すように、サービス提供者（８０３）と発行者（３０２）の間には何らかの情報交換が必要である。つまり、従来のアプリケーション搭載方式に従うならば、サービス提供者はアプリケーション搭載を希望するカードの発行者全てと事前に契約を結ぶ必要があり、アプリケーション搭載時には全てのカード発行者にアプリケーション搭載許可証を発行してもらわなければならない。この実現方法には以下のような難点がある。

【００１１】（１）カード発行者―サービス提供者間の契約関係及び運用時の通信の増大と複雑化

Ｎ個のカード発行者とＭ個のサービス提供者の間で相互にサービスを運用する場合、Ｎ＊Ｍの契約関係が結ばれることになり膨大な量の契約と運用時の通信が発生する

Nintendo Exhibit 1003
Part 2 of 2
Page 00693

ことになる。それに伴うコストや時間が増大し最終的にはＩＣカード単価の増加の原因となる。

【００１２】（２）現実的に契約関係を結べない事業者同士の問題

例えば、インターナショナルカードに国内企業のアプリケーションを搭載する場合、この国内企業がカード発行者と直接コンタクトをとることは現実的にはあまりなく、国内向け窓口業務を行う事業者が間に入ることが考えられる。このようにカード発行者やサービス提供者が属する事業主体により直接契約を結んだり通信を確立することが難しい。

【００１３】

【課題を解決するための手段】本願発明の次の主な二つの形態を有する。これら形態の基本思想を、その要点をごく簡潔に要約すると次ぎの通りである。

【００１４】本願発明の第１の形態は、ＩＣカードの発行者以外の第三者による認証書を用いることである。尚、以下ＩＣカードの発行者以外の第三者による認証書を簡便に代理署名と称する。

【００１５】本願発明の第２の形態は、ＩＣカードが限定機能付きのセキュリティドメインを有することである。この第２の形態は、前記第１の形態の機能を補足し、より有用なＩＣカード・システムを提供することを可能ならしめるものである。

【００１６】そして、本願発明の第３の形態は、前記第１と第２の発明思想を合わせて使用することである。こうして、カード発行者が認めた代理者が、カード発行者が認めた条件でアプリケーション搭載を代行することが可能である。

【００１７】以下に、これらの本願発明の基本思想を軸に、本願発明を説明する。

【００１８】先ず、本願発明の第１の形態についてである。

【００１９】第１の形態は、例えば、第２の機関が、少なくとも、ＩＣカード発行に係わる第１の機関を特徴づけ且つサービス提供の為に持ちいる第１の特徴情報、及び当該第２の機関の第２の特殊情報とを、サービス情報提供に係わる第３の機関に提供が可能であり、前記サービス情報提供に係わる第３の機関は、少なくとも、所望のアプリケーション用プログラムと、前記ＩＣカード発行に係わる第１の機関を特徴づけ且つサービス提供の為に持ちいる第１の特徴情報と、当該第２の機関を特徴づけるところの第２の特殊情報と、をＩＣカードに提供が可能なＩＣカード・システムであると言うことが出来る。

【００２０】この観点は、第１に、わけても、カード発行者　サービス提供者間の契約関係及び運用時の通信の増大と言う前述の難点を解決するためのものである。

【００２１】このためには、発行者とサービス提供者の間に代理者という概念を作り、カード発行者とサービス提供者が直接通信を確立せずともアプリケーション搭載可能な方式が必要となってくる。具体的には、カード発行者が代理者と事前提携をすることで代理者業務を行うことを認証することで、サービス提供者が直接カード発行者と通信をすることなくアプリケーション搭載が実現可能であり、かつ通常のアプリケーション搭載方式と同程度のセキュリティを保つことが可能な方式が必要である。

【００２２】本願発明は、以上の要求を満たすＩＣカードのアプリケーションの搭載方式を提供する。

【００２３】図６は本願発明の方式の骨子を説明する図である。ＩＣカード（１１）はカード発行者（３０２）により発行されている。サービス提供者Ａ（３０３）はカード発行者（３０２）と契約を結んでおり通常のアプリケーション搭載方法によりカードにアプリケーション８０１を発行する。

【００２４】サービス提供者Ｂ（８０３）は代理者（９０１）と契約を結んでいる。

【００２５】本願発明の前提は、前述の難点からして、サービス提供者Ｂ（８０３）とカード発行者（３０２）との間の通信を行わないことである。従って、本願発明は、代理者を通して間接的に発行者と通信を確立する手法である。つまり、カード発行者が行っているアプリケーションの許可を代理者に代行させ、サービス提供者Ｂがカードにアプリケーションを搭載するためには代理者とのみ通信を行えば良いという方法である。しかし、代理者によるアプリケーションへの署名（以降、代理署名と呼ぶ）のみでは、次ぎのような難点を有する。即ち、その第１はカードＡが正当性を検証できない点である。第２は、カード発行者の意志に関わらず不特定者がアプリケーションの搭載が可能であるとの難点である。尚、ここで、署名とは、暗号理論における署名であることは言うまでもない。その代表例はデジタル署名である。例えば、こうしたディジタル署名は、署名アルゴリズムと、確認アルゴリズムの２つの要素から成っている。例えば、秘密の署名アルゴリズムを利用して、文書に署名をすることが出来る。その結果の署名は、その後公開されている確認アルゴリズムを用いて確認できる。確認アルゴリズムは、文書と署名とによって、確認アルゴリズムは署名が本物かどうかを判断し、真もしくは偽という答えを返すことが出来る。こうした署名に関する一般的な理論は、例えば、ダグラス・スチンソン著、桜井幸一訳の「暗号理論の基礎」（共立出版、１９９６）の第６章、わけても第２１７頁より第２２０頁、あるいは「ハンドブック　オブ　アプライド　クリスタルグラフィ（Ｈａｎｄｂｏｏｋ　ｏｆ　Ａｐｐｌｉｅｄ　Ｃｒｙｓｔａｌｇｒａｐｈｙ）」（ＣＲＣ　Ｐｒｅｓｓ、１９９６）、第４３３頁より第４３４頁に見られる。

【００２６】こうした難点を回避する為の本願発明の第１の形態の骨子は、次の諸点に要約される。第１は、カ

Nintendo Exhibit 1003
Part 2 of 2
Page 00694

ード発行者と代理者との間で提携契約を結ぶ際に、カード発行者が代理者に対して代理業務を行うことを認める代理者認定証を渡しておくことである。第２は、代理者がアプリケーション認証する際に、代理署名と代理者認証書を合わせてサービス提供者に渡すことである。こうして本願発明によれば、カードＡ内で代理者認証書がカード発行者による正当なものかを検証することで、不特定者によるアプリケーションの搭載を防ぐことができる。

【００２７】本願発明の主な第２の形態は、前記第１の形態の機能を補足し、より有用なＩＣカード・システムを提供することを可能ならしめるものである。その形態を要約すると、ＩＣカードが、いわゆる限定機能付きのセキュリティドメインを有することである。

【００２８】本願発明の第２の形態は、例えば、第２の機関が、少なくとも、ＩＣカード発行に係わる第１の機関を特徴づけ且つサービス提供の為に持ちいる第１の特徴情報、及び当該第２の機関の第２の特殊情報と、アプリケーション用プログラムのＩＣカードへの格納に関する条件を有するプログラムとを、サービス情報提供に係わる第３の機関に提供が可能であり、前記サービス情報提供に係わる第３の機関は、少なくとも、アプリケーション用プログラムと、前記第１の特徴情報と、前記第２の機関を特徴づけるところの第２の特徴情報と、前記アプリケーション用プログラムのＩＣカードへの格納に関する条件を有するプログラムと、前記格納条件を有するプログラムに係わる第３の特徴情報と、をＩＣカードに提供が可能なＩＣカード・システムであると言うことが出来る。

【００２９】本願発明に係る第１の形態のみでは一度代理者認証書を発行したら、代理者は、それ以降無制限にアプリケーションを搭載できる権限を有する。勿論、提携の契約条件にもよるのだが、カード発行者にとっては、「アプリケーション1個分の搭載許可権限を与えたい」あるいは「有効期限を設定した上での搭載許可権限を与えたい」などの要求は当然発生する。本願発明の第２の形態は、この要求を満たすものである。

【００３０】セキュリティドメインは、前述のようにカード内でアプリケーションの搭載、削除あるいは管理機能などを実現するアプリケーションもしくはデータを有する領域である。従って、本来、セキュリティドメインは、サービス提供者によって作成される。しかし、本願発明では、カード発行者がこのセキュリティドメインを作成し、これを代理者を通してサービス提供者に渡し、サービス提供者のセキュリティドメインとして機能させる。カード発行者は作成の際、アプリケーション搭載回数や有効期限など提携契約に対応する諸条件をセキュリティドメイン内部に設定し、提携先に送信する。もちろん、このセキュリティドメインは改竄防止策を取っている必要がある。

【００３１】図６を参酌すれば、限定機能付きセキュリティドメイン（８０５）は、カード発行者（３０２）により作成され、サービス提供者Ｂ（８０３）のセキュリティドメインとしてカードＡ１１内に格納される。サービス提供者Ｂ（８０３）がアプリケーション（８０２）を搭載する場合、サービス提供者Ｂ（８０３）は代理者（９０１）にアプリケーションの搭載の許可を依頼する。代理者（９０１）は、サービス提供者Ｂ（８０３）に代理者認証書と代理署名を合わせて送信する。サービス提供者Ｂ（８０３）はアプリケーション（３７）と代理者認証書と代理署名を合わせてカードＡ（１１）内に送る。カードＡは、限定機能付きセキュリティドメイン内に設定された限定条件に照らし合わせながら、代理者認証書が正しくカード発行者によることを検証する。その結果、カードＡが、代理署名によりアプリケーションの正当性を確認した場合、当該カードＡはアプリケーションを搭載を可能とする。

【００３２】

【発明の実施の形態】図７は、本発明のＩＣカード（１１）及びＩＣカードアプリケーションの発行システムの構成図である。図７で、符号３０２はカード発行者のサーバであり、符号３０５はカード発行者のデータベースでカード発行者が運用する全てのＩＣカードに関する運用情報である発行管理データを管理する為のものである。同様に符号９０１は代理者のサーバであり、符号９０２は代理者のデータベースで、代理者の搭載の許可を出すＡＰに関する運用管理情報であるＡＰ関連データを管理する。符号３０３はサービス提供者のサーバであり、符号３０６はサービス提供者で自身のアプリケーションの運用情報であるＡＰ関連データを管理する。ＩＣカード（１１）はカード発行者システム内のカード発行処理部（３０８）により発行され、クライアントである利用者（３０１）のもとに配布される。尚、ここでのカードの発行及び配布は、オンライン動作ではなく、通例の意味における発行及び配布である。

【００３３】また、カード発行処理部（３０８）は代理者システムのＡＰ搭載許可処理部（３１０）と事前提携（３１５）を行い、代理者であることを認める代理認定証のやり取りなどを行う。サービス提供者（３０３）がアプリケーションをカードに搭載する場合、サービス提供者システム内のＡＰ搭載処理部（３１２）が代理者のＡＰ搭載許可処理部（３１０）にＡＰ搭載要求（３１９）を行う。代理者は事業者ポリシーに従い、ＡＰの正当性を確認の上ＡＰ搭載許可を発行する（３１６）。サービス提供者は、受け取ったＡＰ搭載許可とアプリケーションを合わせて外部端末（３０４）を通してＩＣカードへ送信する（３１４）。カード発行者サーバ（３０２）と代理者サーバ（９０１）、代理者サーバ（９０１）とサービス提供者サーバ（３０３）は、各々基本的にネットワーク（３１８、３１９）を通して接続されて

Nintendo Exhibit 1003
Part 2 of 2
Page 00695

いる。しかし、運用事業者のポリシーによりフロッピー（登録商標）ディスクなど情報記録媒体の郵送や、書面の郵送などにより情報のやり取りを実現することもある。このように、本願発明のシステムは、カード発行者と代理者は提携契約を結び、代理者がサービス提供者へのサービス提供許可を代行し、サービス提供者サーバからユーザのＩＣカードへアプリケーションが搭載される仕組みとなっている。

【００３４】ここで、以下の説明の理解を容易にする為、本願明細書での特徴的な主要用語について説明する。

【００３５】「代理署名」は、カードの発行者を代理して署名を行うことである。その具体例は署名の対象となる情報に、当該情報を代理者の秘密鍵で暗号化した情報を付与した情報のことである。尚、ここで、前記署名の対象となる情報に代えて、この情報を特徴付ける情報、例えば、そのハッシュ値を用いることも出来る。従って、前記「代理署名」の用語は、この意味をも含むものである。

【００３６】「署名付＊＊」は、「＊＊」に、「＊＊」を署名者の鍵、例えば秘密鍵で暗号化した情報を付与した情報のことである。尚、ここで、前記署名者の鍵に代えて、この鍵情報を特徴付ける情報、例えば、そのハッシュ値を用いることも出来る。従って、前記「署名付＊＊」の用語は、この意味をも含むものである。

【００３７】「代理認定書」は、代理者が発行者によって、その代理であることを認定されたことを示す情報である。その具体例は、代理者の公開鍵に、代理者の公開鍵を発行者の秘密鍵で暗号化した情報を付与した情報である。

【００３８】「アプリケーション認定書」は、アプリケーションをＩＣカードに搭載することを許可する情報である。その具体例は、アプリケーションを特徴付ける特徴情報に、その特徴情報を代理者の秘密鍵で暗号化した情報を付与した情報である。

【００３９】「セキュリティドメイン」はアプリケーションの搭載、削除、あるいはその他の諸管理を行うプログラムのことである。

【００４０】「限定機能付きセキュリティドメイン」は、前記のアプリケーションの搭載、削除、あるいはその他の諸管理にある制限を与える条件を付与したプログラムのことである。

【００４１】尚、本願発明は非対称暗号化機能を有するデータ・システムに関する。

【００４２】非対称暗号化機能とは、例えば、暗号鍵として公開鍵、秘密鍵を使用し、且つこうした非対称暗号鍵により暗号化されたデータの復号化が可能である機能のことである。

【００４３】次に、本願発明の係わる代理署名及びアプリケーションの搭載の具体的方法を説明する。

【００４４】ＩＣカードに対する端末機などは、通例のもので良い。本願発明に係わる各機関は、サーバ、リーダライタなど端末機を有する。本願に係わる各種情報は、このサーバに一旦保持される場合、リーダライタを通してＩＣカードに情報が搭載される場合など各種形態が考えられる。本願発明はこうした動作形態に直接関係するものではない。本願発明はこれらのいずれの場合も適用可能である。

【００４５】図２９は簡単にカード・システム例の概要を示す図である。ＩＣカード５２の中にはチップ５１があって、リーダライタ５３とデータのやりとりを行う例を示している。リーダライタの中にはコントロールプロッセサ５４やデータベースとなる磁気ディスク５５などが存在する。ＩＣカード５２には、例えば、Ｖｃｃ（供給電源）、ＧＮＤ（グランド）、ＲＳＴ（リセット）、Ｉ／Ｏ（入出力）、及びＣＬＫ（クロック）などの諸端子が示されている。又、図中、（１）はリーダライタ５３からＩＣカード５２に対しての、例えば、ＩＤ（ＩＤＥＮＴＩＦＩＣＡＴＩＯＮ）などの各種問い合わせを示す。（２）はＩＣカードが前述の問い合わせに対する返答を示す。こうした、諸情報の伝達は通例のシステムで十分である。従って、その詳細の説明は省略する。

【００４６】尚、ＩＣカード内のＩＣチップにおいて、具体的には、前述のアプリケーション、セキュリティドメインなどは、ＯＳ部、メモリ領域に搭載される。一般に、メモリとしては、ＲＡＭ（Ｒａｎｄｕｍ Ａｃｃｅｓｓ Ｍｅｍｏｒｙ）、フラッシュ・メモリ（Ｆｌａｓｈ Ｍｅｍｏｒｙ）、ＦＲＡＭ（Ｔｅｒｒｏｍａｇｎｅｔｉｃ Ｒａｎｄｕｍ Ａｃｃｅｓｓ Ｍｅｍｏｒｙ）、ＥＥＰＲＯＮ（Ｅｌｅｃｔｒｉｃａｌ Ｅｒａｓａｂｌｅ Ｐｒｏｇｒａｍｍａｂｌｅ ＲｅａｄＯｎｌｙ Ｍｅｍｏｒｙ）あるいはＲＯＭ（Ｒｅａｄ Ｏｎｌｙ Ｍｅｍｏｒｙ）などが用いられる。従って、本願明細書において、「セキュリティドメインを用いて」あるいは「セキュリティドメインを経由して」とは、ＯＳ部が、そのセキュリティドメインなるプログラムを取り出し、このプログラムによって、所定の情報たるデータの諸管理状態を確認し、必要があれば、所定の処理を行うことを意味している。ここで、データの諸管理の項目とは、例えば、サービス情報の搭載容量、搭載回数、搭載するサービス情報の個数、サービス情報の搭載可能な有効期間などである。

【００４７】ＯＳ部などは通例のもので十分である。ＩＣチップ内の信号処理は、通例のＩＣカードのそれをもちいて十分である。

【００４８】又、ＩＣカードには、接触型ＩＣカード、非接触型ＩＣカードなどがあるが、本願発明はこうしたＩＣカードの構成自体によらず適用することが出来る。

【００４９】図８を参酌して、代理署名について説明する。カードＡ（１１）はマルチアプリケーションのダイ

Nintendo Exhibit 1003
Part 2 of 2
Page 00696

ナミックローディングに対応するＯＳが搭載されたＩＣカードであり、カード発行者（３０２）により発行済みである。また、サービス提供者Ｂ（８０３）はＩＣカードにアプリケーションを搭載しサービスを運用する事業主体者であり、代理者（９０１）とサービス提供に関する契約を結んでいる。通常であれば、サービス提供者Ｂ（８０３）がカードＡ（１１）にアプリケーションを搭載するためには、カード発行者（３０２）とサービス提供に関する契約を直接結び、通信を確立した上で許可を受けなければならない。しかし、本願発明のシステムは、カード発行者（３０２）と代理者（９０１）の間で提携契約を結び、代理者（９０１）がカード発行者（３０２）の代理としてアプリケーション搭載許可を行う方法である。提携契約時に、カード発行者（３０２）は代理者（９０１）に代行業務を行うことを認める代理認定証を渡し、以降アプリケーション搭載許可は代理者（９０１）が行う。サービス提供者Ｂ（８０３）はアプリケーション搭載時に、代理者からアプリケーション搭載許可を受け、カードＡにアプリケーションを搭載する。

【００５０】上述した本願発明の第１の形態のシステムが、具体的にどのように行われるか、図９を参酌して説明する。即ち、代理署名を行う際の処理シーケンスを説明する。

【００５１】＜代理署名を行う際の代表的な基本処理シーケンスの例＞カード発行者（３０２）と代理者（９０１）の間では、お互いの運用ポリシーに基づき提携契約を結ばれている。提携契約に合意した場合、代理者はカード発行者に自らの非対称鍵公開鍵を送付し、代理者認証書を要求する（ステップ１２００１）。

【００５２】カード発行者は受け取った公開鍵に対して、自らの非対称鍵秘密鍵で署名を行った代理認定証を代理者に送付する（ステップ１２００２）。

【００５３】次に、カードＡ（１１）へのサービス提供者Ｂ（８０３）によるアプリケーション搭載する際は、まず、サービス提供者Ｂが代理者にカードＡに対するアプリケーション搭載許可要求を行う（ステップ１２００３）。尚、アプリケーション搭載許可要求の具体例は、例えば、搭載するアプリケーションのハッシュ値を送付することである。ハッシュ値とは、当該アプリケーションに対応した固有のデータ値である。アプリケーション搭載許可要求を行う場合、必ずしもハッシュ値でなくとも良いが、ハッシュ値が最も便利であり、多く用いられる。

【００５４】代理者は、サービス提供者Ｂから事前に申請されているアプリケーションの内容を照会し、その正当性を確認する。前述のハッシュ値を用いる場合、この正当性の確認は、例えば、サービス提供者Ｂより送付されたハッシュ値と、当該ＩＣカードの内部で発生させた当該アプリケーションに対応したハッシュ値とを比較することによって行われる。こうしたステップの詳細は後述される。また、カード発行者よりホットリスト（不正カード情報あるいはブラックリスト）などカードに関する情報が提供されていれば、それをも合わせて確認される。

【００５５】代理者は受け取ったアプリケーションのハッシュ値に非対称鍵秘密鍵で代理署名し、代理認定証と合わせてサービス提供者Ｂに送付する（ステップ１２００４）。なお、この秘密鍵は提携契約時にカード発行者に送付した非対称公開鍵に対応するものである。

【００５６】サービス提供者は受け取った代理署名付きアプリケーションハッシュ値、代理認定証、アプリケーション本体を合わせてカードに送信する（ステップ１２００５）。

【００５７】カード内には、カード発行時に、カード発行者の公開鍵が格納されているので、それを用いて代理認定証の署名検証を行い正当性を確認する。

【００５８】これは、代理者の公開鍵に署名を付与したものなので、署名が正しいことを確認した上で、この公開鍵を用いてアプリケーションのハッシュ値を復号化する。それと受け取ったアプリケーションから作成したハッシュ値を比較し同一であれば、アプリケーションをロード・インストールし、アプリケーション搭載完了となる。

【００５９】このように本願発明の第１の形態の手順は、アプリケーションの正当性を代理者が保証し、代理者の正当性をカード発行者が保証することで、間接的にカード発行者の保証のもとにアプリケーションを搭載する仕組みとなっている。図６に例示したように、カード発行者とその他プレーヤの間は、提携契約時、代理認定証の送付のステップがあるが、以降オフラインとなっており、通信を確立する必要がない。ただし、提携契約内容によっては、不正カード情報であるホットリストや各種運用に必要な情報をカード発行者間でやり取りする場合もあるが、その場合でも、カード発行者とサービス提供者との間に通信を直接確立する必要はない。

【００６０】以上説明した代理署名を行う際の基本処理シーケンスに基づく動作の更なる動作例の詳細を、図１０から図１４を用いて説明する。即ち、各図は、各々カード、カード発行者、代理者、サービス提供者の各プレーヤ毎の動作フローチャートである。

【００６１】＜ＩＣカード：ＩＣカードでのアプリケーション搭載処理の例＞図１０は、代理署名を用いたアプリケーション搭載処理を行うＩＣカード側での動作のフローチャートである。

【００６２】カードがアプリケーション搭載処理を開始する（ステップ１３００１）。

【００６３】ＩＣカードは、サービス提供者より、代理認定証と代理者が署名したアプリケーション認証書とアプリケーションを受信する（ステップ１３００２）。このステップは図９のステップ１２００５に対応する。

Nintendo Exhibit 1003
Part 2 of 2
Page 00697

【００６４】代理認定証はカード発行者により署名されている。一方、この代理認定証に対応する鍵がＩＣカード内に格納されており、ＩＣカードは、それを使用して前記の代理認定証を検証し、代理者の鍵を復号化する（ステップ１３００３）。

【００６５】復号化した代理者の鍵を用いて、代理者が署名したアプリケーション認証書の検証を行い、アプリケーションのハッシュ値を取り出す（ステップ１３００４）。

【００６６】また、ＩＣカードは、送信されたアプリケーションのハッシュ値をカード内で計算し、さきほど復号化したハッシュ値と同一かどうかを確認する(ステップ１３００５)。

【００６７】これらが同一であれば、ＩＣカードは、カード発行者が認証した正当なアプリケーションだと認め、これをインストールする（ステップ１３００６）。

【００６８】一方、これらが同一でない場合は、不正なアプリケーションである可能性があるのでインストール処理を中止し、その旨をカード外にエラーメッセージで伝える（ステップ１３００７）。

【００６９】＜カード発行者：カード発行者が代理署名を用いた事前提携処理を行う処理例＞図１１は、カード発行者が代理署名を用いた事前提携処理を行う場合のフローチャートである。このステップは図９における、ステップ１２００１及び１２００２に対応する。

【００７０】カード発行者は、代理者を認める事前提携処理をはじめる。

【００７１】カード発行者は、代理認定証を作成するため、代理者から公開鍵を受け取る（ステップ１４００２）。

【００７２】カード発行者は、ＩＣカード内に格納してある鍵と、これに対応する鍵でこの公開鍵に署名を行い、代理者に戻す（ステップ１４００３）。　以上が、カード発行者　代理者間の事前提携フローである。

【００７３】図１２は代理署名を用いた、代理者の事前提携処理手順を示すフローチャートである。代理者はカード発行者の代行業務を行うために事前提携処理を行う。まずカード発行者に代理者の公開鍵を送信し（ステップ１５００２）、カード発行者の署名が付与された代理認定証を受け取る（ステップ１５００３）。

【００７４】＜代理者：代理署名を用いた、アプリケーション搭載処理の例＞図１３は代理署名を用いた、アプリケーション搭載処理手順を示すフローチャートである。

【００７５】代理者は、サービス提供者からのアプリケーション搭載要求を受領する（ステップ１６００２）。

【００７６】代理者は、事前に登録されていたアプリケーションの内容やサービス提供者の正当性を確認し、アプリケーション搭載を許可できるかどうかを判断する（ステップ１６００３）。

【００７７】搭載を許可する場合、代理者は、アプリケーション認証書をサービス提供者に送り返す（ステップ１６００４）。このアプリケーション認証書は、サービス提供者から送られてきたアプリケーションの特徴を示すデータを、代理者が鍵で暗号化したデータとカード発行者から事前提携時に受け取っている代理認定証を合わせたものである。尚、前記アプリケーションの特徴を示すデータは、例えば、前述のアプリケーションのハッシュ値などである。

【００７８】搭載不可の場合は、代理者は、その旨をサービス提供者に伝える（ステップ１６００５）。

【００７９】＜サービス提供者：サービス提供者のアプリケーション搭載処理の例＞図１４は代理署名を用いた、サービス提供者のアプリケーション搭載処理手順を示すフローチャートである。

【００８０】サービス提供者は、代理者に対して、アプリケーションの特徴を示すデータ、例えばハッシュ値などを送付して搭載許可を要求する（ステップ１７００２）。

【００８１】サービス提供者は、代理者からアプリケーション認証書を受け取り（ステップ１７００３）、アプリケーションと合わせてＩＣカードに送信する（ステップ１７００４）。

【００８２】以上が、代理署名方式を利用してＩＣカードにアプリケーションを搭載する場合の処理手順の例である。

【００８３】次に、前記本願発明の第１の形態の機能を補足する第２の形態の具体例を説明する。前述したように、その形態を要約すると、ＩＣカードが、いわゆる限定機能付きのセキュリティドメインを有することである。

【００８４】上述のように、代理認定証を渡すことで問題点は解決したが新たに起こる問題として、一度代理認定証を発行したら、代理者はそれ以降無制限にアプリケーションを搭載できる権限を持つということがある。ＩＣカード上のアプリケーション搭載領域は無限にあるわけではなく、また提携契約などは期限を設定して結ばれることが通常であるため、カード発行者にとって「アプリケーション1個分の搭載許可権限を与えたい」あるいは「有効期限を設定した上での搭載許可権限を与えたい」などの要求を満たす仕組みが望まれる。

【００８５】そこで、本願の第２の形態が提案された。図１５は「限定機能付きセキュリティドメイン」を用いる場合の全体構成図である。

【００８６】図１５において、ＩＣカード（１１）、カード発行者（３０２）、代理者（９０１）、サービス提供者Ｂ（８０３）のプレーヤの役割と関係は図１１と同様である。カード発行者（３０２）と代理者（９０１）の間で提携契約を結び、代理者（９０１）がカード発行者（３０２）の代理としてアプリケーション搭載許可を

Nintendo Exhibit 1003
Part 2 of 2
Page 00698

行う方式であることは図１１と同じであるが、異なる点としてアプリケーションを搭載する前に、セキュリティドメインを搭載する必要がある点である。以下の説明では、図９の例と異なる点を主に説明し、同じ諸点は説明を省略する。

【００８７】セキュリティドメインとはカード内で、アプリケーション搭載、削除、あるいは管理を行うものである。こうした管理を可能ならしめる前提条件としては、（１）セキュリティドメイン内部に非対称鍵暗号鍵を格納することが可能であることと、（２）アプリケーションはセキュリティドメインを通してカードに格納されることが挙げられる。セキュリティドメインの具体的な形態はアプリケーションの場合もあり、ライブラリの場合もあり、データの場合もある。また、従来はアプリケーション搭載を行う事業者が、セキュリティドメインを作成していた。しかし、本願発明では、カード発行者が作成する。提携契約時にカード発行者（３０２）は代理者（９０１）に代理者として認める代理者認証書の他に限定機能付きセキュリティドメインを渡す。カードＡにアプリケーションを搭載希望するサービス提供者Ｂはまずカードにセキュリティドメインを搭載するため代理者のもとにある限定機能付きセキュリティドメインを受け取りカード内に設定する。その後アプリケーション搭載許可を代理者から受け取りアプリケーションを前記セキュリティドメインを通してカード内に格納する。

【００８８】図１６は、代理署名と限定付きセキュリティドメインの機能を合わせて用い、提携を実現する際の処理シーケンスを示す図である。

【００８９】カード発行者（３０２）と代理者（９０１）の間でお互いの運用ポリシーに基づき提携契約を結ばれている。提携契約に合意した場合、代理者はカード発行者に自らの非対称鍵公開鍵を送付し代理者認証書を要求する（ステップ１９００１）。

【００９０】カード発行者（３０２）は受け取った公開鍵に対して、自らの非対称鍵秘密鍵で署名を行い、代理者認証書を代理者（９０１）に送付する（ステップ１９００２）。

【００９１】次にセキュリティドメインに格納する鍵に関連するやり取りを行う。セキュリティドメインには、運用時にはカード発行者の鍵と代理者の鍵が格納される。まず、代理者（９０１）はセキュリティドメイン用の非対称鍵のペアを作成し、その秘密鍵をカード発行者（３０２）に送付する（ステップ１９００３）。

【００９２】但し、秘密鍵の内容はカード発行者を含め他者には知られてはセキュリティ上問題があるので、秘密鍵をブラインド化し送付する。カード発行者も同様にセキュリティドメイン用の非対称鍵のペアを作成し、この秘密鍵を用いて、受け取った代理者のブラインド化した秘密鍵に署名を行い、代理者に戻す（ステップ１９００４）。

【００９３】これを受け取った代理者はブラインドを解除する。非対称鍵におけるブラインド署名の方式は既に知られている技術である。この方式自体は、例えばＲＳＡブラインド署名などが知られている。それは、例えば、岡本龍明、山本博資著『シリーズ／情報科学の数学　現代暗号』（産業図書株式会社発行、１９９７）に見られる。

【００９４】その後、カード発行者（３０２）は、提携契約に応じた内容の限定条件を設定したセキュリティドメインを作成し、自らのセキュリティドメイン用公開鍵をその内部に格納する。もしくは、あらかじめ提携契約の内容に応じて作成してあったセキュリティドメインにセキュリティドメイン用公開鍵をセットする。このセキュリティドメインにカード発行者の秘密鍵で署名を行い、代理者に限定機能付きセキュリティドメインとして送付する（ステップ１９００５）。こうして、事前提携時のやり取りは完了する。

【００９５】ここで一つ注意すべき点としては、「カード発行者の鍵」と「代理者の鍵」というそれぞれの非対称鍵の他に、「カード発行者のセキュリティドメイン用の鍵」と「代理者のセキュリティドメイン用の鍵」が存在することが挙げられる。

【００９６】次にアプリケーション搭載時のやり取りに関して説明する。アプリケーションはセキュリティドメインがカード内に設定されていることが前提なので、最初にセキュリティドメインの搭載方法を説明する。

【００９７】＜セキュリティドメインの搭載方法の例＞カードＡ（１１）へサービス提供者Ｂ（８０３）のセキュリティドメインを搭載する場合、サービス提供者Ｂのセキュリティドメインとは、事前提携時にカード発行者（３０２）から代理者（９０１）へと送付された限定機能付きセキュリティドメインのことである。代理者（９０１）は、サービス提供者からのセキュリティドメイン搭載許可要求を受ける（ステップ１９００６）。

【００９８】限定機能付きセキュリティドメインをサービス提供者Ｂに送付する（ステップ１９００７）。ただし、あらかじめセキュリティドメインを配布しておくことも可能なので、ステップ１９００６とステップ１９００７に関しては、順序が逆の場合もあり得る。代理者はステップ１９００４において受け取りブラインド解除してあった署名付き秘密鍵とそれに対応する公開鍵をセキュリティドメイン用鍵ペアとしてサービス提供者Ｂに送付する（ステップ１９００８）。

【００９９】このセキュリティドメイン用公開鍵は暗号化や署名はされておらず、サービス提供者Ｂが該セキュリティドメインと通信する上での署名検証や、セキュアな通信確保などに使用する。次にサービス提供者Ｂは限定機能付きセキュリティドメインをカードに格納する（ステップ１９００９）。こうして、ＩＣカードは、このセキュリティドメインの正当性を検証する。限定付き

Nintendo Exhibit 1003
Part 2 of 2
Page 00699

セキュリティドメインは、ステップ１９００５についての記述にもあるように、カード発行者の鍵の署名が付与されており、カード内にはカード発行時にカード発行者の公開鍵が搭載されている。従って、ＩＣカードは、この鍵を用いて当該セキュリティドメインの正当性検証が可能である。

【０１００】正当性が確認できた場合、サービス提供者Ｂよりセキュリティドメイン用秘密鍵を受け取り（ステップ１９０１０）、この鍵の正当性を検証する。ステップ１９００４についての記述にもあるように、このセキュリティドメイン用鍵とは、代理者のセキュリティドメイン用秘密鍵にカード発行者のセキュリティドメイン用秘密鍵で署名を行っているものである。カード内に設定された限定機能付きセキュリティドメインにはカード発行者のセキュリティドメイン用公開鍵が設定されているので、内部で署名の正当性を検証可能である。セキュリティドメイン用鍵の正当性が確認できた場合、鍵は復号化され代理者のセキュリティドメイン用秘密鍵が限定機能付きセキュリティドメイン内部に設定される。

【０１０１】ここまでで、限定付きセキュリティドメインは、ＩＣカードに格納され、鍵も設定されたので機能し始める。当該セキュリティドメイン内部には

（ａ）カード発行者が該セキュリティドメイン用に作成した公開鍵と

（ｂ）代理者が該セキュリティドメイン用に作成した秘密鍵

とが格納されている。前記（ａ）の項目に対応する秘密鍵はカード発行者が保持している。また、前記（ｂ）の項目に対応する公開鍵はサービス提供者Ｂに送付されている。つまり、カード発行者によって作成された限定機能付きセキュリティドメインは２つの主体者の鍵を保持し、カードＡ内で機能するのである。

【０１０２】次にカードＡ（１１）へサービス提供者Ｂ（８０３）によるアプリケーション搭載時の説明を行う。

【０１０３】まず、サービス提供者Ｂが代理者にカードＡに対するアプリケーション搭載許可要求を行う（ステップ１９０１１）。アプリケーション搭載許可要求は、具体的には、搭載するアプリケーションのハッシュ値を送付することである。

【０１０４】代理者（９０１）は、サービス提供者（８０３）と事前に申請されているアプリケーションの内容を照会しその正当性を確認する。また、カード発行者よりホットリスト（不正カード情報・ブラックリスト）などカードＡに関する情報が提供されていればそれを確認する。代理者は受け取ったアプリケーションのハッシュ値に自らの秘密鍵で代理署名し、ステップ１９００２で受け取った代理者認証書と合わせてサービス提供者Ｂに送付する（ステップ１９０１２）。なお、この秘密鍵は提携契約時にカード発行者に送付した非対称公開鍵に対応するものである。

【０１０５】サービス提供者Ｂはステップ１９０００８で取得しているセキュリティドメイン用公開鍵でアプリケーションを暗号化しておき、アプリケーション搭載許可要求の回答として送付されてくる代理署名付きアプリケーションハッシュ値、代理者認証書を、この暗号化したアプリケーション本体を合わせてカードに送信する（ステップ１９０１２）。

【０１０６】カードＡは、以上の情報を限定機能付きセキュリティドメインが受け取り、まず代理者認証書の署名検証を行う。代理者認証書は代理者の公開鍵に対して、カード発行者のセキュリティドメイン用秘密鍵で署名されたもので、限定機能付きセキュリティドメイン内部には対応する公開鍵が保持されているので、これを用いて署名検証し、即ち、復号化する。

【０１０７】ここで、復合化が可能ということは、署名が正しいことを示している。アプリケーションのハッシュ値に代理署名した鍵は、代理者の秘密鍵なので、復号化した代理者の公開鍵を用いてハッシュ値を復号化して取り出す。また一方アプリケーションには代理者のセキュリティドメイン用公開鍵で暗号化されているが、限定機能付きセキュリティドメイン内部にはこれに対応する秘密鍵が格納されているため、これを用いてアプリケーションを復号化し、ハッシュ値をとる。これが先程のハッシュ値と同一であるかを照会し、同一であれば正当なアプリケーションとしてカードＡ内にロード・インストールし、アプリケーション搭載完了となる（１９０１３）。

【０１０８】以上の手順は、アプリケーションの正当性を代理者（９０１）が保証し、代理者の正当性をカード発行者（３０２）が保証することで、間接的にカード発行者（３０２）の保証のもとにアプリケーションを搭載する仕組みとなっている。先に、セキュリティドメインに格納されている２つの鍵を、（ａ）カード発行者作成鍵、（ｂ）代理者作成鍵と記した。

【０１０９】（ａ）の鍵は代理者認証書の検証を行い、一方（ｂ）の鍵はアプリケーションの検証を行っている。

【０１１０】代理署名のみを使用した仕組みとの違いとしては、次ぎの事項が挙げられる。

（１）第１は、カード内で正当性を検証する鍵はセキュリティドメイン内部に保持されていることである。

【０１１１】（２）第２は、上記の鍵はあらかじめ該セキュリティドメイン用にカード発行者と代理者がそれぞれ作成したものであることである。

【０１１２】（３）第３は、アプリケーション搭載・削除・管理や上記鍵の使用は該セキュリティドメイン、もしくはセキュリティドメイン情報を用いたカードＯＳが行うことである。

【０１１３】前述したように、アプリケーション搭載、

Nintendo Exhibit 1003
Part 2 of 2
Page 00700

削除及び管理は、本願発明における限定機能付きセキュリティドメインによりコントロールされている。そして、それをカード発行者が、提携契約条件に従って作成していることから、無制限なアプリケーション搭載は避けることが可能である。代理者が不正に代理署名を行ったとしても、本願発明によれば、次ぎの方策が可能である。即ち、提携契約時に、「アプリケーション１つのみ搭載可能」、「ある日時にまで搭載期限」、「ある容量を超えて搭載不可能」などの適当な条件を付け、その限定機能を付けてセキュリティドメインを、作成すれば良い。カード発行者とその他プレーヤとの間は、提携契約時以降オフラインとなっていて、カード発行者の意志に反してカードにアプリケーションが送信されてきても、カード内の当該セキュリティドメインが搭載を拒否することになる。

【０１１４】以上説明した限定機能付きセキュリティドメインを用いる際の基本処理シーケンスに基づく動作の更なる動作例の詳細を、図１７から図２４を用いて説明する。即ち、各図は、各々ＩＣカード、カード発行者、代理者、サービス提供者の各プレーヤ毎の動作フローチャートである。本図面で“ＳＤ”記載されているのは本願発明で提案する限定機能付きセキュリティドメインのことであり図面の説明においてもＳＤと表す。

【０１１５】＜ＩＣカード：ＳＤ搭載処理の例＞図１７は、本発明方式を利用したＳＤ搭載処理を行うカードのフローチャートである。この処理は、図１９の１９００９及び１９０１０の両ステップに対応する。

【０１１６】ＩＣカードはサービス提供者からＳＤを受信する（ステップ２０００２）。

【０１１７】ＳＤは発行者の鍵により署名されているので、カード内の対応する鍵を用いて復号化する（ステップ２０００３）。

【０１１８】このＳＤの署名が正しければ、ＳＤをカード内にインストールし（ステップ２００５）、正しくない場合はＳＤのインストールを中止しカード外にその旨を伝える（ステップ２００１０）。

【０１１９】ＳＤをインストールした後は、ＳＤ内に設定する鍵を受信する（ステップ２０００６）。これはカード発行者のＳＤ用鍵で署名されているが、対応する鍵がＳＤ作成時に内部に設定されているので、これを用いて署名を復号化する（ステップ２０００７）。

【０１２０】正しい鍵かどうかを検証し(ステップ２０００８)、正しければＳＤ内に鍵を設定し(ステップ２０００９)、不正であれば、ＳＤへの設定を中止しカード外にその旨を伝える（ステップ２００１１）。

【０１２１】＜ＩＣカード：アプリケーション搭載処理の例＞図１８は、本願発明方式を利用したアプリケーション搭載処理を行うカードのフローチャートである。この処理は、図１６の１９０１３のステップに対応する。

【０１２２】カードはアプリケーション搭載処理を開始すると（ステップ２１００１）、代理認定証と代理者が署名したアプリケーション認証書とサービス提供者が持つＳＤ用鍵で暗号化したアプリケーションをサービス提供者より受信する（ステップ２１００２）。

【０１２３】代理認定証はカード発行者により署名されており対応する鍵がＳＤ内に格納されているので、それを使用して代理認定証を検証し、代理者の鍵を復号化する（ステップ２１００３）。

【０１２４】復号化した代理者の鍵を用いて、代理者が署名したアプリケーション認証書の検証を行い、アプリケーションのハッシュ値を取り出す（ステップ２１００４）。

【０１２５】アプリケーションはサービス提供者により暗号化されているが、対応する鍵がＳＤ内にあるのでそれを用いてアプリケーションを復号化し取り出し（ステップ２１００５）、そのハッシュ値をカード内で計算し、さきほど復号化したハッシュ値と同一かどうかを確認する(ステップ２１００６)。

【０１２６】もし、同一であればカード発行者が認証した正当なアプリケーションだと認めインストールする（ステップ２１００７）。

【０１２７】同一でない場合は、不正なアプリケーションである可能性があるのでインストール処理を中止し、その旨をカード外にエラーメッセージで伝える（ステップ２１００８）。

【０１２８】＜カード発行者：事前提携処理の例＞図１９は、カード発行者が、本発明を利用した事前提携処理を行う場合のフローチャートである。この処理は、図１６のステップ１９００１より１９００５にかけてのステップである。

【０１２９】カード発行者は、代理者を認める事前提携処理をはじめると、代理認定証を作成するため代理者から公開鍵を受け取る（ステップ２２００２）。

【０１３０】発行者は、カード内に格納してある鍵と対応する鍵で、この公開鍵に署名を行い、代理者に戻す（ステップ２２００３）。

【０１３１】また、ＳＤ格納用の鍵を代理者から受け取り（ステップ２２００４）、ＳＤ作成時に内部に設定してある鍵と対応する鍵で暗号化し代理者に送信する（ステップ２２００５）。ＳＤを発行者の鍵で署名し代理者に送信する（ステップ２２００６）。

【０１３２】＜代理者：事前提携処理の例＞図２０は本発明を利用した代理者の事前提携処理手順を示すフローチャートである。

【０１３３】まずカード発行者に代理者の公開鍵を送信し（ステップ２３００２）、カード発行者の署名が付与された代理認定証を受け取る（ステップ２３００３）。

【０１３４】ＳＤ搭載用の鍵を生成し、秘密鍵にブラインド化処理を行いカード発行者に送信する（ステップ２３００４）。

Nintendo Exhibit 1003
Part 2 of 2
Page 00701

【０１３５】発行者の署名が付与された上記鍵を受け取りブラインド化解除処理を行う（ステップ２３００５）。

【０１３６】発行者の鍵を内部に含むＳＤを発行者より受け取る（ステップ２３００６）。このＳＤはカード発行者の鍵により署名が付与されている。

【０１３７】＜代理者：ＳＤ搭載要求受付け処理の例＞図２１は本発明を利用した代理者のＳＤ搭載要求受付け処理手順を示すフローチャートである。この処理は、図１９のステップ１９００６より１９００８のステップに対応する。

【０１３８】サービス提供者からのＳＤ搭載要求を受信すると（ステップ２４００２）、発行者から受け取ってある署名付きＳＤをそのままサービス提供者に送信する（ステップ２４００３）。

【０１３９】また次に、あらかじめ作成してあったＳＤ用鍵ペアをサービス提供者に送信する（ステップ２４００４）。このＳＤ用秘密鍵は事前提携処理において発行者に署名を付与されているものである。

【０１４０】＜代理者：アプリケーション搭載要求受付け処理の例＞図２２は本発明を利用した代理者のアプリケーション搭載要求受付け処理手順を示すフローチャートである。

【０１４１】サービス提供者からのアプリケーション搭載要求を受領すると（ステップ２５００２）、事前に登録されていたアプリケーションの内容やサービス提供者の正当性を確認し、アプリケーション搭載を許可できるかどうかを判断する（ステップ２５００３）。

【０１４２】搭載を許可する場合、サービス提供者から送られてきたアプリケーションのハッシュ値などアプリケーションの特徴を示すデータを代理者が鍵で暗号化したデータとカード発行者から事前提携時に受け取っている代理認定証を合わせてアプリケーション認証書として、サービス提供者に送り返す（ステップ２５００４）。搭載不可の場合は、その旨をサービス提供者に伝える（ステップ２５００５）。

【０１４３】＜サービス提供者：ＳＤ搭載処理の例＞図２３は本発明を利用したサービス提供者のＳＤ搭載処理手順を示すフローチャートである。

【０１４４】サービス提供者は、ＳＤ搭載を代理者に要求し（ステップ２６００２）、発行者の署名が付与されているＳＤを受け取り（ステップ２６００３）、カードに送信する（ステップ２６００４）。

【０１４５】カードへのＳＤ設定ができたら、代理者からＳＤ用鍵ペアを受け取る（ステップ２６００５）。

【０１４６】カード発行者の署名が付与されているＳＤ用秘密鍵をカード内のＳＤに送信し設定する（ステップ２６００６）。

【０１４７】＜サービス提供者：アプリケーション搭載処理の例＞図２４は、本発明を利用したサービス提供者のアプリケーション搭載処理手順を示すフローチャートである。

【０１４８】サービス提供者は、代理者に対してアプリケーションのハッシュ値などを送付して搭載許可を要求し（ステップ２７００２）、代理者からアプリケーション認証書を受け取る（ステップ２７００３）。

【０１４９】アプリケーションをＳＤに設定した鍵と対応する鍵で暗号化し、アプリケーション認証書と合わせてカードに送信する（ステップ２７００４）。

【０１５０】以上が、本願発明のアプリケーション搭載方式に従いＩＣカードにアプリケーションを搭載する場合の処理手順である。

【０１５１】次に、本願発明を用いて実現可能なＩＣカードの各種利用形態や運用の諸形態について説明する。

【０１５２】図９において、本発明における提案方式では９０１の代理者が発行者（３０２）の代行業務を行うが、この代行業務を３０２とは別の発行者が行えば、カードＡ（１１）には発行者Ａ（３０２）が契約したサービス提供者だけでなく、発行者（９０１）が契約したサービス提供者のアプリケーションをも搭載することが可能となる。つまり既存の発行者―サービス提供者の仕組みを組み合わせることで本願発明を利用することが可能である。

【０１５３】また、発行者間でお互いがお互いの代理者であれば、双方のカードにどちらのサービス提供者のアプリケーションが搭載可能となる。本発明方式を利用することで、既存カード運用管理システム間の相互運用が可能となる。

【０１５４】次に本願発明のカードシステムのいくつかの具体的な応用例について説明する。

【０１５５】第１の例はテナント業務である。図２５を用いて、このテナント業務と言う本願発明の応用について説明する。本願明細書におけるテナント業務とは、ＩＣカード上の部分領域を、カード発行者以外のテナントビジネス事業者に貸与する、即ち、テナント運用する業務のことを言う。

【０１５６】図２５は、ＩＣカード上の部分領域を運用する概念を示した図である。

【０１５７】この例では、本願発明に係る代理署名及び限定機能付セキュリティドメインとを用いて、ＩＣカードに提携業務の関するアプリケーション搭載を行う。

【０１５８】図２５を参酌して本例を説明する。カード発行者（２８０１）は、ＩＣカード（１１）を発行し、ＩＣカードに係わる責任を全て負う事業者である。カード発行者（２８０１）は、サービス提供者（２８０２）やサービス提供者（２８０３）と契約を結び、ＩＣカード（１１）上へ、各々のアプリケーション搭載を許可する。更に、本例では、テナントビジネス事業者（２８０４）がテナントを希望する事業者（２８０５）に、そのアプリケーションの搭載について認証を与える。こうし

て、テナントを希望する事業者（２８０５）は、前記ＩＣカード（１１）の提携に係わるアプリケーション領域にアプリケーションを搭載する。ここで、テナントを希望する事業者は、これまでの説明におけるサービス提供者である。又、テナントをビジネスとして行おうとする事業者は、これまでの説明における代理者である。

【０１５９】ＩＣカード（１１）には、マルチアプリケーション搭載を可能とし、且つ、ダイナミックローディング可能であるカードＯＳ（２８１０）が搭載されており、カード上の機能を実行する。セキュリティドメインはカードＯＳ上に設定されており、２８１１はサービス提供者（２８０２）の、２８１２はサービス提供者（２８０３）のそれぞれセキュリティドメインである。

【０１６０】サービス提供者（２８０２）はセキュリティドメイン（２８１１）を通して自らのアプリケーション（２８１４、２８１５）を搭載する。同様にサービス提供者（２８０３）はセキュリティドメイン（２８１２）を通して自らのアプリケーション（２８１６、２８１７）を搭載する。ここで、代理者（２８０４）と提携契約を結んでいるサービス提供者（２８０５）がアプリケーションをＩＣカード（１１）に搭載する場合は、限定機能付きセキュリティドメイン（２８１３）を通してアプリケーション（２８１８、２８１９）を搭載する構成となっている。このセキュリティドメイン（２８１３）はある限定機能を付与されており、自らが搭載したアプリケーションを管理しているので、図中斜線部分の領域を考えると、セキュリティドメインにより管理されている領域であるといえる。

【０１６１】搭載許可は代理者（２８０４）が行うので、この搭載領域は、カード発行者（２８０１）より借りうけて、サービス提供者（２８０５）などにサービス提供の場を貸し出している構図となっている。これをビルの所有者からフロアを借り受けて店子を募集するテナントビジネス事業者に対応させると、新しくＩＣカード上の領域を、テナントして運用するＩＣカードの運用形態が考えられる。その場合、代理者（２８０４）は、ＩＣカードを所有するカード発行者（２８０１）からＩＣカード上のある領域を借り受け、その領域上に、サービスを提供するサービス提供者（２８０５）と契約する業務を行う仲介事業者としての機能があれば良い。

【０１６２】以上、本例はＩＣカードの一部の所定領域を、テナントビジネス事業者にその利用を貸与する利用形態である。その具体的なアプリケーションの搭載に関する代理者、サービス提供者の動作については、これまで説明してきた本願発明の一般的な運用と同様である。

【０１６３】また、図２６ではカード発行者（２９０１）が契約関係にあるサービス提供者に限定機能付きセキュリティドメインを対応させた場合の利用形態について説明する。図２５の例との違いは、カード発行者とサービス提供者（２９０５）との間に代理者（２８０４）をおかず、カード発行者（２９０１）が、直接契約しているサービス提供者（２９０５）であることである。その他の点は前述の例と同様である。

【０１６４】この利用形態の利点は、アプリケーション搭載のオフライン認証におけるセキュリティを高くする点である。サービス提供者がアプリケーションを搭載する際に、カード発行者へ認証を要求するのが従来一般的であることは、既に述べた。この場合、要求方法として、ネットワークなどを使用して通信を確立しオンラインで認証してもらう場合と、あらかじめ許可証を大量に発行してもらうオフライン認証がある。オフライン認証の場合、カード発行者のＩＣカードへのコントロールが弱くなる点が問題となっている。しかし、カード上の限定機能付きセキュリティドメインに制限を設定しておくことで、この問題は、コントロール可能となる。また、契約しているサービス提供者間で差異をつけることが可能となり、事業者の信頼度に応じた契約締結ができる。

【０１６５】次に、ＩＣカードにおけるバーチャルカードという本願発明のＩＣカードの応用を提案する。

【０１６６】図２７は、バーチャルカードの概念を説明する図である。バーチャルカードとは、このカードを統括する第３者の認証の下に、複数のカード発行者によるカードの機能に応ずる各々のアプリケションを、一つのカードの所定のアプリケーション領域に搭載したカードの用い方である。従って、一つのカードによって、あたかも複数のカードを持っているかのようにカードを運用することが出来る。

【０１６７】図２７において、カードＡ（３００５）、カードＢ（３００６）、カードＣ（３００７）は、個別のＩＣカードであり、それぞれのカード発行者は、第３者（３００１）と提携契約を結んでいる。以下、この第３者（３００１）を、便宜上バーチャルカード発行者と呼ぶ。バーチャルカード発行者（３００１）は、カードＡ上にアプリケーション搭載領域を借り受けているため、バーチャルカード発行者（３００１）と契約の有るサービス提供者（３００２、３００３、３００４）は搭載条件さえクリアすれば、カードＡ上にアプリケーション搭載可能である。バーチャルカード発行者（３００１）は契約さえ有れば、同様に、カードＢにもカードＣにもアプリケーション搭載可能である。つまり、バーチャルカード発行者（３００１）は、実際のカード発行業務を行っていないにも関わらず、バーチャルカード（３００８）のような概念上のカードを発行しており、バーチャルカード発行者（３００１）と契約すればアプリケーション搭載可能なカードと見なすことができる。サービス提供者側から見ると、複数のカードにアプリケーションを搭載するために、１つのカード発行者のみと契約を結び通信を確立すれば良いので大幅手間が省けコスト削減につながる。

【０１６８】以上、本願諸発明を説明したが、その要点

Nintendo Exhibit 1003
Part 2 of 2
Page 00703

をまとめると次ぎの通りである。

【０１６９】本願発明は、ＩＣカードのカード発行者の許可を直接得ることなく、しかしながら同程度の高セキュリティを保持しながらアプリケーションを搭載することを可能とする。将来、ＩＣカードが普及し、利用数が増加することに伴い発生するであろう下記の諸難点に関しては、カード発行者同士の提携契約とそれに伴う情報のやり取りにより、提携先サービス提供者とカード発行者が直接通信を確立する必要がなくなったため解決される。こうして、本願発明によって、ＩＣカードへの柔軟なサービス提供が可能となった。尚、上記難点とは、第１がカード発行者―サービス提供者間に起こる膨大な量の契約と通信の問題であり、第２が現実的に契約関係を結べない事業者同士の問題である。

【０１７０】また、解決方法の一つである代理署名を用いることにより付随して起こる難点として、第１に一度代理者認証書を発行してしまうと無制限に使用される危険性、第２に搭載回数、搭載領域など条件を付けた提携契約を結ぶ必要性を挙げることが出来る。この難点は、限定機能付きのセキュリティドメインを用いることで回避することが出来る。

【０１７１】さらに、本願発明のアプリケーションの搭載方式を利用することで、各種の新しいＩＣカードの利用形態を可能とした。

【０１７２】こうした新しいＩＣカードの利用形態の例を列挙すれ、下記の通りである。（１）既存システムへの応用方式、（２）発行者間で相互に代行業務を行うことによる相互運用方式、（３）ＩＣカード上の領域を借り受け運用するテナント事業、（４）カード発行者によるサービス提供者間のアプリケーション搭載における差異の設定、（５）バーチャルカード発行事業。

【０１７３】本願は、その形態が多岐に渡るので、以下に主な形態、あるいは概念を整理し、列挙する。

【０１７４】その第１は、非対称暗号化機能を有するＩＣカードを利用した署名生成及び検証方式において、ＩＣカード発行者がＩＣカードにみずからの公開鍵を格納し利用者に配布するステップと、ＩＣカード発行者が第３者（以降代理者と呼ぶ）の公開鍵を受け取り、カード発行者の秘密鍵で署名を付与し代理者に戻すステップと、代理者が自らの秘密鍵でデータに署名を付与するステップと、代理者が前記カード発行者より受け取ったカード発行者署名付き代理者公開鍵とそのデータを合わせてカードに送信するステップと、カード内のカード発行者公開鍵で前記付与したカード発行者署名付き代理者公開鍵を検証するステップと、この代理者公開鍵でデータの署名を検証し正当性を確認するステップを有することを特徴とする署名生成及び検証方式である。

【０１７５】第２は、ＩＣカード発行者がＩＣカードに自らの公開鍵を格納し利用者に配布するステップと、発行者が代理者の公開鍵を受け取り、自らの秘密鍵で署名し代理認定証として渡すというステップと、アプリケーションをカードに発行する業務を行う事業主体者（以降サービス提供者と呼ぶ）がアプリケーションのハッシュ値などアプリケーションの特徴を示すデータを代理者に渡すというステップと、代理者は渡されたアプリケーションのハッシュ値などに代理者の秘密鍵で署名を行うステップと、上記署名付きハッシュ値と発行者より渡されている代理認定証を合わせてアプリケーション認証書としてサービス提供者に戻すというステップと、サービス提供者がアプリケーションとアプリケーション認証書を合わせてカードに搭載するというステップと、カード内の発行者公開鍵により代理認定証を検証するステップと、上記検証された代理者の公開鍵によりカード内でアプリケーションのハッシュ値を検証するステップと、カード内でアプリケーションのハッシュ値を計算するステップと、検証されたハッシュ値と計算されたハッシュ値を比較することでアプリケーションの正当性を確認するステップを有することを特徴とするアプリケーション搭載方式である。

【０１７６】第３は、ＩＣカード内でアプリケーションの搭載・削除・管理機能を行うアプリケーションもしくはデータ（以降セキュリティドメインと呼ぶ）であって、カード発行者により作成され、アプリケーション搭載に関する所定の制限がカード発行者により設定可能であり、内部に発行者の公開鍵が格納されている状態で代理者に渡され、代理者によりサービス提供者に渡され、サービス提供者によりカード内に搭載され、その後サービス提供者により当該セキュリティドメイン内に代理者の秘密鍵を設定され、アプリケーション搭載時に当該セキュリティドメイン内のカード発行者の公開鍵により代理者の公開鍵の署名を検証し、上記代理者の公開鍵を用いてアプリケーションのハッシュ値の署名を検証しアプリケーションの正当性を確認することを特徴とするＩＣカード内でアプリケーションの搭載・削除・管理機能を行うセキュリティドメインである。

【０１７７】第４は、前記署名生成及び検証方式と、前記セキュリティドメインを利用したＩＣカードへのアプリケーション搭載方式である。即ち、発行者が代理者に代理認定証と所定の制限を設定したセキュリティドメインを代理者に渡すステップと、代理者がサービス提供者に、上記セキュリティドメインとセキュリティドメイン用の鍵を渡すステップと、サービス提供者が、該セキュリティドメインをカードに格納し、鍵を設定するステップと、サービス提供者が、代理者にアプリケーション認証書を要求し受け取るステップと、サービス提供者が上記アプリケーション認証書をアプリケーションを合わせてカードに搭載するステップと、カード内でアプリケーション認証書を検証しアプリケーションの正当性を確認するステップを有することを特徴とするアプリケーション搭載方式である。

Nintendo Exhibit 1003
Part 2 of 2
Page 00704

【０１７８】第５は、セキュリティドメインをサービス提供者から受け取り格納する手段と、サービス提供者により該セキュリティドメイン内に鍵を設定する手段と、アプリケーション搭載時にアプリケーションに付与されたアプリケーション認証書を検証する手段と、アプリケーションをインストールする手段を有することを特徴とするアプリケーション搭載可能のＩＣカードである。

【０１７９】第６は、自ら以外の第３者である代理者から渡された公開鍵を受け取り、自らの秘密鍵で署名を付与し代理認定証として代理者に戻し、セキュリティドメインを作成・条件設定した上で代理者に渡すことを特徴とするＩＣカード発行者である。

【０１８０】第７は、ＩＣカードの発行管理業務の一部を代行する事業者（代理者）であり、自らの公開鍵を発行者に渡し、発行者の秘密鍵により署名をされた代理認定証を受け取り、発行者からセキュリティドメインを受け取り、サービス提供者に該セキュリティドメインとセキュリティドメイン用鍵を渡し、サービス提供者からアプリケーションのハッシュ値などアプリケーションの特徴を示すデータを受け取り、それに自らの秘密鍵で署名を付与しそれと代理認定証を合わせてアプリケーション認証書としてサービス提供者に戻すことを特徴とする事業主体者である。

【０１８１】第８は、代理者からセキュリティドメインとセキュリティドメイン用鍵を受け取り、カード内にセキュリティドメインを格納し、鍵を設定し、カードに搭載するアプリケーションのハッシュ値などアプリケーションの特徴を示すデータを、該カードの発行者の代行業務を行う代理者に送信し、代理者からの代理認定証と代理者の秘密鍵で署名を付与されたアプリケーションのハッシュ値などをアプリケーション認証書として受け取り、アプリケーション認証書とアプリケーションを合わせてカードに送信することを特徴とするサービス提供者である。

【０１８２】第９は、前記事業主体者が同時にカード発行管理業務も行い、自らが発行するカードへのアプリケーション認証書の他に、代理認定証を渡されている発行者の発行したカードへのアプリケーション認証書を発行することが可能であることを特徴とするカード発行者間のカード相互運用方式である。

【０１８３】第１０は、カードを発行管理する発行者同士が双方とも前記代理者であることにより、発行者は双方の発行するカードに搭載するアプリケーションの認証書を発行することが可能であることを特徴とするカード発行者間のカード相互運用方式である。

【０１８４】第１１は、ＩＣカード上の部分領域の運用を行い、カード発行者に対してはＩＣカード上の部分領域を借り受けていることへの対価を支払い、サービス提供者に対してはアプリケーション搭載に対する課金や問い合わせなどへのサポート業務を行うことの対価を要求することを特徴とする業務である。

【０１８５】第１２は、セキュリティドメインの機能を使用したＩＣカードへのアプリケーション搭載方式であって、サービス提供者は該セキュリティドメインを通してのみアプリケーション搭載可能であり、該セキュリティドメインに設定される条件はカード発行者がサービス提供者との契約内に応じて設定し、同一カード上で、サービス提供者毎の条件よりアプリケーション搭載条件が異なることを特徴とするアプリケーション搭載方式である。

【０１８６】第１３は、代理者が複数のＩＣカード発行者の代理者であり、サービス提供者からのアプリケーション搭載要求を代理者が受け取り、上記複数カード発行者からの代理認定証を複数付与したアプリケーション認証書を発行し、サービス提供者が該アプリケーション認証書を受け取ることで複数カードへのアプリケーション搭載が可能となることを特徴としているＩＣカード発行管理代行業務である。

【０１８７】第１４は、当該ＩＣカードを発行したＩＣカード発行者が署名した代理者の公開鍵と、代理者が署名したアプリケーションとを受け取る手段と、代理者の公開鍵に付与されたＩＣカード発行者の署名を検証する手段と、上記代理者の公開鍵によりアプリケーションの署名を検証する手段と、を有するアプリケーション搭載可能なＩＣカードである。

【０１８８】第１５は、ＩＣカードの発行管理業務を行うＩＣカード発行者であり、自ら以外の第３者である代理者から渡された公開鍵を受け取り、自らの秘密鍵で署名を付与し、これを代理認定証として代理者に戻すＩＣカード発行者である。

【０１８９】第１６は、ＩＣカードの発行管理業務の一部を代行する事業者（代理者）であり、自らの公開鍵を発行者に渡し、発行者の秘密鍵により署名をされた代理認定証を受け取り、サービス提供者からアプリケーションの、例えば、ハッシュ値などのアプリケーションの特徴を示すデータを受け取り、それに自らの秘密鍵で署名を付与し、それと代理認定証を合わせてアプリケーション認定証としてサービス提供者に戻すことを特徴とするアプリケーション搭載を実施する事業主体者である。

【０１９０】第１７は、アプリケーションをカードに発行する業務を行う事業主体者（サービス提供者）であり、ＩＣカードに搭載するアプリケーションのハッシュ値などアプリケーションの特徴を示すデータを、前記ＩＣカードの発行者の代行業務を行う代理者に送付し、代理者からの代理認定証と代理者の秘密鍵で署名を付与されたアプリケーションのハッシュ値などをアプリケーション認定書として受け取り、アプリケーション認定書とアプリケーションを合わせてカードに送信するアプリケーション搭載を実施するサービス提供者である。

【０１９１】

Nintendo Exhibit 1003
Part 2 of 2
Page 00705

【発明の効果】本願発明の第１の形態は、ＩＣカードのカード発行者の許可を直接得ることなく、カード発行者の許可を得る場合と同程度の高セキュリティを保持しながら、当該ＩＣカードにアプリケーションを搭載することを可能とする。加えて、本願発明の第１の形態は膨大なの契約と通信の量の増大に基づく、現実的各種処理の煩雑さを解消する。

【０１９２】本願発明の第２の形態は、ＩＣカードのカード発行者の許可を直接得ることなく、ＩＣカードにアプリケーションを搭載する場合に付随して生ずるアプリケーションの搭載の無制限性の難点を回避する。

【図面の簡単な説明】

【図１】図１はＩＣカードの基本構成を示す図である。

【図２】図２はＩＣカードの発行、サービス提供に関するこれまでのシステム構成の例である。

【図３】図３はセキュリティドメイン機構を持つＩＣカードの基本構成を示す図である。

【図４】図４は従来のアプリケーション搭載の為のシーケンスの例を示す図である。

【図５】図５は従来方式における難点を説明する為の図である。

【図６】図６は本願発明におけるＩＣカード、発行者、代理者、及びサービス提供者の関係を示す図である。

【図７】図７は本願発明のシステムの基本的な構成を示す図である。

【図８】図８は代理署名を行う場合のＩＣカード、発行者、代理者、及びサービス提供者の関係を示す図である。

【図９】図９は代理署名による提携を実現するシーケンスを示す図である。

【図１０】図１０は、ＩＣカードにアプリケーションを搭載する為の本願発明に係わるシーケンスを示す図である。

【図１１】図１１は、カード発行者が事前提携を行う為の本願発明に係わるシーケンスを示す図である。

【図１２】図１２は代理者が事前提携を行う為の本願発明に係わるシーケンスを示す図である。

【図１３】図１３は代理者がアプリケーションの搭載を行う為の本願発明に係わるシーケンスを示す図である。

【図１４】図１４はサービス提供者がアプリケーションの搭載を行う為の本願発明に係わるシーケンスを示す図である。

【図１５】図１５はセキュリティドメインを有するＩＣカードの基本構成を示す図である。

【図１６】図１６は代理署名及びセキュリティドメインを用いた提携を実現する為の本願発明のシーケンスを示す図である。

【図１７】図１７はＩＣカードにＳＤを搭載する為の本願発明のシーケンスを示す図である。

【図１８】図１８はＩＣカードにアプリケーションを搭載する為の本願発明のシーケンスを示す図である。

【図１９】図１９はカード発行者が事前提携を行う為の本願発明に係わるシーケンスを示す図である。

【図２０】図２０は代理者が事前提携を行う為の本願発明に係わるシーケンスを示す図である。

【図２１】図２１は代理者がＳＤを搭載する為のシーケンスを示す図である。

【図２２】図２２は代理者がアプリケーションの搭載を行う為の本願発明に係わるシーケンスを示す図である。

【図２３】図２３はサービス提供者がＳＤの搭載を行う為の本願発明に係わるシーケンスを示す図である。

【図２４】図２４はサービス提供者がアプリケーションの搭載を行う為の本願発明に係わるシーケンスを示す図である。

【図２５】図２５は本願発明を用いたテナント例を示す図である。

【図２６】図２６は本願発明を用いた制限付きサービスの提供契約を行う例を説明する図である。

【図２７】図２７は本願発明を用いたバーチャルカードの例を説明する図である。

【図２８】図２８はカード・システムの概要を示す図である。

【符号の説明】

１１：ＩＣカード、１０１：ＩＣカード内ハードウェア層

１０２：ＩＣカード内ＯＳ層、１０６：ＩＣカード内アプリケーション

１０７：ＩＣカード内アプリケーション層、１０８：セキュリティドメイン、３０２：カード発行者サーバ、３０３：サービス提供者サーバ

３０４：クライアント側用外部端末、３０５：カード発行者データベース、３０６：サービス提供者データベース、８０１、８０２：アプリケーション、８０４、８０５：セキュリティドメイン、８０３：サービス提供者、９０１：代理者、９０２：代理者のデータベース、２８０１：カード発行者、２８０２、２８０３：サービス提供者、２８０４：代理者＝テナントビジネス事業者、２８０５：サービス提供者＝テナント希望事業者、２８１０：カードＯＳ、２８１１、２８１２：セキュリティドメイン、２８１３：限定機能付きセキュリティドメイン

２８１４～２８１９：アプリケーション、２９０１：カード発行者

２９０２、２９０３、２９０５：サービス提供者

２９０４、２９１０：カードＯＳ

２９１１、２９１２：セキュリティドメイン

２９１３：限定機能付きセキュリティドメイン

２９１４～２９１９：アプリケーション

３００１：バーチャルカード発行者

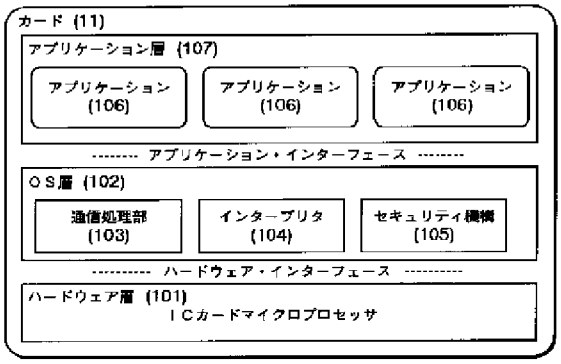
３００２～３００４：サービス提供者、３００５～３００７：ＩＣカード、３０

Nintendo Exhibit 1003
Part 2 of 2
Page 00706

０９～３０１１：提携先の搭載領域。

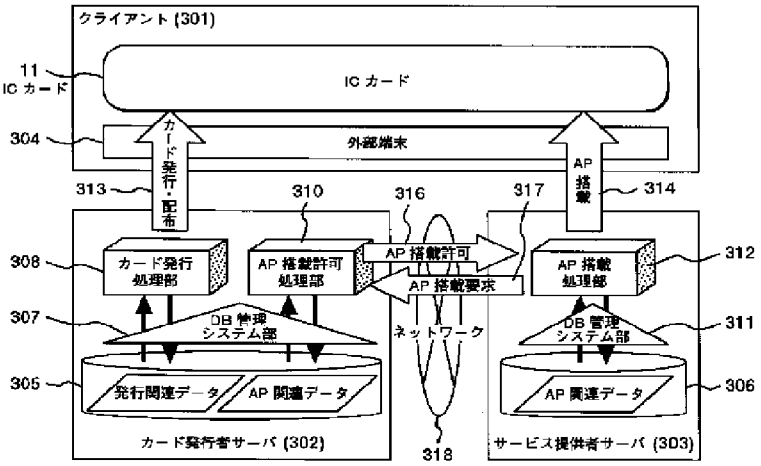
【図１】

図　1



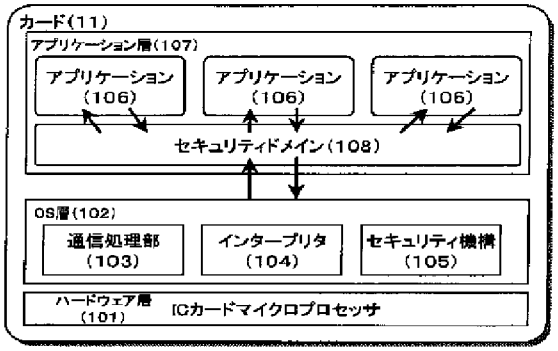
【図２】

図　2



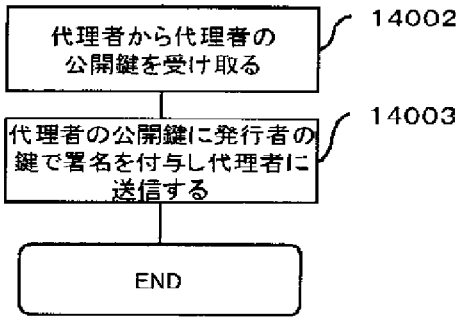
【図３】

図3



【図１１】

図 11



【図４】

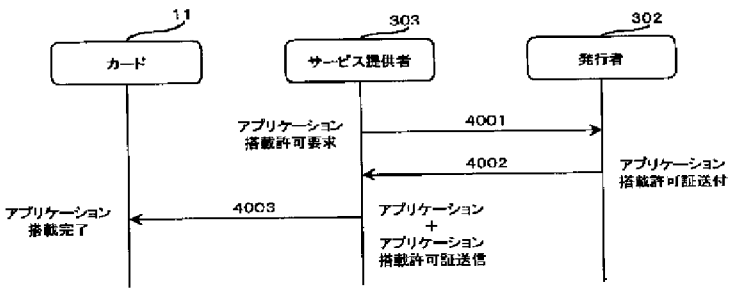
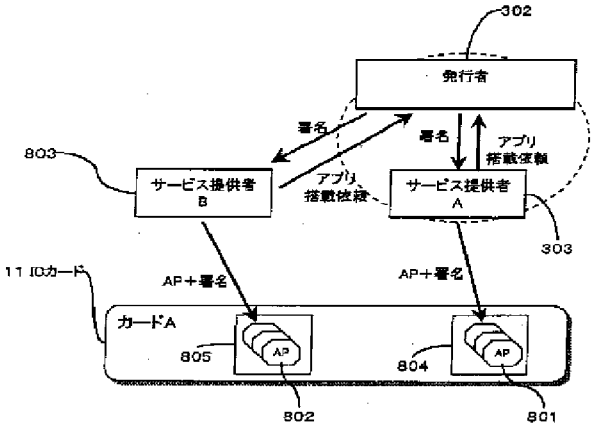


図4

Nintendo Exhibit 1003
Part 2 of 2
Page 00707

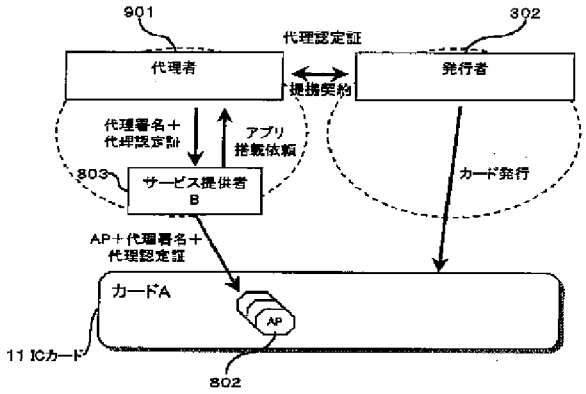
【図５】

図 5



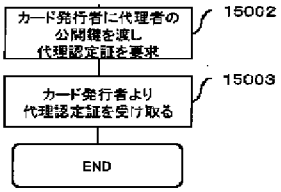
【図８】

図 8



【図１２】

図 12



【図６】

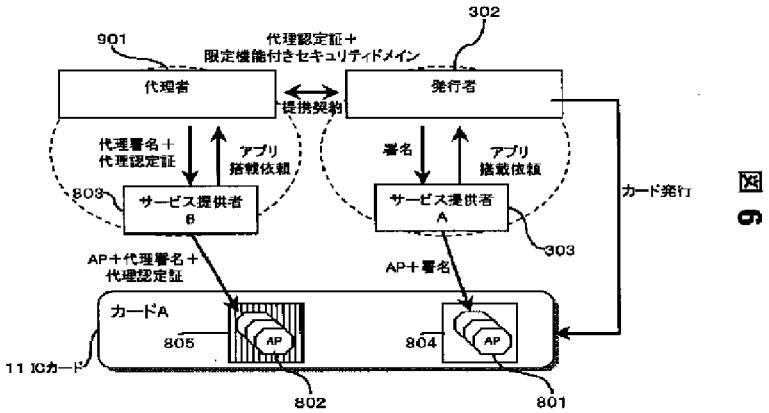
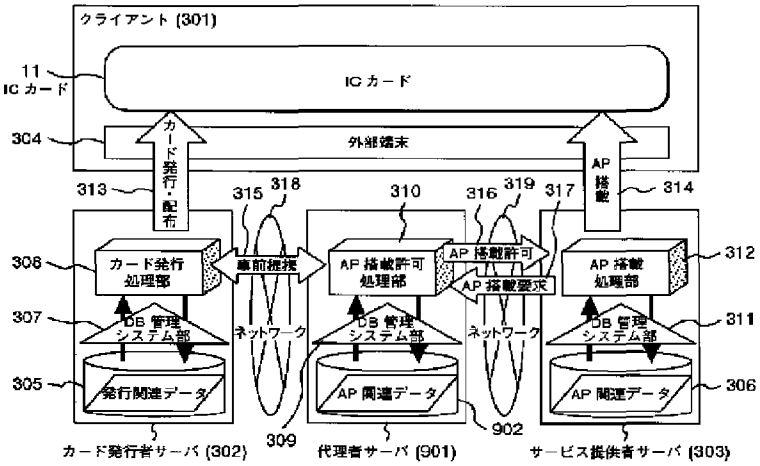


図 6

【図７】

図　7


クライアント (301)
11
IC カード
IC カード
304
外部端末
カード発行・配布
AP 搭載
313
315
318
310
316
319
317
314
308
カード発行処理部
事前提携
AP 搭載許可処理部
AP 搭載許可
AP 搭載要求
AP 搭載処理部
312
307
DB 管理システム部
ネットワーク
DB 管理システム部
ネットワーク
DB 管理システム部
311
305
発行関連データ
AP 関連データ
AP 関連データ
306
309
902
カード発行者サーバ (302)
代理者サーバ (901)
サービス提供者サーバ (303)

【図９】

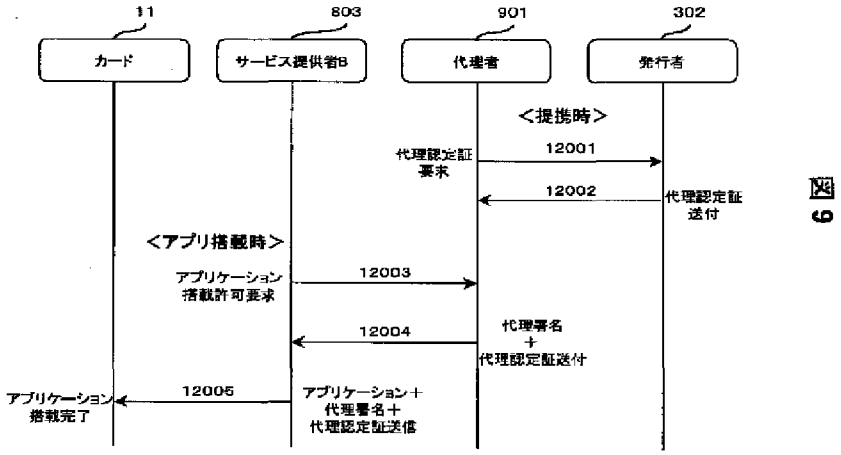
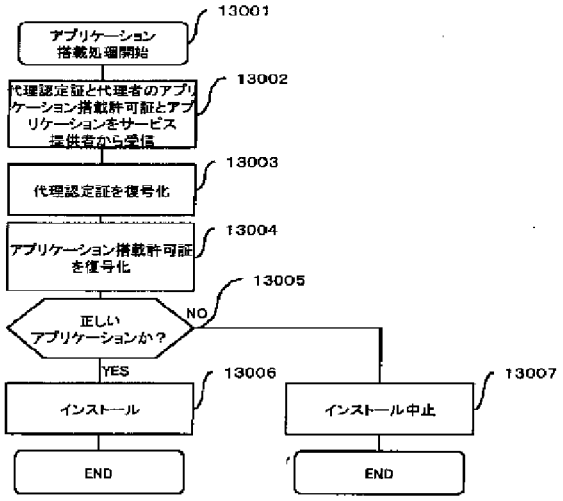

11
803
901
302
カード
サービス提供者B
代理者
発行者
<提携時>
代理認定証要求
12001
12002
代理認定証送付
<アプリ搭載時>
アプリケーション搭載許可要求
12003
12004
代理署名＋代理認定証送付
12005
アプリケーション搭載完了
アプリケーション＋代理署名＋代理認定証送信

図9

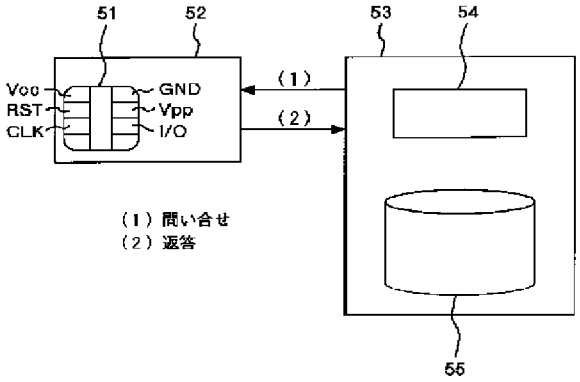
【図１０】

図 10



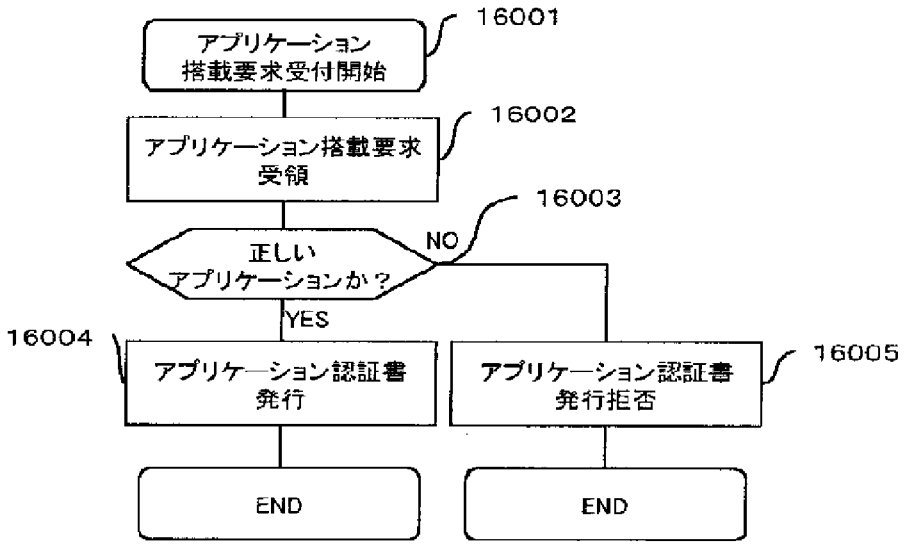
【図２８】

図　２８



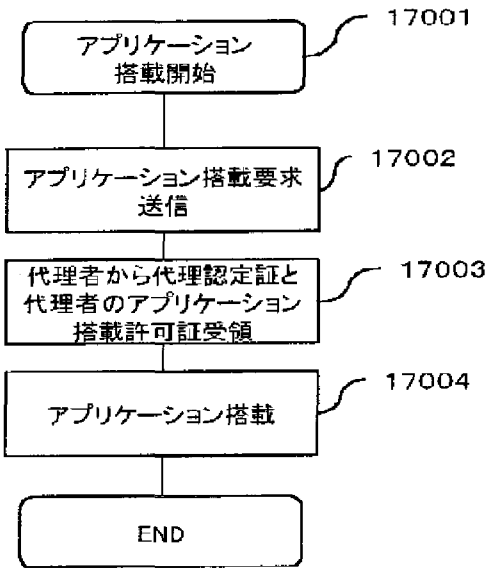
【図１３】

図 13



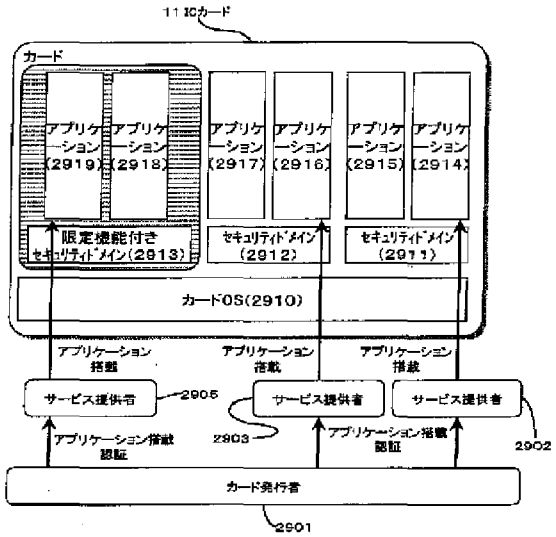
【図１４】

図 14



【図２６】

図 26



【図１５】

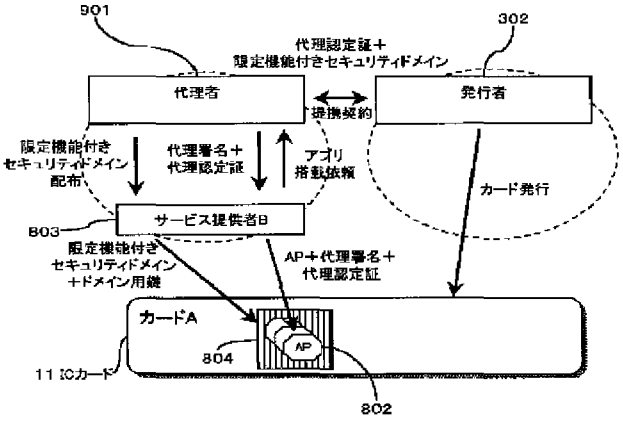
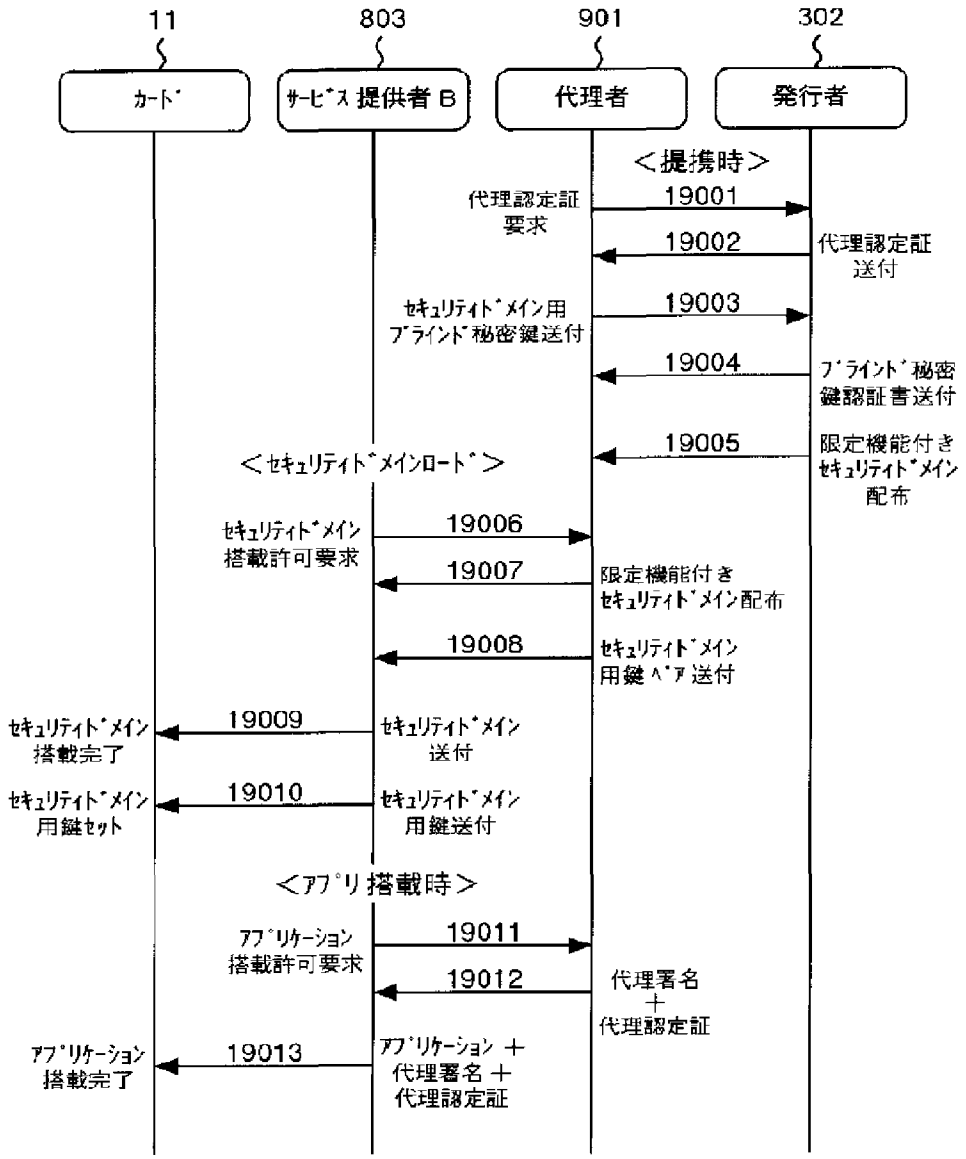


図 15

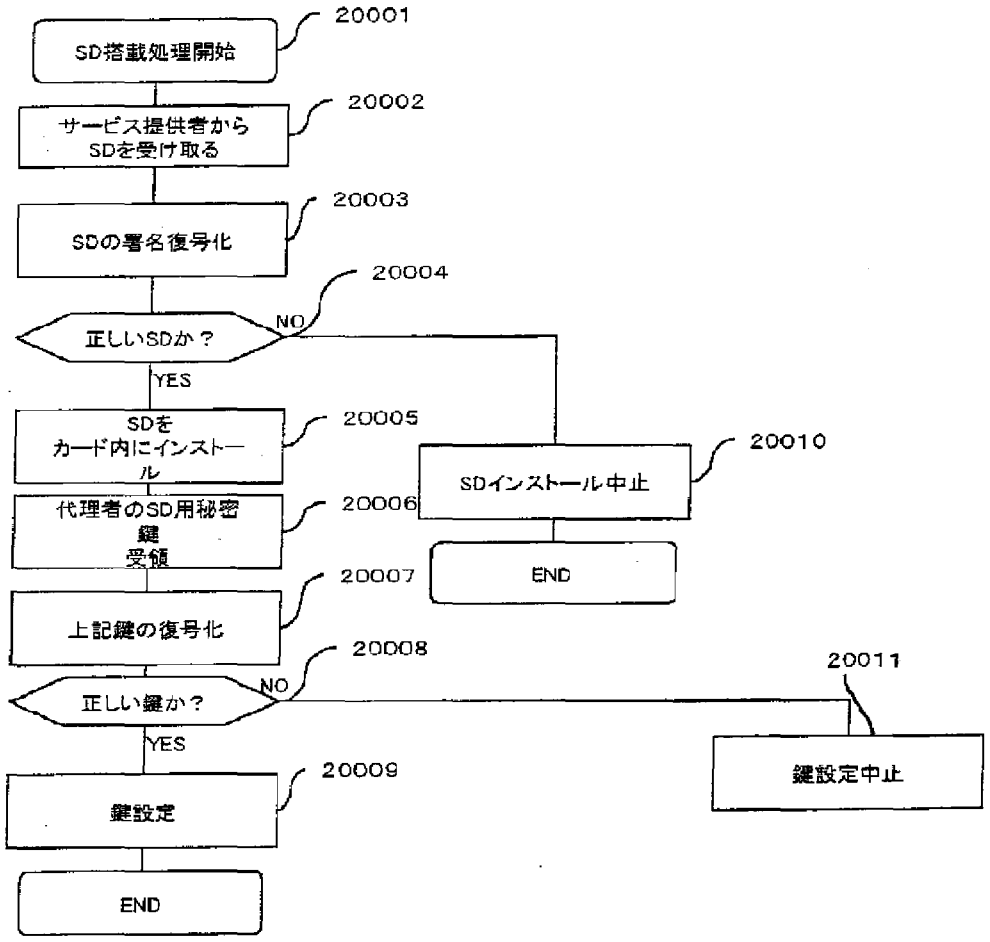
【図１６】

図　１６


11
803
901
302
ｶｰﾄﾞ
ｻｰﾋﾞｽ提供者Ｂ
代理者
発行者
＜提携時＞
代理認定証
要求
19001
19002
代理認定証
送付
ｾｷｭﾘﾃｨﾄﾞﾒｲﾝ用
ﾌﾟﾗｲﾍﾞｰﾄ秘密鍵送付
19003
19004
ﾌﾟﾗｲﾍﾞｰﾄ秘密
鍵認証書送付
19005
限定機能付き
ｾｷｭﾘﾃｨﾄﾞﾒｲﾝ
配布
＜ｾｷｭﾘﾃｨﾄﾞﾒｲﾝﾛｰﾄﾞ＞
ｾｷｭﾘﾃｨﾄﾞﾒｲﾝ
搭載許可要求
19006
19007
限定機能付き
ｾｷｭﾘﾃｨﾄﾞﾒｲﾝ配布
19008
ｾｷｭﾘﾃｨﾄﾞﾒｲﾝ
用鍵ﾍﾟｱ送付
ｾｷｭﾘﾃｨﾄﾞﾒｲﾝ
搭載完了
19009
ｾｷｭﾘﾃｨﾄﾞﾒｲﾝ
送付
ｾｷｭﾘﾃｨﾄﾞﾒｲﾝ
用鍵ｾｯﾄ
19010
ｾｷｭﾘﾃｨﾄﾞﾒｲﾝ
用鍵送付
＜ｱﾌﾟﾘ搭載時＞
ｱﾌﾟﾘｹｰｼｮﾝ
搭載許可要求
19011
19012
代理署名
＋
代理認定証
ｱﾌﾟﾘｹｰｼｮﾝ
搭載完了
19013
ｱﾌﾟﾘｹｰｼｮﾝ ＋
代理署名 ＋
代理認定証

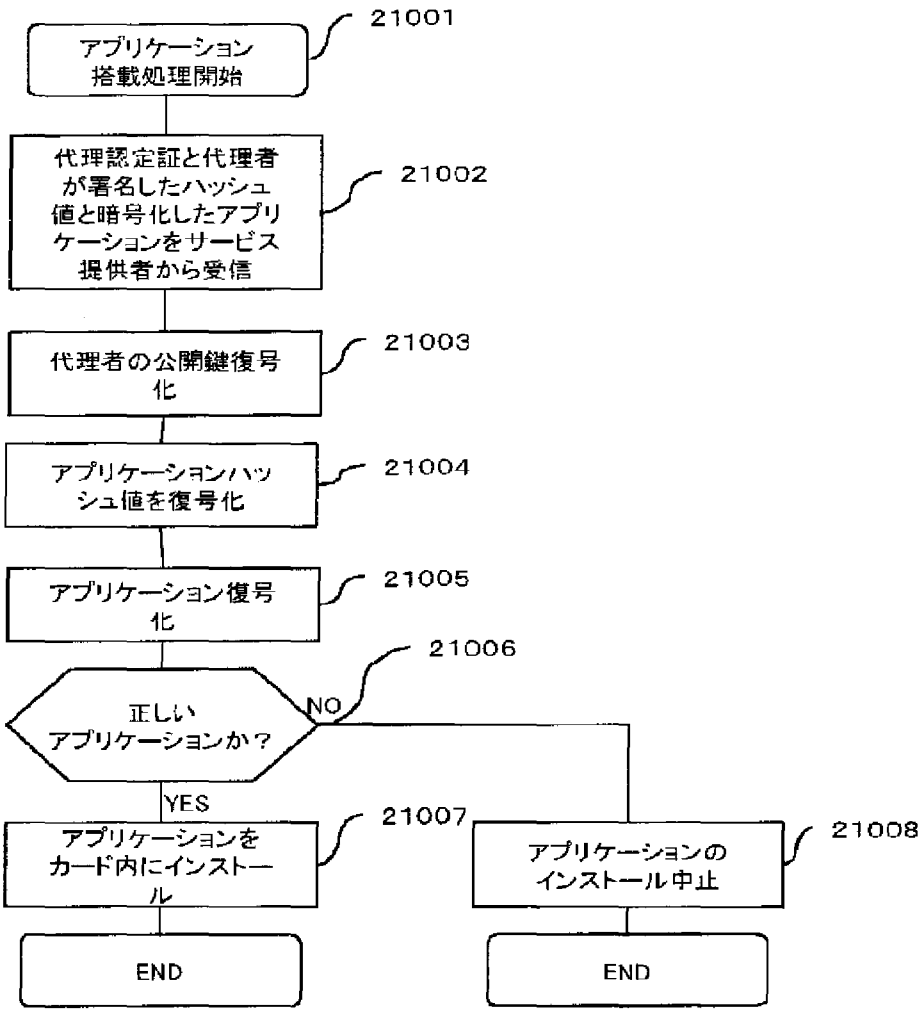
【図１７】

図 17


SD搭載処理開始
20001
サービス提供者から
SDを受け取る
20002
SDの署名復号化
20003
正しいSDか？
20004
NO
YES
SDを
カード内にインストール
20005
SDインストール中止
20010
END
代理者のSD用秘密鍵
受領
20006
上記鍵の復号化
20007
正しい鍵か？
20008
NO
YES
20011
鍵設定中止
鍵設定
20009
END

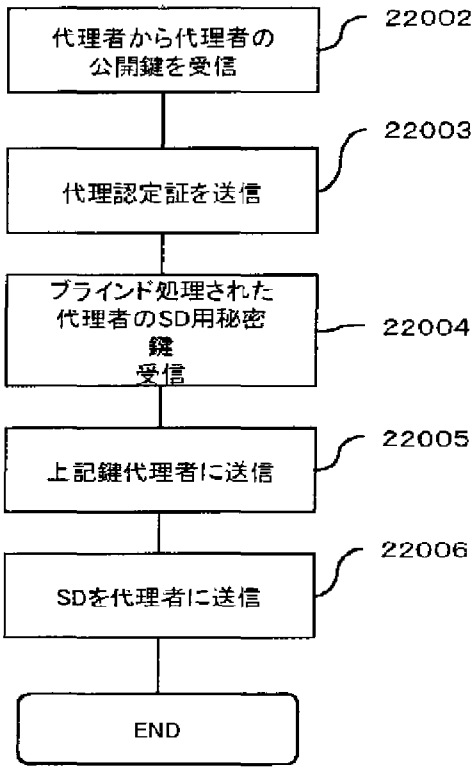
【図１８】

図 18


アプリケーション搭載処理開始
21001
代理認定証と代理者が署名したハッシュ値と暗号化したアプリケーションをサービス提供者から受信
21002
代理者の公開鍵復号化
21003
アプリケーションハッシュ値を復号化
21004
アプリケーション復号化
21005
正しいアプリケーションか？
21006
NO
YES
アプリケーションをカード内にインストール
21007
アプリケーションのインストール中止
21008
END
END

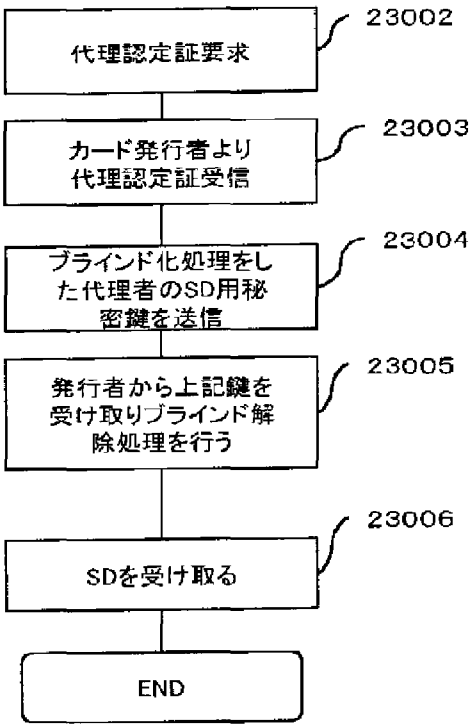
【図１９】

図 19



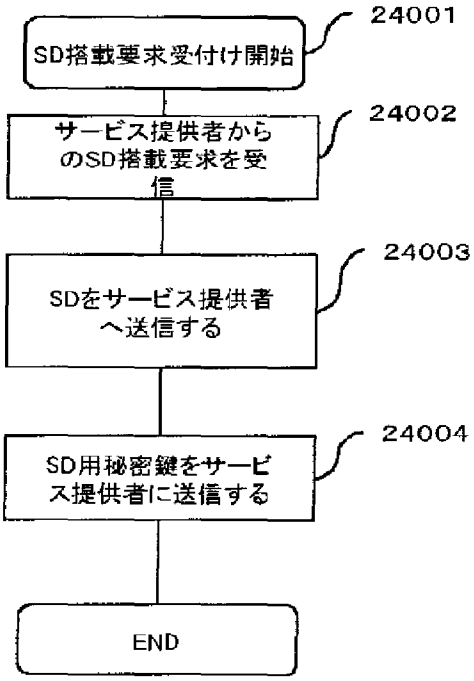
【図２０】

図 20



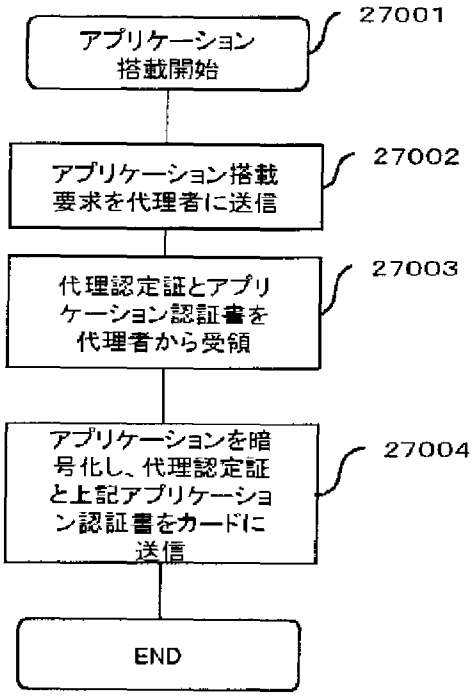
【図２１】

図 21



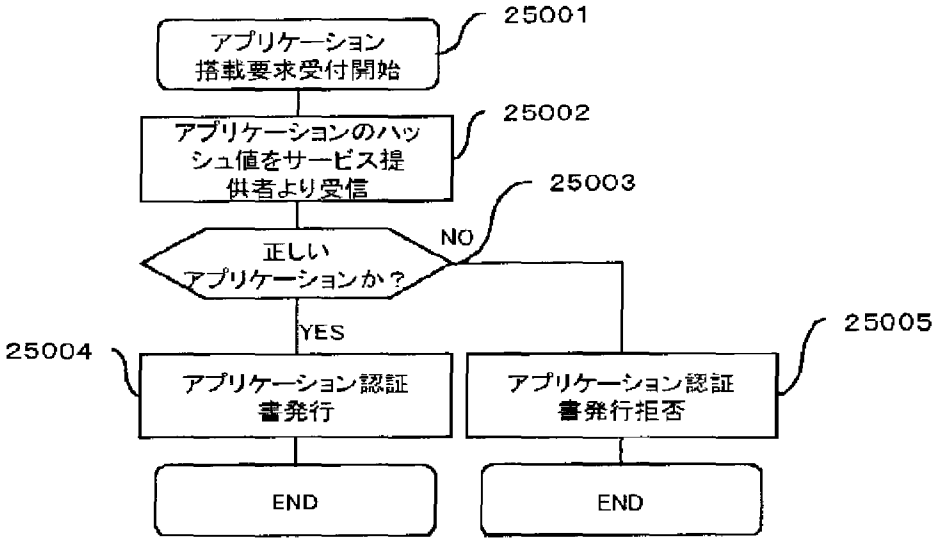
【図２４】

図 24



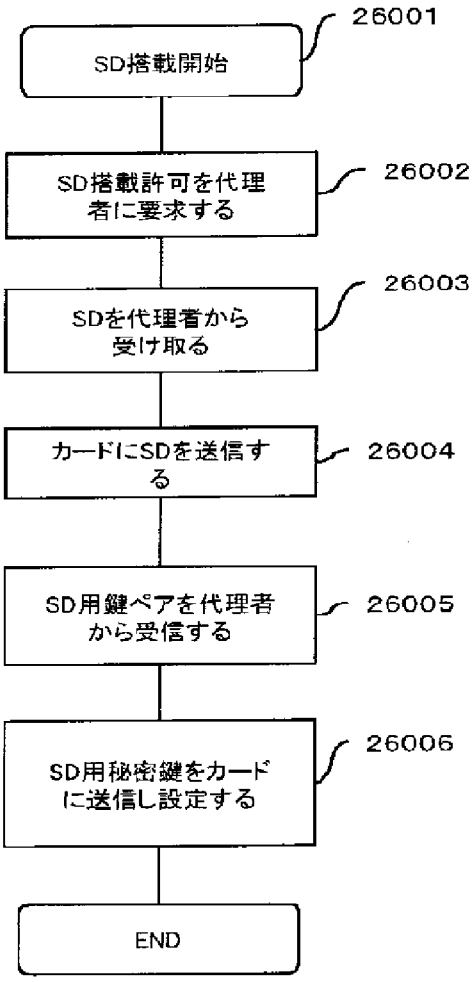
【図２２】

図 22



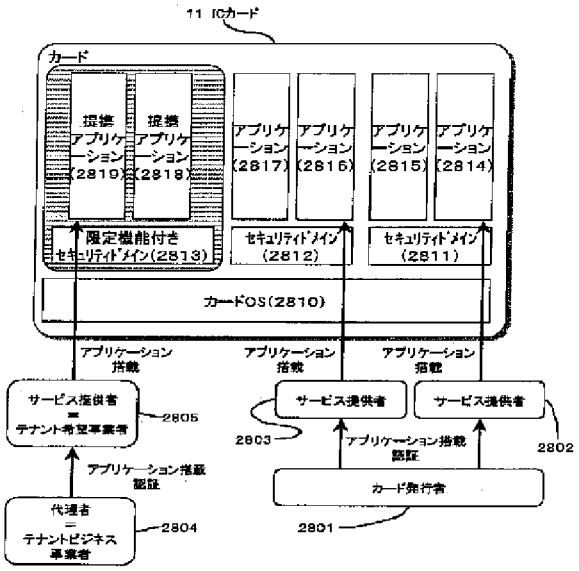
【図２３】

図 23



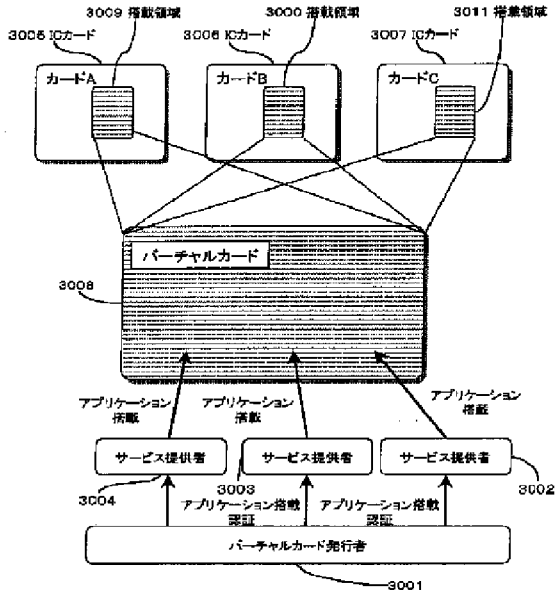
【図２５】

図 25



【図２７】

図 27



フロントページの続き

(51)Int.Cl.[7]	識別記号	FI	テーマコード（参考）
Ｇ０９Ｃ　1/00	６４０	Ｇ０６Ｆ　9/06	６６０Ａ
	６６０	Ｇ０６Ｋ　19/00	Ｎ

(72)発明者　大木　優
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内
(72)発明者　馬場　里美
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内
(72)発明者　松井　隆
神奈川県川崎市幸区鹿島田890番地　株式会社日立製作所内ビジネスソリューション開発本部内

Ｆターム(参考)　2C005 NA02 NA06 SA02 SA05 SA08 SA23
5B035 AA13 BB09 CA11 CA38
5B058 CA25 KA11 KA33
5B076 FA13 FB02
5J104 AA09 LA03 LA06 NA02 NA12 NA35 NA37 NA40

Nintendo Exhibit 1003
Part 2 of 2
Page 00718

(12)特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2003 年7 月17 日 (17.07.2003)

PCT

(10) 国際公開番号
WO 03/058411 A1

(51) 国際特許分類[7]: G06F 1/00, 9/06

(21) 国際出願番号: PCT/JP03/00035

(22) 国際出願日: 2003 年1 月7 日 (07.01.2003)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願2002-1843 2002 年1 月8 日 (08.01.2002) JP

(71) 出願人 (米国を除く全ての指定国について): 株式会社エヌ・ティ・ティ・ドコモ (NTT DOCOMO, INC.) [JP/JP]; 〒100-6150 東京都 千代田区 永田町二丁目11番1号 Tokyo (JP).

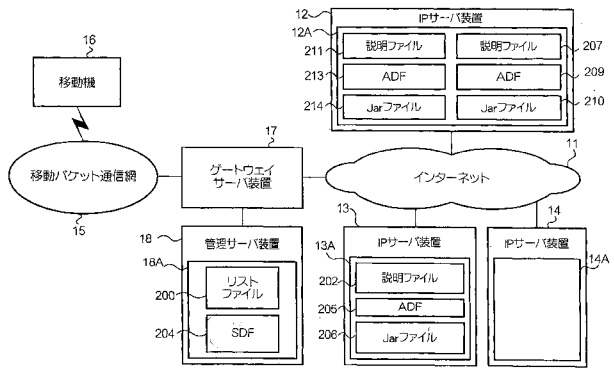
(72) 発明者; および

(75) 発明者/出願人 (米国についてのみ): 山田 和宏 (YAMADA,Kazuhiro) [JP/JP]; 〒100-6150 東京都 千代田区 永田町二丁目１１番１号 山王パークタワー 株式会社エヌ・ティ・ティ・ドコモ 知的財産部内 Tokyo (JP). 渡邉 信之 (WATANABE,Nobuyuki) [JP/JP]; 〒100-6150 東京都 千代田区 永田町二丁目１１番１号 山王パークタワー 株式会社エヌ・ティ・ティ・ドコモ 知的財産部内 Tokyo (JP). 津田 雅之 (TSUDA,Masayuki) [JP/JP]; 〒100-6150 東京都 千代田区 永田町二丁目１１番１号 山王パークタワー 株式会社エヌ・ティ・ティ・ドコモ 知的財産部内 Tokyo (JP). 神谷 大 (KAMIYA,Dai) [JP/JP]; 〒100-6150 東京都 千代田区 永田町二丁目１１番１号 山王パークタワー 株式会社エヌ・ティ・ティ・ドコモ 知的財産部内 Tokyo (JP). 浅井 真生 (ASAI,Mao) [JP/JP]; 〒100-6150 東京都 千代田区 永田町二丁目１１番１号 山王パークタワー 株式会社エヌ・ティ・ティ・ドコモ 知的財産部内 Tokyo (JP). 三浦 史光 (MIURA,Fumiaki) [JP/JP]; 〒100-6150 東京都 千代田区 永田町二丁目１１番１号 山王パークタワー 株式会社エヌ・ティ・ティ・ドコモ 知的財産部内 Tokyo (JP). 鷲尾 諭 (WASHIO,Satoshi) [JP/JP]; 〒100-6150 東京都 千代田区 永田町二丁目１１番１号 山王パークタワー 株式会社エヌ・ティ・ティ・ドコモ 知的財産部内 Tokyo (JP). 富岡 淳樹 (TOMIOKA,Atsuki) [JP/JP]; 〒100-6150 東京都 千代田区 永田町二

[続葉有]

(54) Title: CONTENT DELIVERY METHOD AND CONTENT DELIVERY SYSTEM

(54) 発明の名称: コンテンツ配信方法およびコンテンツ配信システム



16...MOBILE APPARATUS
15...MOBILE PACKET COMMUNICATION NETWORK
17...GATEWAY SERVER APPARATUS
18...MANAGEMENT SERVER APPARATUS
200...LIST FILE
12...IP SERVER APPARATUS
211...EXPLANATION FILE
207...EXPLANATION FILE
214...Jar FILE
210...Jar FILE
11...INTERNET
13...IP SERVER APPARATUS
202...EXPLANATION FILE
206...Jar FILE
14...IP SERVER APPARATUS

(57) Abstract: A mobile apparatus (16) capable of activating Java-AP software receives SDF (Security Description File)(204) from a management server apparatus (18) managed by a reliable organization (a communication business entity managing a mobile packet communication network 15); obtains ADF (205) from an IP server apparatus (13) by use of a URL included in the SDF; obtains a Jar file (206) from the IP server apparatus (13) by use of the ADF (205); and installs into itself Java-AP software including these files. The Java-AP, which is realized by activating the Java-AP software, operates within the range of the right represented by policy information included in the SDF (204).

[続葉有]

WO 03/058411 A1

Nintendo Exhibit 1003
Part 2 of 2
Page 00719

丁目１１番１号 山王パークタワー 株式会社エヌ・ティ・ティ・ドコモ 知的財産部内 Tokyo (JP). 川端 博史 **(KAWABATA,Hiroshi)** [JP/JP]; 〒100-6150 東京都 千代田区 永田町二丁目１１番１号 山王パークタワー 株式会社エヌ・ティ・ティ・ドコモ 知的財産部内 Tokyo (JP). 近藤 隆 **(KONDO,Takashi)** [JP/JP]; 〒100-6150 東京都 千代田区 永田町二丁目１１番１号 山王パークタワー 株式会社エヌ・ティ・ティ・ドコモ 知的財産部内 Tokyo (JP).

(74) 代理人: 川崎 研二 **(KAWASAKI,Kenji)**; 〒103-0027 東京都 中央区 日本橋一丁目２番１０号 東洋ビルディング ７階 朝日特許事務所 Tokyo (JP).

(81) 指定国 *(国内)*: AU, BR, CA, CN, ID, IN, KR, NO, NZ, PH, PL, SG, US.

(84) 指定国 *(広域)*: ヨーロッパ特許 (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, SE, SI, SK, TR).

添付公開書類:
— 国際調査報告書

２文字コード及び他の略語については、定期発行される各*PCT*ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(57) 要約:

Ｊａｖａ－ＡＰソフトウェアを起動することができる移動機１６が、信頼できる機関（移動パケット通信網１５を管理する通信事業者）が管理する管理サーバ装置１８からＳＤＦ（セキュリティ記述ファイル）２０４を受信し、このＳＤＦに内包されているＵＲＬを用いてＩＰサーバ装置１３からＡＤＦ２０５を取得し、このＡＤＦ２０５を用いてＩＰサーバ装置１３からＪａｒファイル２０６を取得し、これらのファイルを内包するＪａｖａ－ＡＰソフトウェアを自身にインストールする。このＪａｖａ－ＡＰソフトウェアを起動することで実現されるＪａｖａ－ＡＰは、ＳＤＦ２０４に内包されているポリシー情報で表される権限の範囲内で動作する。

明細書

コンテンツ配信方法およびコンテンツ配信システム

技術分野

本発明は、端末装置にアプリケーションソフトウェアを配信する技術に関する。

背景技術

Ｊａｖａ（登録商標）プログラミング言語に従って記述されたプログラムを実行してＪａｖａ－ＡＰ（Ｊａｖａアプリケーション）を実行する機能と、この種のプログラムを含むＪａｖａ－ＡＰソフトウェアを、ネットワークを介してダウンロードする機能とを備えた移動機が普及している。

Ｊａｖａ－ＡＰソフトウェアとして、Ｊａｒ（Java Archive）ファイルとＡＤＦ（Application Descriptor File）とがある。ここで、Ｊａｒは、それが実行されることにより、あるＪａｖａ－ＡＰをユーザに提供するプログラムを内包している。また、ＡＤＦはＪａｒファイルに依存しており、例えば、Ｊａｒファイルの格納位置を示すＵＲＬ（以後、パッケージＵＲＬ）、Ｊａｒファイルのサイズを示す情報、Ｊａｒファイルの最終変更日時を示す情報等を必須情報として内包している。

移動機は、次のような手順で、所望のＪａｖａ－ＡＰに関連したソフトウェアのダウンロードを行う。まず、移動機は、ＷＷＷ（World Wide Web）を構成するサーバ装置から所望のＪａｖａ－ＡＰに関連したＡＤＦを取得する。

ＡＤＦを取得した移動機はこのＡＤＦの内容を調べ、所望のＪａｖａ

Nintendo Exhibit 1003
Part 2 of 2
Page 00721

ーＡＰに関連したＪａｒファイルを当該移動機にインストール可能であるか否かを判断する。そして、インストール可能と判断すると、移動機は、ＡＤＦに含まれていたパッケージＵＲＬを用いて、ＷＷＷを構成するサーバ装置からＪａｒファイルを取得する。このＪａｒファイルの取得をもってＪａｖａーＡＰソフトウェアのダウンロードは完了する。その後、移動機においては、ダウンロードされたＪａｖａーＡＰソフトウェアのインストールが行われ、当該Ｊａｖａーソフトウェアは起動要求さえあれば実行される状態となる。

ところで、移動機内で実行されるＪａｖａーＡＰの挙動についての制限は、通信アプリケーションなどの移動機が元から備えているネイティブアプリケーションの挙動についての制限よりも厳しくなっている。例えば、Ｊａｖａーは、移動機内の電話番号データを参照することができないようになっている。このような厳しい制限を課すことにより、悪意をもって作成されたＪａｖａーＡＰ、あるいは不具合を有するＪａｖａーＡＰによって移動機内の秘密性の高い情報が漏洩したり改竄されたりする事態を確実に回避することができる。

しかし、上述した厳しい制限を全てのＪａｖａーＡＰに対して一律に課すだけでは、ユーザやＩＰ（情報提供事業者）の希望を満たすことはできない。例えば、ある程度の信頼性が保証されるのであれば、Ｊａｖａーに移動機に格納された個人情報を参照する権限を与えてもよいと感じるユーザがいると思われる。また、ＩＰにも、移動機に格納されている個人情報や移動機が有する多数の機能を使用するＪａｖａーＡＰを提供したいという希望がある。

これらの希望を満たすべく、移動機のユーザに対して通信サービスを提供する通信事業者等の信頼できる機関がＪａｖａーＡＰに権限を与え、この権限を移動機に通知し、当該権限に基づいて移動機が当該Ｊａｖａーの挙動を制限するという仕組みが考えられる。この仕組みでは、権限の信頼性を保証するために、信頼できる機関以外の他者が権限の付与・管理に関与し得ないようにすべきである。

Nintendo Exhibit 1003
Part 2 of 2
Page 00722

ＪａｖａーＡＰソフトウェアのダウンロード手順に上述の仕組みを適用する場合、ＡＤＦあるいはＪａｒファイルに権限を示す情報を内包させるのが妥当である。ＪａｒファイルはＩＰにより随時更新され、ＩＰが保有するのが適当であることから、信頼できる機関に保有させるならＡＤＦが権限を示す情報を内包させるのに妥当ということになる。

しかし、ＡＤＦはＪａｒファイルに依存した内容となることから、ＩＰが手元のＪａｒファイルを更新すると、信頼できる機関が保有しているＡＤＦの更新も必要になってくる。また、Ｊａｒファイルを更新せずとも、ＡＤＦの更新が必要となることがある。例えば、ＩＰにおいて、あるＪａｒファイルへのアクセスが殺到し、このＪａｒファイルを他のサーバ装置へ移動する場合である。この場合、Ｊａｒファイルの格納位置が変更されるから、ＡＤＦに内包されているパッケージＵＲＬを変更する必要がある。しかしながら、ＡＤＦは信頼できる機関において他者の関与を排するように管理されるのであるから、ＡＤＦの更新作業は繁雑な作業となると予想される。

発明の開示

本発明は、上述した事情に鑑みて為されたものであり、依存関係にある複数のファイルを配信することで配信される、アプリケーションを実現するためのソフトウェアを、ＩＰの自由度を制限することなく、権限に応じた挙動をアプリケーションに対して許可する端末装置へ配信する配信方法および配信システムを提供することを目的としている。

上述した課題を解決するために、本発明は、ファイルの格納位置を通知されると当該ファイルを返送する通信システムが、アプリケーションを実現するためのソフトウェアを内包した実体ファイルに依存した情報と前記実体ファイルの格納位置を示す情報とを含んだアプリケーション記述ファイルの格納位置を示す第１の識別情報と、前記ソフトウェアに従って実行されるアプリケーションの挙動の許容範囲に関する権限情報とを内包したセキュリティ記述ファイルを、当該ファイルを格納

Nintendo Exhibit 1003
Part 2 of 2
Page 00723

した管理サーバ装置から前記セキュリティを確保して前記権限情報によって示された範囲内でアプリケーションの挙動を許可する端末装置へ送信する権限送信過程と、前記端末装置が、前記権限送信過程にて前記通信システムから送信された前記セキュリティ記述ファイルに内包されている前記第１の識別情報を用いて、前記アプリケーション記述ファイルを記憶した１または複数のサーバ装置から、当該アプリケーション記述ファイルを取得する依存情報取得過程と、前記端末装置が、前記依存情報取得過程にて取得した前記アプリケーション記述ファイルを用いて前記通信システムから前記実体ファイルを取得するプログラム取得過程とを有する配信方法を提供する。

この配信方法によれば、端末装置は、アプリケーションに対応したアプリケーション記述ファイルおよび実体ファイルを取得する前に、セキュリティが確保された上で通信システムから送信されるセキュリティ記述ファイルを取得する。このセキュリティ記述ファイルにはアプリケーションに与えられた権限が示されており、端末装置では、取得したセキュリティ記述ファイルに示される権限に応じた挙動が当該セキュリティ記述ファイルに対応するアプリケーションに許可される。

また、本発明は、アプリケーションを実現するためのソフトウェアを内包した実体ファイルと、前記実体ファイルに依存した内容を有し前記実体ファイルの格納位置を示すアプリケーション記述ファイルとを格納した１または複数のサーバ装置と、前記アプリケーション記述ファイルの格納位置を示す第１の識別情報と端末装置が前記ソフトウェアを実行することにより実現されるアプリケーションに与えられた権限を示す権限情報とを内包したセキュリティー記述ファイルを格納した管理サーバ装置とを有し、ファイルの格納位置を通知されると当該ファイルを返送する通信システムと、アプリケーションに与えられた権限に応じた挙動を当該アプリケーションに対して許可する端末装置とを有し、前記管理サーバ装置は、前記セキュリティ記述ファイルを前記端末装置へ、セキュリティを確保して送信し、前記端末装置は、前記通信システ

Nintendo Exhibit 1003
Part 2 of 2
Page 00724

ムから送信された前記セキュリティ記述ファイルに内包されている前記第１の識別情報を用いて前記アプリケーション記述ファイルを取得し、前記アプリケーション記述ファイルを用いて前記通信システムから前記実体ファイルを取得する配信システムを提供する。

この配信システムによれば、端末装置は、アプリケーションに対応したアプリケーション記述ファイルおよび実体ファイルを取得する前に、セキュリティが確保された上で通信システムから送信されるセキュリティ記述ファイルを取得することになる。このセキュリティ記述ファイルにはアプリケーションに与えられた権限が示されており、端末装置では、取得したセキュリティ記述ファイルに示される権限に応じた挙動が当該セキュリティ記述ファイルに対応するアプリケーションに許可される。

図面の簡単な説明

図１は本発明の実施の一形態に係る配信システムの構成を示すブロック図である。

図２は同配信システムに特有のＡＤＦのデータ構成を示す概念図である。

図３は同配信システムを構成する移動機１６の構成を示すブロック図である。

図４は同移動機１６の機能構成を示す概念図である。

図５は同移動機１６がＪａｖａ－ＡＰソフトウェアをダウンロードしインストールする処理を示すフローチャートである。

図６は同配信システムにおいて管理サーバ装置１８に格納されているＳＤＦのデータ構成を示す概念図である。

図７は同ＳＤＦに内包されるポリシー情報の内容を示す概念図である。

図８は同配信システムの動作を説明するためのブロック図である。

図９は配信システムにて配信されるリストページを示す図である。

Nintendo Exhibit 1003
Part 2 of 2
Page 00725

図１０は配信システムを構成するＩＰサーバ装置１２が格納している説明ファイルの内容を示す図である。

図１１は配信システムにて配信される説明ページを示す図である。

図１２はＩＰサーバ装置１２が格納している説明ファイルの内容を示す図である。

図１３は同配信システムにて配信される説明ページを示す図である。

図１４は同配信システムを構成するＩＰサーバ装置１３が格納している説明ファイルの内容を示す図である。

図１５は同配信システムにて配信される説明ページを示す図である。

図１６は同配信システムの動作を説明するためのシーケンス図である。

図１７は同配信システムの動作を説明するためのシーケンス図である。

図１８は同配信システムの動作を説明するためのシーケンス図である。

図１９は同配信システムの他の動作を説明するためのブロック図である。

図２０は同配信システムの他の動作を説明するためのシーケンス図である。

発明を実施するための最良の形態

以下、図面を参照して、本発明の実施の一形態である配信システムについて説明する。なお、図面において、共通する部分には同一の符号が付されている。

（１）構成

図１に示されるように、この配信システムにおいて、ＩＰサーバ装置１２～１４は、インターネット１１に接続されている。ＩＰサーバ装置１２は第１のＩＰ（Internet Provider）によって管理されており、ＩＰサーバ装置１３および１４は第１のＩＰと異なる第２のＩＰにより

Nintendo Exhibit 1003
Part 2 of 2
Page 00726

管理されている。そして、ＩＰサーバ装置１２～１４はＷＷＷを構成しており、それぞれ一般的なＷＷＷサーバ装置と同様のハードウェアおよび機能を有する。移動パケット通信網１５は、通信事業者が移動パケット通信サービスを提供するために用いる網である。移動機１６は、この移動パケット通信網１５との間で無線パケット通信を行うことが可能である。ゲートウェイサーバ装置１７は、移動パケット通信網１５と同じ通信事業者により管理されている。このゲートウェイサーバ装置１７は、移動パケット通信網１５とインターネット１１とを接続する装置であり、一般的なゲートウェイサーバ装置の構成と同様の構成を有する。管理サーバ装置１８は、専用線によりゲートウェイサーバ装置１７に接続されている。この管理サーバ装置１８もまたＷＷＷを構成し、一般的なＷＷＷサーバ装置と同様のハードウェアおよび機能を有する。ゲートウェイサーバ装置１７は、移動パケット通信網１５とインターネット１１との間のパケット中継、管理サーバ装置１８と移動パケット通信網１５との間のパケット中継および管理サーバ装置１８とインターネット１１との間のパケット中継を行う。移動機１６は、この中継機能を利用することにより、移動パケット通信網１５およびインターネット１１を介してＩＰサーバ装置１２～１４とパケット通信を行うことが可能である。なお、実際の配信システムには多数の移動機が存在するが、図面が繁雑になるのを避けるために一つの移動機１６のみが図示されている。これと同様の理由により、３つのＩＰサーバ装置１２～１４のみが図示されている。

この配信システムにおいて、移動機１６は、インターネット１１上の所望のサイトからＪａｖａ－ＡＰソフトウェアを受け取ることができる。この移動機１６が受け取ることができるソフトウェアは、トラステッドＪａｖａ－ＡＰに関するものと、非トラステッドＪａｖａ－ＡＰに関するものに大別される。ここで、トラステッドＪａｖａ－ＡＰソフトウェアは、移動パケット通信網１５を管理する通信事業者が、ＩＰサーバ装置１２～１４を管理するＩＰとの契約に基づいて信頼性を保証し

Nintendo Exhibit 1003
Part 2 of 2
Page 00727

たＪａｖａ－ＡＰソフトウェアである。また、非トラステッドＪａｖａ－ＡＰソフトウェアは、トラステッドＪａｖａ－ＡＰソフトウェア以外のＪａｖａ－ＡＰソフトウェアである。

管理サーバ装置１８は、この配信システム内を流通する各トラステッドＪａｖａ－ＡＰソフトウェアについてＳＤＦ（セキュリティ記述ファイル）を各々記憶している。このＳＤＦは、移動パケット通信網１５を管理する通信事業者によって作成されるファイルであり、移動機のトラステッドＡＰＩ（APplication Interface）を使用するＪａｖａ－ＡＰソフトウェアを移動機にダウンロードする際に必須のファイルである。なお、トラステッドＡＰＩについては後述する。図６に示されるように、ＳＤＦは、トラステッドＪａｖａ－ＡＰソフトウェアを識別するためのＡＰＩＤ、ポリシー情報、当該Ｊａｖａ－ＡＰソフトウェアに対応したＡＤＦの記憶位置を示すＡＤＦ－ＵＲＬ、および当該Ｊａｖａ－ＡＰソフトウェアを提供するＩＰに対してＣＡが付与した公開鍵を有する。ここで、ポリシー情報は、Ｊａｖａ－ＡＰの挙動に対する制限を示す情報である。なお、このポリシー情報およびこれに基づいて行われるＪａｖａ－ＡＰの挙動の制限の詳細については後述する。

本実施形態では、移動機１６からの要求により、ＩＰサーバ装置１２～１４の１つから移動機１６にトラステッドＪａｖａ－ＡＰソフトウェアが配信される場合に、これに対応したＳＤＦが管理サーバ装置１８から移動機１６に配信される。そして、移動機１６において、トラステッドＪａｖａ－ＡＰソフトウェアが実行されるときには、これに対応したＳＤＦに基づいて、トラステッドＪａｖａ－ＡＰの挙動の制限が行われる。これが本実施形態の特徴の１つである。図１に示すとおり、ＳＤＦの送信は移動パケット通信網１５を通して行われ、管理サーバ装置１８とゲートウェイサーバ装置１７は専用線によって接続されている。また、送信に際し、ＳＤＦは暗号化される。

以下、この特徴との関連において、配信システムの各要素の構成を説明する。

Nintendo Exhibit 1003
Part 2 of 2
Page 00728

ＩＰサーバ装置１２、１３及び１４は不揮発性メモリ１２Ａ，１３Ａ及び１４Ａをそれぞれ有する。

不揮発性メモリ１２Ａ、１３Ａおよび１４Ａは、ハードディスク等の不揮発性メモリであり、Ｊａｒファイルおよびａｄｆからなるｊａｖａ－ＡＰソフトウェアと、ｊａｖａ－ＡＰソフトウェアの内容を移動機のユーザに説明するための説明ファイルとを記憶している。

不揮発性メモリ１２Ａ、１３Ａおよび１４Ａに記憶されている個々のｊａｖａ－ＡＰソフトウェアは、トラステッドｊａｖａ－ＡＰソフトウェアであるかも知れないし、非トラステッドｊａｖａ－ＡＰソフトウェアであるかも知れない。トラステッドｊａｖａ－ＡＰであるか非トラステッドｊａｖａ－ＡＰであるかに拘わらず、全てのｊａｖａ－ＡＰソフトウェアのＡＤＦには、ＷＷＷにおけるＪａｒファイルの記憶位置を示すパッケージＵＲＬや、Ｊａｒファイルのサイズを示す情報、Ｊａｒファイルの最終変更日時を示す情報等が記述されている。これらはｊａｖａ－ＡＰソフトウェアのＡＤＦに記述されるべき項目として一般的に知られているものである。そして、トラステッドｊａｖａ－ＡＰソフトウェアのＡＤＦは、これらの一般的に知られた情報の他に、図２に示されるように、そのトラステッドｊａｖａ－ＡＰソフトウェアのＡＰＩＤとＪａｒファイルのハッシュ値とを内包している。さらにトラステッドｊａｖａ－ＡＰソフトウェアのＡＤＦは、当該ソフトウェアを提供するＩＰに対してＣＡ（認証局）から付与された秘密鍵で暗号化されている。

また、説明ファイルは、ＨＴＭＬに従って記述されたテキストファイルである。移動機は、あるｊａｖａ－ＡＰソフトウェアをダウンロードする場合に、それに先だって、このｊａｖａ－ＡＰソフトウェアに対応した説明ファイルをダウンロードする必要がある。説明ファイルには、ｊａｖａ－ＡＰソフトウェアのダウンロードの指示をユーザから受け取るＵＩ（ユーザインターフェイス）を構成するための情報が含まれている。移動機１６は、この情報に従い、ＵＩ画面を表示する。ユーザは、このＵＩ画面中の所望のｊａｖａ－ＡＰを表すオブジェクトを指定す

Nintendo Exhibit 1003
Part 2 of 2
Page 00729

る操作を移動機１６に対して行うことができる。説明ファイルには、このようにしてユーザによって指定されるオブジェクトを、ダウンロード対象であるＪａｖａ－ＡＰソフトウェアに対応するＳＤＦ（ＳＤＦが存在しない場合にはＡＤＦ）のＷＷＷにおける所在を示すＵＲＬに対応付けるように記述されている。

ＩＰサーバ装置１２～１４の各々は、以上説明した各ファイルをＩＰの指示に従って作成および更新する機能を備えている。

管理サーバ装置１８は、ハードディスク等の不揮発性メモリ１８Ａを有する。管理サーバ装置１８は、ＴＣＰコネクションを通信相手との間に確立する。管理サーバ装置１８は、このTCPコネクションを介して、ＨＴＴＰのＧＥＴメソッドを用いた要求メッセージを通信相手から受信すると、当該ＧＥＴメソッドに指定されたＵＲＬで特定されるファイルを不揮発性メモリ１８Ａから読み出し、このファイルを含むＨＴＴＰの応答メッセージを返送して当該コネクションを切断する。

また、上記不揮発性メモリ１８Ａには、ダウンロード可能なＪａｖａ－ＡＰソフトウェアを移動機１６のユーザに紹介するためのリストファイル２００と、このリストファイル２００に挙げられた各Ｊａｖａ－ＡＰソフトウェアに各々対応したＳＤＦとが記憶される。

リストファイル２００は、ＳＤＦと同様、ＩＰサーバ装置１３～１５を管理する各ＩＰと管理サーバ装置１８を管理する通信事業者との間で結ばれた契約に従って通信事業者により作成されたファイルである。このリストファイル２００はＨＴＭＬに従って記述されたテキストファイルである。既に説明したように、移動機は、あるＪａｖａ－ＡＰソフトウェアのＳＤＦをダウンロードする場合に、そのＳＤＦのＵＲＬを含む説明ファイルを取得する必要がある。移動機１６は、この説明ファイルを格納しているＩＰサーバ装置にアクセスするという方法により、この説明ファイルを取得することができる。しかし、このような直接的な方法以外に、本実施形態において移動機１６は、次のような手順により所望のＪａｖａ－ＡＰソフトウェアの説明ファイルを取得すること

Nintendo Exhibit 1003
Part 2 of 2
Page 00730

もできる。まず、移動機１６は、管理サーバ装置１８にアクセスして、このリストファイル２０を取得し、これに従い、ＵＩ画面を表示する。ユーザは、このＵＩ画面中の所望のＪａｖａ－ＡＰを表すオブジェクトを指定する操作を移動機１６に対して行うことができる。リストファイル２００は、このようにしてユーザによって指定されるオブジェクトを、ダウンロード対象であるＪａｖａ－ＡＰソフトウェアの説明ファイルのＷＷＷにおける所在を示すＵＲＬに対応付ける。移動機１６は、このようにリストファイル２００を介して得られるＵＲＬを用いて、ＩＰサーバ装置から説明ファイルを取得するのである。

移動機１６は、図３に示されるように、ＯＳ（オペレーティングシステム）ソフトウェア、Ｊａｖａ－ＡＰを実行する環境を構築するためのＪａｖａ－ＡＰ環境ソフトウェアおよび各種ネイティブＡＰソフトウェア等を記憶したＲＯＭ１６Ａと、ＲＯＭ１６Ａに接続されＲＯＭ１６Ａからプログラムを読み出して実行するＣＰＵ１６Ｂと、ＣＰＵ１６Ｂに接続された表示部１６Ｃと、不揮発性メモリ１６Ｄと、ＲＡＭ１６Ｅと、通信部１６Ｆと、操作部１６Ｇとを有する。

表示部１６Ｃは、例えば液晶表示パネルを有し、ＣＰＵ１６Ｂから供給されるデータを画像として表示する。不揮発性メモリ１６Ｄは例えばＳＲＡＭやＥＥＰＲＯＭであり、ＣＰＵ１６Ｂによりデータを読み書きされる。不揮発性メモリ１６Ｄは、ＷＷＷを構成するサーバ装置（以後、Ｗｅｂサーバ装置）からダウンロードしたＪａｖａ－ＡＰソフトウェア（ＡＤＦおよびＪａｒ）や、ＳＤＦを記憶するために使用される。

通信部１６Ｆは、移動パケット通信網１５と無線パケット通信を行うものであり、ＣＰＵ１６Ｂと移動パケット通信網１５との間でパケットを中継する。また、通信部１６Ｆは、アンテナや無線送受信部の他に、通話のためのＣＯＤＥＣやマイク、スピーカ等を備えている。従って、移動機１６は、この通信部１６Ｆにより、図示せぬ移動通信網を介して回線交換による通話を行うこともできる。操作部１６Ｇは操作子を備え、操作子の操作に応じた信号をＣＰＵ１６Ｂへ供給する。

Nintendo Exhibit 1003
Part 2 of 2
Page 00731

図示せぬ電源が投入されると、ＣＰＵ１６ＢはＲＡＭ１６Ｅをワークエリアとし、ＲＯＭ１６ＡからＯＳソフトウェアに内包されているプログラムを読み出して実行する。これにより、ＣＰＵ１６ＢにはＵＩ等を提供する機能が実行される。すなわち、ＣＰＵ１６ＢはＯＳソフトウェアを起動して移動機１６内にて図４のＯＳを実行する。ＯＳは操作部１６Ｇから供給される信号とＵＩの状態とに基づいてユーザの指示を特定し、この指示に応じた処理を行う。

ユーザの指示がネイティブＡＰソフトウェアである通信ソフトウェアの起動を要求するものであれば、ＯＳは通信ソフトウェアを起動して移動機１６内にて通信ＡＰを実行する。この通信ＡＰを用いることで、ユーザは通話相手と通話をすることができる。

ユーザの指示がネイティブＡＰソフトウェアである電話帳ＡＰの起動を要求するものであれば、ＯＳは電話帳ソフトウェアを起動して移動機１６内にて電話帳ＡＰを実行する。この電話帳ＡＰを用いることで、ユーザは、不揮発性メモリ１６Ｄに記憶された電話帳の内容（以後、電話帳データ）を参照・使用・変更することができる。

ユーザの指示がネイティブＡＰソフトウェアであるＷｅｂブラウザソフトウェアの起動を要求するものであれば、ＯＳはＷｅｂブラウザソフトウェアを起動して移動機１６内にてＷｅｂブラウザを実行する。また、ＷｅｂブラウザはＵＩを提供する。そして、ユーザから操作部１６Ｇの操作により指示があると、ＵＩの状態と操作部１６Ｇから供給される信号とに基づいてユーザの指示を特定し、この指示に応じた処理を行う。例えば、当該指示が指定されたファイルをＷＷＷから取得する旨の場合には、通信部１６Ｆを制御して当該ファイルを記憶したＷｅｂサーバ装置との間にＴＣＰコネクションを確立し、このコネクションを介して、指定された位置を示すＵＲＬでＧＥＴメソッドを用いたＨＴＴＰの要求メッセージを送信し、この要求メッセージに対応する応答メッセージを受信し、当該コネクションを切断する。さらに、Ｗｅｂブラウザは、受信した応答メッセージに内包されているファイルをＨＴＭＬに従っ

Nintendo Exhibit 1003
Part 2 of 2
Page 00732

て解釈し、Ｗｅｂページを内包するＵＩを生成し、ユーザに提供する。また、ユーザの指示がＪａｖａ－ＡＰソフトウェアのダウンロードを要求するものである場合には、この指示を次に述べるＪＡＭ（Java Application Manager）に通知する。具体的には、Ｗｅｂページにおいて、クリック操作またはプレス操作により、オブジェクトタグが指定されているアンカータグで表されるアンカーが指定されると、Ｗｅｂブラウザは当該オブジェクトタグのｄａｔａ属性に指定されているＵＲＬを抽出し、当該ＵＲＬからのＪａｖａ－ＡＰソフトウェアのダウンロードが要求されたことをＪＡＭに通知する。

ユーザの指示がネイティブＡＰソフトウェアであるＪＡＭソフトウェアの起動を要求するものであれば、ＯＳはＪＡＭソフトウェアを起動して移動機１６内にてＪＡＭを実行する。ＪＡＭは、移動機１６にインストールされているＪａｖａ－ＡＰソフトウェアの一覧をユーザに提示し、ユーザにより指定されたＪａｖａ－ＡＰソフトウェアを起動する。具体的には、ＪＡＭに対するユーザの指示がＪａｖａ－ＡＰソフトウェアの起動を要求するものであれば、Ｊａｖａ－ＡＰ環境ソフトウェアが起動されて移動機１６内にＪａｖａ－ＡＰ環境が実行される。そして、指定されたＪａｖａ－ＡＰソフトウェアが起動されてＪａｖａ－ＡＰ環境内にＪａｖａ－ＡＰが実行される。Ｊａｖａ－ＡＰ環境は、携帯端末に適した軽量のＪａｖａ仮想マシンであるＫＶＭと、Ｊａｖａ－ＡＰに対して提供されるＡＰＩとを有する。Ｊａｖａ－ＡＰに対して提供されるＡＰＩは、通信事業者がＩＰとの契約に基づいて信頼性を保証したＪａｖａ－ＡＰ（以後、トラステッドＡＰ）のみに使用が許可されるトラステッドＡＰＩと、あらゆるＪａｖａ－ＡＰに使用が許可される非トラステッドＡＰＩとに分けられる。

（２）動作

以下、本実施形態の動作を説明する。ＪＡＭは、Ｊａｖａ－ＡＰのダウンロードを要求する指示がＷｅｂブラウザから通知されると、Ｊａｖａ－ＡＰソフトウェアを移動機１６にダウンロードしインストールす

Nintendo Exhibit 1003
Part 2 of 2
Page 00733

る処理を行う。この処理の流れを図５に示す。なお、図５では、移動機１６が説明ファイルを取得するまでの過程は省略されている。説明ファイルの取得までの過程は、幾つかの態様があるので、後に具体的な動作例を挙げて説明する。図５に示されるように、ＪＡＭは、まず、ダウンロードしようとするＪａｖａ－ＡＰソフトウェアがトラステッドＪａｖａ－ＡＰソフトウェアであるか否かを判定する（ステップＳ１１）。具体的には、移動機１６が説明ファイルを取得すると、Ｗｅｂブラウザは、この説明ファイルに応じたＵＩをユーザに提供し、Ｊａｖａ－ＡＰソフトウェアのダウンロードに関するユーザからの指示を受け取る。Ｗｅｂブラウザは、このユーザからの指示により特定されたＪａｖａ－ＡＰソフトウェアのＵＲＬをＪＡＭに通知する。ＪＡＭは、Ｗｅｂブラウザから通知されたＵＲＬの末尾のファイル名を参照し、このファイルの拡張子が“sdf”であればトラステッドＪａｖａ－ＡＰソフトウェア“sdf”でなければ非トラステッドＪａｖａ－ＡＰソフトウェアであると判定する。ダウンロードしようとするＪａｖａ－ＡＰソフトウェアがトラステッドＪａｖａ－ＡＰソフトウェアであると判定された場合には、従来と同様のダウンロードおよびインストール処理が行われる（ステップＳ１２）。

ダウンロードしようとするＪａｖａ－ＡＰソフトウェアがトラステッドＪａｖａ－ＡＰソフトウェアと判定された場合には、ＪＡＭは、当該ソフトウェアに対応するＳＤＦを管理サーバ装置１８から取得する（ステップＳ１３）。すなわち、ＪＡＭは、管理サーバ装置１８との間にＴＣＰコネクションを確立し、このコネクションを介して、Ｗｅｂブラウザから通知されたＵＲＬで示される位置に記憶されたＳＤＦの送信を管理サーバ装置１８に要求する内容の要求メッセージを生成・送信し、このメッセージに対する応答メッセージを受信し、上記コネクションを切断する。

そして、ＪＡＭは、応答メッセージに内包されているＳＤＦからＡＰＩＤとＡＤＦ－ＵＲＬと公開鍵を抽出するとともに、当該ＳＤＦを不揮

Nintendo Exhibit 1003
Part 2 of 2
Page 00734

発性メモリ１６Ｄに書き込む。

次に、ＪＡＭはＡＤＦを取得する（ステップＳ１４）。具体的には、ＪＡＭは、ＳＤＦから抽出したＡＤＦ－ＵＲＬで特定されるＡＤＦを記憶したＷｅｂサーバ装置との間にＴＣＰコネクションを確立し、当該ＡＤＦの送信を要求する内容の要求メッセージを生成・送信し、このメッセージに対する応答メッセージを受信し、当該ＴＣＰコネクションを切断する。

既に説明したように、トラステッドＪａｖａ－ＡＰソフトウェアに対応するＡＤＦは、ＡＰＩＤとＪａｒファイルのハッシュ値とを内包し、さらに当該トラステッドＪａｖａ－ＡＰソフトウェアを提供するＩＰに対してＣＡが付与した秘密鍵により署名（暗号化）されている。そこで、ＪＡＭは、応答メッセージに内包されているＡＤＦの署名をＳＤＦから抽出された公開鍵を用いて検証し（復号し）、当該ＡＤＦの正当性を判断する（ステップＳ１５）。

ＡＤＦが正当であると判断した場合には、ＪＡＭは、ＳＤＦから抽出したＡＰＩＤとＡＤＦに内包されているＡＰＩＤとを比較し、両者が一致するか否かを判定する（ステップＳ１６）。両者が一致すると判定された場合には、ＪＡＭは、当該トラステッドＪａｖａ－ＡＰソフトウェアを移動機１６にインストール可能か否かをＡＤＦの内容に基づいて判定する（ステップＳ１７）。この判定の基準は従来と同様である。

インストール可能と判定された場合には、ＪＡＭは、Ｊａｒファイルを取得する。具体的には、ＪＡＭは、当該ＡＤＦを移動機１６に書き込むとともに、当該ＡＤＦからハッシュ値とパッケージＵＲＬを抽出する。さらにＪＡＭは、このパッケージＵＲＬで特定されるＪａｒファイルを記憶したＷｅｂサーバ装置との間にＴＣＰコネクションを確立し、当該Ｊａｒファイルの送信を要求する内容の要求メッセージを生成・送信し、このメッセージに対する応答メッセージを受信し、当該ＴＣＰコネクションを切断する（ステップＳ１８）。

さらに、ＪＡＭは、取得したＪａｒファイルに対するハッシュ値を算

Nintendo Exhibit 1003
Part 2 of 2
Page 00735

出する（ステップＳ１９）。ハッシュ値の算出に使用するハッシュ関数は任意であるが、移動機で使用されるハッシュ関数とＡＤＦに含まれるハッシュ値を算出するためにＩＰが用いるハッシュ関数は一致していなければならない。

ＪＡＭは、ＪＡＭが算出したハッシュ値とＡＤＦから抽出したハッシュ値とを比較し（ステップＳ２０）、両者が一致した場合には、取得したＪａｒファイルを管理サーバ装置１８に書き込み、トラステッドＪａｖａ－ＡＰソフトウェアのインストールに係る各種処理を行い（ステップＳ２１）、インストールに成功した旨をユーザに通知する（ステップＳ２２）。

ＡＤＦが正当でないと判断した場合、ＳＤＦが有するＡＰＩＤとＡＤＦが有するＡＰＩＤが不一致の場合、インストールしようとするＪａｖａ－ＡＰソフトウェアをインストール可能ではないと判断した場合、算出したハッシュ値とＡＤＦが有するハッシュ値とが不一致の場合には、ＪＡＭは、インストールに失敗した旨をユーザに通知するとともに、移動機１６の状態を、ＳＤＦの取得を開始する前の状態に戻す。

また、ＪＡＭは、Ｊａｖａ－ＡＰの挙動を監視し、トラステッドＡＰＩの使用を制限する。この制限は不揮発性メモリ１６Ｄに記憶されるＳＤＦ内のポリシー情報に従って行われる。ＳＤＦ内のポリシー情報は、例えば図７に概念的に示されるような内容となっている。図７に示されるポリシー情報では、移動機に格納された電話帳データを参照するときに必須のトラステッドＡＰＩである“getPhoneList()”と移動機の状態を取得するときに必須のトラステッドＡＰＩである“getMsStatus()”の使用が許可され、移動機に格納された発着信履歴データを参照するときに必須のトラステッドＡＰＩである“getCallHistory()”の使用が禁止されている。

（３）具体的動作例

次に、上述したシステムの動作例について説明する。

なお、以下に述べる動作において、ＴＣＰコネクションの確立および

切断動作についてはＨＴＴＰにおける一般的な動作となることから、それらの説明を省略する。また、前述のＯＳ、Ｗｅｂブラウザ、ＪＡＭ、ＪａｖａーＡＰ、ネイティブＡＰ等が行う動作は移動機１６の動作となることから、以降の説明では、動作の主体を移動機１６とする。

以下説明する動作例では、次のような状況を想定している。まず、図８に示されるように、管理サーバ装置１８の不揮発性メモリ１８Ａには、リストファイル２００とＳＤＦ２０４が記憶されている。ここで、リストファイル２００は、移動機１６において解釈・実行されると図９に示されるリストページ２０１を提供するように記述されている。また、リストファイル２００は、クリックまたはプレス操作により、リストページ２０１を構成する選択肢２０１Ａが指定されると、説明ファイル２０２のＵＲＬ（“http://www.main.bbb.co.jp/ghi.html”）をＧＥＴメソッドのパラメータとして含む要求メッセージが生成されるように記述されている。さらに、リストファイル２００は、リストページ２０１を構成する選択肢２０１Ｂが指定されると、説明ファイル２０７のＵＲＬ（“http://www.ccc.co.jp/jkl.html”）をＧＥＴメソッドのパラメータとして含む要求メッセージが生成されるように記述されている。

また、ＳＤＦ２０４は、ＡＰＩＤとして“0001”、ポリシー情報として図７に示される情報、ＡＤＦ－ＵＲＬとして“http://www.main.bbb.co.jp/viewer.jam”、およびＩＰサーバ装置１３とＩＰサーバ装置１４を管理するＩＰに対してＣＡが付与した公開鍵を内包している。

また、ＩＰサーバ装置１２の不揮発性メモリ１２Ａには、「詰め将棋」(西洋社会における「チェスパズル」に類似するゲーム)なる名称のＪａｖａーＡＰソフトウェア（以後、第１のＪａｖａーＡＰソフトウェア）に対応する説明ファイル２１１、ＡＤＦ２１３およびＪａｒファイル２１４が記憶されている。説明ファイル２１１、ＡＤＦ２１３およびＪａｒファイル２１４はＩＰサーバ装置１２を管理するＩＰによって作成されている。説明ファイル２１１の内容は図１０に示される通りである。説明ファイル２１１は、移動機１６において解釈・実行されると図１１

Nintendo Exhibit 1003
Part 2 of 2
Page 00737

に示される説明ページ２１２を提供するように記述されている。また、ＡＤＦ２１３はパッケージＵＲＬとしてＪａｒファイル２１４のＵＲＬ（“http://www.ccc.co.jp/shogi.jar”）を内包している。

また、ＩＰサーバ装置１２の不揮発性メモリ１２Ａには、「星占い」なる名称のＪａｖａ－ＡＰソフトウェア（以後、第２のＪａｖａ－ＡＰソフトウェア）に対応する説明ファイル２０７、ＡＤＦ２０９およびＪａｒファイル２１０が記憶されているものとする。説明ファイル２０７、ＡＤＦ２０９およびＪａｒファイル２１０はＩＰサーバ装置１２を管理するＩＰによって作成されている。説明ファイル２０７の内容は図１２に示される通りである。説明ファイル２０７は、移動機１６において解釈・実行されると図１３に示される説明ページ２０８を提供するように記述されている。また、ＡＤＦ２０９はパッケージＵＲＬとしてＪａｒファイル２１０のＵＲＬ（“http://www.ccc.co.jp/horoscope.jar”）を内包している。

また、ＩＰサーバ装置１３の不揮発性メモリ１３Ａには、「電話帳ビューア」なる名称のＪａｖａ－ＡＰソフトウェア（以後、第３のＪａｖａ－ＡＰソフトウェア）に対応する説明ファイル２０２、ＡＤＦ２０５およびＪａｒファイル２０６が記憶されている。説明ファイル２０２、ＡＤＦ２０５およびＪａｒファイル２０６は、ＩＰサーバ装置１３およびＩＰサーバ装置１４を管理するＩＰによって作成されている。説明ファイル２０２の内容は図１４に示される通りである。説明ファイル２０２は、移動機１６において解釈・実行されると図１５に示される説明ページ２０３を提供するように記述されている。ＡＤＦ２０５は、ＡＰＩＤとして“0001”、Ｊａｒファイル２０６のハッシュ値、パッケージＵＲＬとしてＪａｒファイル２０６のＵＲＬ（“http://www.main.bbb.co.jp/viewer.jar”）を内包しており、ＩＰサーバ装置１３およびＩＰサーバ装置１４を管理するＩＰに対してＣＡが付与した秘密鍵を用いて署名されている。

また、移動機１６は第１～第３のＪａｖａ－ＡＰソフトウェアをイン

Nintendo Exhibit 1003
Part 2 of 2
Page 00738

ストール可能な状態にある。

（２－１）インストール動作

まず、Ｊａｖａ－ＡＰソフトウェアを移動機１６にインストールする場合の動作例について、Ｊａｖａ－ＡＰソフトウェア毎に説明する。

（２－１－１）第１のＪａｖａ－ＡＰソフトウェア

第１のＪａｖａ－ＡＰソフトウェアのインストール動作は、ユーザが移動機１６を操作し、Ｗｅｂブラウザにより、所望のＪａｖａ－ＡＰソフトウェアを格納しているＩＰサーバ装置を求め、そこから説明ファイル２１１の取得を試みることから始まる。まず、移動機１６では、説明ファイル２１１のＵＲＬ（“http://www.ccc.co.jp/mno.html”）をＧＥＴメソッドのパラメータとして含む要求メッセージｔｍ１２が生成される。この要求メッセージｔｍ１２は、図１６に示されるように、移動機１６から送信されＩＰサーバ装置１２により受信される。ＩＰサーバ装置１２では、この要求メッセージｔｍ１２の内容に対応して説明ファイル２１１を内包した応答メッセージｔｍ１３が生成される。この応答メッセージｔｍ１３はＩＰサーバ装置１２から送信され移動機１６により受信される。移動機１６では、ユーザに対して、説明ファイル２１１の内容に応じたＵＩが提供される。この結果、表示部１６Ｃには、例えば図１１に示すような説明ページ２１２が表示される。

この説明ページ２１２を視たユーザが、説明ページ２１２内のアンカー２１２Ａが押下されるよう移動機１６を操作すると、移動機１６では、図１０の説明ファイル２１１に記述されたアンカータグ（“<A”で始まるタグ）のｉｊａｍ属性に指定されている値がｉｄ属性に指定されているオブジェクトタグ（“<OBJECT”で始まるタグ）を特定する。そして、このオブジェクトタグのｄａｔａ属性に指定されているＵＲＬ（“http://www.ccc.co.jp/shogi.jam”）が抽出され、図５におけるステップＳ１１の判断が行われる。この例では、ＵＲＬの拡張子がｓｄｆではないため、通常の処理（ステップＳ１２）が行われる。すなわち、次の通りである。まず、このＵＲＬで特定されるＡＤＦ２１３の送信を要求す

Nintendo Exhibit 1003
Part 2 of 2
Page 00739

る内容の要求メッセージｔｍ１６が生成される。この要求メッセージｔｍ１６は移動機１６から送信されＩＰサーバ装置１２により受信される。ＩＰサーバ装置１２では、この要求メッセージｔｍ１６の内容に対応してＡＤＦ２１３を内包した応答メッセージｔｍ１７が生成される。この応答メッセージｔｍ１７はＩＰサーバ装置１２から送信され移動機１６により受信される。

移動機１６では、ＡＤＦ２１３の内容に基づいて第１のＪａｖａ－ＡＰソフトウェアをインストール可能か否かが判定される。前述のように、移動機１６は第１のＪａｖａ－ＡＰソフトウェアをインストール可能な状態にあるから、移動機１６では第１のＪａｖａ－ＡＰソフトウェアをインストール可能と判定される。

次に、移動機１６では、ＡＤＦ２１３が不揮発性メモリ１６Ｄ１に書き込まれる。また、移動機１６では、ＡＤＦ２１３からパッケージＵＲＬ（“http://www.ccc.co.jp/shogi.jar”）が抽出され、このパッケージＵＲＬで特定されるＪａｒファイル２１４の送信を要求する内容の要求メッセージｔｍ１８が生成される。この要求メッセージｔｍ１８は移動機１６から送信されＩＰサーバ装置１２により受信される。ＩＰサーバ装置１２では、この要求メッセージｔｍ１８の内容に対応してＪａｒファイル２１４を内包した応答メッセージｔｍ１９が生成される。この応答メッセージｔｍ１９はＩＰサーバ装置１２から送信され移動機１６により受信される。移動機１６ではＪａｒファイル２１４が不揮発性メモリ１６Ｄ１に書き込まれ、第１のＪａｖａ－ＡＰソフトウェアのインストールが完了する。

なお、移動機１６において第１のＪａｖａ－ＡＰソフトウェアをインストール可能ではないと判断された場合、移動機１６の状態は、ＡＤＦ２１３の取得を開始する前の状態に戻る。

（２－１－２）第２のＪａｖａ－ＡＰソフトウェア

第２のＪａｖａ－ＡＰソフトウェアのインストール動作は、ユーザが移動機１６を操作し、説明ファイル２０７の取得を試みることから始ま

る。既に説明したように、説明ファイル２０７は、関連するＩＰサーバ装置への直接アクセスまたはリストファイル２００経由のいずれかにより取得可能であるが、ここでは、リストファイル２００の取得を試みることから始まる動作のみについて説明する。

図１７に示されるように、移動機１６では、リストファイル２００のＵＲＬ（“http://www.aaa.co.jp/def.html”）をＧＥＴメソッドのパラメータとして含む要求メッセージｔｍ２０が生成される。この要求メッセージｔｍ２０は移動機１６から送信され管理サーバ装置１８により受信される。管理サーバ装置１８では、この要求メッセージｔｍ２０の内容に対応してリストファイル２００を内包した応答メッセージｔｍ２１が生成される。この応答メッセージｔｍ２１は管理サーバ装置１８から送信され移動機１６により受信される。移動機１６では、応答メッセージｔｍ２１の受信を契機として、応答メッセージｔｍ２１内のリストファイル２００がＨＴＭＬに従って解釈され、移動機１６のユーザに対して、リストファイル２００の内容に応じたＵＩが提供される。この結果、移動機１６の表示部１６Ｃには、例えば図９に示すようなリストページ２０１が表示される。

このリストページ２０１を視たユーザが、リストページ２０１内の選択肢２０１Ｂが押下されるように移動機１６を操作すると、移動機１６では、選択肢２０１Ｂに対応付けられているＵＲＬ（“http://www.ccc.co.jp/jkl.html”）をＧＥＴメソッドのパラメータとして含む要求メッセージｔｍ２２が生成される。この要求メッセージｔｍ２２は移動機１６から送信されＩＰサーバ装置１２により受信される。ＩＰサーバ装置１２では、この要求メッセージｔｍ２２の内容に対応して説明ファイル２０７を内包した応答メッセージｔｍ２３が生成される。この応答メッセージｔｍ２３はＩＰサーバ装置１２から送信され移動機１６により受信される。移動機１６では、ユーザに対して、説明ファイル２０７の内容に応じたＵＩが提供される。この結果、表示部１６Ｃには、例えば図１３に示すような説明ページ２０８が表示される。

Nintendo Exhibit 1003
Part 2 of 2
Page 00741

この説明ページ２０８を視たユーザが、説明ページ２０８内のアンカー２０８Ａが押下されるよう移動機１６を操作すると、移動機１６では、図１２の説明ファイル２０７に記述されたアンカータグ（“<A”で始まるタグ）のｉｊａｍ属性に指定されている値がｉｄ属性に指定されているオブジェクトタグ（“<OBJECT”で始まるタグ）を特定する。そして、このオブジェクトタグのｄａｔａ属性に指定されているＵＲＬ（“http://www.ccc.co.jp/horoscope.jam”）が抽出され、図５におけるステップＳ１１の判断を行が行われる。この例では、ＵＲＬの拡張子がｓｄｆではないため、通常の処理（ステップＳ１２）が行われる。すなわち、次の通りである。まず、このＵＲＬで特定されるＡＤＦ２０９の送信を要求する内容の要求メッセージｔｍ２６が生成される。この要求メッセージｔｍ２６は移動機１６から送信されＩＰサーバ装置１２により受信される。ＩＰサーバ装置１２では、この要求メッセージｔｍ２６の内容に対応してＡＤＦ２０９を内包した応答メッセージｔｍ２７が生成される。この応答メッセージｔｍ２７はＩＰサーバ装置１２から送信され移動機１６により受信される。

移動機１６では、ＡＤＦ２０９の内容に基づいて第２のＪａｖａ－ＡＰソフトウェアをインストール可能か否かが判定される。前述のように、移動機１６は第２のＪａｖａ－ＡＰソフトウェアをインストール可能な状態にあるから、移動機１６では第２のＪａｖａ－ＡＰソフトウェアをインストール可能と判定される。

次に、移動機１６では、ＡＤＦ２０９が不揮発性メモリ１６Ｄ１に書き込まれる。また、移動機１６では、ＡＤＦ２０９からパッケージＵＲＬ（“http://www.ccc.co.jp/horoscope.jar”）が抽出され、このパッケージＵＲＬで特定されるＪａｒファイル２１０の送信を要求する内容の要求メッセージｔｍ２８が生成される。この要求メッセージｔｍ２８は移動機１６から送信されＩＰサーバ装置１２により受信される。ＩＰサーバ装置１２では、この要求メッセージｔｍ２８の内容に対応してＪａｒファイル２１０を内包した応答メッセージｔｍ２９が生成され

Nintendo Exhibit 1003
Part 2 of 2
Page 00742

る。この応答メッセージｔｍ２９はＩＰサーバ装置１２から送信され移動機１６により受信される。移動機１６ではＪａｒファイル２１０が不揮発性メモリ１６Ｄ１に書き込まれ、第２のＪａｖａ－ＡＰソフトウェアのインストールが完了する。

なお、移動機１６において、第２のＪａｖａ－ＡＰソフトウェアをインストール可能ではないと判断された場合、移動機１６の状態は、ＡＤＦ２０９の取得を開始する前の状態に戻る。

（２－１－３）第３のＪａｖａ－ＡＰソフトウェア

第３のＪａｖａ－ＡＰソフトウェアのインストール動作は、ユーザが移動機１６を操作し、説明ファイル２０２の取得を試みることから始まる。この動作例において、移動機１６は関連するリストファイル２００を取得して、説明ファイル２０２の所在を求め、説明ファイル２０２の取得を試みる。

図１８に示されるように、リストファイル２００の取得を試みることから始まる動作において、移動機１６が応答メッセージｔｍ２１を受信し、例えば図９に示すようなリストページ２０１が表示されるまでは図１７に示す動作と同一の動作が行われる。このリストページ２０１を視たユーザが、リストページ２０１内の選択肢２０１Ａが押下されるように移動機１６を操作すると、移動機１６では、選択肢２０１Ａに対応付けられているＵＲＬ（“http://www.main.bbb.co.jp/ghi.html”）をＧＥＴメソッドのパラメータとして含む要求メッセージｔｍ３２が生成される。この要求メッセージｔｍ３２は移動機１６から送信されＩＰサーバ装置１３により受信される。ＩＰサーバ装置１３では、この要求メッセージｔｍ３２の内容に対応して説明ファイル２０２を内包した応答メッセージｔｍ３３が生成される。この応答メッセージｔｍ３３はＩＰサーバ装置１３から送信され移動機１６により受信される。移動機１６では、ユーザに対して、説明ファイル２０２の内容に応じたＵＩが提供される。この結果、表示部１６Ｃには、例えば図１５に示すような説明ページ２０３が表示される。

Nintendo Exhibit 1003
Part 2 of 2
Page 00743

この説明ページ２０３を視たユーザが、説明ページ２０３内のアンカー２０３Ａが押下されるよう移動機１６を操作すると、移動機１６では、図１４の説明ファイル２０２に記述されたアンカータグ（“<A”で始まるタグ）のｉｊａｍ属性に指定されている値がｉｄ属性に指定されているオブジェクトタグ（“<OBJECT”で始まるタグ）を特定する。そして、このオブジェクトタグのｄａｔａ属性に指定されているＵＲＬ（“http: //www.aaa.co.jp/abc.sdf”）が抽出され、図５におけるステップＳ１１の判断を行が行われる。この例では、ＵＲＬの拡張子がｓｄｆであるため、ステップＳ１３以降の処理が行われる。すなわち、次の通りである。まず、このＵＲＬで特定されるＳＤＦ２０４の送信を要求する内容の要求メッセージｔｍ３４が生成される。この要求メッセージｔｍ３４は移動機１６から送信され管理サーバ装置１８により受信される。管理サーバ装置１８では、この要求メッセージｔｍ３４の内容に対応してＳＤＦ２０４を内包した応答メッセージｔｍ３５が生成される。この応答メッセージｔｍ３５は管理サーバ装置１８から送信され、ゲートウェイサーバ装置１７及び移動パケット通信網１５を介して移動機１６により受信される。管理サーバ装置１８とゲートウェイサーバ装置１７との間の通信路は専用線であり、ゲートウェイサーバ装置１７はセキュリティの確保された移動パケット通信網１５に直接的に接続されていることから、移動機１６に受信されるまでにＳＤＦ２０４が改竄される虞は無い（以上、ステップＳ１３）。

移動機１６において、ＳＤＦ２０４は不揮発性メモリ１６Ｄの不揮発性メモリ１６Ｄ１に書き込まれる。また、移動機１６では、ＳＤＦ２０４からＡＰＩＤ（“0001”）とＡＤＦ－ＵＲＬ（“http://www.main.bbb.co.jp/viewer.jam”）と公開鍵が抽出され、このＡＤＦ－ＵＲＬで特定されるＡＤＦ２０５の送信を要求する内容の要求メッセージｔｍ３６が生成される。この要求メッセージｔｍ３６は移動機１６から送信されＩＰサーバ装置１３により受信される。ＩＰサーバ装置１３では、この要求メッセージｔｍ３６の内容に対応してＡＤＦ２０５を内包し

Nintendo Exhibit 1003
Part 2 of 2
Page 00744

た応答メッセージｔｍ３７が生成される。この応答メッセージｔｍ３７はＩＰサーバ装置１３から送信され移動機１６により受信される（以上、ステップＳ１４）。

移動機１６ではＳＤＦ２０４から抽出された公開鍵を用いてＡＤＦ２０５の正当性が判断される（ステップＳ１５）。前述のように、ＳＤＦ２０４に内包されている公開鍵はＡＤＦ２０５への署名の際に用いた秘密鍵と対応していることから、ＩＰサーバ装置１３内あるいはＩＰサーバ装置１３から移動機１６への通信経路においてＡＤＦ２０５が変更されていない限り、ＡＤＦ２０５が正当であると判断される。

ＡＤＦ２０５が正当であると判断されると、移動機１６では、ＳＤＦ２０４から抽出されたＡＰＩＤとＡＤＦ２０５に内包されているＡＰＩＤとが比較される（ステップＳ１６）。前述のように、ＩＰサーバ装置１３におけるＡＤＦ２０５にはＳＤＦ２０４内のＡＰＩＤと一致するＡＰＩＤが記述されることから、記述ミス等が無い限り、ＳＤＦ２０４から抽出されたＡＰＩＤとＡＤＦ２０５に内包されているＡＰＩＤは一致する。

ＡＰＩＤが一致すると、移動機１６では、ＡＤＦ２０５の内容に基づいて第３のＪａｖａ－ＡＰソフトウェアをインストール可能か否かが判定される（ステップＳ１７）。前述のように、移動機１６は第３のＪａｖａ－ＡＰソフトウェアをインストール可能な状態にあるから、移動機１６では第３のＪａｖａ－ＡＰソフトウェアをインストール可能と判定される。

次に、移動機１６では、ＡＤＦ２０５が不揮発性メモリ１６Ｄ１に書き込まれる。また、移動機１６では、ＡＤＦ２０５からハッシュ値とパッケージＵＲＬ（“http://www.main.bbb.co.jp/viewer.jar”）が抽出され、このパッケージＵＲＬで特定されるＪａｒファイル２０６の送信を要求する内容の要求メッセージｔｍ３８が生成される。この要求メッセージｔｍ３８は移動機１６から送信されＩＰサーバ装置１３により受信される。ＩＰサーバ装置１３では、この要求メッセージｔｍ３８の

内容に対応してＪａｒファイル２０６を内包した応答メッセージｔｍ３９が生成される。この応答メッセージｔｍ３９はＩＰサーバ装置１３から送信され移動機１６により受信される（以上、ステップＳ１８）。

移動機１６ではＪａｒファイル２０６と所定のハッシュ関数とを用いてハッシュ値が算出され（ステップＳ１９）、この算出されたハッシュ値とＡＤＦ２０５から抽出されたハッシュ値とが比較される（ステップＳ２０）。前述のように、ＡＤＦ２０５には当該ＡＤＦ２０５に対応するＪａｒファイルのハッシュ値が記述されることから、記述ミス等がない限り、両ハッシュ値は一致する。両ハッシュ値が一致すると、移動機１６では、Ｊａｒファイル２０６が不揮発性メモリ１６Ｄ１に書き込まれ、第３のＪａｖａ－ＡＰソフトウェアのインストールが完了する（ステップＳ２１およびＳ２２）。

なお、移動機１６においてＡＤＦ２０５が正当でないと判断された場合や、ＳＤＦ２０４から抽出されたＡＰＩＤとＡＤＦ２０５に内包されているＡＰＩＤが不一致の場合、第３のＪａｖａ－ＡＰソフトウェアをインストール可能ではないと判断された場合、算出したハッシュ値とＡＤＦ２０５から抽出されたハッシュ値とが不一致の場合には、ユーザに対して失敗が通知され（ステップＳ２３）、移動機１６の状態は、ＳＤＦ２０４の取得を開始する前の状態に戻る。

（２－２）Ｊａｖａ－ＡＰソフトウェアが起動されている時の移動機１６の挙動

次に、Ｊａｖａ－ＡＰソフトウェアが起動されている時の移動機１６の挙動について説明する。

（２－２－１）第１のＪａｖａ－ＡＰソフトウェア

上述したインストール動作によりインストールされた第１のＪａｖａ－ＡＰソフトウェアが、ＪＡＭが実現された移動機１６において起動され、当該ソフトウェアに対応した機能（以後、第１のＪａｖａ－ＡＰ）が実現されたときの移動機１６の挙動について説明する。

第１のＪａｖａ－ＡＰが使用しようとするＡＰＩが非トラステッド

Nintendo Exhibit 1003
Part 2 of 2
Page 00746

ＡＰＩの場合、当該ＡＰＩの使用はＪＡＭにより許可される。したがって、第１のＪａｖａ－ＡＰは当該ＡＰＩを使用することができる。

また、第１のＪａｖａ－ＡＰが使用しようとするＡＰＩがトラステッドＡＰＩの場合、ＪＡＭは当該Ｊａｖａ－ＡＰに対応するＳＤＦが不揮発性メモリ１６Ｄに記憶されているか否かを調べる。そのようなＳＤＦは不揮発性メモリ１６Ｄに記憶されていないから、ＪＡＭは第１のＪａｖａ－ＡＰによる当該ＡＰＩの使用を禁止する。したがって、第１のＪａｖａ－ＡＰは当該ＡＰＩの使用することができない。

（２－２－２）第２のＪａｖａ－ＡＰソフトウェア

インストールされた第２のＪａｖａ－ＡＰソフトウェアが、ＪＡＭが実現された移動機１６において起動され、当該ソフトウェアに対応した機能が実現されたときの移動機１６の挙動は、第１のＪａｖａ－ＡＰソフトウェアが起動されている時の移動機１６の挙動と同様となる。

（２－２－３）第３のＪａｖａ－ＡＰソフトウェア

インストールされた第３のＪａｖａ－ＡＰソフトウェアが、ＪＡＭが実現された移動機１６において起動され、当該ソフトウェアに対応した機能（以後、第３のＪａｖａ－ＡＰ）が実現されたときの移動機１６の挙動について説明する。

第３のＪａｖａ－ＡＰが使用しようとするＡＰＩが非トラステッドＡＰＩの場合当該ＡＰＩの使用はＪＡＭにより許可される。したがって、第３のＪａｖａ－ＡＰは当該ＡＰＩを使用することができる。

第３のＪａｖａ－ＡＰが使用しようとするＡＰＩがトラステッドＡＰＩの場合、移動機１６の挙動はＡＰＩに依存する。以下、ＡＰＩ毎に移動機１６の挙動を説明する。

（２－２－３－１）getPhoneList()

“getPhoneList()”はトラステッドＡＰＩであるから、当該ＡＰＩの使用の可否は、不揮発性メモリ１６Ｄに記憶されているＳＤＦ２０４内のポリシー情報に基づいてＪＡＭにより決定される。このポリシー情報の内容は図７に示される内容であることから、“getPhoneList()”の使

Nintendo Exhibit 1003
Part 2 of 2
Page 00747

用がＪＡＭにより許可される。したがって、第３のＪａｖａ－ＡＰは“getPhoneList()”を使用することができる。つまり、第３のＪａｖａ－ＡＰは電話帳データを読み出すことができる。

（２－２－３－２）getCallHistory()

“getCallHistory()”はトラステッドＡＰＩであるから、当該ＡＰＩの使用の可否はＳＤＦ２０４内のポリシー情報に基づいてＪＡＭにより決定される。このポリシー情報の内容は図７に示される内容であることから、“getCallHistory()”の使用がＪＡＭにより禁止される。したがって、第３のＪａｖａ－ＡＰは“getCallHistory()”を使用することができない。つまり、第３のＪａｖａ－ＡＰは発着信履歴データを読み出すことができない。

（２－３）第３のＪａｖａ－ＡＰソフトウェアの変更後の動作

次に、ＩＰサーバ装置１３およびＩＰサーバ装置１４を管理するＩＰが第３のＪａｖａ－ＡＰソフトウェアの配信形態や内容を変更した場合の本システム動作について説明する。ただし、ここでの変更は、第３のＪａｖａ－ＡＰソフトウェアの改善等を目的としたＪａｒファイル２０６の内容の変更と、ＩＰサーバ装置１３の負荷の軽減等を目的とした配信形態の変更とを含む。後者の変更を達成するために、ＩＰサーバ装置１３およびＩＰサーバ装置１４を管理するＩＰは、図１９に示すように、変更後のＪａｒファイル２０６（以後、Ｊａｒファイル２１５）をＩＰサーバ装置１４の不揮発性メモリ１４Ａに記憶させ、このＪａｒファイル２１５に対応するようにＡＤＦ２０５の内容を変更してＡＤＦ２１６としている。変更後の第３のＪａｖａ－ＡＰソフトウェアの配信に必要な作業は以上の通りであり、管理サーバ装置１８を管理する通信事業者が行うべき作業は存在しない。

このような変更の後の第３のＪａｖａ－ＡＰソフトウェアのインストール動作は、図２０に示す通りとなる。図２０に示す動作が図１８に示す動作と相違し始めるのは、ＩＰサーバ装置１３において、ＡＤＦ２０５を内包した応答メッセージｔｍ３７ではなく、ＡＤＦ２１６を内包

Nintendo Exhibit 1003
Part 2 of 2
Page 00748

した応答メッセージｔｍ４７を生成してからである。なお、応答メッセージｔｍ４７は応答メッセージｔｍ３７、要求メッセージｔｍ４８は要求メッセージｔｍ３８、応答メッセージｔｍ４９は応答メッセージｔｍ３９に対応している。

ＩＰサーバ装置１３において応答メッセージｔｍ４７を生成して以降の動作が、図１８に示す動作と本質的に異なるのは、ＡＤＦ２１６およびＪａｒファイル２１５が処理の対象となる点と、ＡＤＦ２１６に内包されているパッケージＵＲＬ（“http://www.sub.bbb.co.jp/viewer.jar”）で特定されるＪａｒファイル２１５の送信を要求する内容の要求メッセージｔｍ４８が移動機１６にて生成される点と、この要求メッセージｔｍ４８が移動機１６から送信されＩＰサーバ装置１４により受信される点と、ＩＰサーバ装置１４においてＪａｒファイル２１５を内包した応答メッセージｔｍ４９が生成される点と、この応答メッセージｔｍ４９がＩＰサーバ装置１４から送信され移動機１６により受信される点である。

（３）変形例

上述した配信システムでは、ＡＤＦとＪａｒファイルはＩＰサーバ装置から送信するようにしたが、片方または両方を管理サーバ装置から送信してもよい。

また、上述した配信システムでは、移動機は、秘密鍵による署名データと公開鍵とを用いてＳＤＦとＡＤＦの作成者との対応関係の正当性を確認するようにしたが、システムに要求されるセキュリティレベルによっては、ＳＤＦに公開鍵を内包させず、ＩＰサーバ装置においてはＡＤＦに対する秘密鍵を用いた署名を行わず、かつ移動機においては当該確認処理を省略する、という形態とし、移動機およびＩＰサーバ装置における処理量や、移動機と管理サーバ装置およびＩＰサーバ装置との間の通信量を低減するようにしてもよい。

また、上述した配信システムでは、Ｊａｒファイルのハッシュ値を当該Ｊａｒファイルに対応するＡＤＦに内包させる一方、移動機において

Nintendo Exhibit 1003
Part 2 of 2
Page 00749

ハッシュ値を算出し、両者を比較し、Ｊａｒファイルとａｄｆとの対応関係の正当性を確認するようにしたが、システムに要求されるセキュリティレベルによっては、ａｄｆにハッシュ値を内包させずに当該確認処理を省略する形態とし、移動機およびＩＰサーバ装置における処理量や移動機とＩＰサーバ装置との間の通信量を低減するようにしてもよい。

また、上述した配信システムでは、トラステッドＪａｖａ－ＡＰに固有のＡＰＩＤを使用してＳＤＦとＡＤＦ（およびＪａｒファイル）との対応が正当であるか否かを判定するようにしたが、トラステッドＪａｖａ－ＡＰを提供する情報提供事業者に固有のＣＩＤを用いてＳＤＦとＡＤＦ（およびＪａｒファイル）との対応が正当であるか否かを判定するようにしてもよい。また、システムに要求されるセキュリティレベルによっては、ＡＰＩＤやＣＩＤを用いた判定を省略するようにしてもよい。

また、上述した配信システムではドメインネームを用いてサーバを指定するようにしたが、ＩＰアドレスを用いてサーバを指定するようにしてもよい。

また、移動機において、ＳＤＦの送信元のサーバ装置のドメインネームを予め設定された文字列と比較し、信頼できる機関が管理するサーバ装置のドメインネームである場合にのみ、ＳＤＦを正当と認める態様としてもよい。この態様では、比較対象の文字列（例えば、通信事業者のドメインネームを示す文字列）は移動機のＲＯＭまたは不揮発性メモリに予め格納されることになる。文字列をＲＯＭに予め格納すれば、文字列の書き換えが不可能であるから、より高いセキュリティを確保できる。また、文字列を不揮発性メモリに予め格納する場合では、移動機の売買後に信頼できる機関を格納することができるので、ユーザおよび信頼できる機関に対して優れた利便性を提供することができる。

また、上述した配信システムでは、ＳＤＦの配信に使用する通信路を提供する通信事業者を信頼できる機関として高いセキュリティを確保するようにしたが、本発明は通信路の提供が信頼できる機関により為さ

Nintendo Exhibit 1003
Part 2 of 2
Page 00750

れていない態様をも技術的範囲に含む。例えば、信頼できる機関と移動機とを暗号化通信路により接続し、この通信路を介して信頼できる機関がＳＤＦを配信するようにしてもよい。また、通信路のセキュリティが確保されていなくても、ＳＤＦを暗号化した後に配信し、移動機においてＳＤＦを復号するようにすれば、ある程度のセキュリティを確保してＳＤＦを配信することができる。

上述した配信システムでは、ＨＴＴＰに従ってファイルを送受するようにしたが、ＨＴＴＰＳを使用し、より高いセキュリティを確保するようにシステムを変形してもよい。

また、上述した配信システムにおいて、信頼できる機関がＩＰとなってよいこと、すなわち、管理サーバ装置がＩＰサーバ装置を兼ねるようにしてもよい。

さらに、上述した配信システムでは、Ｊａｖａ－ＡＰによる利用を制限する対象としてＡＰＩを挙げたが、任意のリソースを対象とすることができる。ここでいうリソースはハードウェアリソースであってもよい。また、後述するネットワークリソースやソフトウェアリソースであってもよい。ハードウェアリソースとしては、メモリやスピーカ、マイク、赤外線コントローラ、ＬＥＤ（Light Emitting Diode）等の移動機が備え得るものや移動機と共働するＵＩＭ（User Identity Module）やＳＩＭ（Subscriber Identity Module）等の外部機器なども挙げられる。

次にネットワークリソースについて説明する。前述したように、移動機は移動通信網との間で無線通信を行う。この無線通信時には、移動機は、移動通信網により提供される無線チャネル等の無線リソースを使用する。この無線リソースはネットワークリソースの一種である。また、移動機は無線リソースが属する通信プロトコルレイヤよりも高位の通信プロトコルレイヤにおいて、パケットの伝送路や回線接続の通信路などの通信リソースを使用する。このような通信リソースもネットワークリソースの一種である。

次にソフトウェアリソースについて説明する。ソフトウェアリソース

Nintendo Exhibit 1003
Part 2 of 2
Page 00751

としては、ＡＰＩやクラス、パッケージ等が挙げられる。ソフトウェアリソースが提供する機能は様々であるが、典型的な機能として、暗号演算などの演算処理機能や、Ｗｅｂブラウザ等の他のアプリケーションとの間でデータを送受したりする機能などが挙げられる。また、本発明は、上記外部機器が有するソフトウェアリソースをも利用の制限対象とする態様を技術的範囲に含む。

ところで、Ｊａｖａ－ＡＰによるハードウェアリソースやネットワークリソースの利用は、ソフトウェアリソースを利用して行われるのが一般的である。上述した配信システムにおける移動機も、ハードウェアリソースやネットワークリソースを利用するためのソフトウェアリソースを有しており、このようなソフトウェアリソースの利用を制限することにより、間接的に、ハードウェアリソースやネットワークリソースの利用を制限している。このように、間接的に制限することにより、多様なソフトウェアリソースを用意すれば、トラステッドＪａｖａ－ＡＰについてのみ、Ｊａｖａ－ＡＰの権限を変更する権限を与える、またはダウンロード元のサーバ装置としか通信することができないという制限を外す、あるいは特定の記憶領域に対してアクセスできるようにするといった、複数のリソースの制限を細かく変更しなければ実現できないようなことまで容易に指定できるようになる。なお、移動機に設定されたソフトウェアリソースの利用を制限して上記外部機器のソフトウェアリソースの利用を間接的に制限する態様も本発明の技術的範囲に含まれる。

なお、パーミッションの表現方法としては、一つのリソースと一つのフラグ（許可／禁止）とを対応付けるようにしてもよいし、複数のリソースのパーミッションを一つの表現で示すようにしてもよい。

また、本発明では、複数の種類を持つリソースについて、利用を許可（あるいは禁止）するようにパーミッションを示すことも可能である。この場合、移動機において、より木目細かな制御が実現される。例えば、メモリには２つの形態(読み出しと書き込み)があるから、非トラステッ

Nintendo Exhibit 1003
Part 2 of 2
Page 00752

ドＪａｖａ－ＡＰには読み出しでしか利用されないが、トラステッドＪａｖａ－ＡＰには読み出し及び書き込みの両方で利用され得るようにすることもできる。また、例えば、１つのパケット伝送路を複数のアプリケーションが共用可能な移動機において、パケット伝送路を利用する権限を有するＪａｖａ－ＡＰが起動されている間にＷｅｂブラウザ等が起動された場合、当該Ｊａｖａ－ＡＰが「パケット伝送路の利用を排他的に行う」ことを許可されていないＪａｖａ－ＡＰであればＷｅｂブラウザ等によるパケット伝送路の共用を排除することはできないが、「パケット伝送路の利用を排他的に行う」ことを許可されているＪａｖａ－ＡＰであればパケット伝送路を占有して使用することができる、といった制御が可能となる。また、上記変形例をさらに変形することで、以下の制御も可能となる。即ち、ある種のパーミッションを与えられたＪａｖａ－ＡＰはユーザに許可を求めることなくパケット通信路を排他的に利用することが可能となる。また、別のパーミッションを与えられたＪａｖａ－ＡＰはユーザに許可を求めることなくパケット通信路を利用することが可能だがパケット通信路を排他的に利用するためにはユーザの許可を得ることが必要となる。また、別のパーミッションを与えられたＪａｖａ－ＡＰはユーザに許可を求めることなくパケット通信路を利用することが可能だがパケット通信路を排他的に利用することは不可能となる。また、別のパーミッションを与えられたＪａｖａ－ＡＰはユーザの許可を得て初めてパケット通信路を利用することが可能となる。また、別のパーミッションを与えられたＪａｖａ－ＡＰはパケット通信路を利用することすら不可能となる。これらの例から明らかなように、本発明における「利用の種類」には、リソースを利用する際に経る手順の種類（ユーザの許可を得る手順／ユーザの許可を得ない手順）も含まれる。

また、上述した配信システムでは全ての移動機に対して同一のリストページが提供されるが、移動機毎に異なるリストページを提供するようにしてもよい。

Nintendo Exhibit 1003
Part 2 of 2
Page 00753

また、上述の配信システムでは、Ｊａｖａ－ＡＰの実行時にＪａｖａ－ＡＰの挙動を制限するようにした。その代わりに、ＩＰサーバ装置に格納されているＪａｒファイルにポリシー情報を内包させ、移動機において、Ｊａｒファイルのダウンロード時に、このポリシー情報とＳＤＦ中とのポリシー情報とを比較し、両者が一致しない場合には、当該Ｊａｒファイルに対応するＪａｖａ－ＡＰを起動できないように、あるいは当該Ｊａｒファイルを含むＪａｖａ－ＡＰソフトウェアをインストールできないようにしてもよい。もちろん、両ポリシー情報の一致する項目についてのパーミッションのみを有効とするようにしてもよい。

また、通信事業者がＣＡにより付与された秘密鍵を用いて署名してからＳＤＦを配信し、移動機においてＣＡが通信事業者に付与した公開鍵を用いてＳＤＦの署名を検証するようにしてもよい。もちろん、通信事業者の公開鍵は予め移動機に格納されていなければならない。公開鍵は、通信により配信し予め不揮発性メモリに書き込むことが可能である。また、ＲＯＭに書き込んだ後に移動機を販売することも可能である。

また、上述の配信システムではソフトウェアは移動機へ配信されるが、本発明の技術的範囲には、移動機以外の端末装置へ配信する態様も含まれる。

Nintendo Exhibit 1003
Part 2 of 2
Page 00754

請求の範囲

1．　ファイルの格納位置を通知されると当該ファイルを返送する通信システムが、アプリケーションを実現するためのソフトウェアを内包した実体ファイルに依存した情報と前記実体ファイルの格納位置を示す情報とを含んだアプリケーション記述ファイルの格納位置を示す第１の識別情報と、前記ソフトウェアに従って実行されるアプリケーションの挙動の許容範囲に関する権限情報とを内包したセキュリティ記述ファイルを、当該ファイルを格納した管理サーバ装置から前記セキュリティを確保して前記権限情報によって示された範囲内でアプリケーションの挙動を許可する端末装置へ送信する権限送信過程と、

前記端末装置が、前記権限送信過程にて前記通信システムから送信された前記セキュリティ記述ファイルに内包されている前記第１の識別情報を用いて、前記アプリケーション記述ファイルを記憶した１または複数のサーバ装置から、当該アプリケーション記述ファイルを取得する依存情報取得過程と、

前記端末装置が、前記依存情報取得過程にて取得した前記アプリケーション記述ファイルを用いて前記通信システムから前記実体ファイルを取得するプログラム取得過程と

を有する配信方法。

2．　前記アプリケーション記述ファイルが、前記依存情報取得過程にて前記管理サーバ装置から取得される請求項１に記載の配信方法。

3．　前記実体ファイルが、前記プログラム取得過程にて前記管理サーバ装置から取得される請求項１に記載の配信方法。

4．　前記アプリケーション記述ファイルが前記依存情報取得過程にて

Nintendo Exhibit 1003
Part 2 of 2
Page 00755

前記管理サーバ装置から取得され、

前記実体ファイルが前記プログラム取得過程にて前記管理サーバ装置から取得される請求項１に記載の配信方法。

５．　前記通信システムが前記セキュリティ記述ファイルを暗号化する暗号化過程と、

前記端末装置が、前記権限送信過程にて前記通信システムから送信された前記セキュリティ記述ファイルを復号する復号過程とを有し、

前記権限送信過程では、前記暗号化過程にて暗号化された前記セキュリティ記述ファイルを前記端末装置へ送信し、

前記依存情報取得過程では、前記端末装置が、前記復号過程にて復号された前記セキュリティ記述ファイルを用いて前記アプリケーション記述ファイルを取得する

請求項１に記載の配信方法。

６．　前記権限情報は資源の利用に関する制限を示す

請求項１に記載の配信方法。

７．　前記資源は前記端末装置内部のハードウェア資源である

請求項６に記載の配信方法。

８．　前記資源は前記端末装置外部の、前記端末装置が使用可能なハードウェア資源である

請求項６に記載の配信方法。

９．　前記資源は前記端末装置内部のソフトウェア資源である

請求項６に記載の配信方法。

１０．　前記資源は前記端末装置外部の、前記端末装置が使用可能なソ

フトウェア資源である

請求項６に記載の配信方法。

１１．　前記資源は、前記端末装置が使用可能なネットワーク資源である

請求項６に記載の配信方法。

１２．　前記権限情報は資源の利用の種類を示す

請求項１に記載の配信方法。

１３．　前記アプリケーションに対応するアプリケーション記述ファイルは前記アプリケーションを提供する情報提供事業者に対して認証局が与えた秘密鍵で署名されており、

前記アプリケーションに対応するセキュリティ記述ファイルは前記情報提供事業者に対して認証局が与えた公開鍵を内包し、

前記プログラム取得過程では、前記端末装置が、前記依存情報取得過程で取得したアプリケーション記述ファイルの正当性を前記公開鍵を用いて検証し、正当性が検証された場合にのみ、当該アプリケーション記述ファイルを用いて前記通信システムから前記実体ファイルを取得する

請求項１に記載の配信方法。

１４．　前記アプリケーション記述ファイルおよび前記セキュリティ記述ファイルは、前記管理サーバ装置を管理する管理者が付与するアプリケーション識別子を内包し、

前記プログラム取得過程では、前記端末装置が、前記権限送信過程で前記管理サーバ装置から送信されたセキュリティ記述ファイルに内包されたアプリケーション識別子と前記依存情報取得過程で取得したアプリケーション記述ファイルに内包されたアプリケーション識別子と

Nintendo Exhibit 1003
Part 2 of 2
Page 00757

を比較し、両者が一致した場合にのみ、当該アプリケーション記述ファイルを用いて前記通信システムから前記実体ファイルを取得する

請求項１に記載の配信方法。

１５．　前記通信システムは、さらに、前記セキュリティ記述ファイルの格納位置を示す第２の識別情報を内包したダウンロード用ファイルを格納した情報提供サーバ装置を有し、

前記通信システムが端末装置へ前記ダウンロード用ファイルを送信する事前送信過程と、

前記端末装置が、前記事前送信過程にて前記通信システムから送信された前記ダウンロード用ファイルを用いて前記セキュリティ記述ファイルの送信を前記通信システムに要求する権限送信要求過程とを有し、

前記権限送信過程では、前記通信システムが、前記権限送信要求過程にて要求された前記セキュリティ記述ファイルを前記端末装置へ送信する

請求項１に記載の配信方法。

１６．　前記依存情報送信過程にて送信されるセキュリティ記述ファイルの格納位置が前記管理サーバ装置内の場合にのみ前記依存情報取得過程以降の過程を実行する

請求項１に記載の配信方法。

１７．前記端末装置は移動機である

請求項１～請求項１６のいずれか一の請求項に記載の配信方法。

１８．　アプリケーションを実現するためのソフトウェアを内包した実体ファイルと、前記実体ファイルに依存した内容を有し前記実体ファイルの格納位置を示すアプリケーション記述ファイルとを格納した１または複数のサーバ装置と、前記アプリケーション記述ファイルの格納位

Nintendo Exhibit 1003
Part 2 of 2
Page 00758

置を示す第１の識別情報と端末装置が前記ソフトウェアを実行することにより実現されるアプリケーションに与えられた権限を示す権限情報とを内包したセキュリティー記述ファイルを格納した管理サーバ装置とを有し、ファイルの格納位置を通知されると当該ファイルを返送する通信システムと、

アプリケーションに与えられた権限に応じた挙動を当該アプリケーションに対して許可する端末装置とを有し、

前記管理サーバ装置は、前記セキュリティ記述ファイルを前記端末装置へ、セキュリティを確保して送信し、

前記端末装置は、前記通信システムから送信された前記セキュリティ記述ファイルに内包されている前記第１の識別情報を用いて前記アプリケーション記述ファイルを取得し、前記アプリケーション記述ファイルを用いて前記通信システムから前記実体ファイルを取得する

配信システム。

１９．　前記管理サーバ装置が前記アプリケーション記述ファイルを格納する請求項１８に記載の配信システム。

２０．　前記管理サーバ装置が前記実体ファイルを格納する請求項１８に記載の配信システム。

２１．　前記管理サーバ装置が前記アプリケーション記述ファイルを格納し、

前記管理サーバ装置が前記実体ファイルを格納する請求項１８に記載の配信システム。

Nintendo Exhibit 1003
Part 2 of 2
Page 00759

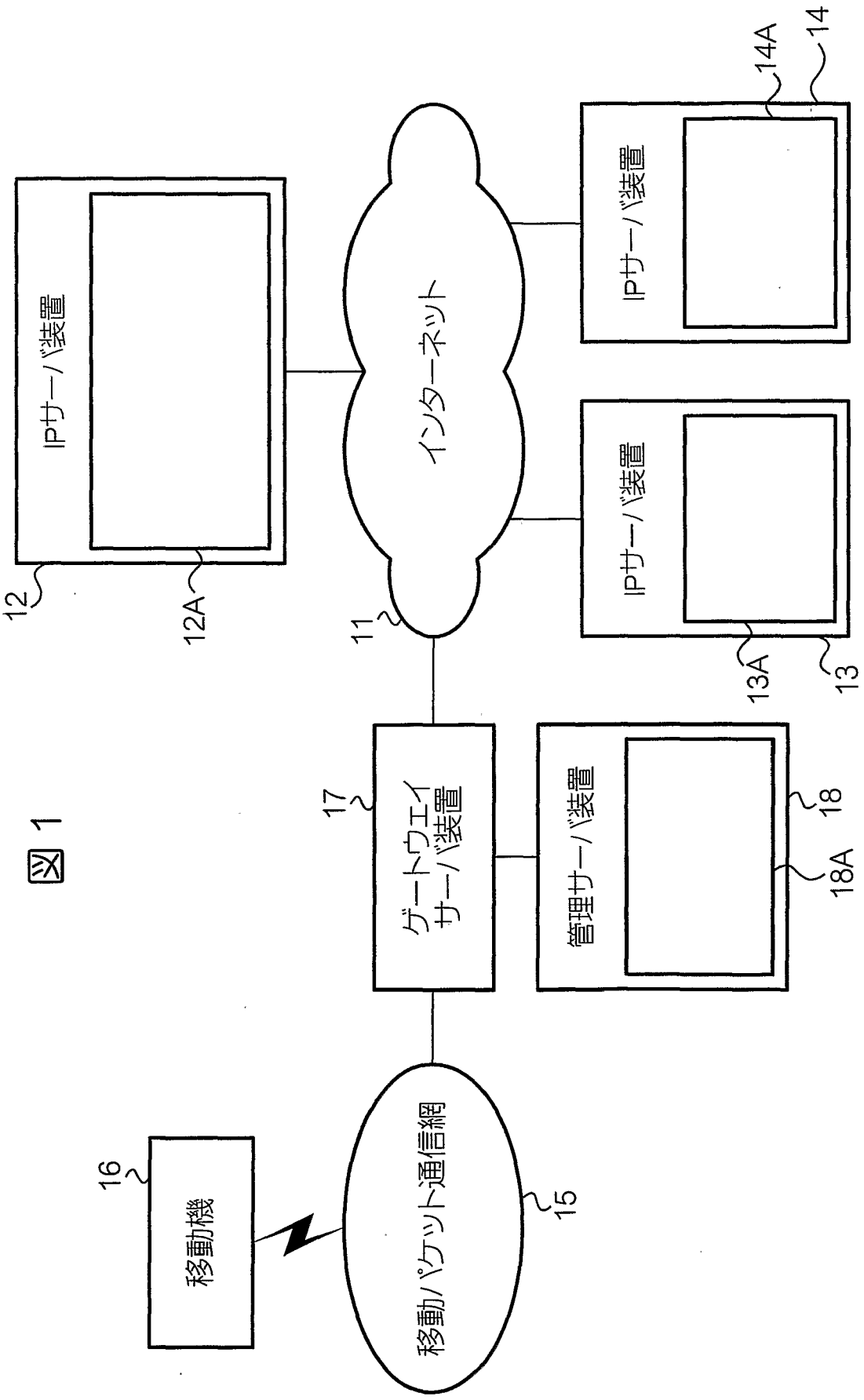

IPサーバ装置
12
12A
インターネット
11
IPサーバ装置
14A
14
IPサーバ装置
13A
13
ゲートウエイ
サーバ装置
17
管理サーバ装置
18
18A
移動/パケット通信網
15
移動機
16

図 1

図 2

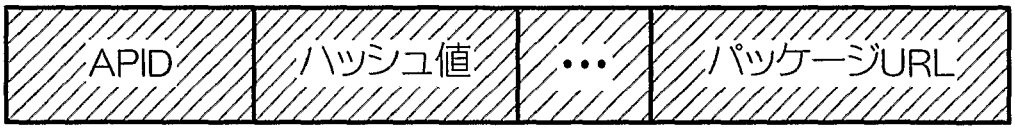

APID
ハッシュ値
…
パッケージURL

図 3

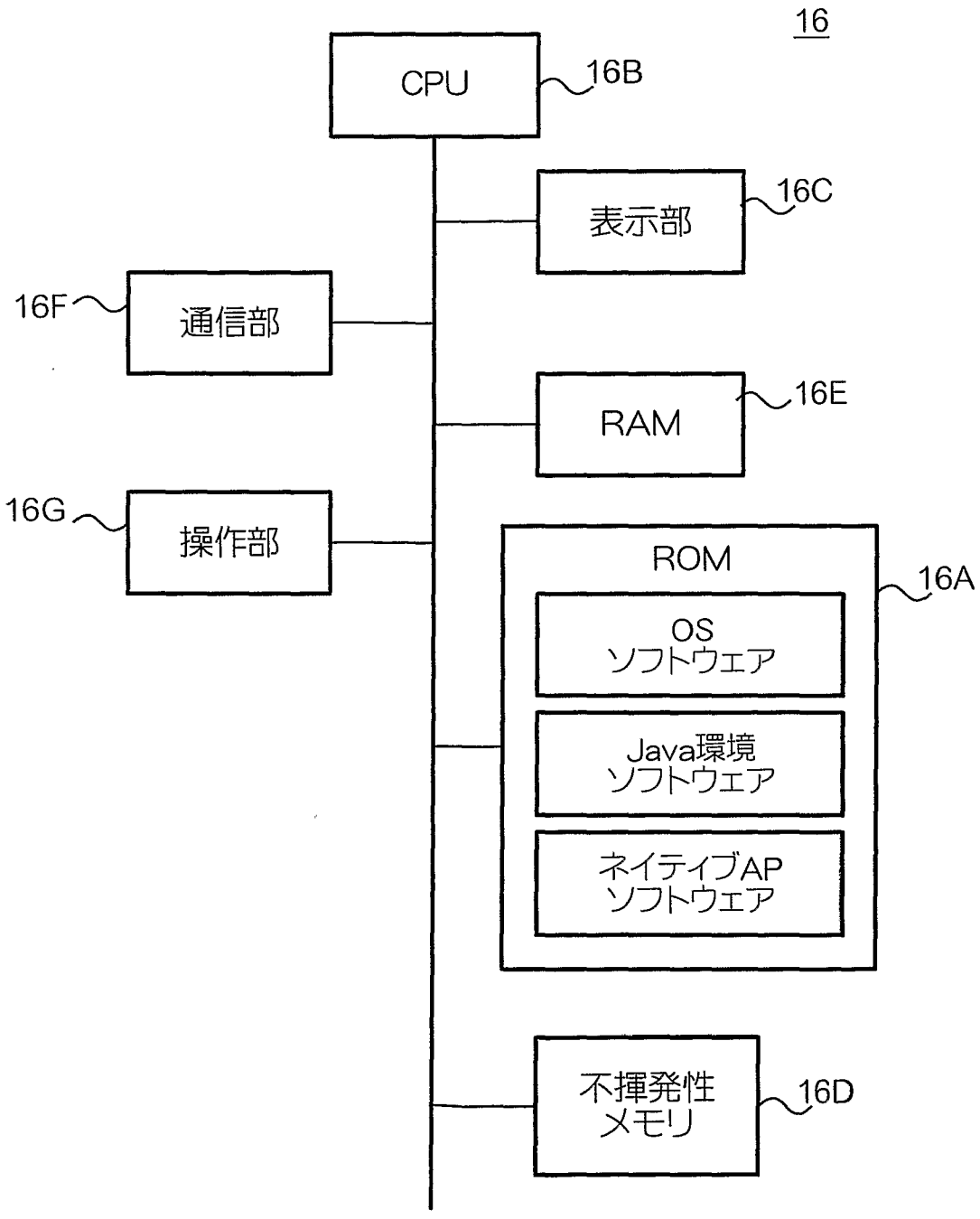

16
CPU
16B
表示部
16C
16F
通信部
RAM
16E
16G
操作部
ROM
16A
OS
ソフトウェア
Java環境
ソフトウェア
ネイティブAP
ソフトウェア
不揮発性
メモリ
16D

図4

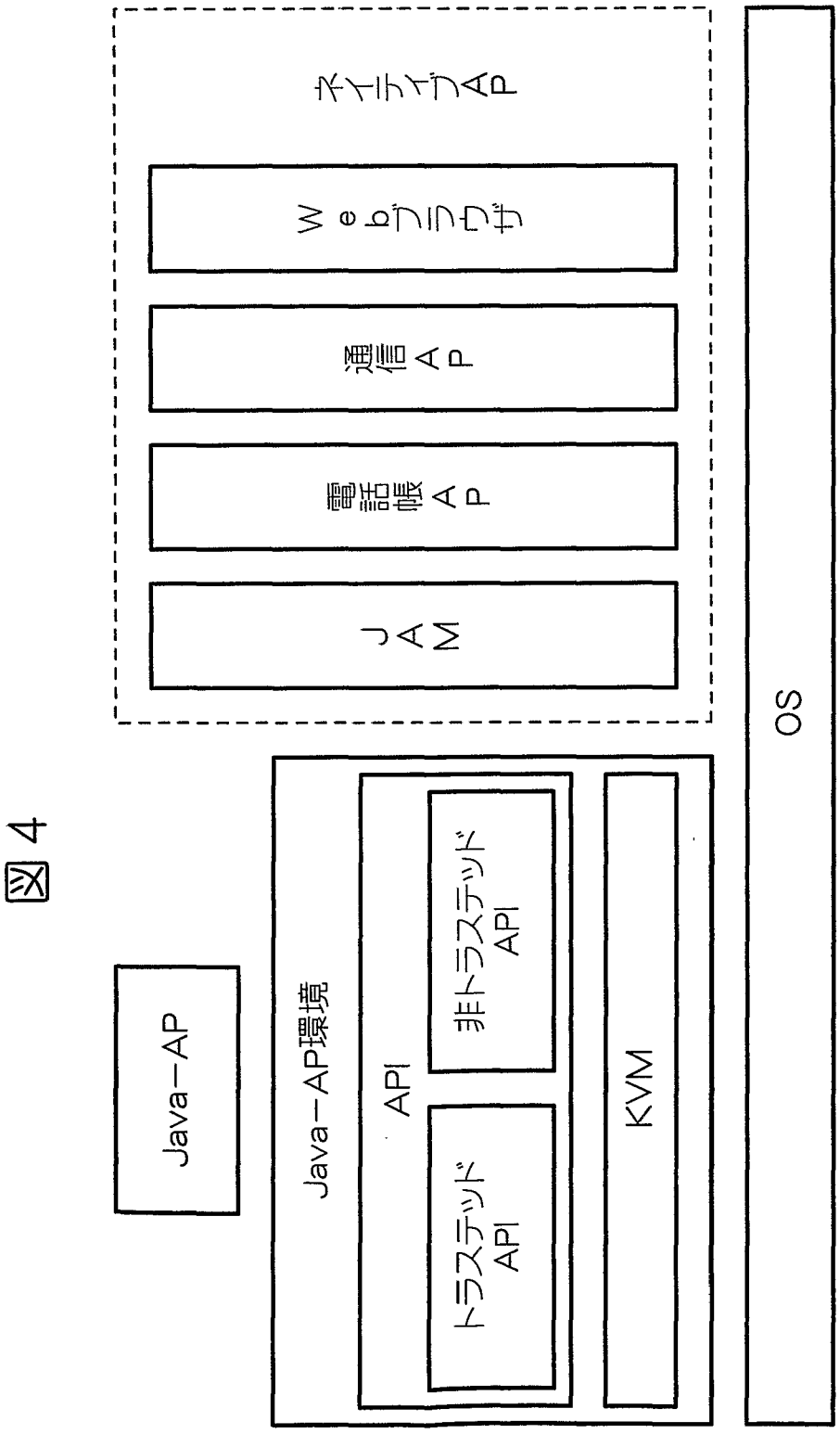

ネイティブAP
Webブラウザ
通信AP
電話帳AP
JAM
OS
Java−AP
Java−AP環境
API
非トラステッド API
トラステッド API
KVM

図 5

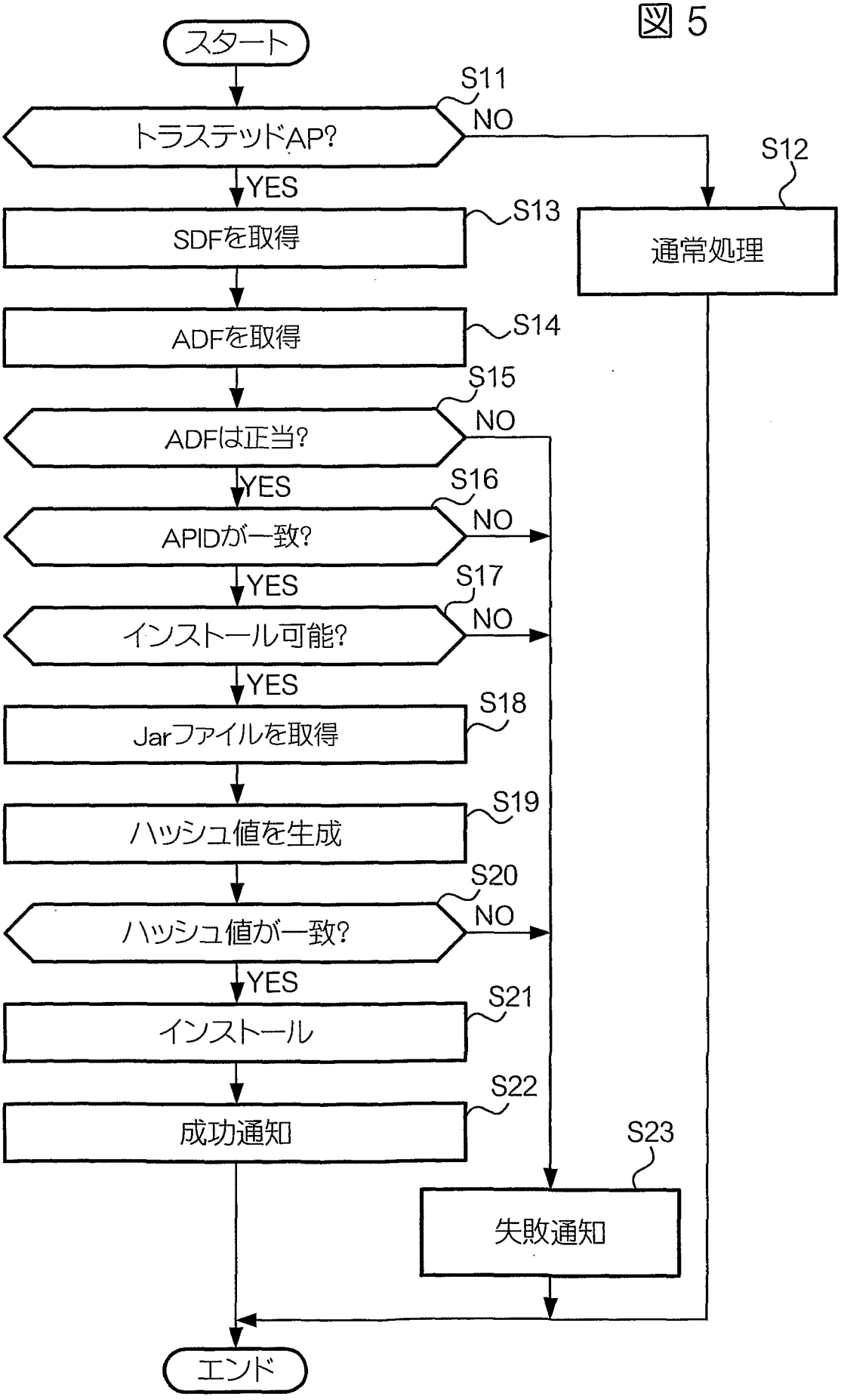

スタート
S11
トラステッドAP?
NO
S12
YES
S13
SDFを取得
通常処理
S14
ADFを取得
S15
ADFは正当?
NO
YES
S16
APIDが一致?
NO
YES
S17
インストール可能?
NO
YES
S18
Jarファイルを取得
S19
ハッシュ値を生成
S20
ハッシュ値が一致?
NO
YES
S21
インストール
S22
成功通知
S23
失敗通知
エンド

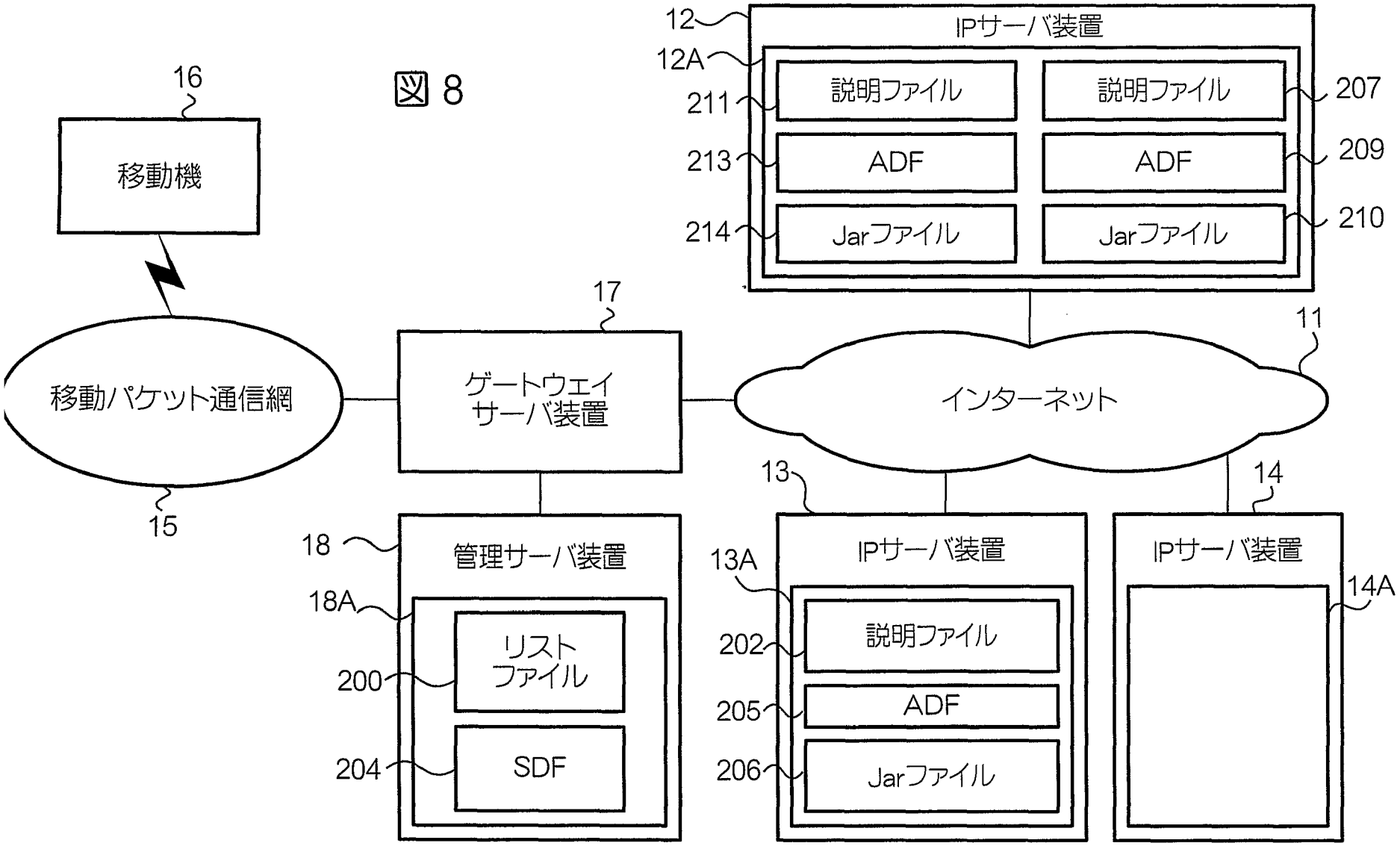
図 6

APID	ポリシー情報	ADF－URL	公開鍵

図 7

トラステッドAPI	パーミッション
getPhoneList()	○
getCallHistory()	×
getMsStatus()	○

図 8



Nintendo Exhibit 1003
Part 2 of 2
Page 00765

図 9

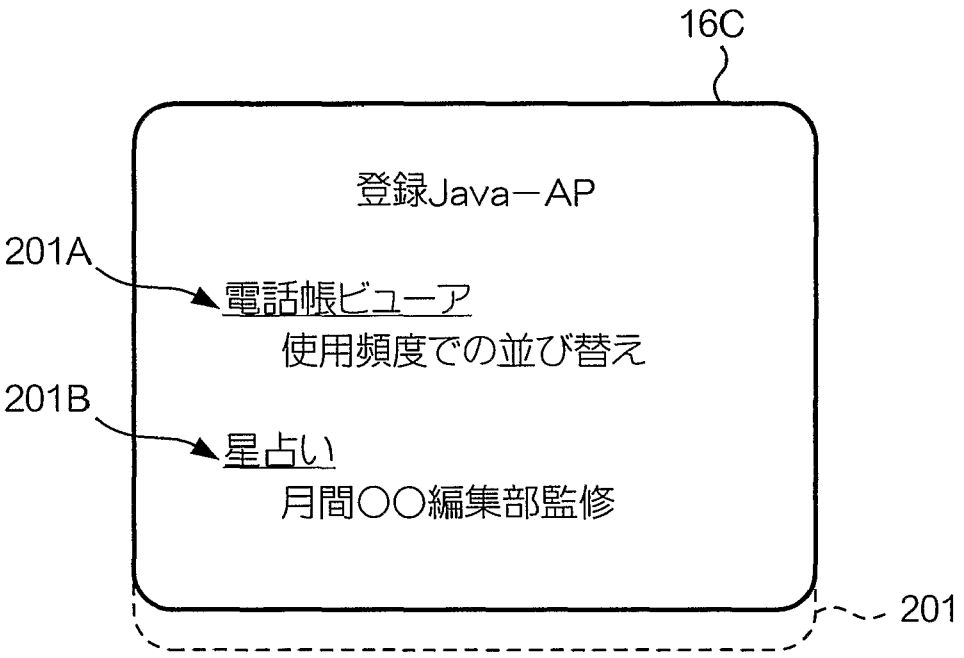


図 10

<OBJECT declare id="application.declaration"
data="http://www.ccc.co.jp/horoscope.jam>
詰め将棋
</OBJECT>
～するソフトウェアです。ダウンロードするには
<A ijam="#application.declaration">ここ</A>
をクリック。

図 11

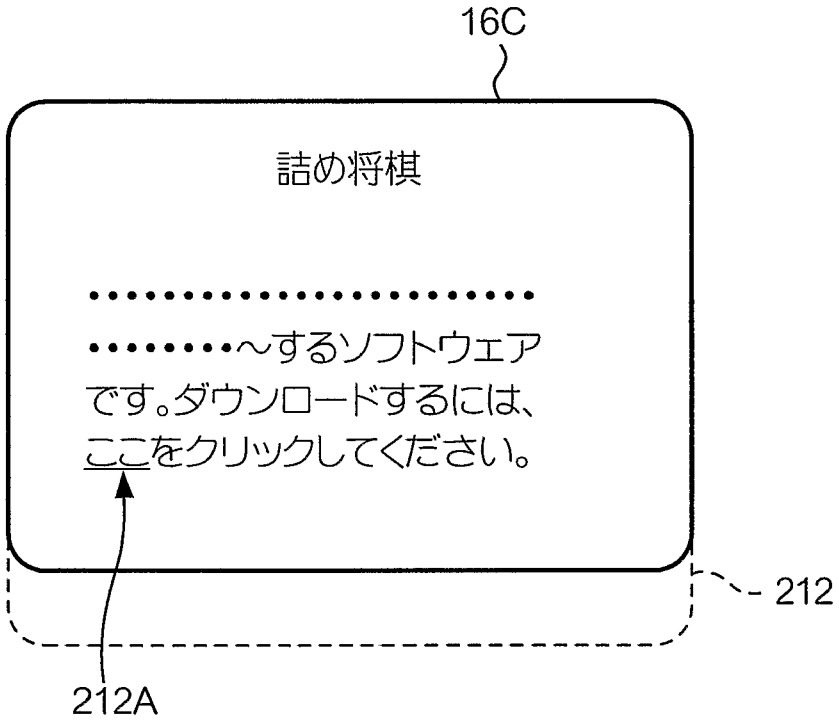


図 12

<OBJECT declare id="application.declaration"
data="http://www.ccc.co.jp/viewer.jam>
星占い
</OBJECT>
～するソフトウェアです。ダウンロードするには
<A ijam="#application.declaration">ここ</A>
をクリック。

図 13

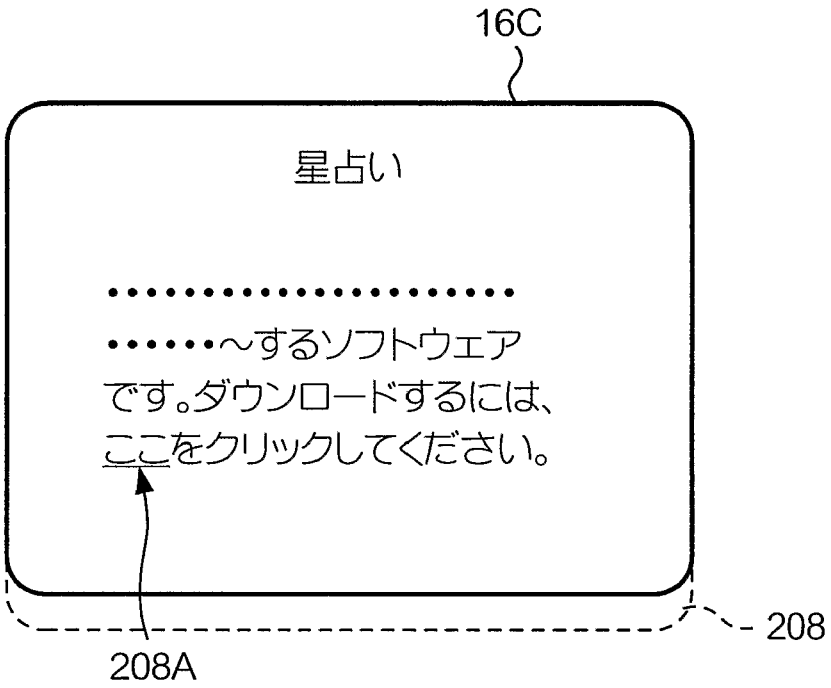


図 14

<OBJECT declare id="application.declaration"
data="http://www.aaa.co.jp/abc.sdf"
type="application/x-jam">
電話帳ビューア
</OBJECT>
～するソフトウェアです。ダウンロードするには
<A ijam="#application.declaration">ここ</A>
をクリック。

図 15

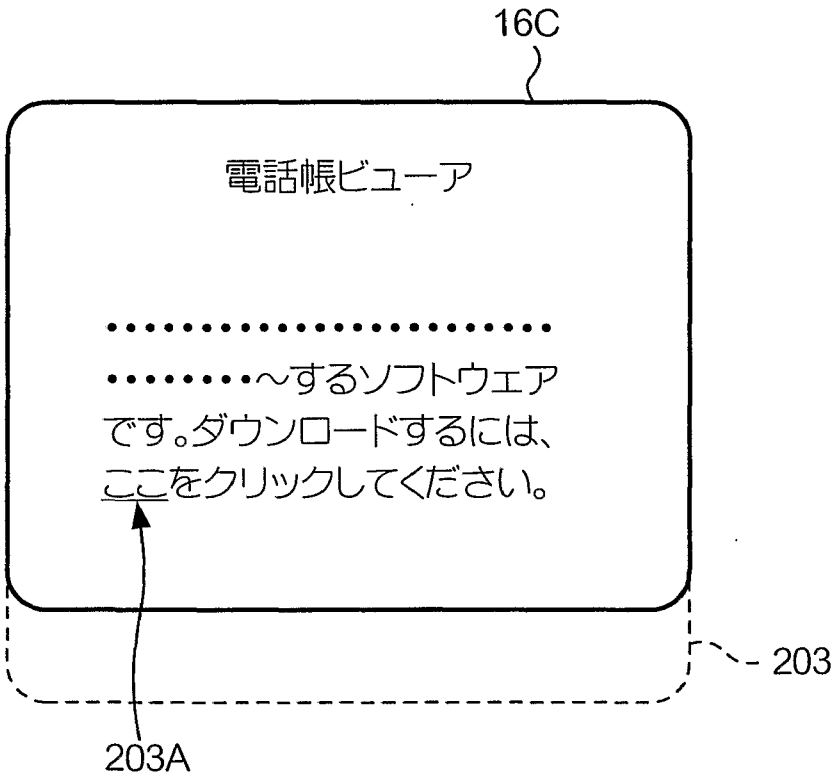

16C
電話帳ビューア
・・・・・・・・・・・・・・・・・・・・・・・・
・・・・・・・・～するソフトウェア
です。ダウンロードするには、
ここをクリックしてください。
203
203A

図 16

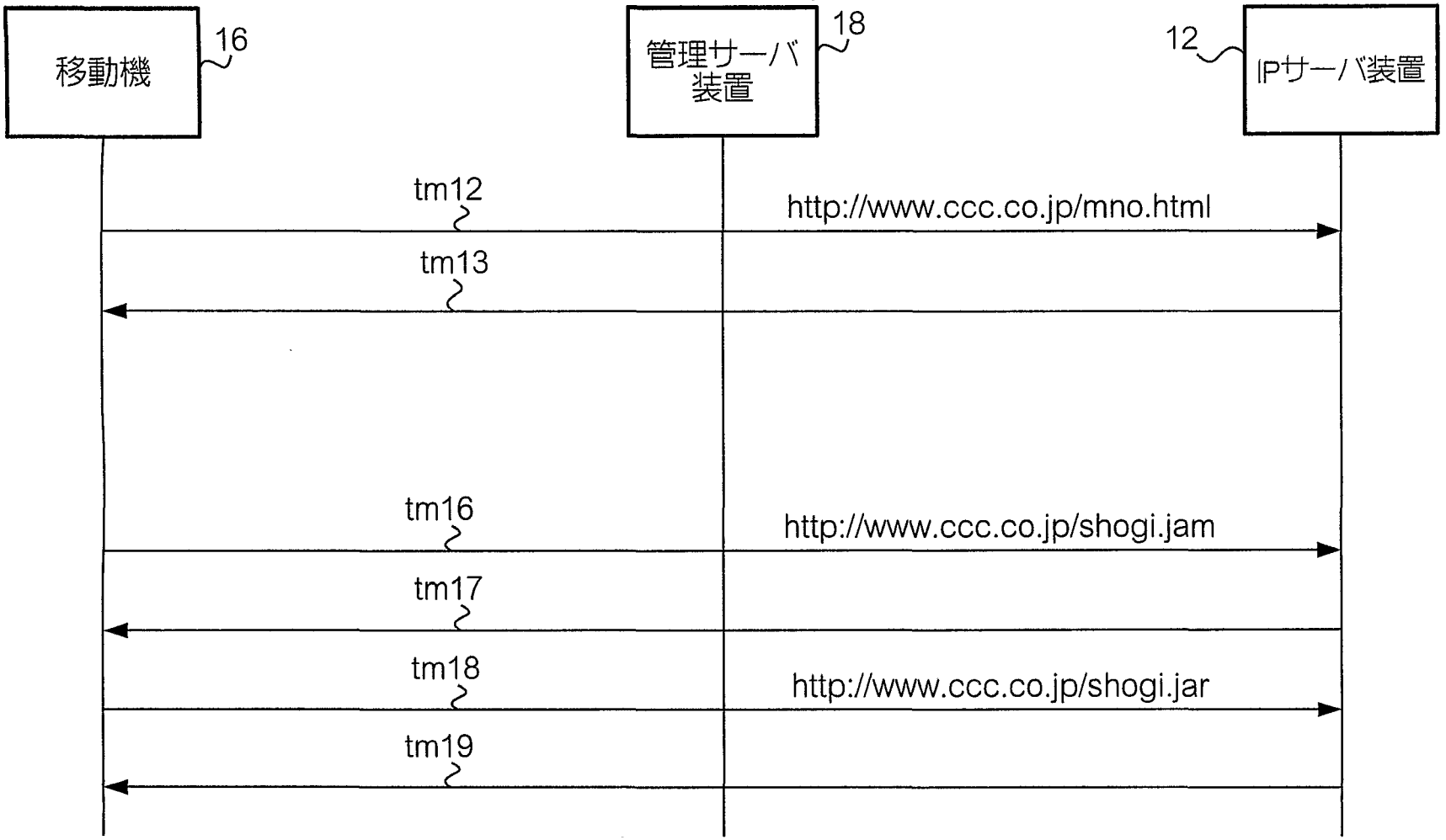

移動機
16
管理サーバ装置
18
12
IPサーバ装置
tm12
http://www.ccc.co.jp/mno.html
tm13
tm16
http://www.ccc.co.jp/shogi.jam
tm17
tm18
http://www.ccc.co.jp/shogi.jar
tm19

図 17

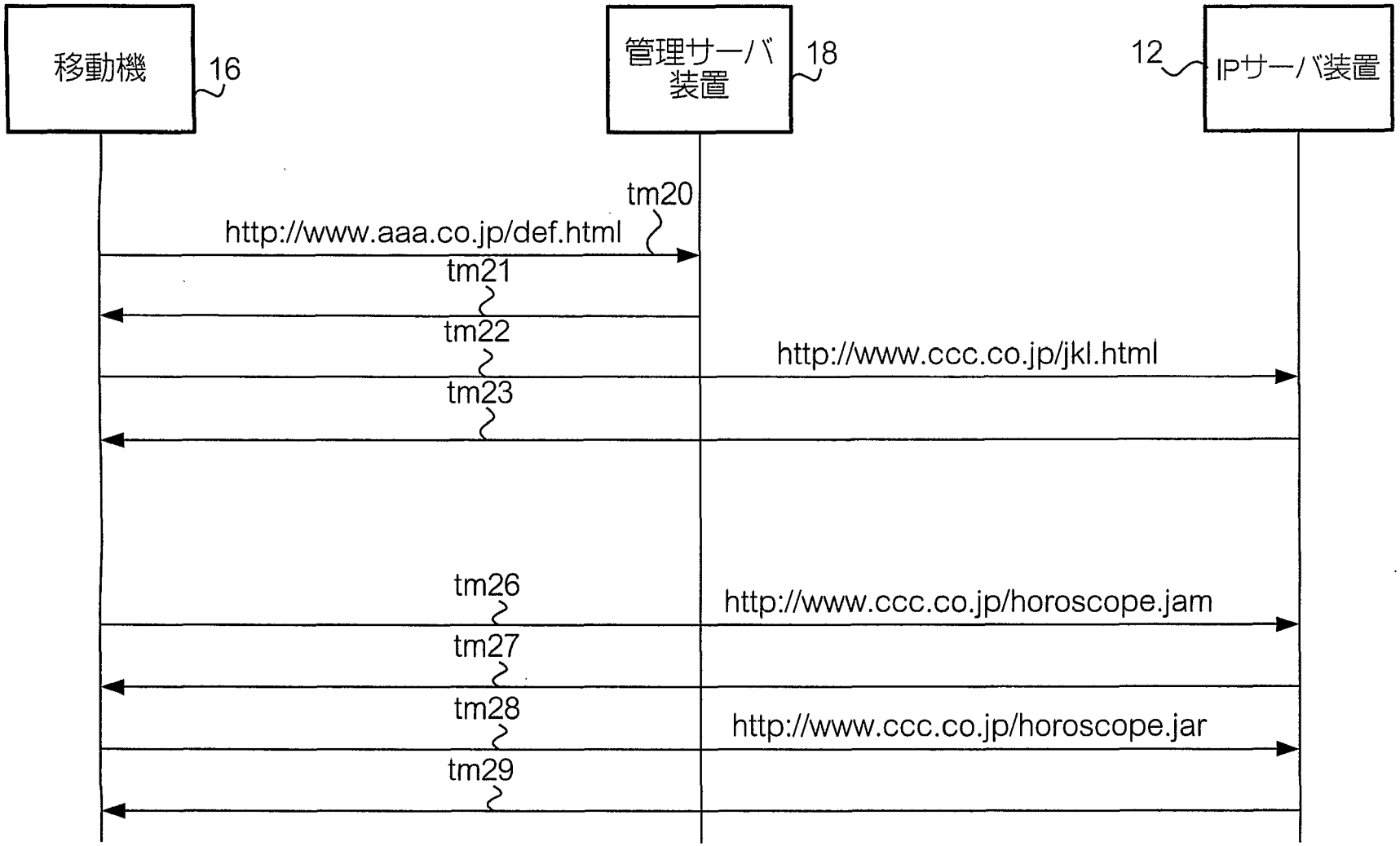
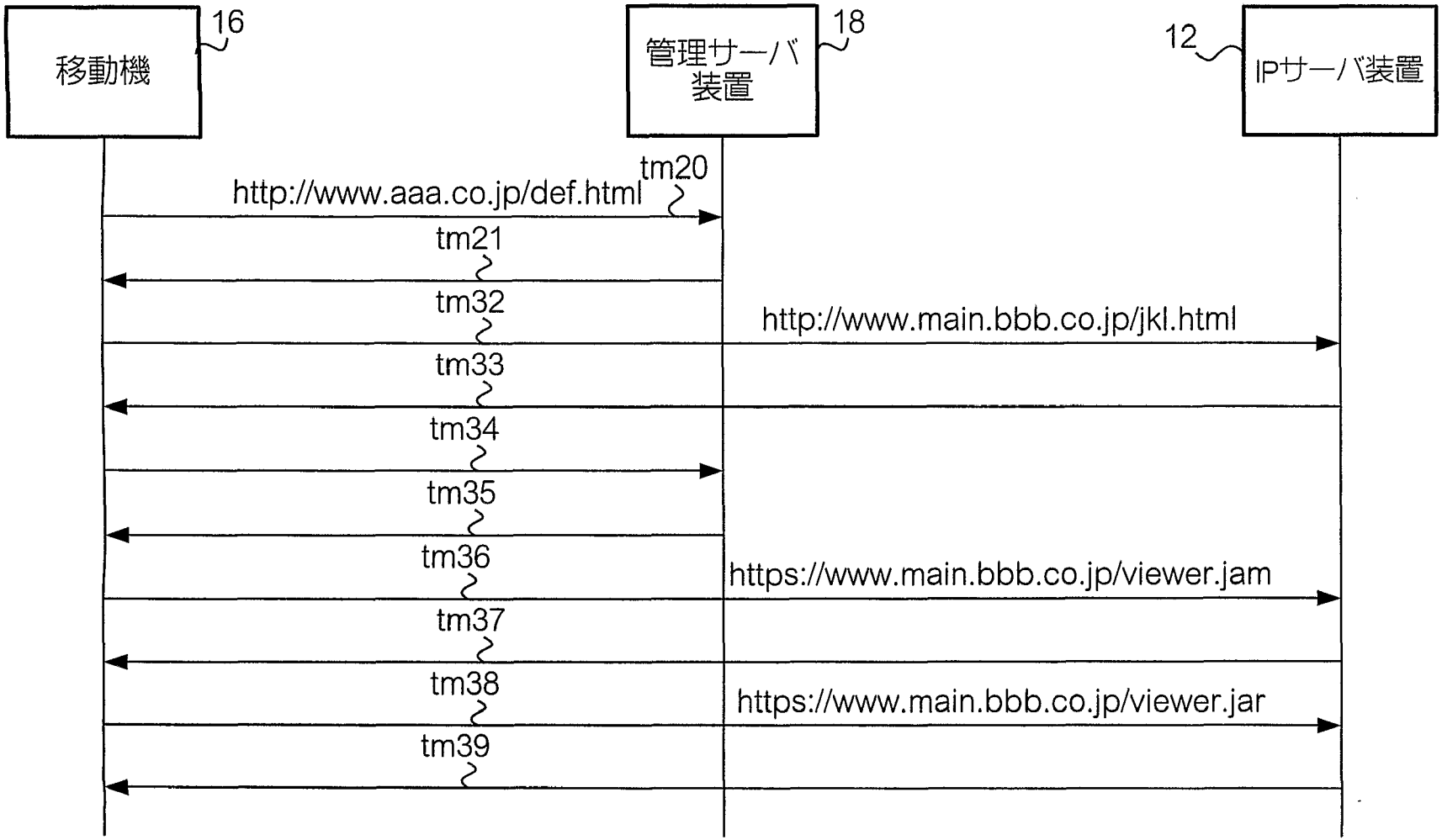
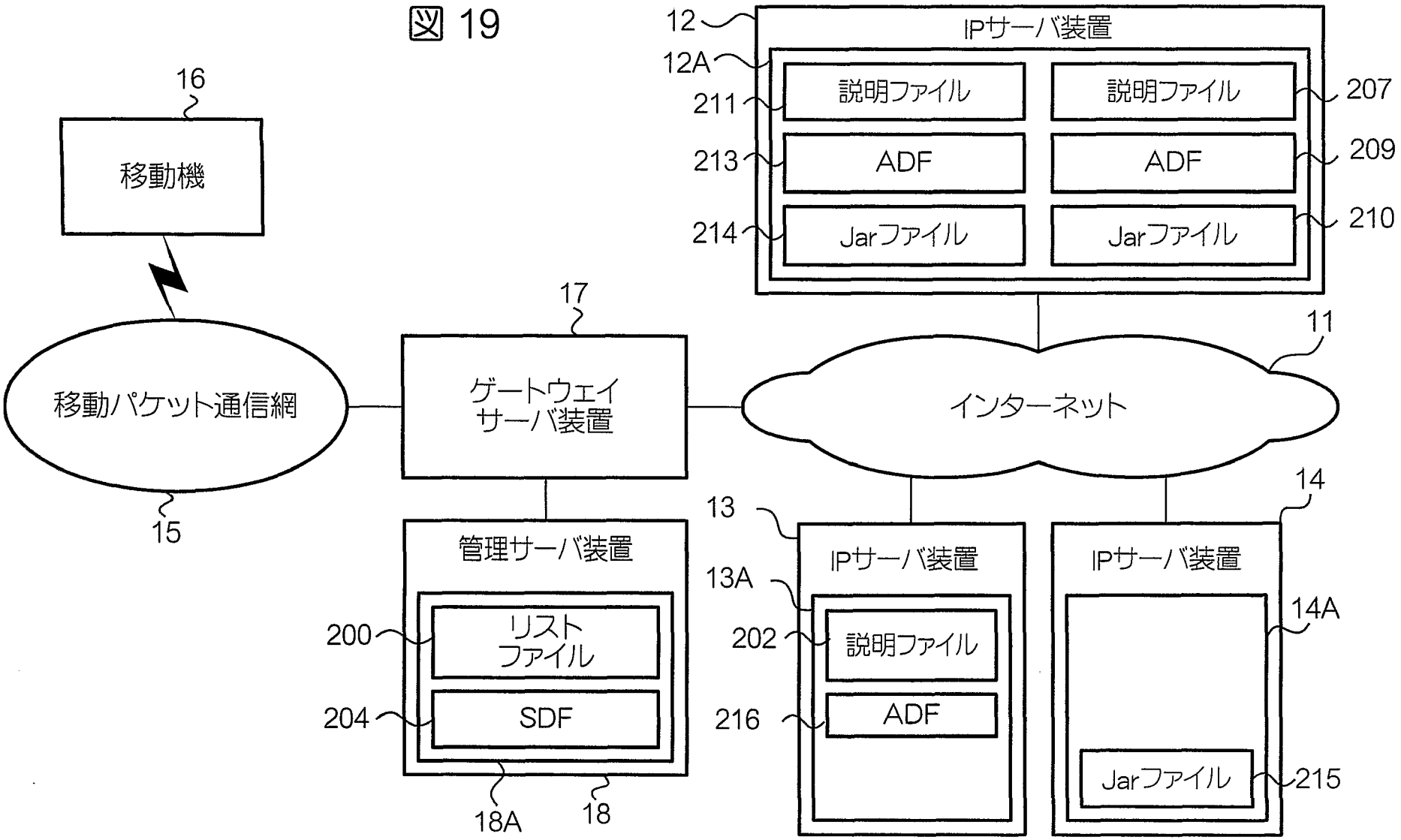

移動機
16
管理サーバ装置
18
12
IPサーバ装置
tm20
http://www.aaa.co.jp/def.html
tm21
tm22
http://www.ccc.co.jp/jkl.html
tm23
tm26
http://www.ccc.co.jp/horoscope.jam
tm27
tm28
http://www.ccc.co.jp/horoscope.jar
tm29

図 18



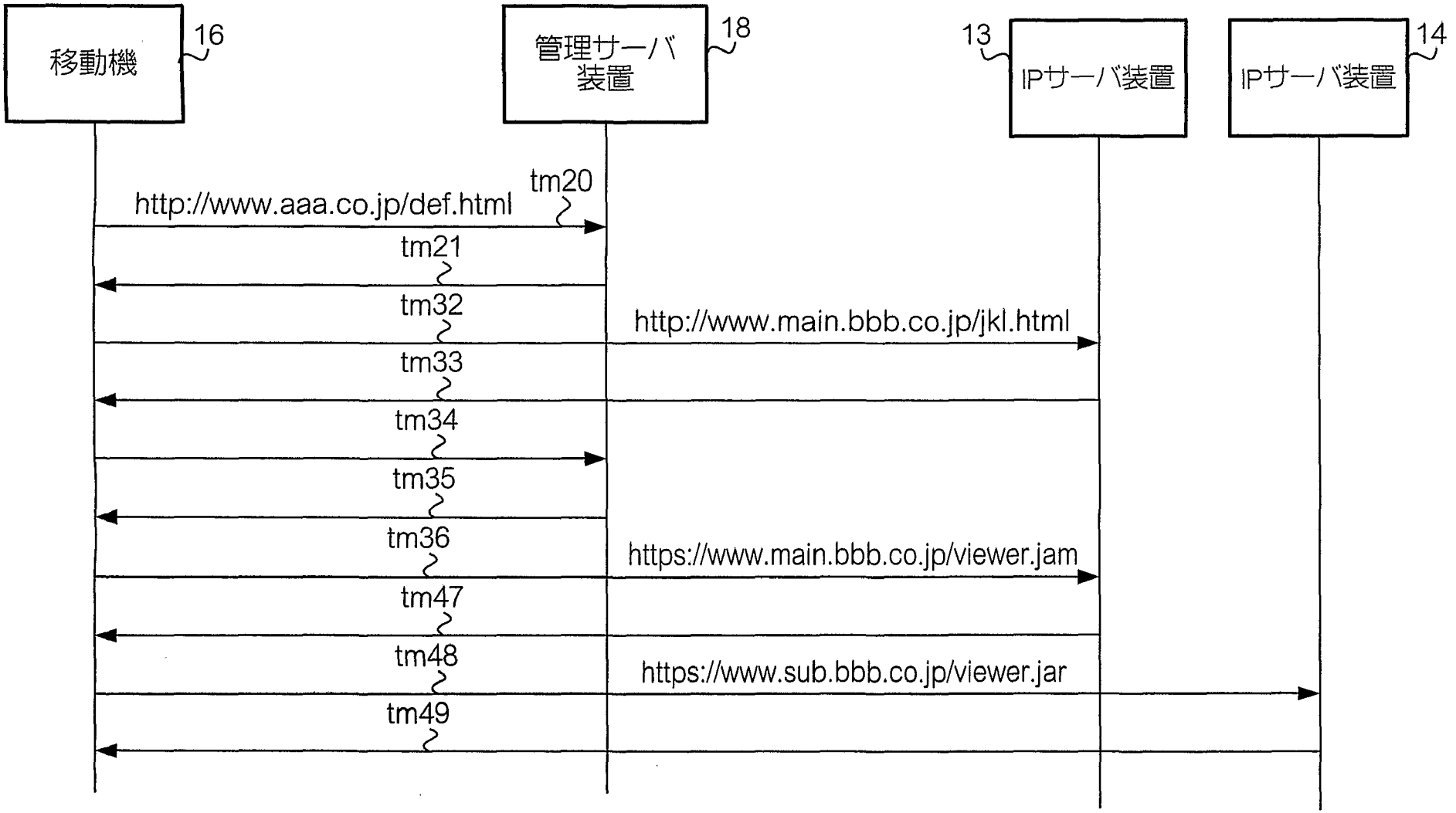
Nintendo Exhibit 1003
Part 2 of 2
Page 00772

図 19



Nintendo Exhibit 1003
Part 2 of 2
Page 00773

図 20



Nintendo Exhibit 1003
Part 2 of 2
Page 00774

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP03/00035

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl[7] G06F1/00, 9/06

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)
Int.Cl[7] G06F1/00, 9/06

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

Jitsuyo Shinan Koho	1922-1996	Toroku Jitsuyo Shinan Koho	1994-2003
Kokai Jitsuyo Shinan Koho	1971-2003	Jitsuyo Shinan Toroku Koho	1996-2003

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 98/021683 A2 (FINJAN SOFTWARE, LTD.), 22 May, 1998 (22.05.98), Full text; all drawings & WO 99/035583 A2 & US 6092194 A & US 6154844 A & US 6167520 A & EP 965094 A & IL 129729 D & JP 2003-514326 A Full text; all drawings	1-21
A	WO 00/042498 A1 (Hitachi, Ltd.), 20 July, 2000 (20.07.00), Full text; all drawings & AU 1889699 A	1-21

☒ 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 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
03 April, 2003 (03.04.03)	15 April, 2003 (15.04.03)
Name and mailing address of the ISA/ Japanese Patent Office	Authorized officer

Nintendo Exhibit 1003
Part 2 of 2
Page 00775

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP03/00035

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6317742 B1 (Sun Microsystems, Inc.), 13 November, 2001 (13.11.01), Full text; all drawings & EP 853279 A2 & SG 085092 A & JP 10-254783 A Full text; all drawings	1-21
A	JP 2001-243062 A (Nippon Telegraph And Telephone Corp.), 07 September, 2001 (07.09.01), Full text; all drawings (Family: none)	1-21
A	JP 2001-117769 A (Matsushita Electric Industrial Co., Ltd.), 27 April, 2001 (27.04.01), Full text; all drawings (Family: none)	1-21
A	Tetsuya KAKU, Masahiro YAMADA, Hiroaki ITO, "Hajimete no i-mode Java Programming Junbi kara Haifu made 'i Appli' Kaihatsu no Subete", first edition, Introduction to Java Programming in i-mode, 26 March, 2001 (26.03.01), pages 37 to 41	1-21
A	"21 Seiki ni Mukete DoCoMo no Mobile Multimedia no Sekai i-mode o Sarani Omoshiroku, 'i Appli' Service & 503i Series", Business Communication, 01 February, 2001 (01.02.01), Vol.38, No.2, pages 44 to 47	1-21
A	Jemie Jaworski, "Java 2 Security Programming Kiso Gainen kara Jisso no Shosai made Shoban", Java Security Handbook, 1st Edition, 25 April, 2001 (25.04.01), pages 79 to 90, 529 to 541	1-21
E,A	JP 2003-50641 A (NEC Corp.), 21 February, 2003 (21.02.03), Full text; all drawings (Family: none)	1-21

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ０３／００００３５

Ａ．　発明の属する分野の分類（国際特許分類（ＩＰＣ））
Ｉｎｔ.Ｃｌ⁷　Ｇ０６Ｆ１／００，９／０６

Ｂ．　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））
Ｉｎｔ.Ｃｌ⁷　Ｇ０６Ｆ１／００，９／０６

最小限資料以外の資料で調査を行った分野に含まれるもの
日本国実用新案公報　　１９２２－１９９６年
日本国公開実用新案公報　　１９７１－２００３年
日本国登録実用新案公報　　１９９４－２００３年
日本国実用新案登録公報　　１９９６－２００３年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

Ｃ．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
A	WO 98/021683 A2(FINJAN SOFTWARE,LTD.) 1998.05.22,全文,全図 & WO 99/035583 A2 & US 6092194 A & US 6154844 A & US 6167520 A & EP 965094 A & IL 129729 D & JP 2003-514326 A 全文,全図	1-21
A	WO 00/042498 A1(株式会社日立製作所 HITACHI,LTD.) 2000.07.20,全文,全図 & AU 1889699 A	1-21

☒ Ｃ欄の続きにも文献が列挙されている。　　☐ パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー
「Ａ」特に関連のある文献ではなく、一般的技術水準を示すもの
「Ｅ」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「Ｌ」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「Ｏ」口頭による開示、使用、展示等に言及する文献
「Ｐ」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献
「Ｔ」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「Ｘ」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Ｙ」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「＆」同一パテントファミリー文献

国際調査を完了した日　０３．０４．０３

国際調査報告の発送日　15.04.03

国際調査機関の名称及びあて先
日本国特許庁（ＩＳＡ／ＪＰ）
郵便番号１００－８９１５
東京都千代田区霞が関三丁目４番３号

特許庁審査官（権限のある職員）　５Ｂ　９４６２
中野　裕二
印

電話番号　０３－３５８１－１１０１　内線　３５４５

様式ＰＣＴ／ＩＳＡ／２１０（第２ページ）（１９９８年７月）

Nintendo Exhibit 1003
Part 2 of 2
Page 00777

国際調査報告	国際出願番号　ＰＣＴ／ＪＰ０３／００００３５

C（続き）．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
A	US 6317742 B1(Sun Microsystems, Inc.) 2001.11.13, 全文, 全図 & EP 853279 A2 & SG 085092 A & JP 10-254783 A 全文, 全図	1-21
A	JP 2001-243062 A(日本電信電話株式会社) 2001.09.07, 全文, 全図(ファミリーなし)	1-21
A	JP 2001-117769 A(松下電器産業株式会社) 2001.04.27, 全文, 全図(ファミリーなし)	1-21
A	加来 徹也, 山田 昌宏, 伊藤 広明, はじめてのｉモードＪａｖａプログラミング 準備から配布まで「ｉアプリ」開発のすべて 1版 Introduction to Java programming in i-mode, 2001.03.26, p37-41	1-21
A	21世紀に向けて ＤｏＣｏＭｏのモバイルマルチメディアの世界 ｉモードをさらに面白く、「ｉアプリ」サービス＆５０３ｉシリーズ，ビジネスコミュニケーション, 2001.02.01, 第38巻, 第2号, p44-47	1-21
A	ジョウォルスキー ジェミー Jemie Jaworski, Ｊａｖａ２セキュリティプログラミング 基礎概念から実装の詳細まで 初版 Java Security Handbook, 1st Edition, 2001.04.25, p79-90, 529-541	1-21
EA	JP 2003-50641 A(日本電気株式会社) 2003.02.21, 全文, 全図(ファミリーなし)	1-21

Nintendo Exhibit 1003
Part 2 of 2
Page 00778

WO0171590

Publication Title:

COMMUNICATION SYSTEM

Abstract:

Abstract of WO 0171590

(A1) A communication system enabling safe, simple electronic commerce. A sales management center (300) sends commodity information for commodity introduction to a relay terminal (400). The relay terminal (400) forwards the commodity information to a user terminal (500). If a purchaser orders a desired commodity by operating the user terminal (500), the user terminal (500) creates an order electronic message according to commodity information and sends it to a central management center (100). The central management center (100) carries out electronic settlement with a settlement center (200) in response to the order electronic message and sends a processing result message describing the result of the settlement to the user terminal (500) and the sales management center (300).; Since a closed communication system is set up, leakage of personal information to the third party can be prevented, and excellent taffeur resistance is exhibited.

Courtesy of http://v3.espacenet.com

This Patent PDF Generated by Patent Fetcher(R), a service of Stroke of Color, Inc.

(12)特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2001 年9 月27 日 (27.09.2001)

PCT

(10) 国際公開番号
WO 01/71590 A1

(51) 国際特許分類[7]: G06F 17/60

(21) 国際出願番号: PCT/JP01/02274

(22) 国際出願日: 2001 年3 月22 日 (22.03.2001)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願2000-80453 2000 年3 月22 日 (22.03.2000) JP

(71) 出願人 (米国を除く全ての指定国について): バサラ・リサーチ株式会社 (VASARA RESEARCH INC.) [JP/JP]; 〒150-0001 東京都渋谷区神宮前6-10-11 原宿ソフィアビル7階 Tokyo (JP).

(72) 発明者; および
(75) 発明者/出願人 (米国についてのみ): 永島道夫 (NAGASHIMA, Michio) [JP/JP]. 森永和美 (MORINAGA, Kazumi) [JP/JP]. 藤井克磨 (FUJII, Katsuma) [JP/JP]. 生越昌己 (OGOSHI, Masami) [JP/JP]; 〒150-0001 東京都渋谷区神宮前6-10-11 原宿ソフィアビル7階 バサラ・リサーチ株式会社内 Tokyo (JP). 松本高明 (MATSUMOTO, Takaaki) [JP/JP]; 〒165-0034 東京都中野区大和町4-14-3 Tokyo (JP).

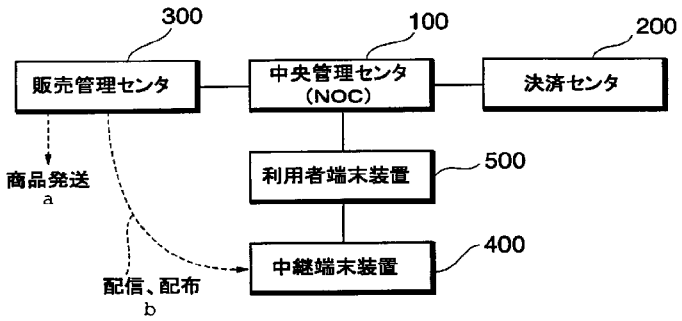
(74) 代理人: 弁理士 小橋信淳(KOBASHI, Nobukiyo); 〒150-0022 東京都渋谷区恵比寿南1丁目6番10号 恵比寿MFビル14号館4階 小橋特許事務所 Tokyo (JP).

(81) 指定国 (国内): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ,

[続葉有]

(54) Title: COMMUNICATION SYSTEM

(54) 発明の名称: 通信システム



300...SALES MANAGEMENT CENTER
100...CENTRAL MANAGEMENT CENTER (NOC)
200...SETTLEMENT CENTER
a...DELIVERY OF COMMODITY
b...SENDING
500...USER TERMINAL
400...RELAY TERMINAL

(57) Abstract: A communication system enabling safe, simple electronic commerce. A sales management center (300) sends commodity information for commodity introduction to a relay terminal (400). The relay terminal (400) forwards the commodity information to a user terminal (500). If a purchaser orders a desired commodity by operating the user terminal (500), the user terminal (500) creates an order electronic message according to commodity information and sends it to a central management center (100). The central management center (100) carries out electronic settlement with a settlement center (200) in response to the order electronic message and sends a processing result message describing the result of the settlement to the user terminal (500) and the sales management center (300). Since a closed communication system is set up, leakage of personal information to the third party can be prevented, and excellent taffeur resistance is exhibited.

[続葉有]

WO 01/71590 A1

PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) 指定国 (広域): ARIPO 特許 (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), ユーラシア特許 (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI 特許 (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:
— 国際調査報告書

２文字コード及び他の略語については、定期発行される各*PCT*ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(57) 要約:

本発明の通信システムは、安全且つ簡易な電子商取引を可能にする通信システムを提供するものである。

販売管理センタ３００から中継端末装置４００に商品紹介用の商品情報を配信又は配布する。中継端末装置４００は配信又は配布された商品情報を利用者端末装置５００へ転送する。購入者が利用者端末装置５００を操作して所望の商品を注文すると、利用者端末装置５００が商品情報に基づいて注文電文を生成して中央管理センタ１００に送信する。中央管理センタ１００は注文電文に基づいて決済センタ２００との間で電子決済を行い、その決済の結果を示す処理結果電文を利用者端末装置５００と販売管理センタ３００に対して通知する。閉じられた通信システムが実現されることから、個人情報等の第三者への漏洩を未然に防止しし得る優れた耐タフー性を発揮する。

【選択図】　第１図

明 細 書

通信システム

技術分野

本発明は、例えば電子商取引等を行うための通信システムに関する。

技術背景

従来、新聞や、カタログ、ラジオ放送、テレビジョン放送などのメディアを通じて商品情報を提供し、その商品情報に基づいて商品購入の注文を受ける通信販売の方式が一般的に知られている。

例えば、テレビジョン放送による通信販売にあっては、販売者は、販売しようとする商品の情報と、注文受付用の電話やファクシミリの番号を放送し、一方、購入者は、電話やファクシミリによって希望の商品を指定して注文するようになっている。

そして、販売者は、購入者の代金支払いを確認した後、購入者に対して商品を発送するという方法が採られている。

また、販売者が購入者へ商品の引き渡しを行った後、購入者が販売者へ代金支払いを行うようにする方法や、販売者から依託された配送業者が購入者に商品を引き渡すのと引き替えに購入者から代金を受け取るという方法などが採られている。

また、代金の支払い方法として、現金による場合の他、銀行などの金融機関に振り込む方法や、購入者が所有するクレジットカードによって支払う方法などが知られている。

ところが、従来の通信販売では、購入者は注文の際に、氏名や住所、電話番号などの個人情報を販売者に提示する必要があった。このため、購入者の個人情報が悪意の第三者に漏洩して悪用されやすいなどの問題があった。

Nintendo Exhibit 1003
Part 2 of 2
Page 00782

また、購入者がクレジットカードなどを用いて支払いを行う場合にも、購入者は販売者や依託された配送業者に対して、クレジットカードの番号などの秘密性の高い個人情報を提示しなければならなかった。このため、悪意の第三者に個人情報が漏洩して悪用されやすいなどの問題があった。

また、購入者が金融機関に代金を振り込む場合には、販売者が実際に代金を受け取るまでに時間がかるため、販売者にとって営業上の負担を強いられる場合が生じるという問題があった。

本発明はこうした従来の問題点を克服し、安全な電子商取引などを可能にする通信システムを提供することを目的とするものである。

発明の開示

請求項１に係る本発明の通信システムは、電子決済の処理をする中央管理手段と、商品を販売する販売者が所有する販売管理手段と、商品を購入する購入者が所有する利用者端末手段とを備え、上記販売者が配布する商品情報の中から上記購入者が上記利用者端末装置を用いて所望の商品を上記中央管理手段に対して注文すると、上記中央管理手段が注文に応じて電子決済し、電子決済の処理結果を上記販売管理手段と上記利用者端末手段に通知することを特徴とする。

請求項２に係る本発明の通信システムは、請求項１に係る発明において、上記販売管理手段は、上記販売者が販売する商品の情報を利用者端末装置に提供し、上記利用者端末装置は、上記購入者の注文に応じて、上記提供された商品の情報に基づいて注文電文を生成して上記中央管理手段に送信し、上記中央管理手段は、上記送信された注文電文に基づいて上記電子決済を行うと共に、上記電子決済の処理結果を上記販売管理手段と上記利用者端末手段に通知することを特徴とする。

請求項３に係る本発明の通信システムは、請求項１又は請求項１に係る発明において、上記販売管理手段は、上記購入者が所有する中継

Nintendo Exhibit 1003
Part 2 of 2
Page 00783

端末装置に上記商品の情報を提供し、上記中継端末装置は、提供された上記商品の情報を上記利用者端末装置に提供することを特徴とする。

これら請求項１，２，３に係る発明の通信システムによると、中央管理手段が注文に応じて電子決済し、その電子決済の処理結果を販売者が所有する販売管理手段と購入者が所有する利用者端末手段との両方に通知する。

従来のように、購入者が販売者に対して商品の申し込みを行い、販売者が購入者に対して商品を渡したときに、販売者と中央管理センタ等の決済センタで電子決済を行ったのでは、決済処理の内容を購入者が知り得る状態とはならないため、購入者の個人情報が漏洩しているのか否か知ることができないが、本通信システムによると、中央管理手段の電子決済の処理結果が販売管理手段と利用者端末手段との両方に通知されるので、利用者端末手段を所有している購入者は、個人情報の安全確保を図るべく確認をすることができ、更に適切な電子決済が成されたか否かを知ることができる。

つまり、販売者が所有する販売管理手段と電子決済をする中央管理手段との間に、利用者端末装置が介在した通信システム構成が実現されることになるため、購入者は、販売管理手段から提供される商品情報に基づいて、利用者端末装置を用いて中央管理手段に対して、安心して商品の注文をすることが可能になる。

更に、販売管理手段にも電子決済の行われた結果が中央管理手段から通知されることで、適切な電子決済が成されたか否かことを知ることができ、適切な電子決済が成された場合に、注文された商品を購入者に配送することが可能になる。

請求項４に係る本発明の通信システムは、電子決済の処理をする中央管理手段と、商品を販売する販売者が所有する販売管理手段と、商品を購入する購入者が所有する利用者端末手段と、販売する商品の情報を上記販売管理手段から供給される中継端末装置とを備え、上記販

Nintendo Exhibit 1003
Part 2 of 2
Page 00784

売者が配布する商品情報の中から上記購入者が上記販売管理手段に対して所望の商品を注文するのに応じて、上記販売者が購入者に商品を配送する際に、配送すべき商品に付された商品情報を上記中継端末手段が読み取って上記利用者端末装置に送信し、上記利用者端末装置は送信された上記商品情報に基づいて上記中央管理手段に注文の情報を転送し、上記中央管理手段は、注文の情報に応じて電子決済し、電子決済の処理結果を上記販売管理手段と上記利用者端末手段に通知することを特徴とする。

請求項４に係る発明の通信システムによれば、購入者から販売者に商品の注文が行われて、注文された商品を購入者に引き渡す際に、商品に付されている商品情報を中継端末装置で読み取ることで、その読み取った商品情報が利用者端末装置に転送される。そして、利用者端末装置は転送されてきた商品情報に基づいて中央管理手段に対して商品の注文をすることになる。

このため、販売者が所有する販売管理手段と電子決済をする中央管理手段との間に、利用者端末装置が介在した通信システム構成が実現されることになるため、購入者は個人情報が漏洩するという問題を回避して所望の商品を注文することができる。

更に、電子決済の行われた結果が中央管理手段から利用者端末装置に通知されることで、購入者は、適切な電子決済が成されたか否かことを知ることができる。

更に、販売管理手段にも電子決済の行われた結果が中央管理手段から利用者端末装置に通知されることで、販売者側も、適切な電子決済が成されたか否かことを知ることができる。このため、販売者にとっても、注文された商品が購入者に引き渡されたことと、適切に電子決済が成されたことを知ることができ、極めて安全性の高い通信システムが実現される。

Nintendo Exhibit 1003
Part 2 of 2
Page 00785

図面の簡単な説明

第１図は、発明を実施するための最良の形態の通信システムの基本構成を示すブロック図である。

第２図は、第１図に示すの中央管理センタの構成を示すブロック図である。

第３図は、第１図に示す販売管理センタの構成を示すブロック図である。

第４図は、第１図に示す中継端末装置の構成を示すブロック図である。

第５図は、第１図に示す利用者端末装置の構成を示すブロック図である。

第６Ａ図は、利用者端末装置の外観構成を示す平面図である。

第６Ｂ図は、変形例の利用者端末装置の外観構成を示す平面図である。

第７図は、通信システムの動作を説明するためのフローチャートである。

第８図は、通信システムの動作を更に説明するためのフローチャートである。

第９図は、通信システムの動作を更に説明するためのフローチャートである。

第１０図は、通信システムの動作を更に説明するためのフローチャートである。

第１１図は、通信システムの動作を更に説明するためのフローチャートである。

第１２図は、通信システムの動作を更に説明するためのフローチャートである。

第１３図は、通信システムの動作を更に説明するためのフローチャートである。

Nintendo Exhibit 1003
Part 2 of 2
Page 00786

第１４図は、通信システムの動作を更に説明するためのフローチャートである。

発明を実施するための最良の形態

以下、発明を実施するための最良の形態を図１ないし図１４を参照して説明する。尚、本実施の形態として、本発明の通信システムを説明するに当たり、本発明の通信システムを用いて通信販売を行う場合について説明する。

第１図は本実施の形態の通信システムの基本構成と、本通信システムを用いた通信販売の方法を示す説明図である。

第１図において、中央管理センタ１００は、ＮＯＣ（Network Operation Center）と呼ばれ、通信機能を有するコンピュータシステムを備えて構成されている。また、中央管理センタ１００は、決済センタ２００と販売管理センタ３００及び利用者端末装置５００との間で通信を行い、電子商取引のための処理を統括的に管理する。

決済センタ２００は、通信機能を有するコンピュータシステムを備えて構成され、中央管理センタ１００からの決済要求に応じて、金融機関などとの間で電子決済の処理を代行する。

販売管理センタ３００は、販売者側に設置される通信機能を有するコンピュータシステムを備えて構成され、商品を紹介するための配布情報を通信電文（以下、配布電文という）にして、インターネット等の通信網を介して中継端末装置４００へ配信する。また、電子決済の結果を中央管理センタ１００から受信するようになっている。

尚、販売者は、上記の配布電文を通信網を介して配信するだけでなく、配布電文をテレビジョン放送やラジオ放送等の放送系メディアによって配信したり、配布電文をバーコードにして、新聞や雑誌、販売カタログ、広告等の紙媒体に掲載したり、配布電文をＣＤ（Compact Disk）等の情報記録媒体に記録し、その情報記録媒体を雑誌や販売カ

Nintendo Exhibit 1003
Part 2 of 2
Page 00787

タログに添付して配布する。

つまり、販売者は、商品を紹介する手段として、インターネット等の通信系メディア、テレビジョン放送やラジオ放送等の放送系メディア、紙媒体や情報記録媒体を選択的に利用できるようになっている。

中継端末装置４００は、販売管理センタ３００と利用者端末装置５００との間に介在する中継器であり、販売管理センタ３００から通信系メディアや放送系メディアを通じて配信される配布電文を受信し、受信した配布電文を利用者端末装置５００へ転送する。また、紙媒体に掲載されているバーコードを読み取るバーコードリーダや、ＣＤ等の情報記録媒体に記録されている配布電文を読み取る再生装置を備え、読み取った情報を配布電文にして利用者端末装置５００へ転送する。

尚、上記のバーコードリーダや再生装置、通信系メディアや放送系メディアを通じて配信される配布電文を受信するための受信装置等は、購入者の希望に応じてオプション設定されるようになっている。

また、中継端末装置４００は、購入者の自宅などに設置されて利用されることを想定した端末装置であり、卓上型の端末装置として提供されたり、所謂パーソナルコンピュータの拡張スロットに接続することでパーソナルコンピュータの一部として動作させることが可能な拡張ボートとして提供されたり、テレビジョンシステムの受信機に接続されるユニットなどとして提供されるようになっている。

利用者端末装置５００は、中継端末装置４００より転送されてくる配布電文を受信し、商品を紹介するための情報を購入者に提示する。また、購入者が商品を選択して注文すると、その商品を注文するための「注文電文」を作成して中央管理センタ１００へ送信する。また、注文に応じて中央管理センタ１００が電子決済の処理を行うと、中央管理センタ１００からの電子決済に関する情報を受信し、適切な決済が行われたかの判断を行うための情報を提示する。

中央管理センタ１００と決済センタ２００の間と、中央管理センタ

Nintendo Exhibit 1003
Part 2 of 2
Page 00788

１００と販売管理センタ３００の間は、専用の通信網のほか、商用通信網を介して接続される。

また、中継端末装置４００と利用者端末装置５００の間も、無線又は有線によって接続される。購入者が中継端末装置４００に設けられている通信用の接続端子に利用者端末装置５００を接続すれば有線による通信が可能となり、有線による接続をしない場合には、両者に設けられている無線通信機能によって無線通信できるようになっている。

また、利用者端末装置５００と中央管理センタ１００の間は、利用者の利便性を考慮して移動体通信網を介して通信が行われるようになっている。但し、商用の有線通信網を介して中央管理センタ１００と通信できるようにもなっている。

次に、かかる構成の通信システムを利用した通信販売の方法を、より詳細に説明する。

まず、販売者は、上記したように、商品を紹介するための配布情報を通信系メディアや、放送系メディア、紙媒体や情報記録媒体などによって配信又は配布する。

尚、通信系メディアや放送系メディアによって配布情報を配信する場合には、「商品情報」と「販売者情報」を有する配布情報を、「配布電文」として配信する。ここで、「商品情報」は、商品名、商品の単価、商品の外観などを示すためのイメージ情報、商品を識別するために付された商品コード、予定している商品の販売個数、販売期間などの情報であり、「販売者情報」は、販売に関わっている店舗を識別するための店舗コード、販売者公開鍵の情報などの情報であり、これらの「商品情報」と「販売者情報」を結合して販売者秘密鍵で電子署名することによって「配布電文」を生成して配信する。

また、ＣＤ等の情報記録媒体を配布する場合にも、「配布電文」に販売者秘密鍵で電子署名を付けて記録する。

また、紙媒体で配布する場合には、上記の「配布電文」をバーコー

Nintendo Exhibit 1003
Part 2 of 2
Page 00789

ドの形態にして印刷等することにより、配布情報として提供する。

次に、通信系メディアによって配信された「配布電文」を中継端末装置４００が受信すると、中継端末装置４００は、その取得した「配布電文」に中継端末装置４００に予め設定されている「固有情報（利用者を識別するためのＩＤ情報など）」を付加して、利用者端末装置５００へ送信する。

また、放送メディアによって中継端末装置４００に「配布電文」が供給されると、中継端末装置４００は、その取得した「配布電文」に中継端末装置４００に予め設定されている「固有情報」を付加して、利用者端末装置５００へ送信する。

また、購入者にＣＤ等の情報記録媒体が配布される場合には、購入者がその情報記録媒体を中継端末装置４００の再生装置に装填して情報再生を行わせると、中継端末装置４００は情報記録媒体から「配布電文」を再生し、予め設定されている「固有情報」を付加して、利用者端末装置５００へ送信する。

また、購入者に販売カタログ等の紙媒体が配布される場合には、購入者が上記バーコードリーダによって「配布電文」を読み取らせると、中継端末装置４００は、読み取った「配布電文」に「固有情報」を付加して、利用者端末装置５００へ送信する。

こうして中継端末装置４００を識別するための「固有情報」の付けられた「配布電文」が中継端末装置４００から利用者端末装置５００へ送信されると、利用者端末装置５００は、受信した「配布電文」中の商品情報を表示する。つまり、「配布電文」に含まれている商品名や価格、商品コードなどの商品に関する情報を表示する。

そして、購入者がその表示を見て所望の商品を選択して注文のための操作をすると、利用者端末装置５００は、「注文電文」を作成して中央管理センタ１００側へ送信する。

より詳細に述べれば、利用者端末装置５００は、通信販売で必要と

Nintendo Exhibit 1003
Part 2 of 2
Page 00790

なる「発送先（購入者の住所など）」の情報を「販売者公開鍵」で暗号化し、購入者の希望する商品とその個数などの「注文事項」に関する情報に添付する。更に、これらの情報を中央管理センタ１００のＮＯＣ公開鍵で暗号化することによって上記の「注文電文」を作成し、中央管理センタ１００側へ送信する。

中央管理センタ１００は、「注文電文」によって購入者からの注文がなされると、適正な注文がなされたか否かなどの判断処理を行い、適正な注文であれば決済センタ２００に対して電子決済を要求する。

そして、中央管理センタ１００は、電子決済の完了を示す情報を利用者端末装置５００と販売管理センタ３００に通知することにより、購入者と販売者に対して電子決済が行われた結果を知らせる。つまり、購入者の注文とほぼ同時に電子決済が完了し、あたかも現金支払いが行われたのと同様の処理が行われる。

そして、販売管理センタ３００側の販売者は、中央管理センタ１００からの電子決済の完了を示す情報に従って、注文された商品を購入者に発送する。

詳細については後述するが、中央管理センタ１００は、利用者端末装置５００から中央管理センタ１００側へ送信される上記の「注文電文」中に含まれている「発送先」の情報を、決済完了を示す情報と共に販売管理センタ３００に送信する。このため、販売管理センタ３００は注文主である購入者の住所などを知ることができ、購入者の住所（発送先）に商品を発送できるようになっている。

このように、この通信システムによれば、従来の通信販売と同様に、販売管理センタ３００から利用者（購入者）に、商品を紹介するための配布情報を広範に配信又は配布するという特徴を有している。

そして、配信又は配布された情報に基づいて購入者が利用者端末装置５００を用いて中央管理センタ１００に注文を出すと、販売者に関する情報と購入者に関する情報とが含まれた「注文電文」が中央管理

Nintendo Exhibit 1003
Part 2 of 2
Page 00791

センタ１００に送信され、中央管理センタ１００と決済センタ２００が決済処理を行って、その決済完了の情報を販売管理センタ３００と利用者端末装置５００に通知するので、守秘義務を有する中央管理センタ３００の立ち会いの下で、販売者と購入者とがほぼ同時に商取引を行うことが可能となる。

これにより、販売者と購入者はほぼ同時に取引の状況を知ることができるため、両者間における代金支払いのトラブルを未然に防止することができる。

また、従来の銀行振り込みやクレジットカードによる代金支払いでは、決済に長時間を要する場合が生じるが、本通信システムでは即時に決済が行われるため、販売者に対して営業上の負担を強いるという問題を解消することができる。

また、購入者は、クレジットカードの番号などの秘密性の高い個人情報を販売者に提示する必要がないので、不正な第三者が個人情報を入手するなどの問題を未然に防止することができる。つまり、本通信システムでは、販売者側に位置する販売管理センタ３００と中央管理センタ１００の間に、購入者が所有する利用者端末装置５００が介在し、利用者端末装置５００から中央管理センタ１００に直接注文するようになっている。このため、購入者の個人情報が第三者に漏洩することを未然に防止できる。

また、中央管理センタ１００が決済完了の情報を販売管理センタ３００と利用者端末装置５００に通知するので、中央管理センタ１００と利用者端末装置５００との間でのみ決済が行われるというような販売者にとって不利な状況を招くことがない。このため、販売者にとって極めて安全な通信販売の方法を提供することができる。

次に、本通信システムの構成と動作の詳細を、第２図ないし第１４図を参照して説明する。

まず、第２図に基づいて中央管理センタ１００の基本的な構成を説

Nintendo Exhibit 1003
Part 2 of 2
Page 00792

明する。中央管理センタ１００は、制御部１０、メモリ部１１、公開鍵データベース１２、鍵管理部１３、通信処理部１４、送受信部１５、通信処理部１６、送受信部１７を備えて構成されている。

ここで、制御部１０は、中央管理センタ１００全体を集中管理する。メモリ部１１は、決済データ等を記憶する。公開鍵データベース１２は、公開鍵データを記憶し管理する。鍵管理部１３は、通信に関わるデータの暗号化と復号処理等を行う。

通信処理部１４は、制御部１０から供給されるデータを変調して送受信部１５へ供給し、また、送受信部１５から供給されるデータを復調して制御部１０へ供給する。送受信部１５は、通信処理部１４からの変調データを移動体通信網ＮＴ１へ送信し、また、移動体通信網ＮＴ１からのデータを受信して通信処理部１４へ転送する。つまり、通信処理部１４と送受信部１５は、移動体通信網ＮＴ１を介して利用者端末装置５００と通信するために設けられている。

通信処理部１６は、制御部１０から供給されるデータを変調して送受信部１７へ供給し、また、送受信部１７からのデータを復調して制御部１０へ供給する。

送受信部１７は、通信処理部１６からのデータをローカルエリアネットワーク（Local Area Network：ＬＡＮ）等の通信網ＮＴ２へ送信し、また、通信網ＮＴ２からのデータを受信して通信処理部１６へ転送する。また、送受信部１７は、通信処理部１６からのデータを専用通信網やインターネット等の通信網ＮＴ３へ送信する。

つまり、通信処理部１６と送受信部１７は、通信網ＮＴ２を介して決済センタ２００と通信すると共に、通信網ＮＴ３を介して販売管理センタ３００と通信するために設けられている。

次に、第３図に基づいて販売管理センタ３００の基本的な構成を説明する。販売管理センタ３００は、制御部３０、メモリ部３１、鍵管理部３２、通信処理部３３，３５、送受信部３４，３６、表示部３７

Nintendo Exhibit 1003
Part 2 of 2
Page 00793

を備えて構成されている。

ここで、制御部３０は、販売管理センタ３００全体を集中管理する。メモリ部３１は、演算データや商品紹介用の配布情報のデータ、決済データ等を記憶する。

鍵管理部３２は、通信に関わる配布電文の暗号化と、中央管理センタ１００からの決済に関する電文を復号処理する。

通信処理部３３は、制御部３０から供給されるデータを変調して送受信部３４へ供給し、また、送受信部３４から供給されるデータを復調して制御部３０へ供給する。送受信部３４は、インターネット等の通信網ＮＴ４を介して購入者側の中継端末装置４００と通信する。

通信処理部３５は、制御部３０から供給されるデータを変調して送受信３６へ供給し、また、送受信部３６からのデータを復調して制御部３０へ供給する。送受信部３６は、通信網ＮＴ３を介して中央管理センタ１００と通信する。

かかる構成により、販売管理センタ３００は、通信網ＮＴ４を介して中継端末装置４００と通信し、通信網ＮＴ３を介して中央管理センタ１００と通信するようになっている。

尚、第３図には、典型例として、商品紹介のための情報をインターネット等の通信網ＮＴ４を通じて購入者側へ配信する場合の構成を示している。しかし、前述したように、販売者は紙媒体やＣＤ等の情報記録媒体、放送系メディア等を通じて配布や配信を行うことも可能となっている。

次に、第４図に基づいて中継端末装置４００の基本的な構成を説明する。

中継端末装置４００は、制御部４０、メモリ部４１、表示部４２、通信処理部４３、送信部４４、データ形式変換部４５を備えて構成されている。また、購入者の希望に応じて、販売管理センタ３００からの配布電文を受信するＴＶチューナやラジオチーナ４８ｂと、バーコ

Nintendo Exhibit 1003
Part 2 of 2
Page 00794

ーダリーダ４８ｃと、ＣＤ等の情報記録媒体を再生するための再生部４８ｄを選択的に組み付けることができるようになっている。また、パーソナルコンピュータ４８ｅに接続することできるようになっている。

ここで、制御部４０は、中継端末装置４００全体を集中管理する。メモリ部４１は、「配布電文」のデータや各種の演算データ等を記憶する。表示部４２は、「配布電文（又は配布情報）」中に含まれている商品紹介のための情報を表示する。通信処理部４３は、制御部４０から供給されるデータを変調して送信部４４へ供給する。送信部４４は、光通信と電波による無線通信を行うための通信ポートを構成しており、利用者端末装置５００との間で無線アンテナ４６又は光通信ポート４７のいずれかを介して通信する。

データ形式変換部４５は、ＴＶチューナやラジオチーナ４８ｂ、バーコードリーダ４８ｃ、再生部４８ｄから供給される「配布電文（又は配布情報）」を本通信システムで扱える所定フォーマット形式の「配布電文」に変換して制御部４０へ供給する。

かかる構成により、中継端末装置４００は、販売者側から提供される「配布電文（又は配布情報）」を取得し、所定フォーマット形式の「配布電文」に変換して利用者端末装置５００へ送信する。

次に、第５図に基づいて利用者端末装置５００の基本的な構成を説明する。利用者端末装置５００は、制御部５０、メモリ部５１、入力部５２、表示部５３、通信処理部５４、送受信部５５を備えて構成されている。更に、制御部５６、中継端末装置４００からの「配布電文」や、「注文電文」のデータ等を記憶するメモリ部５７、データの暗号化と複号処理を行う鍵管理部５８、通信処理部５９、送受信部６０を備えて構成されている。

制御部５０は、利用者端末装置５００全体の動作を集中管理すると共に、通信処理部５４と送受信部５５及び無線アンテナ６１を介して、

Nintendo Exhibit 1003
Part 2 of 2
Page 00795

移動体通信網ＮＴ１を介して中央管理センタ１００との間で双方向通信を行う。

制御部５６は、中継端末装置４００側から送信されてくるデータを受信アンテナ６２，６３と送受信部６０及び通信処理部５９を介して入力し、鍵管理部５８へ転送して復号処理を行わせる。

尚、送受信部６０は、光通信と電波による無線通信を行うための通信ポートを構成しており、無線アンテナ６２又は光通信ポート６３のいずれかを介して中継端末装置４００との間で通信を行う。

ここで、第６Ａ図の平面図に示すように、利用者端末装置５００は携帯性を有する小型携帯電話と同様の形状をしており、第５図中の点線で囲った部分５００Ａ内の構成要素５６～６０，６２，６３が、中継端末装置４００と中央管理センタ１００との間で通信する機能を有し、残余の構成要素５０～５５，６１は、周知の無線電話機能を有して、筐体（符号省略）内に一体に内蔵されている。

また、購入者が入力部５２の操作釦等を操作して電話機能を指令をすると、制御部５０がその指令に従って通話のため処理を行う。また、購入者が入力部５２の操作釦等を操作して商品を注文するための指令をすると、制御部５０がその指令に従って制御部５６を制御して「注文電文」を作成する。更に、制御部５０は、その「注文電文」を移動体通信網ＮＴ１を介して中央管理センタ１００へ送信する。また、購入者が入力部５２の操作釦等を操作して中継端末装置４００との間での通信を指令すると、制御部５０が制御部５６を制御して、中継端末装置４００からの上記「配布電文」を取得させ、更に、取得した「配布電文」のデータをメモリ部５７に記憶させる。

尚、変形例として、第６Ｂ図の平面図に示すように、第５図中の点線で囲った部分５００Ａ内の構成要素５６～６０，６２，６３を別体の着脱自在なモジュール５００Ａにし、このモジュール５００Ａを既存の携帯電話本体に装着することで、利用者端末装置５００を実現で

きるようになっている。

次に、第７図ないし第１２図のフローチャートを参照して、本通信システムと、中央管理センタ１００、決済センタ２００、販売管理センタ３００、中継端末装置４００、利用者端末装置５００の動作を説明する。尚、利用者端末装置５００を所有する購入者が、通信販売によって所望の商品を注文する場合の動作を説明する。

第７図は、販売管理センタ３００が、商品紹介のための配信電文を作成して配信あるいは配布する際の処理を示している。

ステップＳ１００において、販売者が商品紹介のための「商品情報」と「販売者情報」のデータを第３図に示したメモリ部３１に記憶させて、購入者側への情報配信又は配布の開始を指示すると、ステップＳ１０２において、制御部３０が、メモリ部３１の上記「商品情報」と「販売者情報」のデータを結合して、「配布情報」のデータDSELを作成する。

次に、鍵管理部３２が乱数ＲＤを生成する（ステップＳ１０４）。更に、この乱数ＲＤを中央管理センタ１００が発行する公開鍵（以下、ＮＯＣ公開鍵という）D'KYNOCOPENで暗号化することで、暗号化した乱数データＲＤSHOPを生成する（ステップＳ１０６）。

次に、制御部３０が、上記の暗号化した乱数データＲＤSHOPと、販売者を識別するための販売者識別データＩＤSHOPと、「配布情報」のデータDSELを結合して、配布電文ＤＴSELを作成する（ステップＳ１０８）。

次に、販売管理センタ３００の操作者が、ＰＩＮコード（個人識別コード）を入力すると（ステップＳ１１０）、鍵管理部３２がこのＰＩＮコードの正当性を確認し（ステップＳ１１２）、正しくない場合には、ステップＳ１１０から再入力となる。一方、ＰＩＮコードが正しい場合には、ステップＳ１１４の処理に移行する。

ステップＳ１１４では、鍵管理部３２は、配布電文ＤＴSELに販売者

Nintendo Exhibit 1003
Part 2 of 2
Page 00797

の秘密鍵ＤKYSHOPで電子署名を行い、署名された配布電文ＤSELsigを生成する。

そして、ステップＳ１１６において、制御部１３が、署名した配布電文ＤSELsigを通信処理部３３に供給し、送受信部３４を通じて中継端末装置４００側へ配信する。また、ステップＳ１１８に示すように、配布電文ＤSELsigは、放送系メディアや情報記録媒体や紙媒体によっても購入者側へ配布される。

尚、ステップＳ１１６の処理を継続して行うことにより、中継端末装置４００が何時でも配布電文ＤSELsigを受信できるようになっている。

こうして販売管理センタ３００から配信又は配布がなされると、次に、中継端末装置４００は、第８図に示す処理を行うことで、販売管理センタ３００からの情報の取得と、取得した情報の利用者端末装置５００への送信を行う。

まず、第８図のステップＳ２００において、中継端末装置４００のデータ形式変換部４５が、受信した配布電文ＤSELsigを所定のフォーマット形式に変換し、メモリ部４１に記憶させる。また、ステップＳ２０２に示すように、放送メディアで配信された場合にも同様にして配布電文を所定のフォーマット形式に変換し、メモリ部４１に記憶させる。また、紙媒体や情報記録媒体によって配布された配布電文もバーコードリーダや再生装置によって読み取り、メモリ部４１に記憶させる。

次に、ステップＳ２０４において、表示部４２がメモリ部４１に記憶された配布電文ＤSELsigに基づいて商品情報を表示する。

次に、ステップＳ２０６において、制御部４０が中継端末装置４００に予め設定されている中継端末識別データＩＤTRを配布電文ＤSELsigに追加し、ステップＳ２０８において、送受信部４４等が利用者端末５００へ送信する。

Nintendo Exhibit 1003
Part 2 of 2
Page 00798

こうして中継端末装置４００から利用者端末装置５００に、中継端末識別データＩＤTRの追加された配布電文ＤSELsigが送信されると、購入者は利用者端末装置５００を用いて商品の注文が可能となる。

まず、第９図のステップＳ３００において、利用者端末装置５００は、送受信部６０によって配布電文ＤSELsigを受信し、メモリ部５７に記憶させる。

次に、ステップＳ３０２において、表示部５３が、配布電文ＤSELsigの商品情報を表示する。

次に、ステップＳ３０４において、購入者が入力部５２を操作して希望の商品の指定と注文をすると、ステップＳ３０６において制御部５６がその注文の指示を確認し、ステップＳ３０８に移行する。一方、購入者が注文しない旨の入力をすると、制御部５６がこれを検知して処理を終了する。

ステップＳ３０８では、制御部５６が、配布電文ＤSELsigの中から注文すべき商品に関する配布電文を抽出し、注文した時刻のデータを追加することで注文情報のデータＤBUYを生成する。

次に、制御部５６が、ステップＳ３１０において、注文情報のデータＤBUYに重複検出用の値を表すデータ（以下、重複検出データという）ＤTMを追加し、更にステップＳ３１２において、利用者端末識別データＩＤUSERを追加することで注文電文ＤTBUYを生成する。

次に、購入者が入力部５２からＰＩＮコードを入力すると（ステップＳ３１４）、鍵管理部５８がＰＩＮコードの正当性を確認し（ステップＳ３１６）、正しくない場合には再入力のためにステップＳ３１４の処理に移行し、ＰＩＮコードが正しい場合にはステップＳ３１８に移行する。

ステップＳ３１８では、鍵管理部５８が、購入者秘密鍵ＤKYUSERで注文電文ＤTBUYに電子署名し、更にステップＳ３２０において、電子署名した注文電文ＤTBUYsigをＮＯＣ公開鍵ＤKYNOCOPENで暗号化する。

Nintendo Exhibit 1003
Part 2 of 2
Page 00799

次に、ステップＳ３２２において、暗号化した注文電文ＤTBUYKYsigを移動体通信網ＮＴ１を介して中央管理センタ１００へ送信する。

そして、ステップＳ３２４において、中央管理センタ１００から電子決済の返信が来るまで待機状態となる。

尚、利用者端末装置５００は、中央管理センタ１００から電子決済の返信が送られてくると、第１３図に示す処理を行うが、この処理については後述する。

次に、中央管理センタ１００の動作を第１０図ないし第１２図に基づいて説明する。

まず、第１０図のステップＳ４００において、中央管理センタ１００は、送受信部１５により利用者端末装置５００からの注文電文ＤTBUYKYsigを受信してメモリ部１１に記憶する。

次に、鍵管理部１３がその注文電文ＤTBUYKYsigを央管理センタ１００の秘密鍵データＤKYNOCで復号する（ステップＳ４０２）。

次に、ステップＳ４０４において、制御部１０が、上記の復号した注文電文ＤTBUYsig'から利用者識別データＩＤUSERを取り出し、公開鍵データベース１２を検索する。

更に、制御部１０は、利用者識別データＩＤUSERが公開鍵データベース１２に存在するか否かチェックし（ステップＳ４０６）、存在しない場合はエラー処理を行う。一方、存在する場合はステップＳ４０８に移行する。

ステップＳ４０８では、重複検出データＤTMをチェックし、送られてきた注文電文ＤTBUYKYsigが重複したものか否か判定する。重複した注文電文であった場合にはエラー処理を行う。一方、重複した注文電文でなければ、正しい注文電文であると判定して、ステップＳ４１０に移行する。

ステップＳ４１０では、制御部５が、公開鍵データベース１２から利用者の公開鍵ＤKYUSEROPENを取り出し、更に、ステップＳ４１２に

Nintendo Exhibit 1003
Part 2 of 2
Page 00800

おいて、利用者公開鍵ＤKYUSEROPENによって上記の復号した注文電文ＤTBUYsig'の購入者の電子署名を検証する。そして、ステップＳ４１４で、電子署名が正しいか否か判断し、正しくない場合はエラー処理を行う。一方、正しい場合は、第１１図に示すステップＳ４１６に移行する。

次に、第１１図のステップＳ４１６では、制御部１０が、注文電文ＤTBUYsig'から販売者識別データＩＤSHOPを取り出し、公開鍵データベース１２を検索する。更に、ステップＳ４１８において、販売者識別データＩＤSHOPが公開鍵データベース１２に存在するか否かチェックし、存在しない場合はエラー処理を行い、存在する場合はステップＳ４２０に移行する。

ステップＳ４２０で、公開鍵データベース１２から販売者公開鍵ＤKYSHOPOPENを取り出し、更にステップＳ４２２において、注文電文ＤTBUYsig'の販売者の電子署名を検証する。

そして、ステップＳ４２４において、電子署名が正しいか否か判断し、電子署名が正しくない場合はエラー処理を行う。一方、正しい場合はステップＳ４２６に移行する。

次に、ステップＳ４２６において、注文電文ＤTBUYsig'から注文情報のデータＤBUYと注文時の時刻情報を取り出し、メモリ部１１に記憶する。

更に、ステップＳ４２８において、注文電文ＤTBUYsig'から販売者乱数ＲＤSHOPを取り出し、更にステップＳ４３０で、鍵管理部１３が販売者乱数ＲＤSHOPを中央管理センタ１００の秘密鍵データＤKYNOCで復号してメモリ部１１に記憶し、第１２図のステップＳ４３２に移行する。

ステップＳ４３２では、制御部１０が、注文時の時刻情報と販売者識別データＩＤSHOPと、利用者識別データＩＤUSER、注文情報のデータＤBUYをまとめて決済要求電文ＤACKを生成する。そして、この決済

Nintendo Exhibit 1003
Part 2 of 2
Page 00801

要求電文ＤACKを送受信部１７から通信網ＮＴ２を介して決済センタ２００に送信し（ステップＳ４３４）、決済センタ２００の決済処理が行われるまで待機する（ステップＳ４３６）。これにより、あたかも販売者と購入者がその場で現金決済するのと同様な電子決済、つまり、決済センタ２００が立ち会いの下で、販売者と購入者が同時決済するような電子決済が行われることになる。

そして、決済センタ２００から中央管理センタ１００に、決済の結果を示す決済結果電文ＤACKRTNが通信網ＮＴ２を介して送られてくると、ステップＳ４３８において、送受信部１７が決済結果電文ＤACKRTNを受信する。

次に、ステップＳ４４０において、制御部１０が決済結果電文ＤACKRTNをチェックし、種々の原因で決済が正常になされなかった場合（決済不成立の場合）には、エラー処理を行う。一方、正常に決済がなされた場合（決済成立の場合）は、ステップＳ４４２～Ｓ４５０の処理を行うことにより、利用者端末装置５００と販売管理センタ３００へ処理結果電文を送信する。

まず、ステップＳ４４２において、制御部１０が、メモリ部１１に記憶されている販売者から送付されている乱数ＲＤSHOPを販売者公開鍵ＤKYSHOPOPENで暗号化する。

次に、ステップＳ４４４において、暗号化した乱数ＲＤSHOP’と、決済結果電文ＤACKRTNと、中央管理センタ２００が決済処理を行ったことを示す決済処理の結果と、上記ステップＳ４０２で復号した注文電文ＤTBUYsig’を結合して、処理結果電文ＤRTNを生成する。

次に、ステップＳ４４６において、処理結果電文ＤRTNを中央管理センタ１００の秘密鍵ＤKYNOCで電子署名する。

更に、ステップＳ４４８において、鍵管理部１３が、上記の電子署名した処理結果電文ＤRTNsigを利用者公開鍵ＤKYUSEROPENと販売者公開鍵ＤKYSHOPOPENで暗号化する。

Nintendo Exhibit 1003
Part 2 of 2
Page 00802

そして、ステップＳ４５０において、暗号化した処理結果電文ＤRTNsig'を送受信部１５から移動体通信網ＮＴ１を介して利用者端末装置５００へ通知すると共に、送受信部１７から通信網ＮＴ３を介して販売管理センタ３００へ通知する。

次に、中央管理センタ１００が処理結果電文ＤRTNsig'を通知すると、前述した待機状態の利用者端末装置５００は、第１３図に示す処理を行い、販売管理センタ３００は、第１４図に示す処理を行う。

利用者端末装置５００は、第１３図のステップＳ３２６において、処理結果電文ＤRTNsig'を受信し、更にステップＳ３２８において、処理結果電文ＤRTNsig'を利用者秘密鍵ＤKYUSERで復号する。

次に、ステップＳ３３０において、上記の復号した処理結果電文ＤRTNsig'を中央管理センタ１００の公開鍵ＤKYNOCOPENによって電子署名を検証する。

次に、ステップＳ３３２において、上記の電子署名が正しいか否かチェックし、正しくない場合にはエラー処理を行い、正しい場合にはステップＳ３３４に移行して、表示部５３が、注文情報のデータＤBUYと、中央管理センタ１００からの決済処理の結果を表示する。

これにより、購入者は、表示部５３の表示を見ることで、自分が注文した商品が適切に注文され且つ電子決済がなされたことを知ることができる。

一方、販売管理センタ３００は、第１４図のステップＳ１２０において、処理結果電文ＤRTNsig'を受信し、更にステップＳ１２２において、処理結果電文ＤRTNsig'の電子署名を中央管理センタ１００の公開鍵ＤKYNOCOPENで検証する。

次に、ステップＳ１２４において、上記の電子署名が正しいか否かチェックし、正しくない場合にはエラー処理を行い、正しい場合にはステップＳ１２６に移行する。

ステップＳ１２６では、処理結果電文ＤRTNsig'中の暗号化されて

Nintendo Exhibit 1003
Part 2 of 2
Page 00803

いる乱数ＲＤSHOPを販売者秘密鍵ＤKYSHOPで復号し、次に、ステップＳ１２８において、復号した乱数ＲＤが中央管理センタ１００に既に登録しておいた乱数ＲＤと等しいか否かチェックする。ここで、上記の乱数ＲＤが等しくなければエラー処理を行い、等しければステップＳ１３０に移行する。

ステップＳ１３０では、表示部３７が、注文された商品の情報ＤBUYと、中央管理センタ１００が決済した結果を表示すると共に、ステップＳ１３２において、処理結果電文ＤRTNsig'に含まれている「注文された商品」と「購入者の住所」等のデータを「受注」データとして、商品配送のための処理を行う。

このように、処理結果電文ＤRTNsig'が中央管理センタ１００から利用者端末装置５００と販売管理センタ３００へ通知されるため、購入者は適切に注文と電子決済が行われたか否かを知ることができ、販売者は適切に電子決済が行われて発送すべき商品の確認をすることができる。

尚、説明の都合上、利用者端末装置５００と販売管理センタ３００へ同じ処理結果電文ＤRTNsig'が通知されることとして説明したが、利用者端末装置５００には購入者が適切に注文と電子決済が行われたか否かを知るのに必要な最小限の情報を通知し、販売管理センタ３００には販売者が適切に電子決済が行われて発送すべき商品の確認をすることができるのに必要な最小限の情報を通知してもよい。

本実施の形態の通信システムによれば、次のような効果が発揮される。

まず、販売管理センタ３００は、販売しようとする商品の情報に販売者の電子署名を行って配信又は配布し、利用者端末装置５００から注文を行うと、中央管理センタ１００がこの電子署名を検証してその検証結果を「処理結果電文」に含めて利用者端末装置５００に通知する。したがって、購入者は、販売者から商品が発送されてきたときに、

Nintendo Exhibit 1003
Part 2 of 2
Page 00804

「処理結果電文」と突き合わせを行うことにより、抵当名販売者からの商品であるか否かの確認をすることができる。

また、利用者端末装置５００で商品を注文する際、購入者の住所（発送先）等の情報を付加した「注文電文」を中央管理センタ１００に送信し、中央管理センタ１００が電子決済を行うと、購入者の住所（発送先）等の情報を含んだ「処理結果電文」が販売管理センタ３００に通知されるので、販売者は注文された商品を誤りなく購入者に発送することができる。

また、商品毎の価格などの商品情報が含められた「配布電文」や「配布情報」が販売管理センタ３００から中継端末装置４００を介して利用者端末装置５００に提供されるので、購入者は中継端末装置４００や利用者端末装置５００の表示部を見るだけで、所望の商品を選択することができる。つまり、利用者端末装置５００等によって多くの商品を見ることができるという優れた利便性を提供することができる。

また、利用者端末装置５００から中央管理センタ１００へ、利用者識別データを含んだ「注文電文」が送信されるので、中央管理センタ１００は、購入者を誤ることなく電子決済の処理をすることができる。

また、購入者の「注文電文」は販売者の「販売者公開鍵」によって暗号化されて中央管理センタ１００に送信される。このため、購入者の住所など情報は第三者に漏洩することなく、中央管理センタ１００を介して販売管理センタ３００に送信することができる。

また、守秘義務を有する中央管理センタ１００の立ち会いの下で、利用者端末装置５００と販売管理センタ３００との間の電子商取引が即座に行われる。このため、購入者と販売者はあたかも現金決済を行ったのと同様な利便性が得られる。

また、実際には現金を用いて決済が行われないので、例えば従来の商品配送業者が商品引き渡しと同時に現金を受け取るといった煩雑な処理を行う必要がなくなる。また、配送業者が現金を携帯しなくて済

Nintendo Exhibit 1003
Part 2 of 2
Page 00805

むので、安全な商取引を可能にする。

また、購入者の個人情報は、利用者端末装置５００内で処理されるので、従来のクレジットカードのようなＰＩＮコードが第三者に漏洩するという問題を未然に防止することができる。

また、中央管理センタ１００が利用者端末装置５００と販売管理センタ３００には、電子署名された「処理結果電文」が通知されるので、仮に何らかのトラブルが発生した場合でも、この「処理結果電文」によってトラブルを解決することが可能となる。

また、販売者と購入者の電子署名だけでなく、中央管理センタ１００でも電子署名した「処理結果電文」を利用者端末装置５００と販売管理センタ３００に通知するので、確実且つ安全な電子商取引の方法を提供することができる等の様々な効果を発揮する。

尚、本実施の形態では、中継端末装置４００と利用者端末装置５００を分離して購入者に提供する場合を説明したが、中継端末装置４００を利用者端末装置５００に内蔵して一体化してもよい。また、第６Ｂ図に示した着脱自在なモジュール５００Ａに中継端末装置４００を内蔵して提供してもよい。

また、変形例として、商品配送者が中継端末装置４００を携帯するようにしてもよい。この場合には、商品配送者が中継端末装置４００を携帯し易いような携帯端末装置にする。

そして、販売者は、テレビ放送やラジオ放送などの放送系メディアやカタログ等の紙媒体によって、商品情報を配布する。また、購入者が配布された情報から所望の商品を選択し、販売管理センタ３００に対し電話やファクシミリで商品を注文する。

そして、販売者が注文された商品を購入者に対して配送する際（出荷の際）に、発送すべき商品の情報である前述の「商品情報」と「販売者情報」を有する「配布電文」を作成し、この「配布電文」をバーコードや磁気データとして商品にて添付する。そして、配送者がこの

Nintendo Exhibit 1003
Part 2 of 2
Page 00806

商品と共に中継端末装置４００を携帯して購入者宅へ配送する。

ここで、配送者が購入者に商品を引き渡す際に、上記のバーコードや磁気データを中継端末装置４００に設けられているバーコードリーダや磁気読取り装置で読み取る。

こうすると、中継端末装置４００で読みとられた「商品情報」と「販売者情報」が、前述した「配布電文」と同様に利用者端末装置５００に転送される。そして、購入者が利用者端末装置５００を操作して注文の操作をすると、「注文電文」が中央管理センタ１００に送信され、前述した電子決済が行われることになる。また、電子決済が行われた結果を示す「処理結果電文」が利用者端末装置５００と販売管理センタ３００に通知されることになる。

したがって、利用者端末装置５００に通知された「処理結果電文」の内容を配達者と購入者が共に確認することで、適切な電子決済がなされたか否かの判断をすることができ、適切な電子決済がなされたとの確認をした後、配達者は商品を購入者に引き渡すことができる。また、販売管理センタ３００にも「処理結果電文」が通知されているので、配達者は購入者から代金を受け取る必要がない。このため、この変形例によれば、従来の現金引き替えで商品を引き渡す場合に比べて、安全な商取引を可能にする通信システムを提供することができる。

また、従来の通信販売では、販売者（配達者）が購入者に商品を引き渡すと共に所定の振り込み用紙を渡し、商品引き渡し後にその振り込み用紙によって金融機関に代金を支払う方法が採られる場合があった。この従来の場合には、購入者はわざわざ金融機関に出向く必要があり、また、販売者は振り込みによって代金支払いがなされて金融機関からその通知が送られてくるまでは、商品と代金支払いの突き合わせを行うことができないという不便さがあった。

しかし、この変形例によれば、購入者への商品引き渡しと電子決済がほぼ同時に行われることになるため、購入者にとっても販

Nintendo Exhibit 1003
Part 2 of 2
Page 00807

売者にとっても従来のような煩雑な処理が不要となり、優れた利便性を提供することができる等の効果を発揮する。

産業上の利用可能性

本発明の通信システムは、販売管理手段と中央管理手段との間に利用者端末装置が介在し、販売管理手段から提供される商品情報に基づいて利用者端末装置が中央管理手段に対して注文を行い、更に、注文に応じて中央管理手段が決済の結果を販売管理手段と利用者端末装置に通知するようにしたので、閉鎖した通信システム内で情報通信を行うことができる。このため、個人の秘密情報の保護等を極めて高い信頼性をもって実現することができる。特に、従来のように、購入者が商品購入や注文等を行う際、自分の氏名や住所、電話番号、クレジットカードの番号等の秘密性の高い個人情報を提示しなければならないといった問題を解消し、耐タフー性の優れた通信システムを提供できる。

また、販売管理手段側の販売者と利用者端末装置側の購入者との両者に対し、迅速で、且つ安全な電子商取引などを行うことが可能な通信システムを提供することができる。

Nintendo Exhibit 1003
Part 2 of 2
Page 00808

請求の範囲

1．　電子決済の処理をする中央管理手段と、商品を販売する販売者が所有する販売管理手段と、商品を購入する購入者が所有する利用者端末手段とを備え、

前記販売者が配布する商品情報の中から前記購入者が前記利用者端末装置を用いて所望の商品を前記中央管理手段に対して注文すると、前記中央管理手段が注文に応じて電子決済し、電子決済の処理結果を前記販売管理手段と前記利用者端末手段に通知することを特徴とする通信システム。

2．　請求項1に記載の通信システムであって、前記販売管理手段は、前記販売者が販売する商品の情報を利用者端末装置に提供し、前記利用者端末装置は、前記購入者の注文に応じて、前記提供された商品の情報に基づいて注文電文を生成して前記中央管理手段に送信し、前記中央管理手段は、前記送信された注文電文に基づいて前記電子決済を行うと共に、前記電子決済の処理結果を前記販売管理手段と前記利用者端末手段に通知することを特徴とする通信システム。

3．　請求項1又は請求項2に記載の通信システムであって、前記販売管理手段は、前記購入者が所有する中継端末装置に前記商品の情報を提供し、前記中継端末装置は、提供された前記商品の情報を前記利用者端末装置に提供することを特徴とする通信システム。

4．　電子決済の処理をする中央管理手段と、商品を販売する販売者が所有する販売管理手段と、商品を購入する購入者が所有する利用者端末手段と、販売する商品の情報を前記販売管理手段から供給される中継端末装置とを備え、

前記販売者が配布する商品情報の中から前記購入者が前記販売管理手段に対して所望の商品を注文するのに応じて、前記販売者が購入者に商品を配送する際に、配送すべき商品に付された商品情報を前記中

Nintendo Exhibit 1003
Part 2 of 2
Page 00809

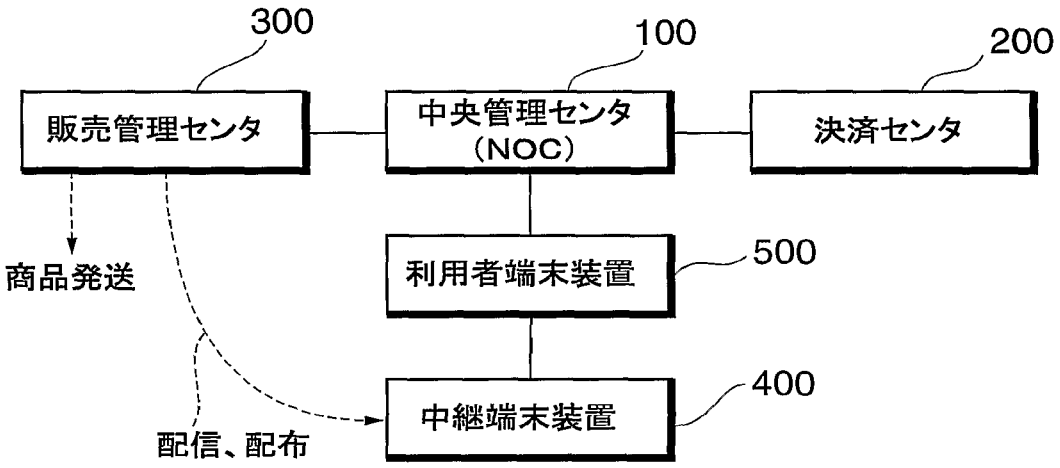
継端末手段が読み取って前記利用者端末装置に送信し、

前記利用者端末装置は、送信された前記商品情報に基づいて前記中央管理手段に注文の情報を転送し、

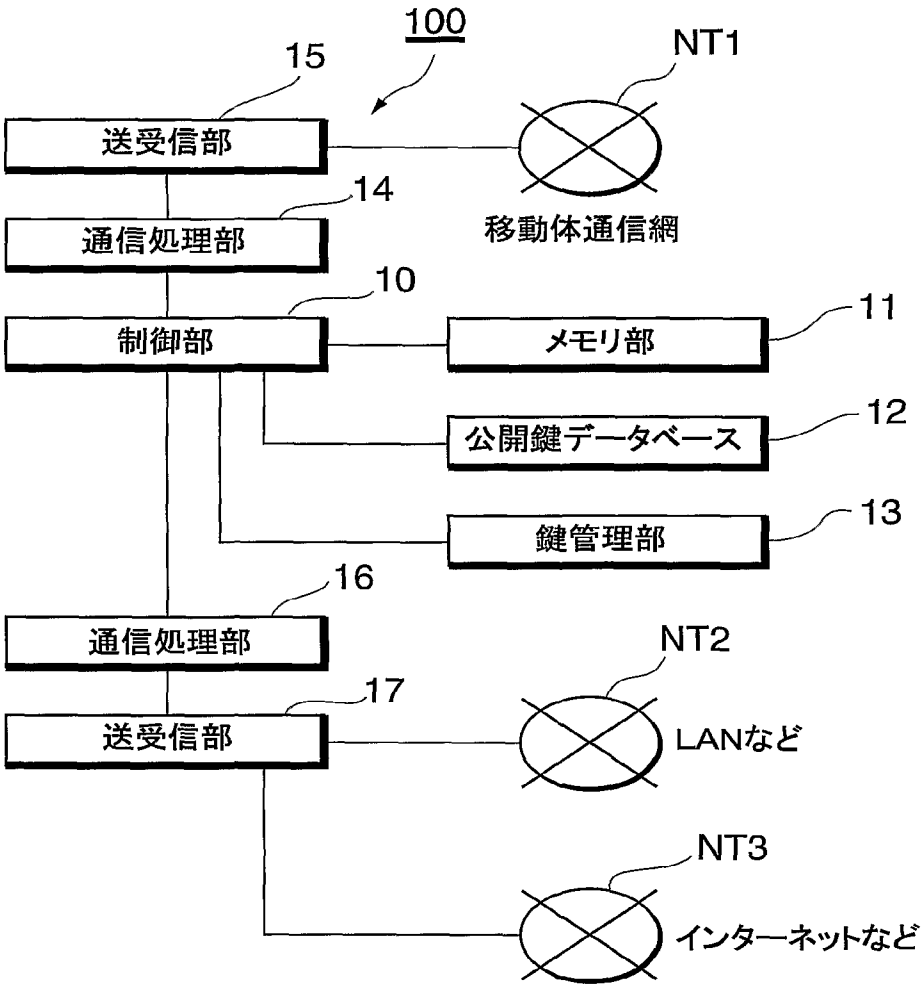
前記中央管理手段は、注文の情報に応じて電子決済し、電子決済の処理結果を前記販売管理手段と前記利用者端末手段に通知することを特徴とする通信システム。

Nintendo Exhibit 1003
Part 2 of 2
Page 00810

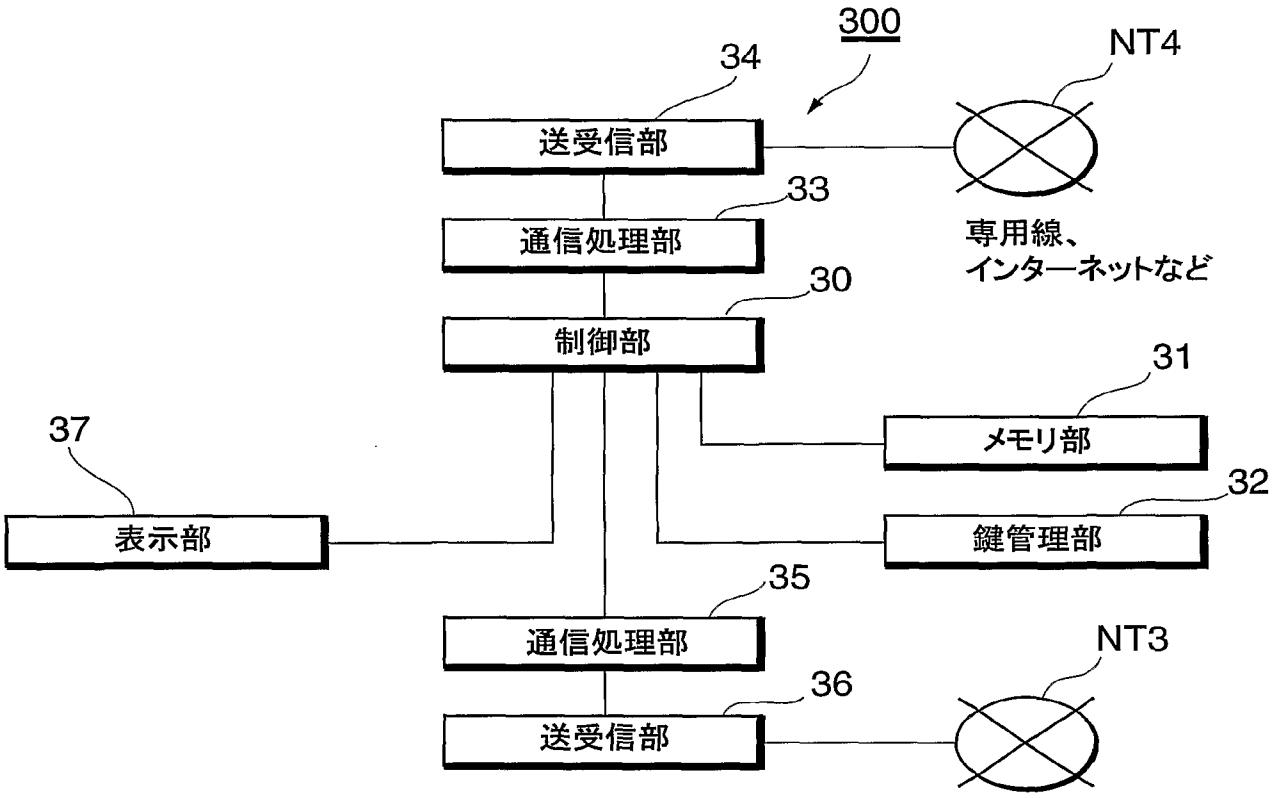
第１図


300
100
200
販売管理センタ
中央管理センタ
(NOC)
決済センタ
商品発送
利用者端末装置
500
中継端末装置
400
配信、配布

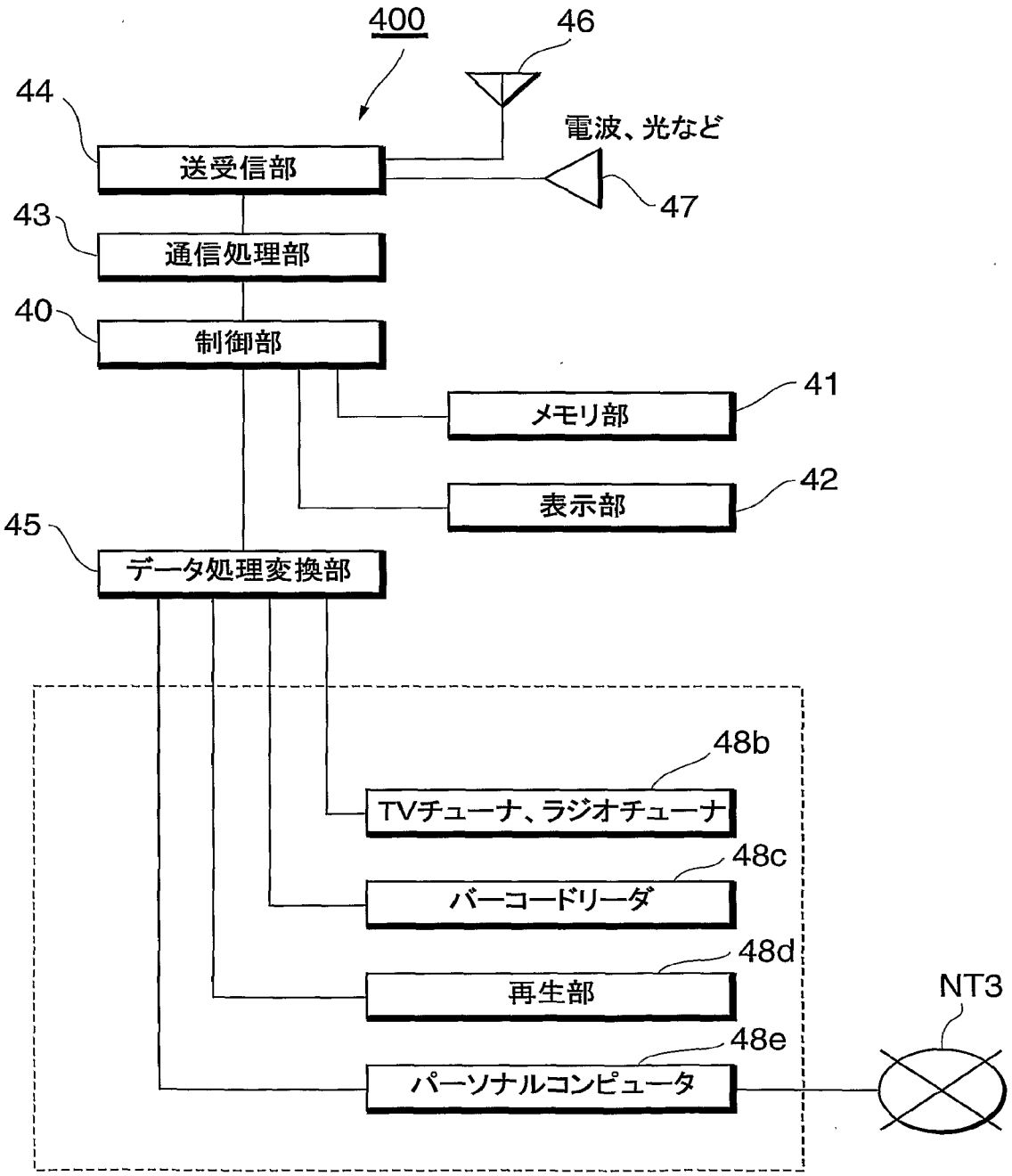
第２図


100
15
NT1
送受信部
14
通信処理部
移動体通信網
10
制御部
メモリ部
11
公開鍵データベース
12
鍵管理部
13
16
通信処理部
NT2
17
送受信部
LANなど
NT3
インターネットなど

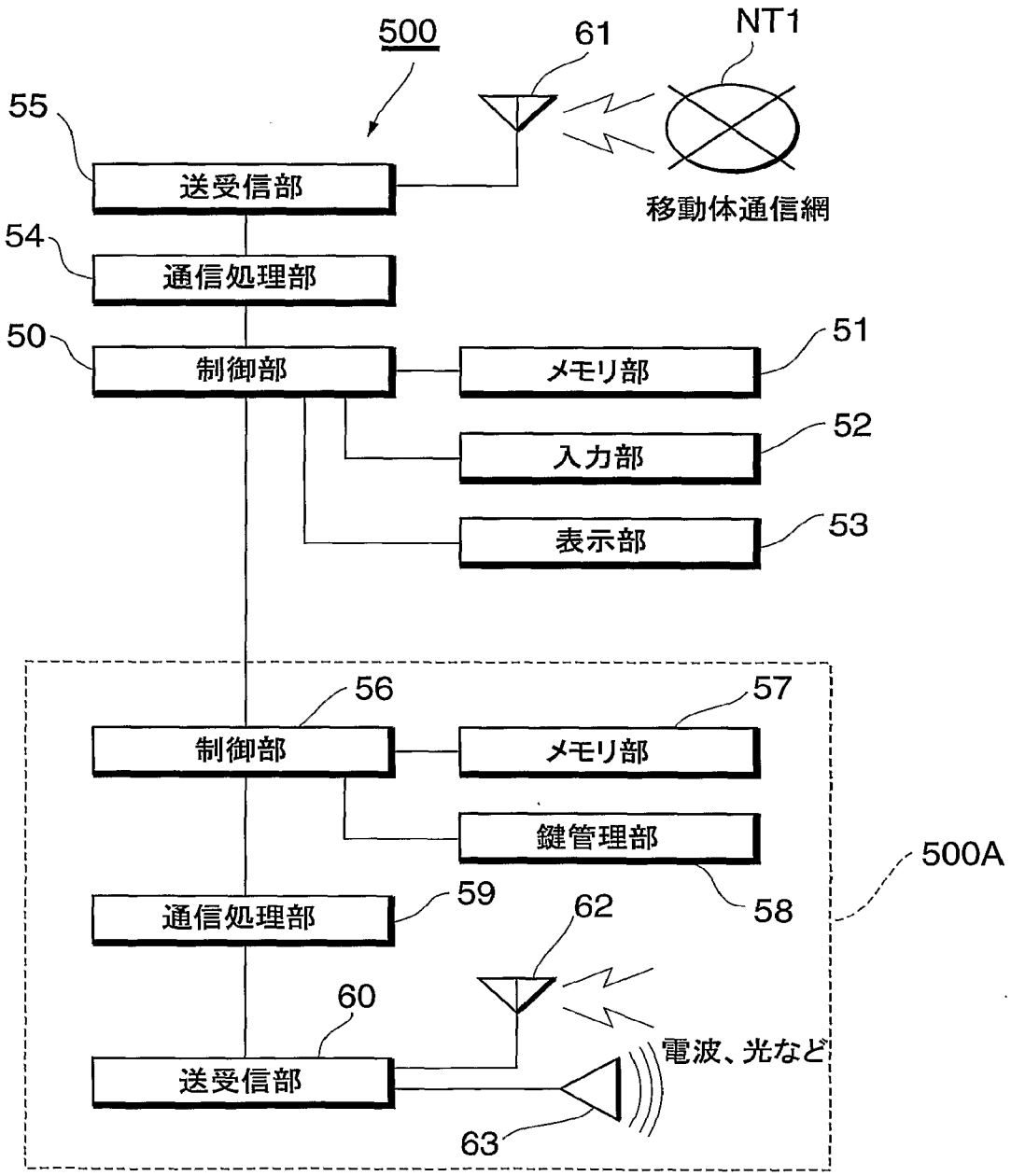
第３図


300
34
NT4
送受信部
33
通信処理部
専用線、
インターネットなど
30
制御部
31
メモリ部
37
表示部
32
鍵管理部
35
通信処理部
NT3
36
送受信部

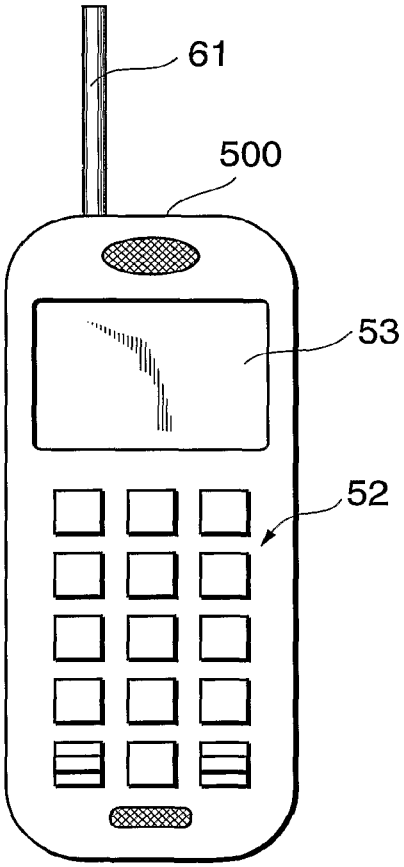
第４図


400
46
44
電波、光など
送受信部
47
43
通信処理部
40
制御部
41
メモリ部
42
表示部
45
データ処理変換部
48b
TVチューナ、ラジオチューナ
48c
バーコードリーダ
48d
再生部
NT3
48e
パーソナルコンピュータ

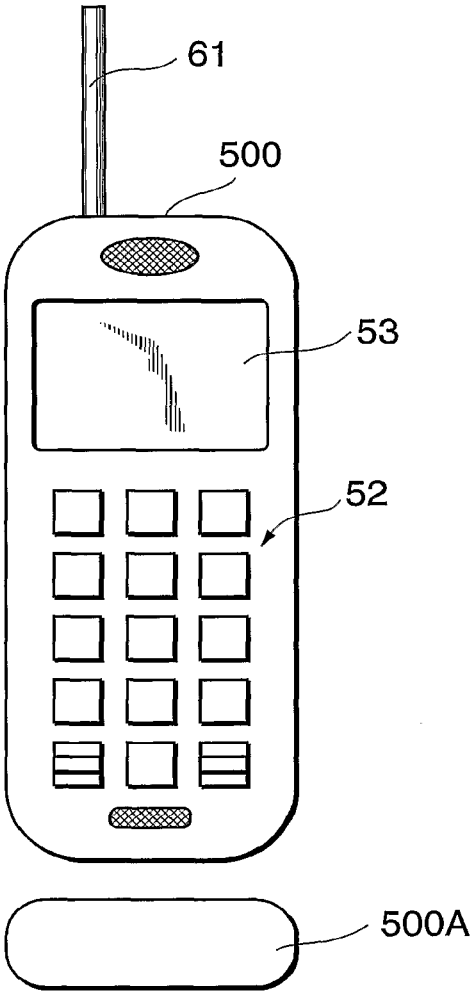
第５図


500
61
NT1
55
送受信部
移動体通信網
54
通信処理部
50
制御部
51
メモリ部
52
入力部
53
表示部
56
57
制御部
メモリ部
鍵管理部
500A
59
58
通信処理部
62
60
電波、光など
送受信部
63

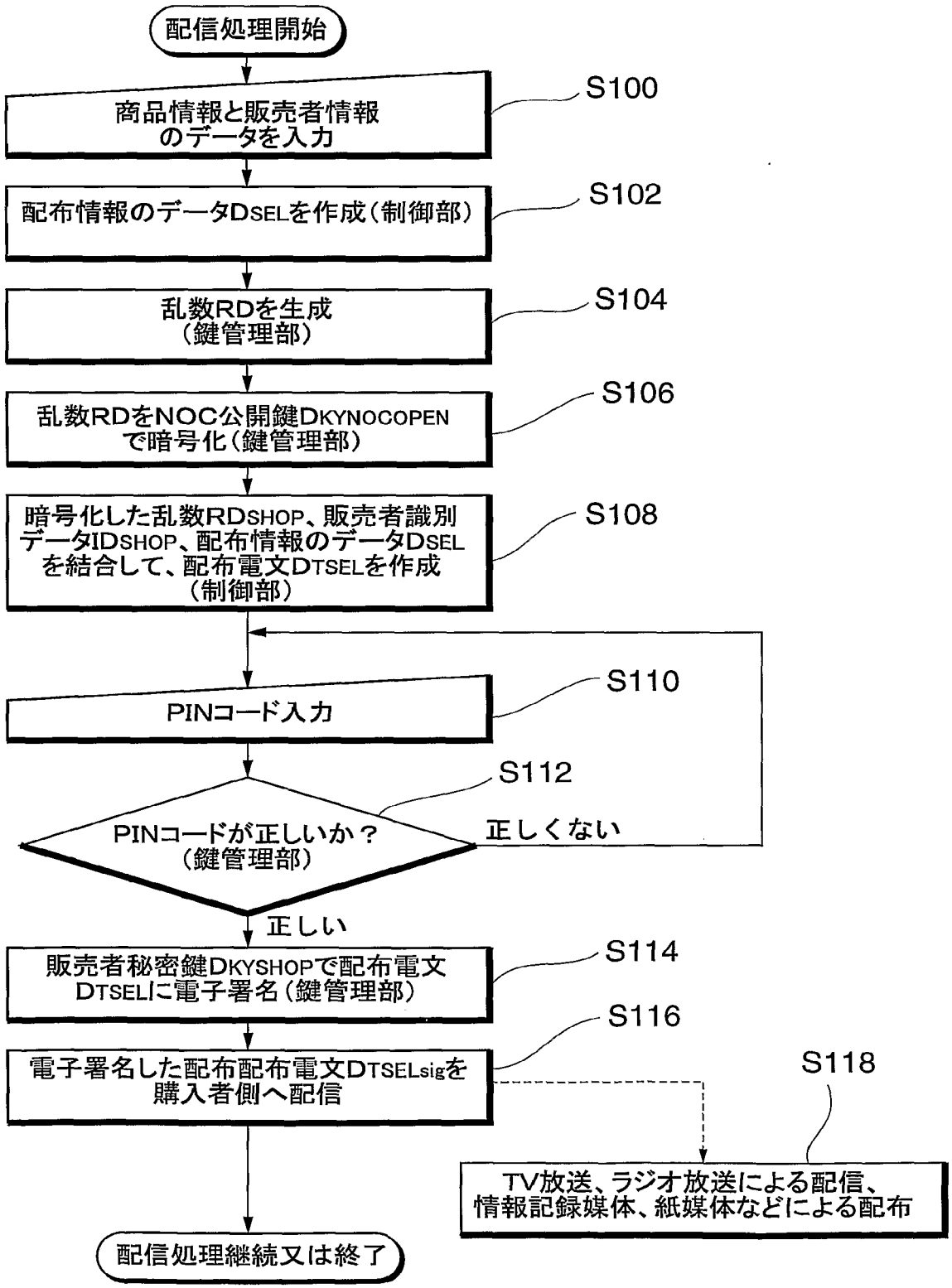
第６Ａ図



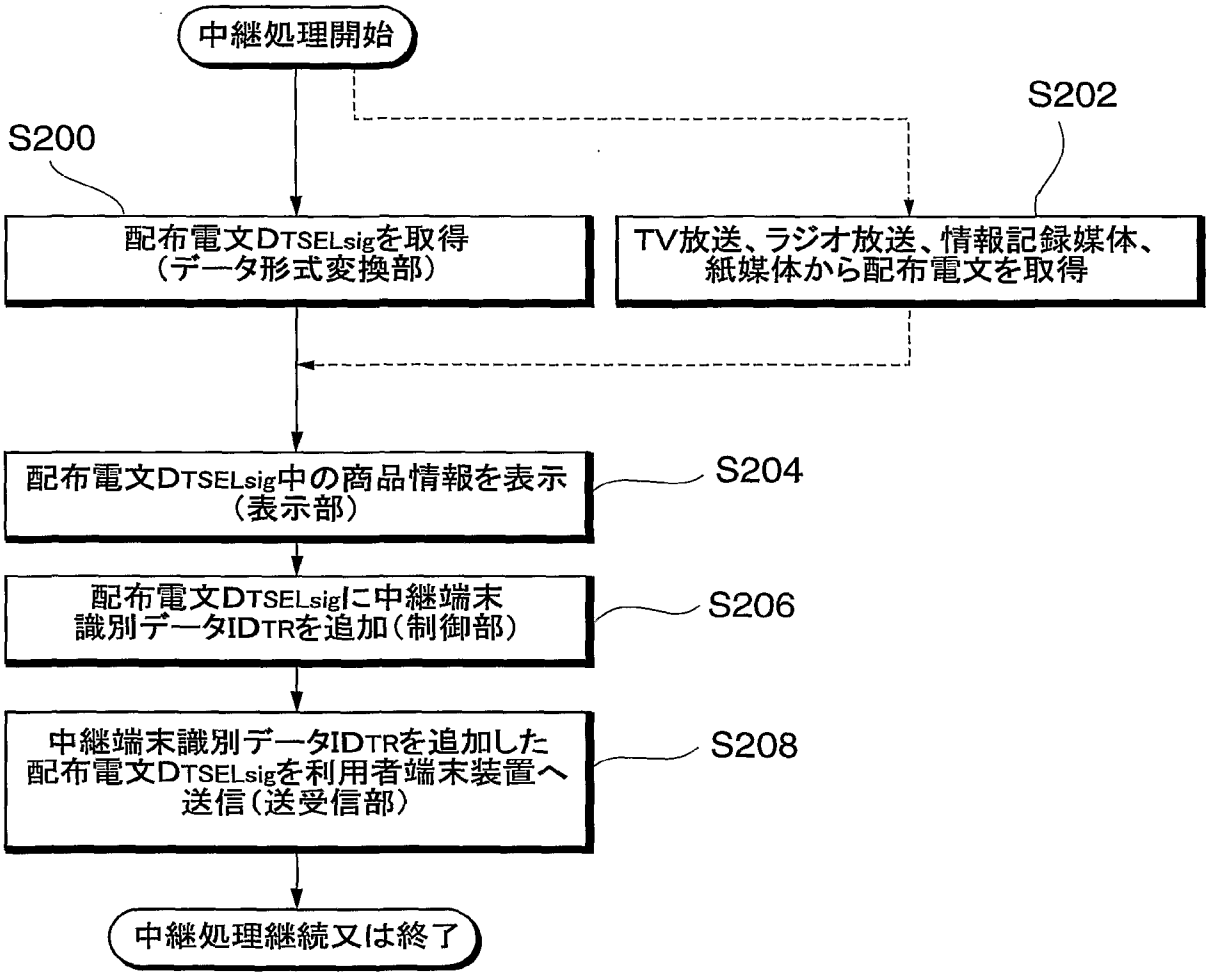
第６Ｂ図



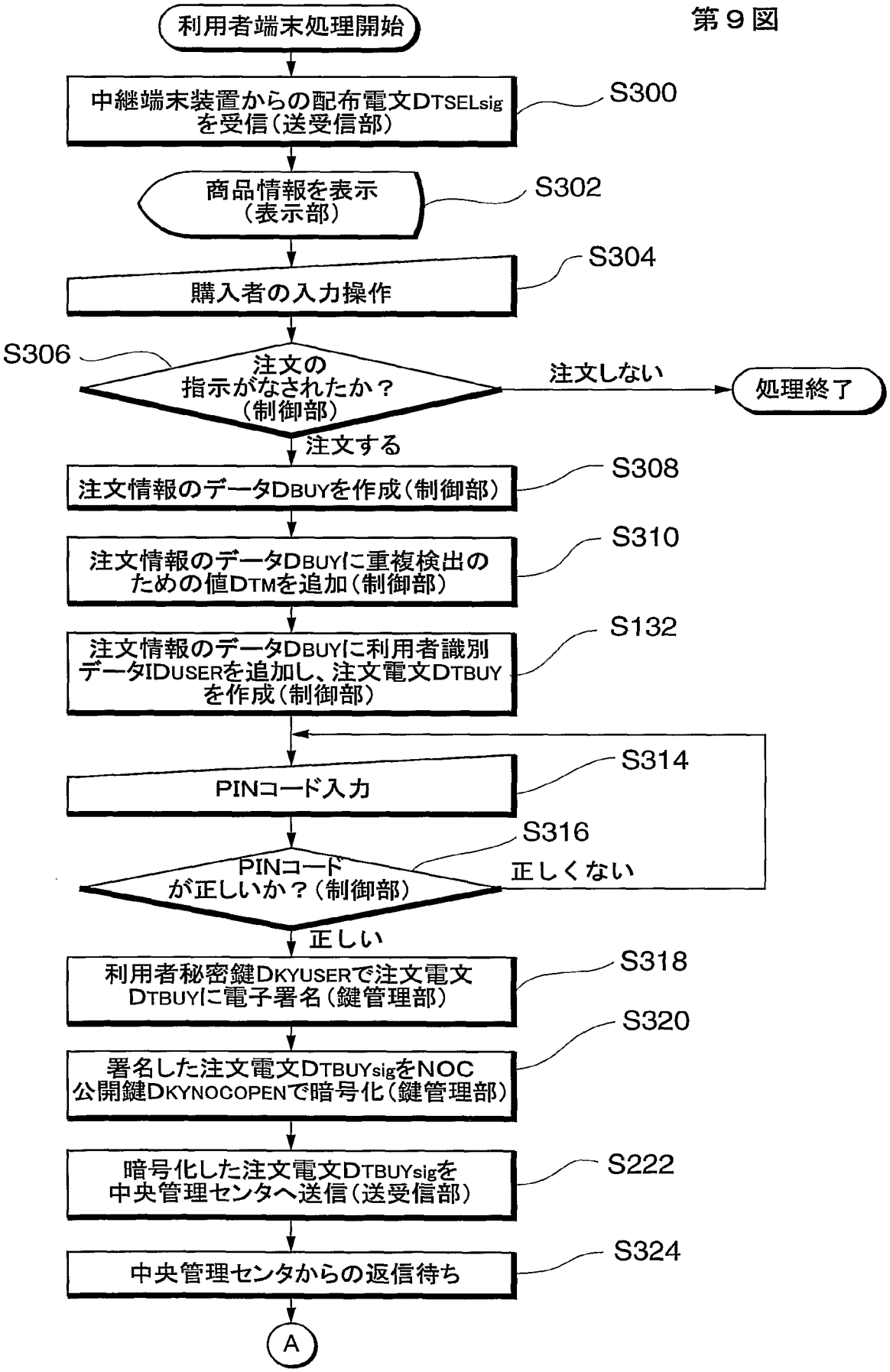
第７図


配信処理開始
商品情報と販売者情報のデータを入力
S100
配布情報のデータDSELを作成（制御部）
S102
乱数RDを生成（鍵管理部）
S104
乱数RDをNOC公開鍵DKYNOCOPENで暗号化（鍵管理部）
S106
暗号化した乱数RDSHOP、販売者識別データIDSHOP、配布情報のデータDSELを結合して、配布電文DTSELを作成（制御部）
S108
PINコード入力
S110
PINコードが正しいか？（鍵管理部）
S112
正しくない
正しい
販売者秘密鍵DKYSHOPで配布電文DTSELに電子署名（鍵管理部）
S114
電子署名した配布配布電文DTSELsigを購入者側へ配信
S116
S118
TV放送、ラジオ放送による配信、情報記録媒体、紙媒体などによる配布
配信処理継続又は終了

第８図

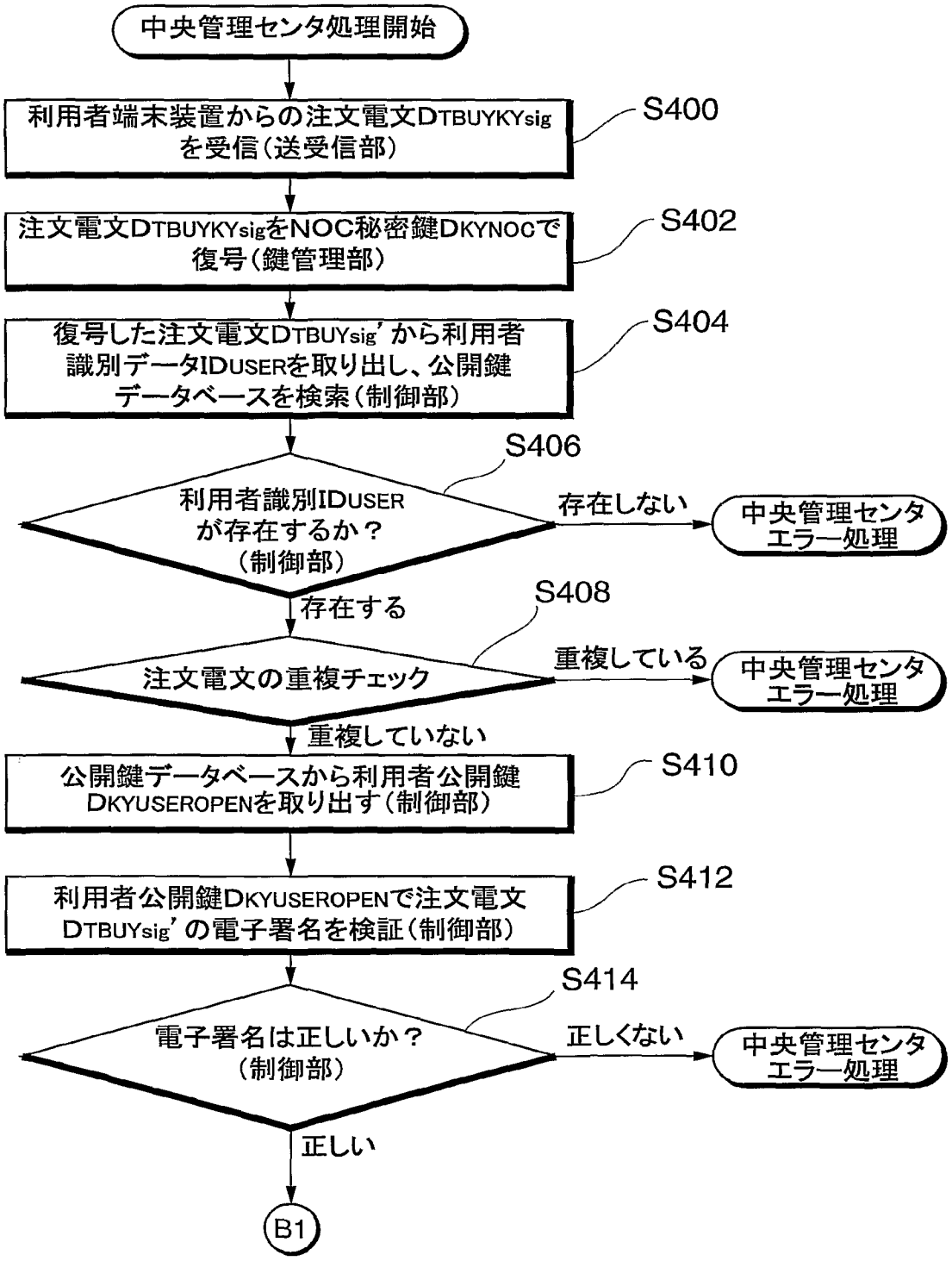

中継処理開始
S200
S202
配布電文DTSELsigを取得
（データ形式変換部）
TV放送、ラジオ放送、情報記録媒体、
紙媒体から配布電文を取得
配布電文DTSELsig中の商品情報を表示
（表示部）
S204
配布電文DTSELsigに中継端末
識別データIDTRを追加（制御部）
S206
中継端末識別データIDTRを追加した
配布電文DTSELsigを利用者端末装置へ
送信（送受信部）
S208
中継処理継続又は終了

第９図

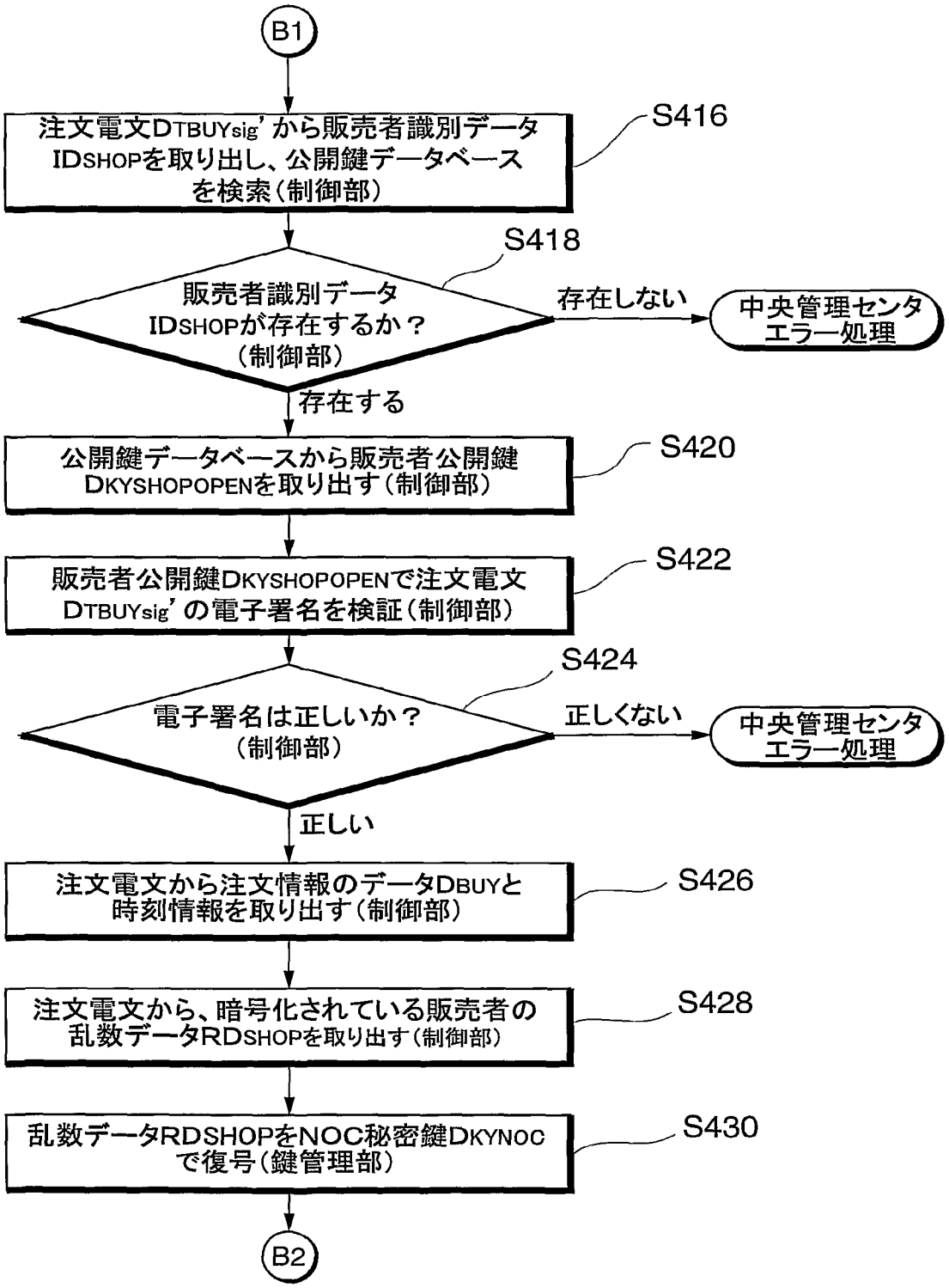


Nintendo Exhibit 1003
Part 2 of 2
Page 00819

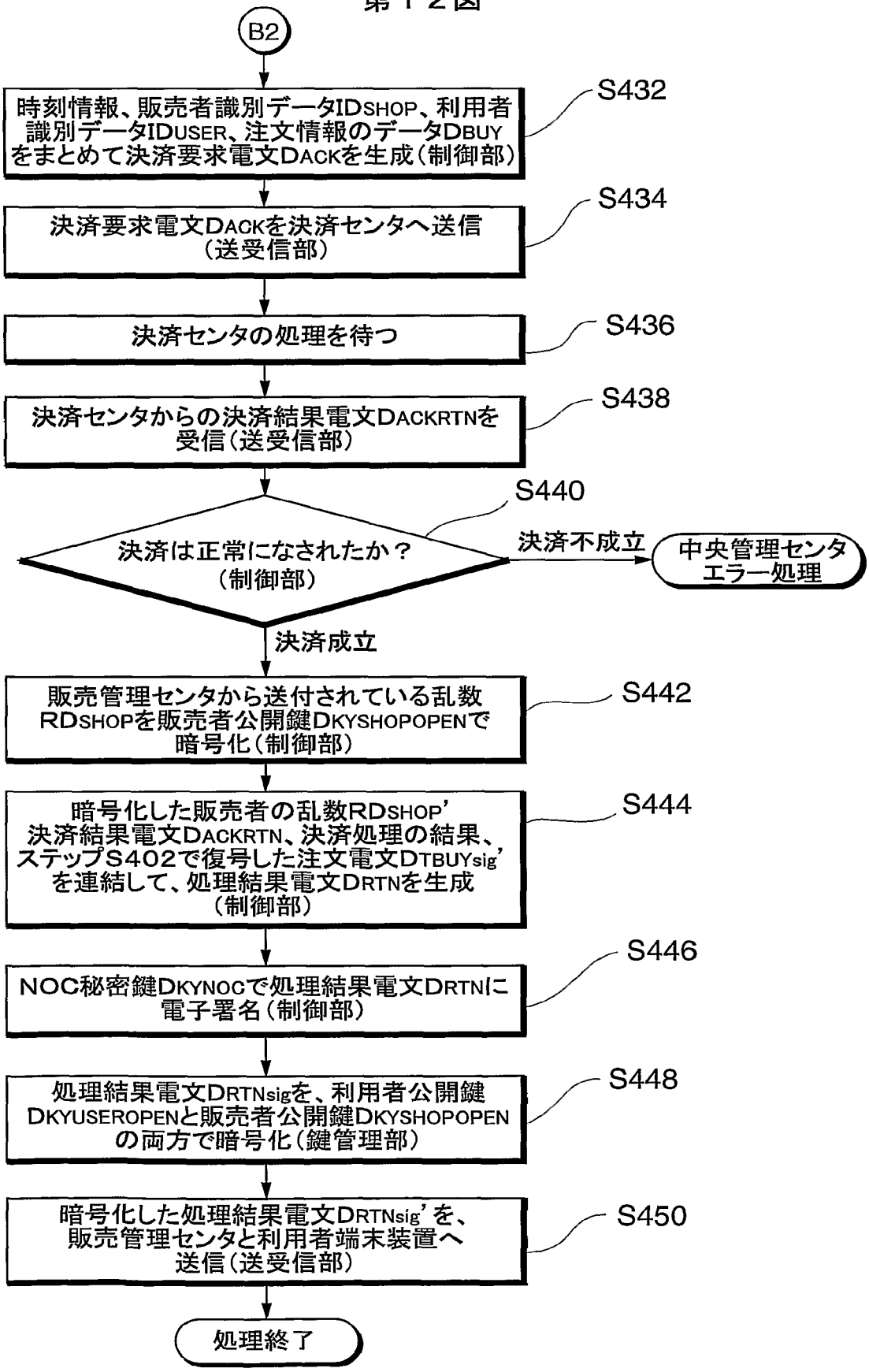
第１０図


中央管理センタ処理開始
利用者端末装置からの注文電文DTBUYKYsigを受信（送受信部）
S400
注文電文DTBUYKYsigをNOC秘密鍵DKYNOCで復号（鍵管理部）
S402
復号した注文電文DTBUYsig'から利用者識別データIDUSERを取り出し、公開鍵データベースを検索（制御部）
S404
S406
利用者識別IDUSERが存在するか？（制御部）
存在しない
中央管理センタエラー処理
存在する
S408
注文電文の重複チェック
重複している
中央管理センタエラー処理
重複していない
公開鍵データベースから利用者公開鍵DKYUSEROPENを取り出す（制御部）
S410
利用者公開鍵DKYUSEROPENで注文電文DTBUYsig'の電子署名を検証（制御部）
S412
S414
電子署名は正しいか？（制御部）
正しくない
中央管理センタエラー処理
正しい
B1

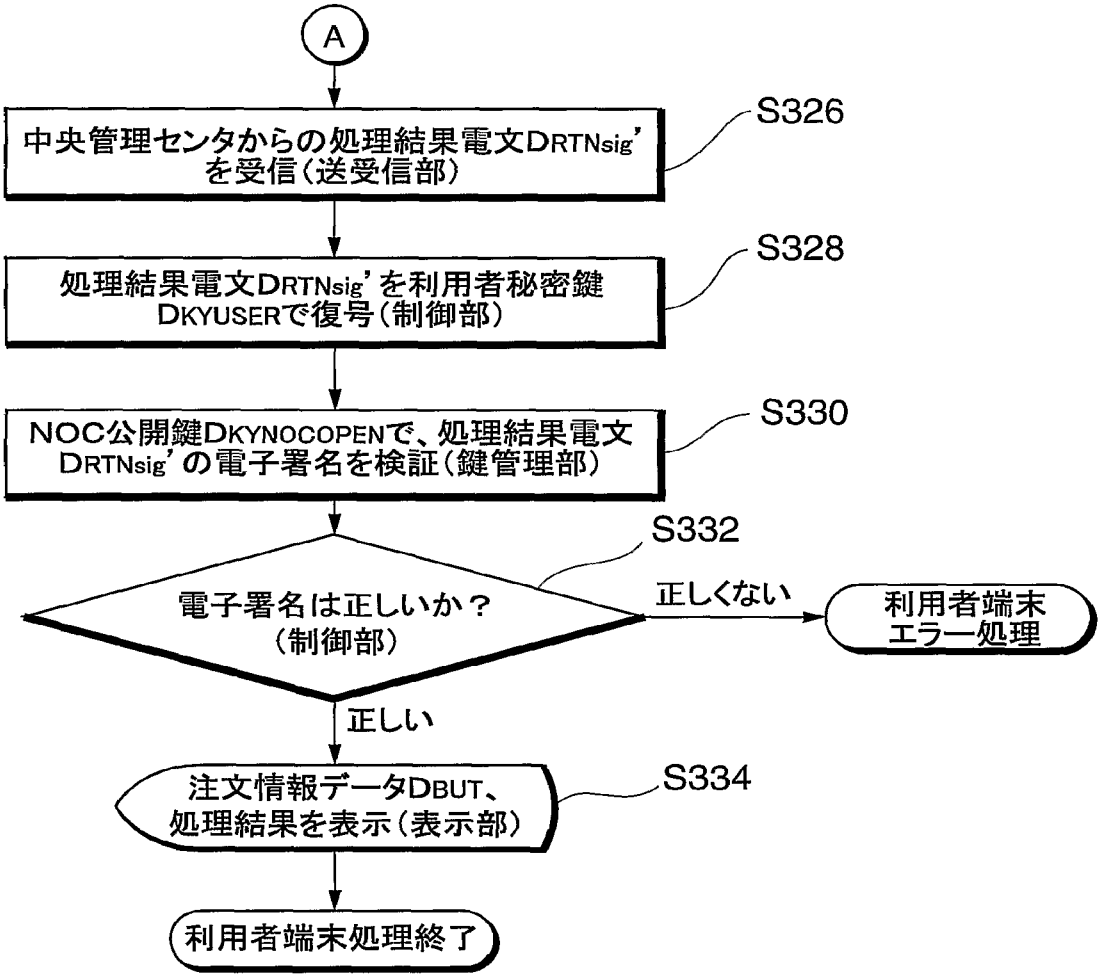
第１１図


B1
注文電文DTBUYsig'から販売者識別データIDSHOPを取り出し、公開鍵データベースを検索（制御部）
S416
S418
販売者識別データIDSHOPが存在するか？（制御部）
存在しない
中央管理センタエラー処理
存在する
公開鍵データベースから販売者公開鍵DKYSHOPOPENを取り出す（制御部）
S420
販売者公開鍵DKYSHOPOPENで注文電文DTBUYsig'の電子署名を検証（制御部）
S422
S424
電子署名は正しいか？（制御部）
正しくない
中央管理センタエラー処理
正しい
注文電文から注文情報のデータDBUYと時刻情報を取り出す（制御部）
S426
注文電文から、暗号化されている販売者の乱数データRDSHOPを取り出す（制御部）
S428
乱数データRDSHOPをNOC秘密鍵DKYNOCで復号（鍵管理部）
S430
B2

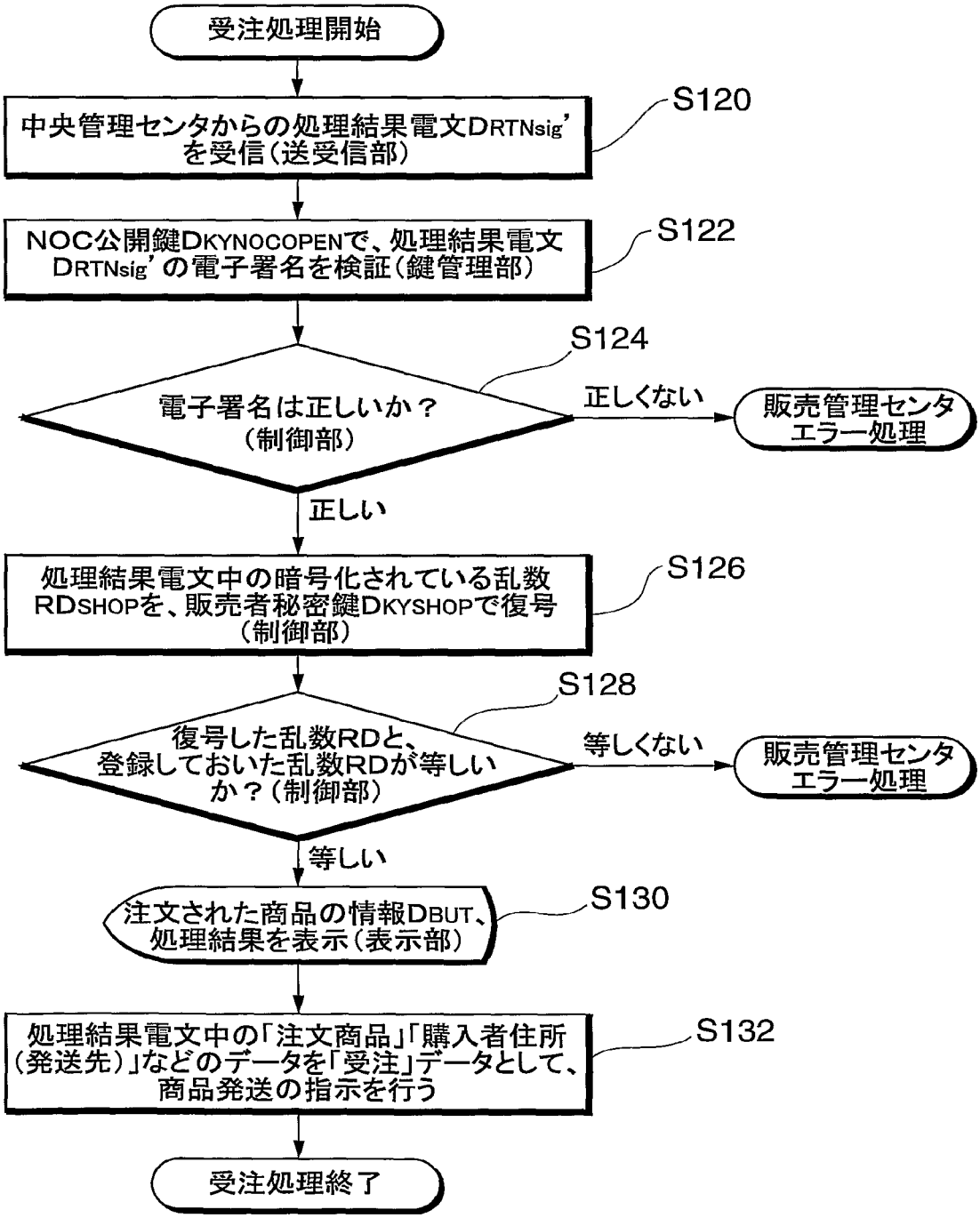
第１２図


B2
時刻情報、販売者識別データIDSHOP、利用者識別データIDUSER、注文情報のデータDBUYをまとめて決済要求電文DACKを生成（制御部）
S432
決済要求電文DACKを決済センタへ送信（送受信部）
S434
決済センタの処理を待つ
S436
決済センタからの決済結果電文DACKRTNを受信（送受信部）
S438
決済は正常になされたか？（制御部）
S440
決済不成立
中央管理センタエラー処理
決済成立
販売管理センタから送付されている乱数RDSHOPを販売者公開鍵DKYSHOPOPENで暗号化（制御部）
S442
暗号化した販売者の乱数RDSHOP'決済結果電文DACKRTN、決済処理の結果、ステップS402で復号した注文電文DTBUYsig'を連結して、処理結果電文DRTNを生成（制御部）
S444
NOC秘密鍵DKYNOCで処理結果電文DRTNに電子署名（制御部）
S446
処理結果電文DRTNsigを、利用者公開鍵DKYUSEROPENと販売者公開鍵DKYSHOPOPENの両方で暗号化（鍵管理部）
S448
暗号化した処理結果電文DRTNsig'を、販売管理センタと利用者端末装置へ送信（送受信部）
S450
処理終了

第１３図


A
中央管理センタからの処理結果電文DRTNsig'を受信（送受信部）
S326
処理結果電文DRTNsig'を利用者秘密鍵DKYUSERで復号（制御部）
S328
NOC公開鍵DKYNOCOPENで、処理結果電文DRTNsig'の電子署名を検証（鍵管理部）
S330
電子署名は正しいか？（制御部）
S332
正しくない
利用者端末エラー処理
正しい
注文情報データDBUT、処理結果を表示（表示部）
S334
利用者端末処理終了

第１４図



Nintendo Exhibit 1003
Part 2 of 2
Page 00824

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP01/02274

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl[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)
Int.Cl[7] G06F17/60

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1926-1996 Jitsuyo Shinan Toroku Koho 1996-2001
Kokai Jitsuyo Shinan Koho 1971-2001 Toroku Jitsuyo Shinan Koho 1994-2001

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US, 5608778, A (Lucent Technologies Inc.), 04 March, 1997 (04.03.97), Figs. 2, 7 & CA, 2156206, A, C & JP, 8-096043, A & EP, 708547, A2	1-3
Y	WO, 99/09502, A1 (Matsushita Electric Ind. Co., Ltd.), 25 February, 1999 (25.02.99), Fig. 84 (8410, 8418, 8421), & AU, 86484/98, A1 & EP, 950968, A1 & CN, 1246941, T	1-3
Y	WO, 92/11598, A1 (Motorola Inc.), 09 July, 1992 (09.07.92), Fig. 5(D); Fig. 1 (116) & AT, 188562, E & GR, 3032864, T, C & CA, 2096730, A & KR, 97-07003, B1 & DE, 69131897, T & JP, 6-501329, A & EP, 564469, A1, A4, B1 & JP, 3010069, B2 & EP, 940760, A1 & US, 5221838, A & ES, 2141092, T	1-3

☒ 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 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
19 June, 2001 (19.06.01)	03 July, 2001 (03.07.01)
Name and mailing address of the ISA/ Japanese Patent Office	Authorized officer
Facsimile No.	Telephone No.

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

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP01/02274

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO, 98/56181, A1 (Telia AB), 10 December, 1998 (10.12.98), Fig. 3 & EP, 986911, A1 & SE, 9702146, A & NO, 995670, A & SE, 509582, C2	1-3
Y	EP, 804012, A2 (Nokia Mobile Phones Ltd.), 29 October, 1997 (29.10.97), Fig. 1 & JP, 10-098706, A & FI, 103450, B & EP, 804012, A3 & US, 6172673, B1 & FI, 961750, A	1-3
A	US, 5324922, A (VeriFone Inc.), 28 June, 1994 (28.06.94), Fig. 1 (Family: none)	4
A	WO, 97/41499, A1 (Jerry R. Martinez), 06 November, 1997 (06.11.97), page 17, ("Another way a delivery person..."), & AU, 28027/97, A	4

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

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ０１／０２２７４

A.　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int. Cl.[7]　G06F17/60

B.　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Int. Cl.[7]　G06F17/60

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1926–1996 年
日本国公開実用新案公報	1971–2001 年
日本国実用新案登録公報	1996–2001 年
日本国登録実用新案公報	1994–2001 年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C.　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
Y	US, 5608778, A (Lucent Technologies Inc) 4．3月．1997 (04.03.97) *図 2; 図 7* &　CA, 2156206, A, C　&　JP, 8-096043, A &　EP, 708547, A2	1–3

☒ C欄の続きにも文献が列挙されている。　☐ パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの

「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）

「O」口頭による開示、使用、展示等に言及する文献

「P」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」同一パテントファミリー文献

国際調査を完了した日 19.06.01	国際調査報告の発送日 03.07.01
国際調査機関の名称及びあて先 日本国特許庁（ＩＳＡ／ＪＰ） 郵便番号１００－８９１５ 東京都千代田区霞が関三丁目４番３号	特許庁審査官（権限のある職員） 阿波　進　印　5L　9168 電話番号　０３－３５８１－１１０１　内線　３５６１

様式ＰＣＴ／ＩＳＡ／２１０（第２ページ）（１９９８年７月）

Ｃ（続き）．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
Y	WO, 99/09502, A1 （松下電器産業株式会社） 25．2月．1999 (25.02.99) *図 84（8410, 8418, 8421 を見よ）* & AU, 86484/98, A1 & EP, 950968, A1 & CN, 1246941, T	1–3
Y	WO, 92/11598, A1 (Motorola Inc) 9. 7月. 1992 (09.07.92) *図 5D; 図 1（116 を見よ）* & AT, 188562, E & GR, 3032864, T, C & CA, 2096730, A & KR, 97-07003, B1 & DE, 69131897, T & JP, 6-501329, A & EP, 564469, A1, A4, B1 & JP, 3010069, B2 & EP, 940760, A1 & US, 5221838, A & ES, 2141092, T	1–3
Y	WO, 98/56181, A1 (Telia AB) 10. 12月. 1998 (10.12.98) *図 3* & EP, 986911, A1 & SE, 9702146, A & NO, 995670, A & SE, 509582, C2	1–3
Y	EP, 804012, A2 (Nokia Mobile Phones Ltd) 29. 10月. 1997 (29.10.97) *図 1* & JP, 10-098706, A & FI, 103450, B & EP, 804012, A3 & US, 6172673, B1 & FI, 961750, A	1–3
A	US, 5324922, A (VeriFone Inc) 28. 6月. 1994 (28.06.94) *図 1* (ファミリーなし)	4
A	WO, 97/41499, A1 (Jerry R Martinez) 6. 11月. 1997 (06.11.97) *ページ 17（"Another way a delivery person..." の段落を見よ）* & AU, 28027/97, A	4

Nintendo Exhibit 1003
Part 2 of 2
Page 00828

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

(11) EP 1 168 141 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
02.01.2002 Bulletin 2002/01

(51) Int Cl.[7]: **G06F 1/00**

(21) Application number: **01305484.6**

(22) Date of filing: **25.06.2001**

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

(30) Priority **23.06.2000 US 213495 P**

(71) Applicant **Franklin Electronic Publishers, Incorporated**
Burlington, NJ 08016 (US)

(72) Inventors:
- **Yianilos, Peter**
  Princeton, New Jersey 08540 (US)
- **Kilian, Joseph**
  Princeton Junction, NJ 08550 (US)

(74) Representative: **Hale, Peter et al**
Kilburn & Strode 20 Red Lion Street
London WC1R 4PJ (GB)

(54) A secure and open computer platform

(57) A computer platform is described that provides control features to allow for the protection of intellectual property rights and prevent malfunctioning of the platform. The platform uses 1) a secure operating system including a secure memory management system, 2) public key encryption. 3) data authentication through digital signatures and 4) application/data approval through a flexible access policy through the use of object handlers and an application program approval process. Through these four control features. the platform provides the ability to control access to data and minimize the effects of computer malfunctions.

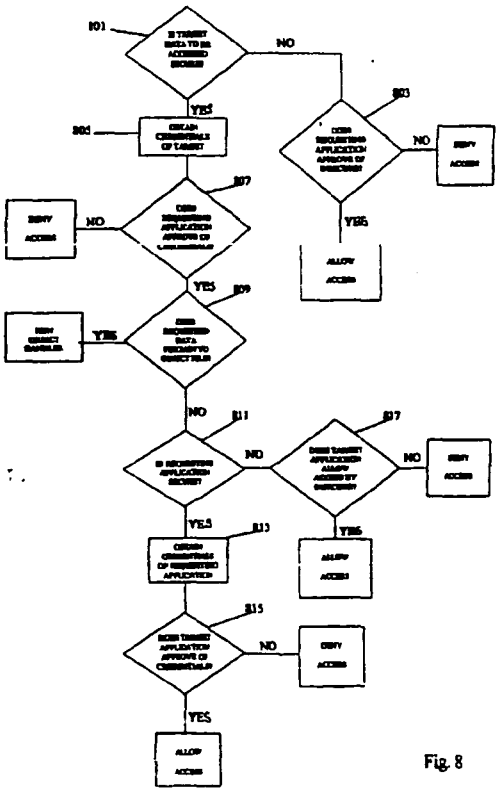


Fig. 8

EP 1 168 141 A2

Description

[0001] This invention relates to an apparatus and system for providing a computer platform. in particular. one that controls the operation of the application programs and object files to adequately protect against computer malfunctions.

[0002] In creating a computer platform. one concern is the amount of control that the platform retains over how application programs operate and data files are accessed on the platform. A computer platform can have control features that implement rules and restrictions on how application programs run and data files are accessed on that platform. Each application program operating on this platform would then usually be written in accordance with these rules or restrictions.

[0003] Control features of a platform are intended to effect some type of security to the platform user and/or the application program or object file writer. For example. a computer platform can contain certain control features to prevent undesirable computer malfunctions such as ones caused by a computer virus or a badly written application program. In other instances, a computer platform can also have control features to prevent violations of a person's intellectual property rights that can occur by unauthorized duplication of copyrighted material.

[0004] The more control features that are implemented in a given platform, the less flexible the platform is in accommodating application programs. If an application program is written to comply with a particular platform's rules, that application program might only be capable of being used on that platform. Conversely, application programs written to comply with a different platform's rules might not operate on this platform.

[0005] An application programmer will usually have to determine which platform the program application is being written for before writing the application program. Consequently, an application programmer will usually favor the platforms that are the least restrictive because it increases the chance of the application program being able to be run on the most platforms. Thus, it is more advantageous for a platform to use the least amount of rules in implementing the desired control features.

[0006] If a computer platform is intended to handle sensitive copyrighted material, then prevention of unauthorized copying becomes paramount and other control issues are less important. With the advent and popularity of digital publications and electronic distribution of publications to be read on electronic readers, protection of copyrights on a computer platform has become important.

[0007] Consequently, it would be advantageous to provide a computer platform that provides control features to prevent unwanted violations of intellectual property rights and still allows enough flexibility for application programs and object files. It would also be advantageous to provide a computer platform that provides control features that protect against malfunctions of program applications on the platform.

[0008] The current invention involves a system that provides enough control features to create a secure platform and yet maintain the flexibility to be able to operate and run a large variety of applications.

[0009] The present invention is defined in the accompanying independent claims. Some preferred features are recited in the dependent claims.

[0010] In a preferred embodiment the invented system entails utilizing control features which in combination protect against malfunctions in a computer platform and provides the ability to prevent unauthorized access to copyrighted material. These security measures include:

1) A secure operating system. including a secure memory management system;
2) Public key encryption;
3) Data authentication through digital signatures; and
4) Application program/object file approval.

[0011] Providing a secure operating system can entail two aspects: 1) ensuring that the operating system is approved for the platform and 2) creating firewalls around application programs and object files that operate on the platform. The term "firewall" is used herein to refer to the arrangement in the computer platform that employs certain memory management components of the operating system to bar access to an application program or an object file in memory. Access to the object file or application program can be granted only if the required permission is obtained.

[0012] Encryption generally entails reformatting data using an "encryption" key such that no one else will be able to utilize or read that data if they do not have an appropriate "decryption" key.

[0013] Data authentication entails the ability of verifying the author and title of data to ensure that the data is properly authored and has not been tampered with from the time of creation by that author to the receipt by the platform.

[0014] Application/data approval allows application programs to reject the type of object files that it will operate on and also allows the object files to reject the application through a flexible access policy through the use of object handlers and an application program approval process.

[0015] In the embodiment of the invented system, these four types of control features are available to be accessed by an application program or data file while on the platform. Application programs and files can operate on the platform without utilizing the control features. Through these four types of control features, a computer platform that is both open to accept various application programs and to protect intellectual property is provided.

EP 1168141A2 I >

Nintendo Exhibit 1003
Part 2 of 2
Page 00830

[0016] The present invention can be put into practice in various ways, some of which will now be described with reference to the accompanying drawings, in which:

Fig. 1 depicts the layout of an apparatus that implements one embodiment of the present invention;
Fig.2 depicts a flow chart demonstrating the creation of a digital signature in accordance with one embodiment of the present invention;
Fig. 3 depicts a flow chart demonstrating the signature authentication process in accordance with one embodiment of the present invention;
Fig. 4 depicts a flow chart demonstrating the creation of a digital signature along with the further step of encryption in accordance with one embodiment of the present invention;
Fig. 5 depicts a flow chart demonstrating the signature authentication process along with the further step of decryption in accordance with one embodiment of the present invention;
Fig. 6 depicts a flow chart demonstrating the creation of a digital signature along with the further step of encryption in accordance with another embodiment of the present invention
Fig. 7 depicts a flow chart demonstrating the signature authorization process along with the further step of decryption in accordance with another embodiment of the present invention; and
Fig. 8 depicts a flow chart demonstrating the steps of obtaining permission to access data in memory in accordance with one embodiment of the present invention.

[0017] Fig. 1 depicts the layout of a device that implements one embodiment of the present invention. Computer platform 101 is provided with an input interface 103 and an output interface 105 to allow the platform to obtain and transmit data. Input interface 103 and output interface 105 can be the same physical interface so long as it has the capability of both receiving and transmitting data.

[0018] Hardware 107 is contained within computer platform 101 and implements firmware 109. Hardware 107 also comprises memory registers 111. The computer platform's operating system will be loaded onto the hardware 107.

[0019] The operating system ("O/S") is the foundation for the platform's security and digital rights management infrastructure. Firmware 109 contains an O/S verifier 113 to authenticate the O/S before the O/S is loaded onto the hardware 107. Every time that the O/S is loaded or modified, the O/S verifier 113 must authenticate the O/S. O/S verification prevents potential circumvention of the security features that are implemented by the O/S by unauthorized modification or substitution of the O/S. O/S verification does not impose any restrictions that would affect application programs running on the computer platform

[0020] O/S verification can be accomplished by various methods. One method of O/S verification that can be implemented uses digital signatures. Algorithms for creating and verifying signatures, and for generating the public/private signing key pairs, and well known in the literature (c.f. Bruce, Schneier. "Applied Cryptography, second edition", John Wiley & Sons, Inc. New York (1996). Figs. 2 and 3 depict digital signature verification for the O/S. This technique uses public/secret signature keys. The first step 201 of the digital signature process is to create the data packet to be signed. The data packet is not necessarily just the O/S. Other information may be included as part of the data packet. For example, the credentials of the data can be included along with the data packet. These credentials can include various items to identify the O/S, such as, the author, version and date of creation.

[0021] After determining the data packet to be signed, the next step 203 is to apply a hash function to the data packet. A hash function essentially creates a value of fixed length called the hash value. The hash value is derived from characteristics of the data packet. Different data packets will usually create different hash values. Furthermore, it is computationally unfeasible to produce two different packets that have the same hash value.

[0022] Each author is associated with one or more public/private pairs of signing keys. After the hash value is created (203), the hash value and the private signing key are input to a fixed, publicly known signing algorithm, that produces a digital signature as its output (205). Each private signing key is unique to and known by a single author. The resulting signature is unique to the author who knows the private signing key.

[0023] Fig. 3 depicts the steps that are taken to authenticate a signed data packet. The first step (301) is to recalculate the hash value of the data packet. This calculation reapplies the hash function to the data packet. The recalculated hash value, the signature and the public signing key are given as inputs to a fixed, publicly known signature verification algorithm (303). This algorithm outputs one of two values: "accept", meaning that the signature is to be accepted and "reject", meaning that the signature is to be rejected.

[0024] The combination of creating a hash code and signing the hash code to create the digital signature allows the ability of ensuring that the signer created the signature and that the data packet has not been altered since the signature was created.

[0025] For the O/S verifier, the authorized O/S programmer's public signature key is burned into the hardware of the computer platform. This is necessary since the O/S is usually the first software that is loaded onto the computer platform and the computer platform must be able to verify that O/S when it is loaded. The authorized O/S programmer is usually the manufacturer of the computer platform.

[0026] The procedures of creating a digital signature and authenticating the digital signature detailed in Figs.

BNSDOCID <EP 1168141A2 I >

Nintendo Exhibit 1003
Part 2 of 2
Page 00831

2 and 3 are not limited to O/S verification. These procedures can also be applicable to the process of authenticating application programs and object files. However, in addition to the procedures in Figs 2 and 3, authenticating application programs and object files can entail additional steps.

[0027] Figs. 4 and 5 depict the procedures that can be implemented when authenticating signatures of application programs and object files. Similar to the procedures outlined in Fig. 2, the first three steps of creating a signature for application programs and object files are the steps of creating the data packet 401, creating the hash value 403 and creating the digital signature 405. For application programs and object files, credentials should be included with the data packet and those credentials should include at least the name of the author of the data. After the signature of the data packet is created, the data packet and the signature are both encrypted with a public encryption key in step 407.

[0028] Similar to the public/secret signature keys, the public encryption key is distinctively related to a private decryption key. However, both encryption/decryption keys are unique to a particular computer platform rather than a particular programmer. Another difference is that the public encryption key is used to encrypt the data while the private decryption key is used to decrypt the data. The private decryption key is kept secret and is only known to that particular computer platform. This permits only that computer platform access to the encrypted data that has been encrypted with the public key unique to the platform.

[0029] To authenticate the application program or object file, the first step 501 is to decrypt the encrypted data with the computer platform's private decryption key. After the data has been decrypted, the decrypted data should consist of the data packet and the digital signature. The next steps are to recreate the hash value of the data packet 503, and obtain the output from the publicly known signature verification algorithm 505. If the hash values do not match, then the data is erased from memory.

[0030] Encryption of the data packet by using the public/private key technique provides rigorous protection against unauthorized access to the encrypted data by using two separate keys. This process, however, can utilize a substantial amount of the platform's processing resources. In addition, by encrypting the entire data packet, the encryption and decryption process can take a substantial amount of time to perform.

[0031] Another alternative to using the public/private keys to encrypt the data packet is to utilize a less complicated encryption technique in addition to the public/private key encryption. Figs. 6 and 7 depict the process of authenticating program applications and object files utilizing the second encryption technique.

[0032] The first step 601 is to create the data packet. The second and third steps 603, 605 are to create the hash value and digital signatures of the data packet. The fourth step 607 is to encrypt the data packet and signature with a single encryption/decryption key. This single key is used to both encrypt and decrypt. The single decryption key, itself, is then encrypted in the next step 609 with the public encryption key that pertains to the particular platform.

[0033] To decrypt the data at the platform, the platform will first have to decrypt the single encryption/decryption key by using its private decryption key in step 701. With the decrypted single encryption/decryption key, the platform now has the single encryption/decryption key and the data packet can then be decrypted. By separately encrypting the single encryption/decryption key with the public/private keys, the benefit of using the more secure public/private encryption system to protect the entire data packet is gained while maintaining a lower level of complexity.

[0034] Once an application program or object file has been authenticated by the procedures detailed in Figs. 4 or 6 or other similar procedures, it is considered to be secure data. If an application program or object file is not authenticated, it is considered to be insecure.

[0035] By having the encryption/decryption keys unique to each computer platform, the invented system can restrict the availability of data to a particular computer platform. Any computer platform that does not have the correct private decryption key for an encrypted data packet cannot properly decrypt it. This ability, in combination with the signature authentication procedures, provides the ability to exert complete control over data transmissions to the computer platform by: 1) ensuring that any data transmission to the computer platform can only be read by that platform, and 2) ensuring that the data transmission has not been tampered with since the author of the data signed it.

[0036] The computer platform's private decryption key must be burned into the hardware so that the platform will always be able to decrypt encrypted data packets. In addition, authentication of application programs and object file should be done every time that an application program or object files is placed into the computer platform's memory. By performing this authentication every instance that data is loaded onto the platform, the system ensures that all data loaded on the computer platform will be properly characterized as either secure or insecure.

[0037] Although only the manufacturer's public signature is burned into the hardware, the ability to designate application programs and object files as being secure is not restricted to the manufacturer of the computer platform. Any programmer that writes an application program or object file to be used on the computer platform can designate it as being secure data. To accommodate different programmers in designating data as secure, the programmers can send all data to be designated as secure to the manufacturer. After authenticating the data from the programmer, the manufacturer will then digitally sign the data with its secret signature key that cor-

Nintendo Exhibit 1003
Part 2 of 2
Page 00832

responds to the public signature key that has been burned into the computer platform's hardware.

[0038] Another embodiment entails sending the application programmer's public signature key to the manufacturer. The manufacturer can then digitally sign this public signature key. This signature can then be appended to signatures using the application programmer's signature key. Since it has been signed by the manufacturer, the computer platform will accept the programmer's public signature key as an additional signature key that can be used to authenticate application programs and object files as being secure. These additional public signature keys and the corresponding identities are stored in a list by the O/S. This list will be accessed by the O/S and the identity contained in the credentials will be used to determine the appropriate public signature key.

[0039] This procedure of approving another programmer's public signature key is not applicable to O/S verification. It is important to always control the O/S because it ensures that the security features imposed by the O/S cannot be circumvented.

[0040] One security feature implemented by the O/S that should not be circumvented is creating firewalls around data in the memory registers 111 to preserve the integrity and privacy of the application programs and the object files. Both an application program's memory and an object files' memory are automatically shielded from all other applications that may be running on the computer platform 101. No special programming by the programmers is needed to enjoy this protection. To breach a firewall, the O/S will seek permission from two places: 1) the application program requesting the breach and 2) the data in memory that is to be accessed.

[0041] Fig. 8 depicts the process of obtaining the appropriate permission before allowing access to application programs or object files in memory. For the first step 801, the O/S determines if the data in memory to be accessed belongs to a secure object file or application program. If the data in memory is not secure, then the O/S informs the requesting application program that it is not secure. The O/S will permit the requesting application program to gain access to the insecure data if the requesting application has been written to allow access to insecure data as determined in step 803.

[0042] If the data in memory pertains to secure data, then the O/S provides the credentials of the secure target data to the requesting application program in step 805 in order for the requesting application program to determine if the credentials are approved in step 807.

[0043] The O/S will not automatically provide access to the secure data if the requesting application program approves of the credentials. With secure data, the O/S then determines whether the data pertains to an application program or an object file in step 809.

[0044] If the memory belongs to an object file, then the O/S will usually run an object handler associated with the object file. The term "object handler" used herein refers to a program that is associated with object files that determine if access to that object file is permitted. For example, an object handler associated with a particular object file might permit access through the fire wall for an application program which derives from the publisher of the object file. The object handler can use a number of parameters (publisher, expiration date, platform identifications, etc.) as criteria for access. If the object handler determines that permission to access the desired memory is permitted, the O/S will allow access.

[0045] Object handlers are usually created by the author of the data packet and are included along with the data packet that will be digitally signed. However, default object handlers can created by the O/S for secure object files that have no object handler in another embodiment of the present invention.

[0046] If the target data in memory belongs to an application program, then the O/S must then determine if the target application approves of the requesting application program. The next step 811 is to determine if the requesting application is secure. Depending on the requesting application's secure status, either the credentials of the requesting application is communicated to the target application program in step 813 or the fact that it is insecure is communicated to the target application program. In either case, the target program application must determine if access to its memory is granted as depicted in steps 815 and 817 respectively.

[0047] Obtaining permission to breach a firewall must be obtained every new instance in which an application program requests access to data in memory. Once an application program obtains permission to access the memory, the application program does not have to obtain permission again until it terminates its access. Once access is terminated, a successive request for access to data in memory will treated as a new instance and the requisite permission must be obtained.

[0048] By rigorously protecting the integrity of application programs and object files in memory, the system minimizes the damage caused by computer malfunctions. Any malfunctioning application program or computer viruses can be blocked from affecting other program applications and object files. The firewall essentially isolates potential adverse effects to application programs and object files.

[0049] Firewalls also allow the system to extend the control over data transmissions exerted through data encryption and signature authentication to cover the data while in the computer platform's memory. Once in the computer platform's memory, unauthorized access is blocked by the combination of the firewalls and the approval process through object handlers and application program approvals. Thus, unauthorized access to the data is prevented from the time of the creation of the digital signature through to the transmission of the data to the computer platform and use on the platform.

[0050] Unauthorized access to the data can also be prevented when the data is exported out of the computer

EPODOC <EP 1168141A2 I >

Nintendo Exhibit 1003
Part 2 of 2
Page 00833

platform by applying the encryption/decryption process to all data being transmitted out of the computer platform. Every time that any data marked as secure is transferred out of the computer platform, the computer platform encrypts the data with the computer platform's encryption key. By encrypting the data, this prevents anyone who does not have that computer platform's decryption key from improperly accessing the data that has been transferred out of the platform.

[0051] A variety of data may be designated as sensitive by the OS, meaning that it is undesirable for other entities to learn the contents of this data. For example, the OS may designate as sensitive data that it receives in encrypted form. The OS may automatically deem portions of the memory used by a secure application to be sensitive. Or it may act on a request, generated by an application or the OS itself, that data be designated as being sensitive.

[0052] Normally, sensitive date is kept within the computer platform. However, due to memory limitations or to enable backups, it may be necessary to export sensitive data from the computer platform. In the preferred embodiment of the invention, information that is designated as sensitive by the OS is encrypted when it is transmitted, in its entirety or in part, out of the computer platform, and decrypted when it is reimported into the computer platform. In the preferred embodiment of the invention, the OS performs these encryption and decryption steps automatically. Other modes of operation are possible and may be more suitable for specific contexts. For example, the OS may query the computer platform's user or applications residing on the computer platform before performing the encryption or decryption operations. By controlling access to data that has been transferred out of the platform, the invented system effects control over access to the data in all instances of potential access after the digital signature is created. Besides fraudulently securing permission by accessing the secret and private keys, a person must pass the implemented security measures to gain access to secure data. These security features do not impose restrictive rules upon the application programmer because insecure application programs and object files can still operate on the invented system.

[0053] The present invention is not to be considered limited in scope by the preferred embodiments described in the specification. The software for the system by which the method of the invention is executed can be stored on any suitable computer readable medium such as floppy disc, computer hard drive, CD-ROM, Flash ROM, non-volatile ROM and RAM. The medium can be magnetically or optically readable. Additional advantages and modifications, which readily occur to those skilled in the art from consideration and specification and practice of this invention are intended to be within the scope of the following claims.

Claims

1. A data control system comprising

   a computer platform operable to authenticate an operating system to be loaded on the platform and preventing the operating system from being loaded onto the platform when the operating system is not authenticated;
   a memory in which application programs and object files can be stored, the operating system being operable to create a firewall around data in memory pertaining to application programs and object files to control access to the application programs and object files;
   an input interface connected with the platform to allow input data to be received by the platform, the operating system being capable of decrypting the input data and of authenticating the input data, and the firewalls around the data in memory being capable of allowing the application programs to access the data in memory when approval of access is obtained from the application program and from the data in memory; and
   an output interface connected to the platform to allow the platform to transmit output data out of the platform, which output data is encrypted when transmitted.

2. A data control system as claimed in claim 1, wherein the platform authenticates the operating system by verifying a digital signature associated with the operating system.

3. A data control system as claimed in claim 1 or 2, wherein the operating system decrypts the input data with a private decryption key unique to the platform.

4. A data control system as claimed in claim 3, further comprising a sending station capable of encrypting data with a public encryption key; said public encryption key being directly related to said private decryption key of said computer platform.

5. A data control system as claimed in any of claims 1 to 4, wherein the operating system authenticates the input data by verifying a digital signature associated with the input data with a public signature key and input data that is not authenticated by said operating system is classified as insecure data.

6. A data control system as claimed in any of claims 1 to 5, wherein the output interface encrypts the output data when the output data includes at least a portion of data that has been authenticated by the operating system.

Nintendo Exhibit 1003
Part 2 of 2
Page 00834

7. A data control system as claimed in claim 5, further comprising a sending station capable of creating a digital signature with a secret signature key: the secret signature key being distinctively associated with the sending station.

8. A data control system as claimed in any of claims 1 to 4, wherein the operating system is capable of authenticating the input data by using a hash function.

9. A data control system as claimed in any of claims 1 to 8, wherein the data in memory gives approval for access through an object handler associated with each of the object files when the data in memory pertains to the object files.

10. A data control system as claimed in any of claims 1 to 9, wherein the output data is encrypted with a public encryption key unique to the platform.

11. A data control system as claimed in claim 10, wherein the output data is decrypted with a private decryption key associated with the public encryption key.

12. A data control system as claimed in any of claims 1 to 11 in which the computer platform includes hardware for authenticating the operating system and preventing the operating system being loaded when it is not authenticated.

13. A data control system comprising:

a sending station, including: (a) a plurality of application programs, (b) a plurality of object files, (c) a plurality of handler programs, each associated with a separate one of said object files; and (d) a plurality of secret key encoded signatures, each distinctive to a subset of said application programs and said object files;
a plurality of receiving platforms, each having firmware and an operating system, the firmware being operable to authenticate the operating system,
each of the receiving platforms being adapted to receive the application programs, object files, handlers and signatures, and each of the receiving platforms having: (a) a public signature identification key to authenticate the signatures and (b) firewalls associated with the application programs and object files to control access to each of the application programs and object files, one of the handler programs associated with each of the object files being operable to permit access to the associated object files by an appropriate one or more of the application programs; and each of the handler programs being programmable to permit multi-parameter control over access to the associated object files

14. The data control system of claim 13 wherein: the object files and the application programs at the sending station are encrypted with a public key unique to the receiving platform being addressed and wherein the encrypted object files and application programs are decrypted with a private key, at the receiving platform.

15. The data control system of claim 13 or 16, wherein the signature identification is provided through a signature creation algorithm and a secret key at the sending station, and through a signature verification algorithm and a public key at each receiving platform.

16. The system of any of claims 13 to 15 wherein,

the sending station has a plurality of secret key encoded signatures, each signature being distinctive to a separate set of application programs and data texts,
each receiving platform having a plurality of public signature identification keys to correspond to the plurality of secret keys at the sending station.

17. A method for providing a data control system, comprising the steps of:

authenticating an operating system to be loaded on a computer platform; the authenticating step to be performed every time an operating system is loaded on the computer platform;
verifying credentials of data transmitted to the computer platform before loading the data into memory of the computerplatform;
creating firewalls around data loaded into memory of the computer platform;
decrypting data transmitted to the computer platform with a private decryption key unique to the computer platform; and
encrypting data transferred out of the computer platform with a public encryption key unique to the computer platform and associated with the public decryption key.

18. A method as claimed in claim 17, wherein the authenticating step is performed by verifying a digital signature associated with the operating system.

19. A method as claimed in claim 17 or 18, further comprising the step of obtaining permission before allowing an application program to access data loaded into memory.

EPODOC <EP 1168141A2 I_>

Nintendo Exhibit 1003
Part 2 of 2
Page 00835

20. A method as claimed in claim 19 wherein the obtaining step is performed through object handlers.

21. A computer readable medium having computer-executable instructions for performing the method of any of claims 16 to 20

Nintendo Exhibit 1003
Part 2 of 2
Page 00836

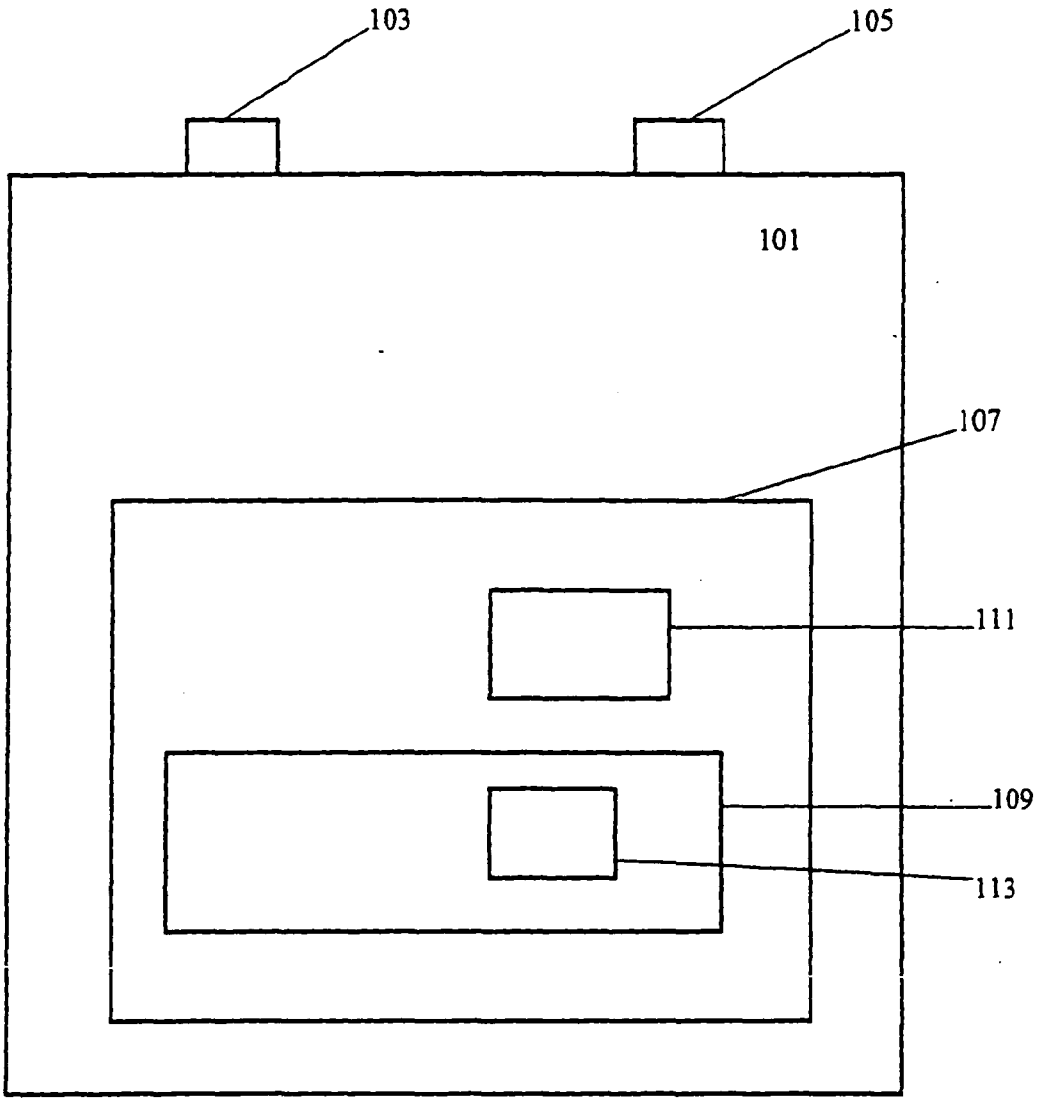


Fig 1

BNSDOCID <EP 1168141A2 I >

Nintendo Exhibit 1003
Part 2 of 2
Page 00837

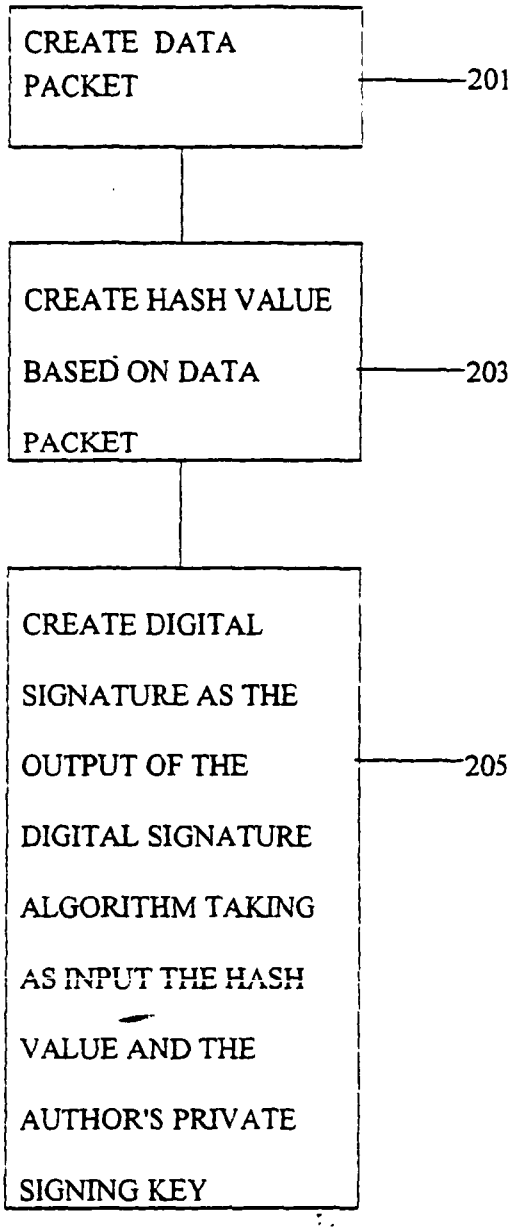


Fig. 2

Nintendo Exhibit 1003
Part 2 of 2
Page 00838

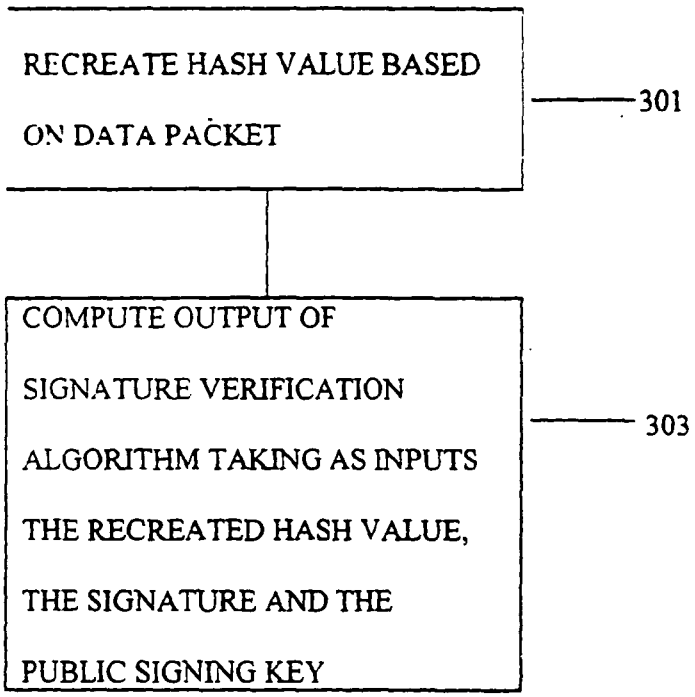


Fig. 3

EPODOC <EP_____ 1168141A2_I >

Nintendo Exhibit 1003
Part 2 of 2
Page 00839

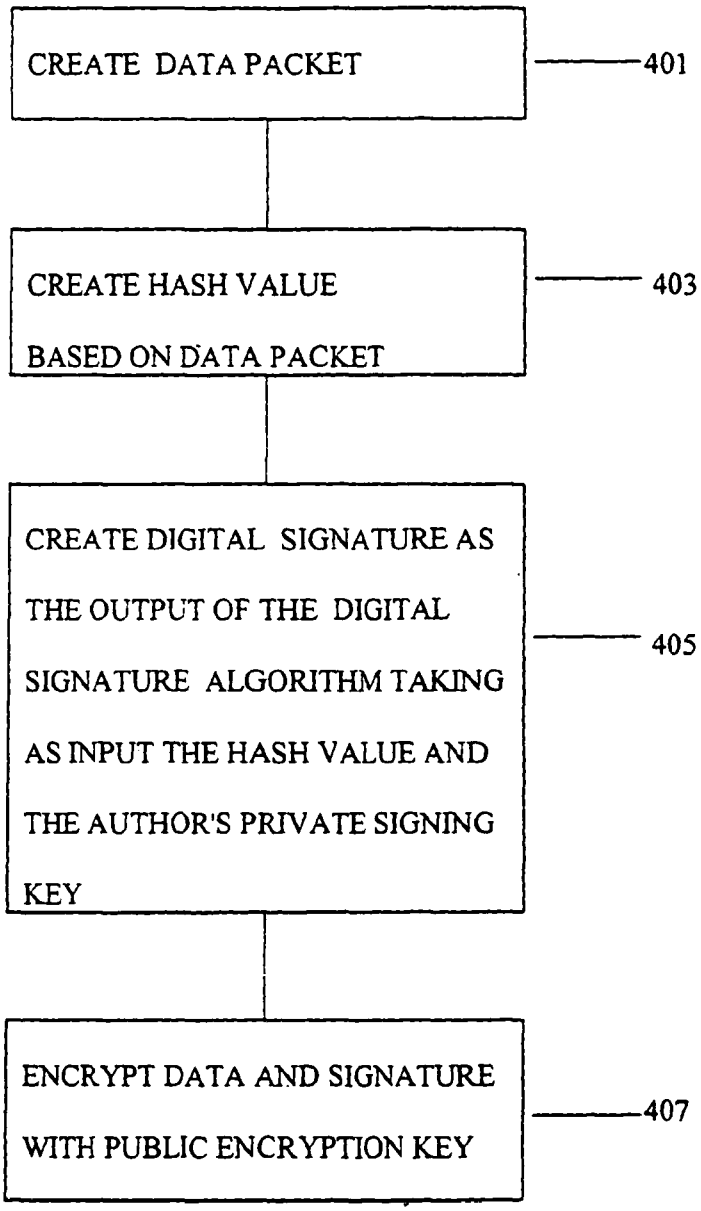


Fig. 4

Nintendo Exhibit 1003
Part 2 of 2
Page 00840

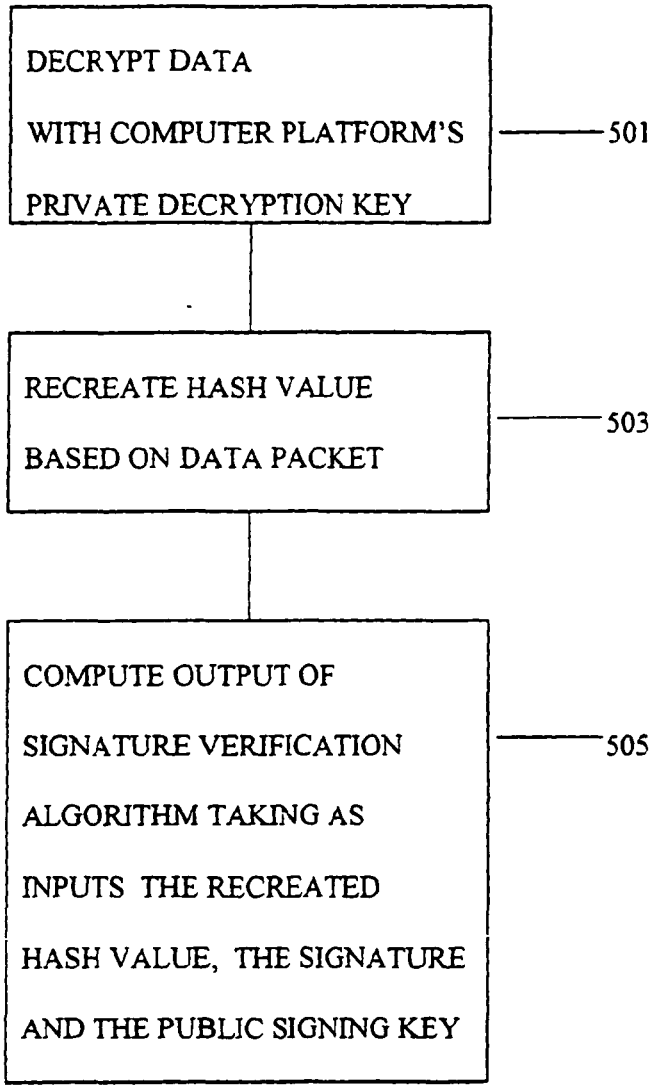


Fig. 5

EPODOC <EP 1168141A2 1 >

Nintendo Exhibit 1003
Part 2 of 2
Page 00841

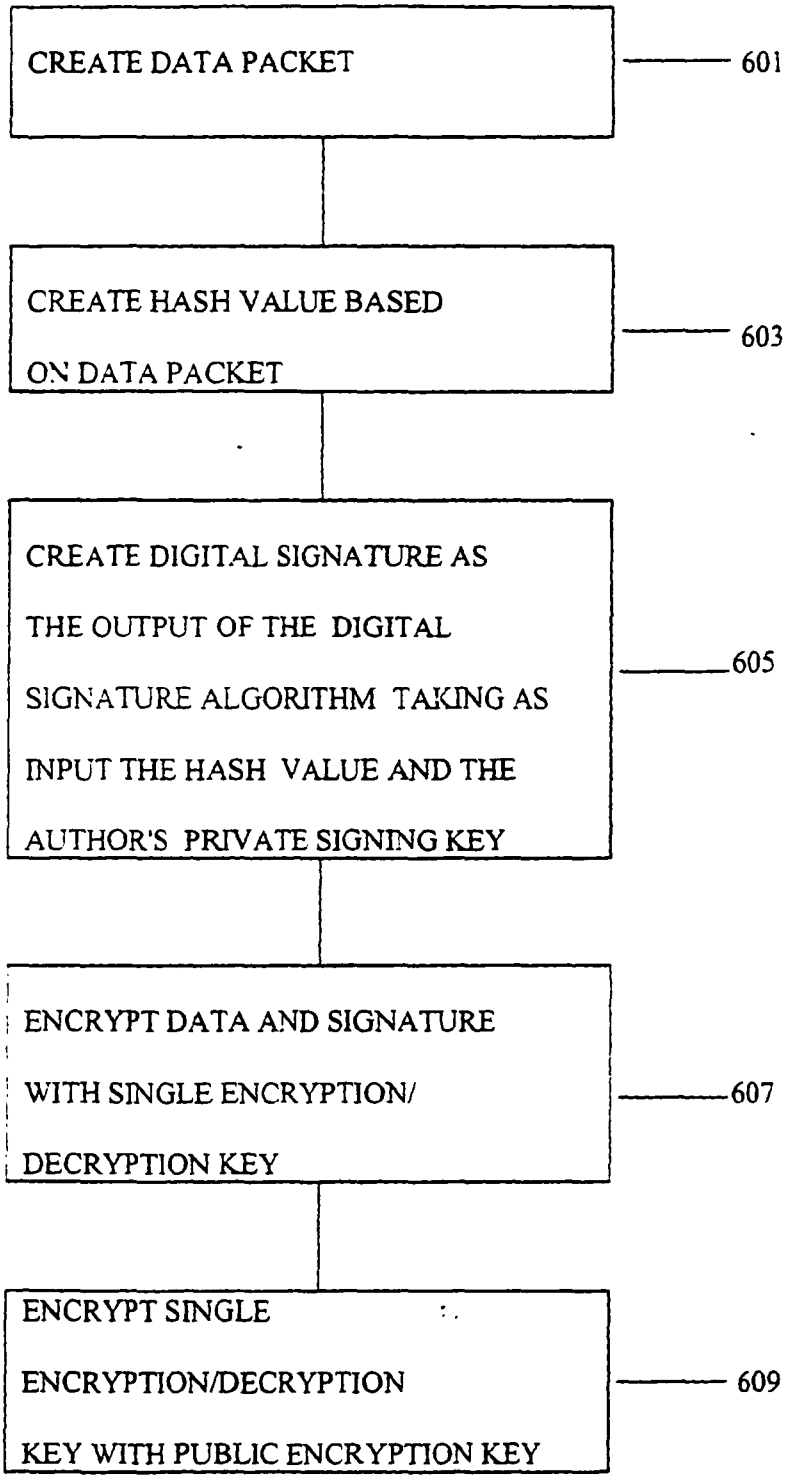


Fig. 6

Nintendo Exhibit 1003
Part 2 of 2
Page 00842

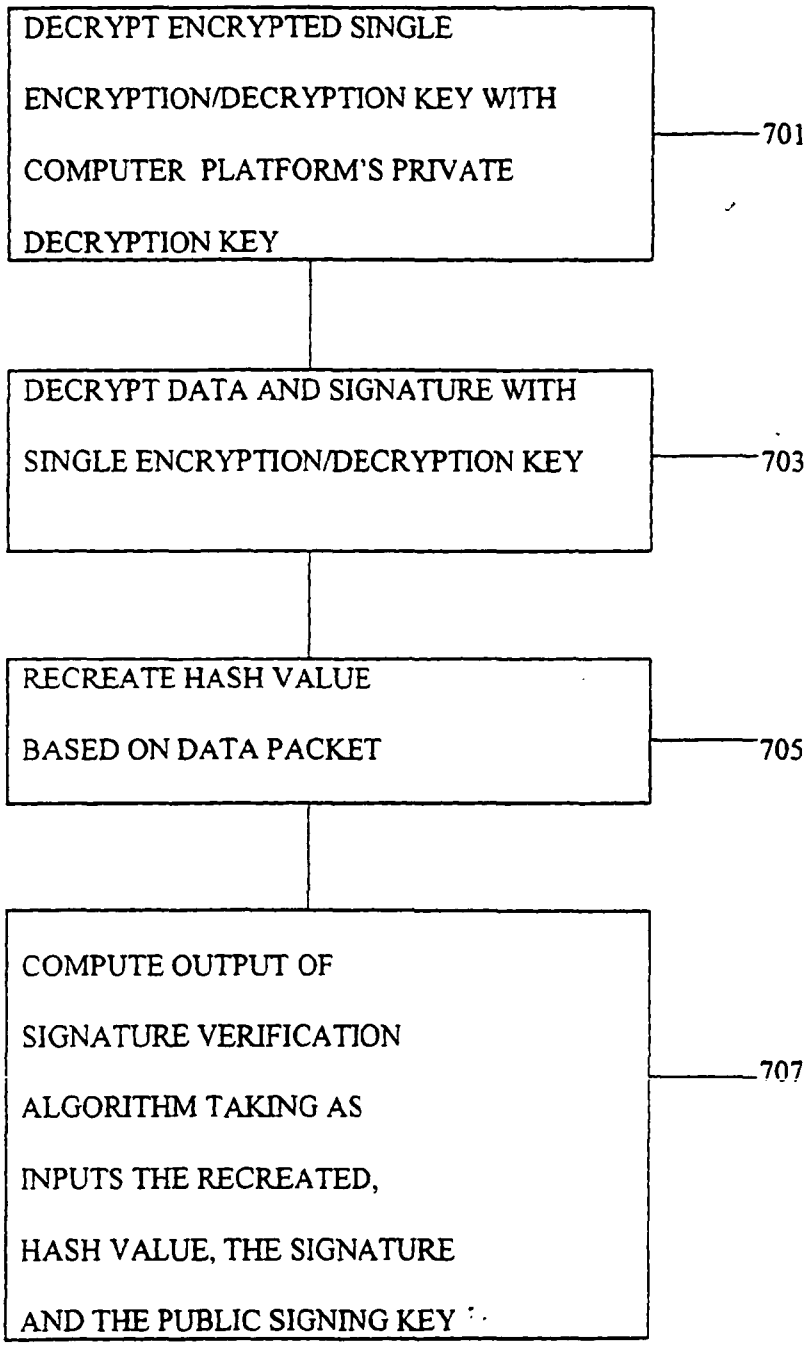


Fig. 7

EPODOC <EP 1168141A2 I >

Nintendo Exhibit 1003
Part 2 of 2
Page 00843

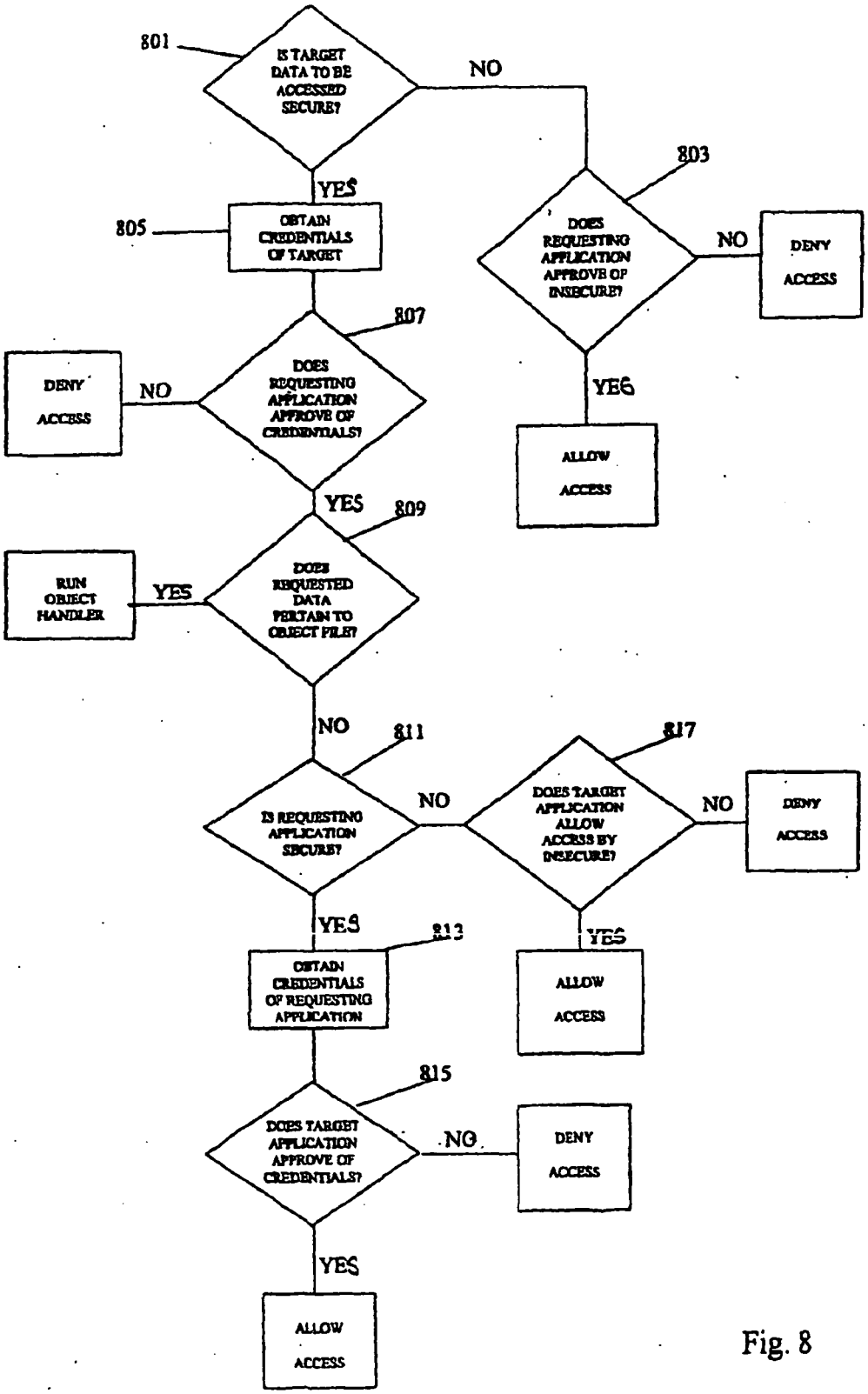


Fig. 8

Nintendo Exhibit 1003
Part 2 of 2
Page 00844

Electronic Acknowledgement Receipt

EFS ID:	22539799
Application Number:	14715227
International Application Number:	
Confirmation Number:	5874
Title of Invention:	System and Method of Owner Application Control of Electronic Devices
First Named Inventor/Applicant Name:	Neil P. Adams
Customer Number:	82313
Filer:	J. Robert Brown/Rami Law
Filer Authorized By:	J. Robert Brown
Attorney Docket Number:	11913-US-CNT3 4214-25714
Receipt Date:	04-JUN-2015
Filing Date:	18-MAY-2015
Time Stamp:	16:35:04
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Foreign Reference	EP0384610.pdf	410277 4448237ada4a3ffb64b4a14233304d3a3a707bf7	no	8

Warnings:	
Information:	

2	Foreign Reference	GB2312767.pdf	2479517 e54f147c2b57d3125ff8971018290cefc4046089	no	20
Warnings:					
Information:					
3	Foreign Reference	EP0828209.pdf	1756948 a073321c8739409a2fab1eae29936bb674292b5e	no	12
Warnings:					
Information:					
4	Foreign Reference	WO9844404.pdf	2172226 c20e09f3bb4e650eb1c49efe12aa7e07bba37ae7	no	22
Warnings:					
Information:					
5	Foreign Reference	WO0060434.pdf	3340148 a386a366a988287aa01d192abf4cae748c7c012a	no	32
Warnings:					
Information:					
6	Foreign Reference	GB2355819.pdf	3755189 956f6a3e58200883a74cfa4e5074719fed928a16	no	35
Warnings:					
Information:					
7	Foreign Reference	WO0171590.pdf	6432890 3f8913b94f0437223ba7ac8a180993df9f090d85	no	50
Warnings:					
Information:					
8	Foreign Reference	WO2001077811.pdf	4221182 4188616571b0e20365f43f4f67c1dc8d6084d8ac	no	33
Warnings:					
Information:					
9	Foreign Reference	EP1168141.pdf	2006610 db572705acd6e70638981643953cfcafdfd8c53	no	16
Warnings:					
Information:					
10	Foreign Reference	JP200256360.pdf	5472684 915ab8b9bf32770250e13aaf28a39f9e6227da48	no	31
Warnings:					
Information:					

11	Foreign Reference	JP2002091598.pdf	1846999 653d4269021a70a9d4aa330293f7a395b7ba952e	no	9
Warnings:					
Information:					
12	Foreign Reference	JP2002170063.pdf	3178149 7bdd4f10127fca038c7b6fc464dcb1b3006246fb	no	15
Warnings:					
Information:					
13	Foreign Reference	JP2002182963.pdf	1830564 ba3505ecf26b7a82235a78639c93a511ad0a225b	no	11
Warnings:					
Information:					
14	Foreign Reference	WO02097620.pdf	4386391 2e7efc240438d78d41eee9a28be8d94c713b4ee5	no	37
Warnings:					
Information:					
15	Foreign Reference	FR2825565.pdf	1885392 dc10ff299ae4ceccc9ed4b6c7ca7ccdd052d12a8	no	16
Warnings:					
Information:					
16	Foreign Reference	GB2378780.pdf	7409705 8389ad8cdf469c7ddadbd5ba59136d04f9da1a45	no	68
Warnings:					
Information:					
17	Foreign Reference	GB2333865.pdf	4811680 30e56d8037c6c378c7d7dd3a78c146e8c80f1ad2	no	43
Warnings:					
Information:					
18	Foreign Reference	WO200358411.pdf	7821598 8b9981e1910c78cacc111544436516c9555e7e65	no	60
Warnings:					
Information:					
19	Foreign Reference	WO200363524.pdf	6817256 a956e50e4e72df862832d8e29ee839513360ec0a	no	64
Warnings:					
Information:					

20	Foreign Reference	EP1185923.pdf	1636671 f02c7a38735b7edc18dad63063e1ca6fda7174a8	no	37
Warnings:					
Information:					
21	Foreign Reference	EP1427166.pdf	3458186 d0fd1ea781b3900ef0567a5b98007dc4ef7891dd	no	28
Warnings:					
Information:					
22	Foreign Reference	EP1465041.pdf	769396 c6c4ce242e08a4b185f4078ea4b61738fe5462bc	no	23
Warnings:					
Information:					
23	Foreign Reference	EP0813132.pdf	659252 502fd54feb2d9e969ec99f6a26b5d7a538222a14	no	15
Warnings:					
Information:					
24	Foreign Reference	EP2429148A2.pdf	4116419 6da6cdba5f22d120c46fa04ee63d398f9ecd2b86	no	31
Warnings:					
Information:					
25	Non Patent Literature	Takaragi_EncryptionSystem_June1991.pdf	1582137 32bcb5deed05e70441bb3e7114b6f8656ddef921	no	11
Warnings:					
Information:					
26	Non Patent Literature	Bos_CSS_Extended_Tiling_May2004.pdf	275411 7ade5d802f52a5073a167602df409558e3b3d5b4	no	2
Warnings:					
Information:					
27	Non Patent Literature	W3C_CSS3_Module_Border_November2002.pdf	1676832 ba11d7e91e723ae1f751feb6ba4a360070dcbf4e	no	14
Warnings:					
Information:					
28	Non Patent Literature	W3C_CSS3_Backgrounds_and_Borders_February2005.pdf	3650676 699ba52acfdfd9525ee02be343b807974ea6dc64	no	30
Warnings:					
Information:					

29	Non Patent Literature	Samarati_Access_Control_Policies_Models.pdf	744005 94fc0a8a6656743271ea27461bc314962025aa86	no	60
Warnings:					
Information:					
30	Non Patent Literature	Microsoft_Managing_Authorization_Access_Control.pdf	1273802 216070d8efc88bdb21ec7b7080d4fb589f502b85	no	84
Warnings:					
Information:					
31	Non Patent Literature	Clarkson_Access_Control_Cornell.pdf	193334 1b11550fd9d785f767f83405663343d70a8567508	no	8
Warnings:					
Information:					
32	Non Patent Literature	Wikipedia_Access_Control_List.pdf	109025 9a22792282e69a767a667f71997488fed6c10883	no	3
Warnings:					
Information:					
33	Non Patent Literature	Wikipedia_Access_Control_List_June_2013.pdf	62623 e8685a303284f109cc9068f76cb5d5131991f239	no	1
Warnings:					
Information:					
34	Other Reference-Patent/App/Search documents	OA_27101_061107.pdf	1523397 21a60cc33e92df801bc91f638cb070e291c4d35d	no	13
Warnings:					
Information:					
35	Other Reference-Patent/App/Search documents	OA_27101_022208.pdf	1041674 9122e15872f0ebb822b3770ab013a89ec93b1609	no	11
Warnings:					
Information:					
36	Other Reference-Patent/App/Search documents	FOA_27101_092308.pdf	814764 a3c5fc53b7764389f3421d0e12fe4ba9c3c27d49	no	9
Warnings:					
Information:					
37	Other Reference-Patent/App/Search documents	AA_27101_120808.pdf	306664 33aaece73675905f16c0996f1e9488bb25b0da39	no	3
Warnings:					
Information:					

38	Other Reference-Patent/App/Search documents	OA_27101_040209.pdf	849350 be2f025e31c76d2a566e691df42d083cb7e94437	no	10
Warnings:					
Information:					
39	Other Reference-Patent/App/Search documents	FOA_27101_102809.pdf	831674 31af8fd2e5f39afd1dd205e01e1113fb1a02235f	no	10
Warnings:					
Information:					
40	Other Reference-Patent/App/Search documents	AA_27101_011910.pdf	325931 312e26da6c97355916e4cf64d519e97415e4b319	no	3
Warnings:					
Information:					
41	Other Reference-Patent/App/Search documents	NOA_27101_050410.pdf	819582 17ac04f2aaad2025bc9208ef12b9eefd9e0fedc1	no	7
Warnings:					
Information:					
42	Other Reference-Patent/App/Search documents	OA_27115_080311.pdf	2409189 46f269e6350680fc3e1e807361e95883c9ae2b55	no	20
Warnings:					
Information:					
43	Other Reference-Patent/App/Search documents	OA_27115_011912.pdf	1107560 6e1349b1583a3ae0e9ce3ac692f4a6e1e01c3a0a	no	11
Warnings:					
Information:					
44	Other Reference-Patent/App/Search documents	NOA_27115_061312.pdf	1065037 8f25fc4c5550fbcce7e8b94455a9705e04f1091b	no	9
Warnings:					
Information:					
45	Other Reference-Patent/App/Search documents	OA_27116_061713.pdf	684483 61684b11f061df650c1b65e29aa517f63730490d	no	19
Warnings:					
Information:					
46	Other Reference-Patent/App/Search documents	OA_082107.pdf	2284465 c690fc257b04b5ae6b99935f263b700a3d2cedfa	no	21
Warnings:					
Information:					

47	Other Reference-Patent/App/Search documents	FOA_27116_101013.pdf	535758 f951b57478171178451ae46e1378da36db40ea19	no	6
Warnings:					
Information:					
48	Other Reference-Patent/App/Search documents	OA_25701_052908.pdf	409060 42ff6ed7f666933013ef6d1537b58131a8c6b558	no	4
Warnings:					
Information:					
49	Other Reference-Patent/App/Search documents	OA_25701_031009.pdf	238013 8f671d744ba464b0974e5aed08c5276ddc4736ba	no	6
Warnings:					
Information:					
50	Other Reference-Patent/App/Search documents	OA_25701_082009.pdf	1681356 edf0b5871cf01efd61cae30e54b5f62a5d81c747	no	15
Warnings:					
Information:					
51	Other Reference-Patent/App/Search documents	NOA_25701_022310.pdf	322860 7bf6eb0e4cc46f325d7c32f4e196e44cd85fb943	no	7
Warnings:					
Information:					
52	Other Reference-Patent/App/Search documents	NOA_25701_061010.pdf	948410 0a181391ff50f35bc18b49b8928b77cb01b18e2b	no	8
Warnings:					
Information:					
53	Other Reference-Patent/App/Search documents	OA_25702_040708.pdf	446144 11820f3861bb26b672b7a65eb42c7b6ec14544f1	no	11
Warnings:					
Information:					
54	Other Reference-Patent/App/Search documents	FOA_25702_102708.pdf	390727 86429dbb0a9e236a4fe82677446f04106e9b2f81	no	10
Warnings:					
Information:					
55	Other Reference-Patent/App/Search documents	NOA_25702_020909.pdf	306445 cff7830d2e37571c409bfd79806fce07038063af	no	6
Warnings:					
Information:					

56	Other Reference-Patent/App/Search documents	OA_25703_032210.pdf	692417 433f4200f22be3da74a6b2243fc856f62e5d1053	no	12
Warnings:					
Information:					
57	Other Reference-Patent/App/Search documents	NOA_25703_081110.pdf	711720 2e7b1518e60c3bebf8ebd630e3f5895ac8d45987	no	12
Warnings:					
Information:					
58	Other Reference-Patent/App/Search documents	OA_25712_111213.pdf	1209373 cd12fd1a74e68e02be5e434c43684824f30d8cc9	no	32
Warnings:					
Information:					
59	Other Reference-Patent/App/Search documents	OA_25712_040314.pdf	313332 927409d67799fcac631093b7ce0f832783d74f93	no	9
Warnings:					
Information:					
60	Information Disclosure Statement (IDS) Form (SB08)	11913-US-CNT3_2015-06-04_IDS1_a.pdf	618090 12afff9941dddac4b675cda9bfc3aaaa83945d05	no	12
Warnings:					
Information:					
		Total Files Size (in bytes):	115130649		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 1/00 — A1

(11) International Publication Number: WO 00/60434

(43) International Publication Date: 12 October 2000 (12.10.00)

(21) International Application Number: PCT/US00/08868

(22) International Filing Date: 4 April 2000 (04.04.00)

(30) Priority Data:
60/127,968 6 April 1999 (06.04.99) US

(71) Applicant: MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, WA 98052-6399 (US).

(72) Inventors: HUAPAYA, Luis, M.; Apartment L201, 14430 N.E. 35th Street, Bellevue, WA 98007 (US). LEATHAM, Scott, R.; 13029 326th Avenue N.E., Duvall, WA 98019 (US). SWEDBERG, Gregory, D.; 500 Kalmia P1 NW, Issaquah, WA 98027 (US). VERES, James, E.; 14029 171st Lane N.E., Woodinville, WA 98072 (US).

(74) Agents: KOEHLER, Steven, M. et al.; Westman, Champlin & Kelly, P.A., International Centre, Suite 1600, 900 Second Avenue South, Minneapolis, MN 55402-3319 (US).

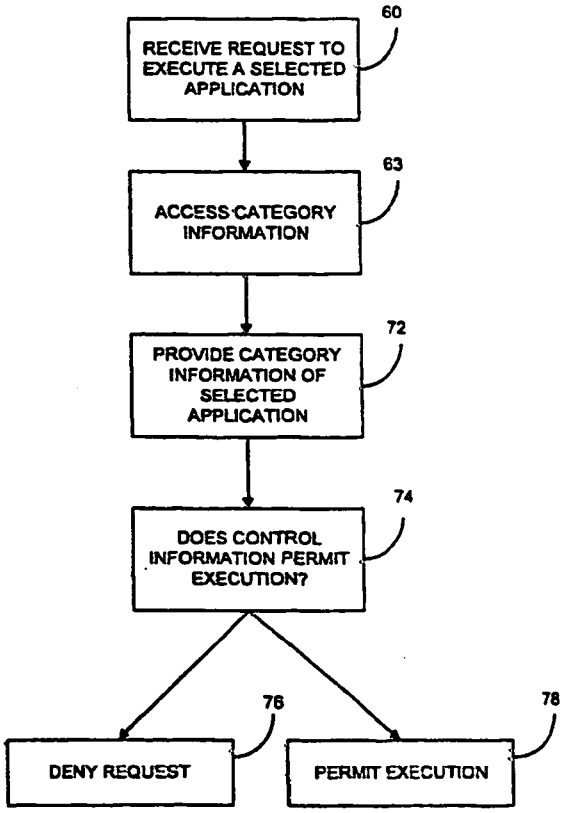
(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, 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
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: METHOD AND SYSTEM FOR CONTROLLING EXECUTION OF COMPUTER PROGRAMS

(57) Abstract

A computer implemented method includes receiving requests (52) related to executing a selected application; accessing a database (54) having control information related to a computer user's ability to execute a category of applications; and providing an output (56) indicating whether the selected application should be executed. Instructions or routines can be provided on a computer readable medium to implement the method.



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		

(54) Title: METHOD AND SYSTEM FOR CONTROLLING EXECUTION OF COMPUTER PROGRAMS

BACKGROUND OF THE INVENTION

The present invention relates to computers. More particularly, the present invention relates to controlling execution of applications on a computer.

Computers are now widely found in many homes. The home computer is typically a desktop personal computer, although with greater processing power and storage capabilities, notebook computers and even hand held devices may be ever more prevalent in the home. In whatever form the computer may take, the computer is a multi-purpose processing appliance that can be used by many to fulfill different needs. For instance, an adult can use the computer to maintain financial or other personal information, perform online investing, perform work related tasks, etc. A school age child may use the computer for completing homework assignments, research or as a form of entertainment. Even preschoolers are being taught to use a computer for educational games.

Although the computer can fulfill many needs for users of different ages, the applications stored or executed thereon are not typically appropriate for all users. In addition, the computer can be a tempting distraction to children, when other activities, such as studying is desired. It is usually impossible for a parent or other guardian to monitor what the computer is being used for throughout the day. In some households, the child may even have his or her own computer, for example, in his or her room. Although conveniently placed for quiet and effective studying, monitoring by the parent or guardian is highly impractical.

Nintendo Exhibit 1003
Part 2 of 2
Page 00855

Password protection has been used in the past to limit access to a particular application. In yet other systems, an application could be restricted to a particular time of use. However, since new applications can be installed easily on a computer, these types of systems will not prevent a child from executing other programs, which the parent or guardian may not believe is appropriate.

There thus is an ongoing need to provide computer administrators, such as parents, guardians and teachers, the ability to control applications that are executed on a computer. Such a system would allow children to use the computer at appropriate times of the day and for appropriate subject matter without imposing continuous monitoring upon the administrator.

SUMMARY OF THE INVENTION

A computer implemented method includes receiving requests related to executing a selected application; accessing a database having control information related to a computer user's ability to execute a category of applications; and providing an output indicating whether the selected application should be executed. Instructions or routines can be provided on a computer readable medium to implement the method.

A second aspect of the present invention includes a computer-implemented method that accesses a database having control information related to a computer user's ability to execute a category of applications. An output is provided indicating when an executing application should be at least hindered. Instructions or routines can be provided on a computer readable medium to implement this method.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of an exemplary computing environment for the present invention.

Fig. 2 is a block diagram illustrating aspects of the present invention.

Fig. 3 is a flow chart illustrating exemplary steps of the present invention.

DETAILED DESCRIPTION OF THE ILLUSTRATIVE EMBODIMENTS

Prior to a detailed discussion of the present invention, an overview of an exemplary operating environment may be helpful. FIG. 1 and the related discussion provide a brief, general description of a suitable computing environment in which the invention can be implemented. Although not required, the invention will be described, at least in part, in the general context of computer-executable instructions, such as program modules, being executed by a personal computer. Generally, program modules include routine programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Tasks performed by the programs and modules are described below and with the aid of block diagrams and flow charts. Those skilled in the art can implement the descriptions, block diagrams and flow charts as processor executable instructions, which can be written on any form of a computer readable medium. In addition, those skilled in the art will appreciate that the invention can be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention can also be

Nintendo Exhibit 1003
Part 2 of 2
Page 00857

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 can be located in both local and remote memory storage devices.

With reference to FIG. 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a conventional personal computer 10, including a processing unit 11, a system memory 12, and a system bus 13 that couples various system components including the system memory to the processing unit 11. The system bus 13 can 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) 14 and a random access memory (RAM) 15. A basic input/output system 16 (BIOS), containing the basic routine that helps to transfer information between elements within the personal computer 10, such as during start-up, is stored in ROM 14. The personal computer 10 further includes a hard disk drive 17 for reading from and writing to a hard disk (not shown), a magnetic disk drive 18 for reading from or writing to a removable magnetic disk 19, and an optical disk drive 20 for reading from or writing to a removable optical disk such as a CD ROM or other optical media. The hard disk drive 17, magnetic disk drive 18, and optical disk drive 20 are connected to the system bus 13 by a hard disk drive interface 22, magnetic disk drive interface 23, and an optical drive interface 24,

respectively. The drives and the associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for the personal computer 10.

Although the exemplary environment described herein employs the hard disk, the removable magnetic disk 19 and the removable optical disk 21, it should be appreciated by those skilled in the art that other types of computer readable media, which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories (RAMs), read only memory (ROM), and the like, can also be used in the exemplary operating environment.

A number of program modules can be stored on the hard disk, magnetic disk 19, optical disk 21, ROM 14 or RAM 15, including an operating system 65, one or more application programs 26, other program modules 67, and program data 28. A user can enter commands and information into the personal computer 10 through input devices such as a keyboard 30, a handwriting tablet 31, a pointing device 32 and a microphone 33. Other input devices (not shown) can include a joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 11 through a serial port interface 36 that is coupled to the system bus 13, but can be connected by other interfaces, such as a sound card, a parallel port, a game port or a universal serial bus (USB). A monitor 37 or other type of display device is also connected to the system bus 13 via an interface, such as a video

Nintendo Exhibit 1003
Part 2 of 2
Page 00859

adapter 38. In addition to the monitor 37, personal computers typically include other peripheral output devices such as a speaker 35 and a printer (not shown).

The personal computer 10 can operate in a networked environment using logic connections to one or more remote computers, such as a remote computer 39. The remote computer 39 can be another personal computer, a server, a router, a network PC, a peer device or other network node, and typically includes many or all of the elements described above relative to the personal computer 10, although only a memory storage device 40 has been illustrated in FIG. 1. The logic connections depicted in FIG. 1 include a local area network (LAN) 41 and a wide area network (WAN) 42. Such networking environments are commonplace in offices, enterprise-wide computer network Intranets and the Internet.

When used in a LAN networking environment, the personal computer 10 is connected to the local area network 41 through a network interface or adapter 43. When used in a WAN networking environment, the personal computer 10 typically includes a modem 44 or other means for establishing communications over the wide area network 42, such as the Internet. The modem 44, which can be internal or external, is connected to the system bus 13 via the serial port interface 36. In a network environment, program modules depicted relative to the personal computer 10, or portions thereof, can be stored in the remote memory storage devices. As appreciated by those skilled in the art, the network connections shown are exemplary and other means of establishing a

communications link between the computers can be used.

Fig. 2 schematically illustrates, in block diagram form, a parental control system or module 50 for controlling execution of applications on the computer 10. Generally, in a first aspect of the present invention, the parental control module 50 receives requests 52 related to executing a selected application. Parental control module 50 accesses a database 54 located on any of the storage devices described above. The database 54 includes control information related to a computer user's ability to execute a category of applications. For instance, control information can include at least one of a time of day and a duration of execution for each of the categories of applications. Parental control module 50 provides as an output 56 an indication as to whether the selected application forming the request 52 should be executed. The output 56 can be used by the operating system of the computer 10 to permit the application to be started. In another embodiment, the application receives the output 56 and allows further operation, if permitted.

In another aspect of the present invention, parental control module 50 monitors execution of a selected application and provides as output 56 an indication when the executing application should be terminated or at least hindered in order to encourage the computer user to stop execution.

Fig. 3 illustrates the general steps described above as well as other steps that can be performed in a specific embodiment. At step 60, parental control module 50 receives the request 52

Nintendo Exhibit 1003
Part 2 of 2
Page 00861

related to executing a selected application. Request 52 can originate from any one of a number of sources in various computer architectures. For instance, request 52 can originate from an application currently running on the computer 10, or from other computers communicating with computer 10 through a communications link such as a modem or a network. In one typical environment, request 52 originates from the operating system used to manage resources and data flow in the computer 10. In this example, the operating system provides information indicating which application has been selected for execution. Typically, an Application Programming Interface ("API") is used to pass the application's information between the operating system and parental control module 50. APIs are well known communication mediums for passing information between applications and/or the operating system. The information can be the name of the executable file for the selected application. In other embodiments, additional information can also be provided such as the location of the application in the computer 10 (e.g. the storage device and the folder or directory where the application is stored), or on a remote computer connected to computer 10.

In the embodiment illustrated in Fig. 2, parental control module 50 communicates with an application manager module 66. In this embodiment, application manager module 66 is used to manage information stored in a database 68 related to applications that can be executed on computer 10. It should be noted that application manager module 66 is illustrated separately; however, if desired tasks performed by application manager module 66 can be

incorporated into parental control module 50.

Database 68 stores information related to applications that can be executed on computer 10. Generally, the information stored in database 68 organizes or at least categorizes the applications. For example and without limiting the present invention, applications can be categorized as entertainment applications, educational applications, multimedia applications, reference applications, educational/entertainment applications, financial applications, etc. If desired, categories can be subdivided. For instance, entertainment applications (e.g. games, etc.) can be categorized based on content such as the level of violence depicted, or other criteria. For example, the criteria can include a rating level from a third party such as the Entertainment Software Rating Board. At step 63, application manager module 66 accesses database 68 via 69 to ascertain the category of the selected application forming request 52. Application manager module 66 provides as an output 70, the category of the selected application at step 72. Application manager module 66 can also provide a "null" value, or other selected value, if the selected application belongs to the "unknown" category.

At this point, it should be mentioned that applications generally can be classified as "legacy" and "non-legacy" applications. As used herein, "legacy" refers to applications that cannot or will not honor requests from parental control module 50 to stop execution. This aspect is discussed below. In contrast, "non-legacy" applications refer to applications that accept and honor requests to stop execution. Database 68 can store category information for both "legacy" and

Nintendo Exhibit 1003
Part 2 of 2
Page 00863

"non-legacy" applications. In one embodiment, when applications are stored on computer 10, the application is "registered" with application manager module 66, wherein category information is recorded in database 68. The category information can be automatically included in database 68 upon installation of the application on computer 10. The categories can be chosen by the software developer and read from the installation data. One or more categories may be considered relevant for each application. For instance, a software developer may consider a particular application to be both entertainment and entertainment/educational. If desired, application manager module 66 or parental control module 50 can include editing routines with suitable user interfaces to allow the parent or guardian to override the initial categorizations.

It should also be noted that database 68 can also include category information for applications that are not currently stored on, or registered with, computer 10. In this manner, if computer 10 is directed to execute an application, such as another legacy application that was unknown to the parent or guardian, parental control module 50 will obtain information about the application from database 68 for use in determining whether the application should be executed. Database 68 can be periodically updated with new information about new or old applications as desired. Updates to database 68 can be through any of the mass storage devices depicted in Fig. 1, or such information can be obtained remotely and downloaded, such as from the Internet.

Referring back to Fig. 3, at step 74,

parental control module 50 accesses database 54 and obtains control information as a function of the category information obtained from database 68. As mentioned above, control information in database 54 indicates whether or not a category of applications is executable. If the category of applications is executable, parental control module 50 may also obtain current time and date information 76 (Fig. 2) from suitable resources of computer 10 in order to ascertain if the selected application forming request 52 can be executed at the present time. If desired, database 54 can store information pertaining to the total time that computer 10 has been used to execute each category of applications. Parental control module 50 can then ascertain if any time remains for executing a specific category of applications. Step 74 generally denotes examination of the control information for the category information provided at step 72.

As appreciated by those skilled in the art, control parameters in the control information can vary from category to category, using various forms. For example, without limiting the present invention, some categories may be executable at any time of the day, while certain users may never execute other applications falling in other categories. Likewise, certain categories can be designated for execution only during certain times of the week or day. In addition to, or in the alternative, other parameters, such as duration of execution in a week or a day can also be set for each category of applications and used to control execution. Based on the control information for the category corresponding to request 52, the request to execute the selected application is denied at step

BNSDOCID: <WO_____0060434A1_I_>

Nintendo Exhibit 1003
Part 2 of 2
Page 00865

76, or permitted at step 78. It should be noted that if the request is denied, communication to the user can be performed using a number of techniques. If desired, parental control module 50 can display a suitable dialogue box on the monitor 37 stating that the request has been denied and the reason for denial. If the request was denied based on an inappropriate time to execute the selected application, a suggested time that the user can return to the computer for execution can also be displayed. If desired, any or all of this information can be provided to the application forming the request 42 wherein the application displays the information to the user.

Another aspect of the present invention includes monitoring execution of a selected application and providing an output, based on the control information stored in database 54, indicating that the executing application should be terminated or at least hindered in order to encourage the computer user to stop execution. As discussed above, applications can be classified into one of two categories "legacy" and "non-legacy". Non-legacy applications are designed to work with parental control module 50, and thus, will honor requests that the applications terminate execution. Parental control module 50 can provide the request to terminate execution using any of a number of techniques. Based on the current time, the duration of execution or other control parameters, parental control module 50 can monitor these parameters with respect to control information for the category of the application executing. When any one of the control parameters has been exceeded, parental control module 50 can provide a message request to the application to terminate

execution. The application receives the message request and informs the user that execution will be terminated with any relevant information saved for later use. In yet other non-legacy applications, parental control module 50 can assess the relevant control information and provide time information to the executing application indicating when the execution should be automatically terminated. This time information can be provided initially when permission is granted, or during application execution.

However, legacy applications may be more difficult to terminate in view that they have not been written to honor requests from parental control module 50. Nevertheless, parental control module 50 can take actions designed to at least hinder execution of the application, and thus, encourage the user to stop execution. In one embodiment, parental control module 50 can initiate temporary suspension of the application and display of a dialogue box requesting that the user terminate execution. If the user does not terminate execution within a selected time period, repeated display of the dialogue box and suspension of the executing application will frustrate the user and inhibit execution.

Another technique for hindering execution includes reducing the priority level of the application in a multi-processing environment where simultaneous applications can be executed. Using this technique, the legacy application runs slower in order to encourage the user to terminate the application.

If desired, the operating system can be instructed to terminate the executing application abruptly. In view that abrupt termination can result in

Nintendo Exhibit 1003
Part 2 of 2
Page 00867

loss of data, a dialogue box can be displayed warning the user that abrupt termination will occur in a selected time period, and thus in order to save relevant data, the application should be closed. In some situations, the computer user may not be in front of the monitor when the dialogue box warning of abrupt termination is displayed. Therefore, in order to prevent unnecessary loss of data, the time period before abrupt termination occurs begins with acknowledgement of the user to the warning such as by the user closing the dialogue box.

As discussed in the background section, it is common for a computer to be used by two or more individuals. In these circumstances, it is also common for some individuals to log into the computer and provide a password or other means to identify the individual to the computer. Database 54 can be organized to accommodate multiple users wherein control information related to execution of categories can be individually set for each user. In this manner, control information such as which applications are allowed to be executed, which types or rating levels of applications are allowed to be executed, when they can be executed and for how long each category of applications can be executed, is separately adjustable for each user. For example, a parent or guardian can be given full access to all applications, while a teenager is limited in content, duration and at what times during the day execution will be allowed.

Some users, such as young children, may not be capable of logging into the computer and providing the necessary password. In such a case, a set of control information can be used by parental control

module 50 when a particular individual has not been identified. This allows the computer to be used by the youngest child wherein a limited set of categories are allowed. Those individuals having the ability to log in would have a unique set of control information adjusted for their needs, presumably allowing more access to executable applications. Suitable user interfaces are provided to allow the parent or guardian the ability to individually adjust the set of control information for each user. For example, if a child has performed particularly well in school, additional time can be provided to execute games on the computer.

Although the present invention has been described with reference to preferred embodiments, workers 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.

Nintendo Exhibit 1003
Part 2 of 2
Page 00869

WHAT IS CLAIMED IS:

1. A computer implemented method comprising:
receiving requests related to executing a selected application;
accessing a database having control information related to a computer user's ability to execute a category of applications; and
providing an output indicating whether the selected application should be executed.

2. The computer implemented method of claim 1 wherein the control information comprises at least one of a time of day and duration of execution for each of the categories of applications.

3. The computer implemented method of claim 2 and further comprising receiving current time information.

4. The computer implemented method of claim 3 and further comprising receiving current date information.

5. The computer implemented method of claim 1 wherein the database includes control information for each of a plurality of users.

6. The computer implemented method of claim 5 and further comprising receiving user information capable of identifying a user of the computer, and wherein accessing the control information includes

accessing the control information as a function of the user information.

7. The computer implemented method of claim 1 and further comprising providing a second output indicating when an executing application should be at least hindered.

8. The computer implemented method of claim 7 wherein providing the second output comprises providing a message to the executing application requesting the application to terminate.

9. The computer implemented method of claim 7 wherein providing the second output includes initiating a notice to the user that the executing application will be terminated at a selected time.

10. The computer implemented method of claim 9 and further comprising receiving acknowledgement by the user of the notice and initiating a timer upon receipt of the acknowledgement.

11. The computer implemented method of claim 1 wherein the control information includes allowable rating levels of applications.

12. A computer readable medium having instructions readable by a processor that when implemented comprise:

receiving requests related to executing a selected application;

accessing a database having control

Nintendo Exhibit 1003
Part 2 of 2
Page 00871

information related to a computer user's ability to execute a category of applications; and

providing an output indicating whether the selected application should be executed.

13. The computer readable medium of claim 12 wherein the control information comprises at least one of a time of day and duration of execution for each of the categories of applications.

14. The computer readable medium of claim 13 and further comprising receiving current time information.

15. The computer readable medium of claim 14 and further comprising receiving current date information.

16. The computer readable medium of claim 12 wherein the database includes control information for each of a plurality of users.

17. The computer readable medium of claim 16 and further comprising receiving user information capable of identifying a user of the computer, and wherein accessing the control information includes accessing the control information as a function of the user information.

18. The computer readable medium of claim 12 and further comprising providing a second output indicating when an executing application should be at least hindered.

19. The computer readable medium of claim 18 wherein providing the second output comprises providing a message to the executing application requesting the application to terminate.

20. The computer readable medium of claim 18 wherein providing the second output includes initiating a notice to the user that the executing application will be terminated at a selected time.

21. The computer readable medium of claim 20 and further comprising receiving acknowledgement by the user of the notice and initiating a timer upon receipt of the acknowledgement.

22. The computer readable medium of claim 12 wherein the control information includes allowable rating levels of applications.

23. A computer readable medium having data and instructions comprising:

a database having control information related to a computer user's ability to execute a category of applications; and

a routine receiving requests related to executing a selected application, the routine accessing the database and, as a function of the control information, providing an output indicating whether the selected application should be executed.

24. The computer readable medium of claim 23 wherein the control information comprises at least one of a time of day and duration of execution for each of the categories of applications.

25. The computer readable medium of claim 24 wherein the routine receives current time information.

26. The computer readable medium of claim 25 wherein the routine receives current date information.

27. The computer readable medium of claim 23 wherein the database includes control information for each of a plurality of users.

28. The computer readable medium of claim 27 wherein the routine receives user information capable of identifying a user of the computer, the routine accessing the control information as a function of the user information.

29. The computer readable medium of claim 23 wherein the routine provides a second output indicating when an executing application should be at least hindered.

30. The computer readable medium of claim 29 wherein the second output comprises a message to the executing application requesting the application to terminate.

31. The computer readable medium of claim 29 wherein the second output initiates a notice to the

user that the executing application will be terminated within a prescribed time period.

32. The computer readable medium of claim 31 wherein the routine receives acknowledgement by the user of the notice, the routine initiating a timer upon receipt of the acknowledgement.

33. A computer implemented method comprising:
accessing a database having control information related to a computer user's ability to execute a category of applications; and
providing an output indicating when an executing application should be at least hindered.

34. The computer implemented method of claim 33 wherein the control information comprises at least one of a time of day and duration of execution for each of the categories of applications.

35. The computer implemented method of claim 33 wherein providing the output comprises providing a message to the executing application requesting the application to terminate.

36. The computer implemented method of claim 33 wherein providing the output includes initiating a notice to the user that the executing application will be terminated at a selected time.

37. The computer implemented method of claim 36

and further comprising receiving acknowledgement by the user of the notice and initiating a timer upon receipt of the acknowledgement.

38. A computer readable medium having instructions readable by a processor that when implemented comprise:

accessing a database having control information related to a computer user's ability to execute a category of applications; and

providing an output indicating when an executing application should be at least hindered.

39. The computer readable medium of claim 38 wherein the control information comprises at least one of a time of day and duration of execution for each of the categories of applications.

40. The computer readable medium of claim 38 wherein providing the output comprises providing a message to the executing application requesting the application to terminate.

41. The computer readable medium of claim 38 wherein providing the output includes initiating a notice to the user that the executing application will be terminated at a selected time.

42. The computer readable medium of claim 41 and further comprising receiving acknowledgement by the user of the notice and initiating a timer upon receipt

of the acknowledgement.

43. A computer readable medium having data and instructions comprising:

a database having control information related to a computer user's ability to execute a category of applications; and

a routine accessing the database and, as a function of the control information, providing an output indicating when an executing application should be at least hindered.

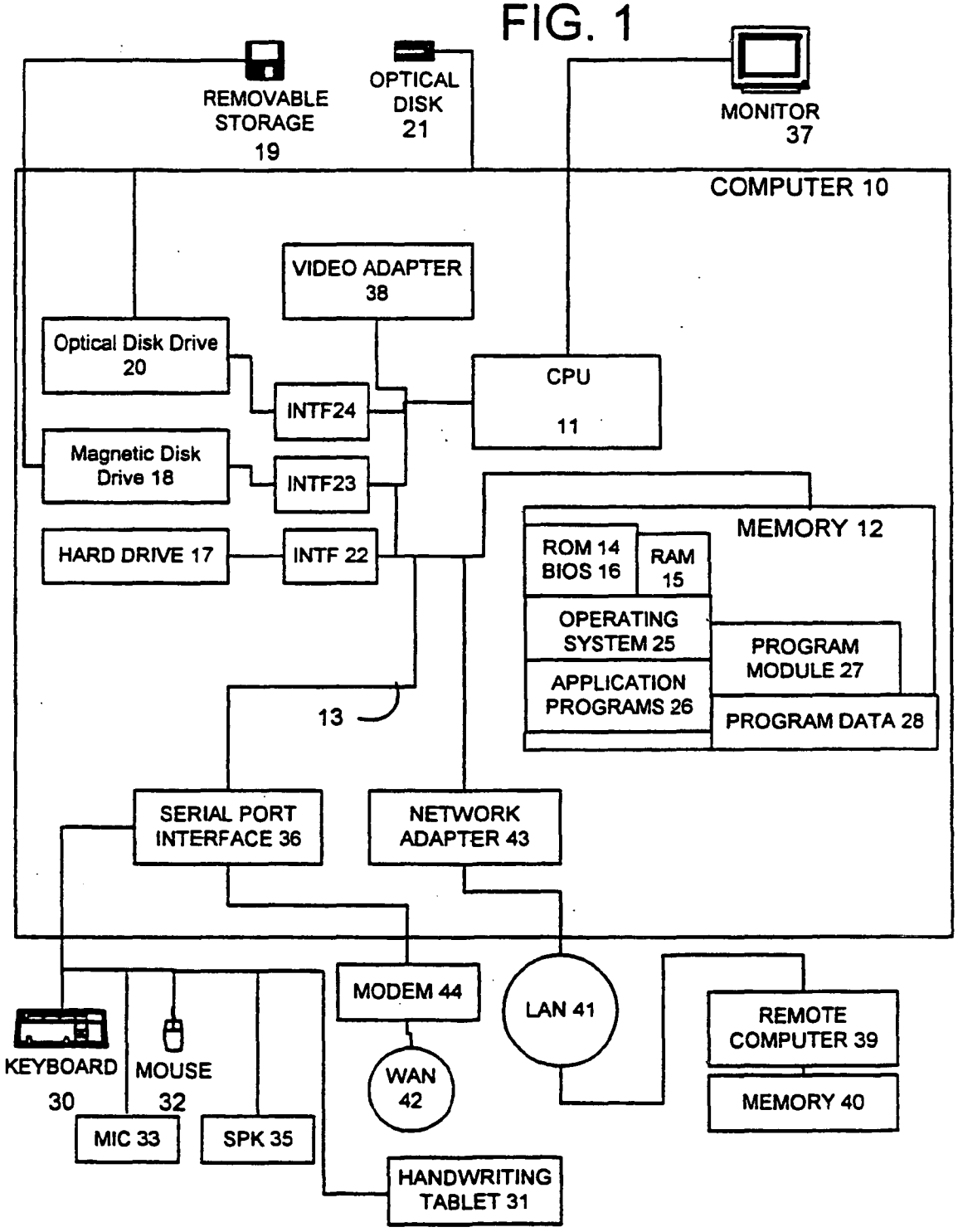
44. The computer readable medium of claim 43 wherein the control information comprises at least one of a time of day and duration of execution for each of the categories of applications.

45. The computer readable medium of claim 43 wherein the output comprises a message to the executing application requesting the application to terminate.

46. The computer readable medium of claim 43 wherein the output initiates a notice to the user that the executing application will be terminated at a selected time.

47. The computer readable medium of claim 46 wherein the routine receives acknowledgement by the user of the notice, the routine initiating a timer upon receipt of the acknowledgement.

FIG. 1



Nintendo Exhibit 1003
Part 2 of 2
Page 00878

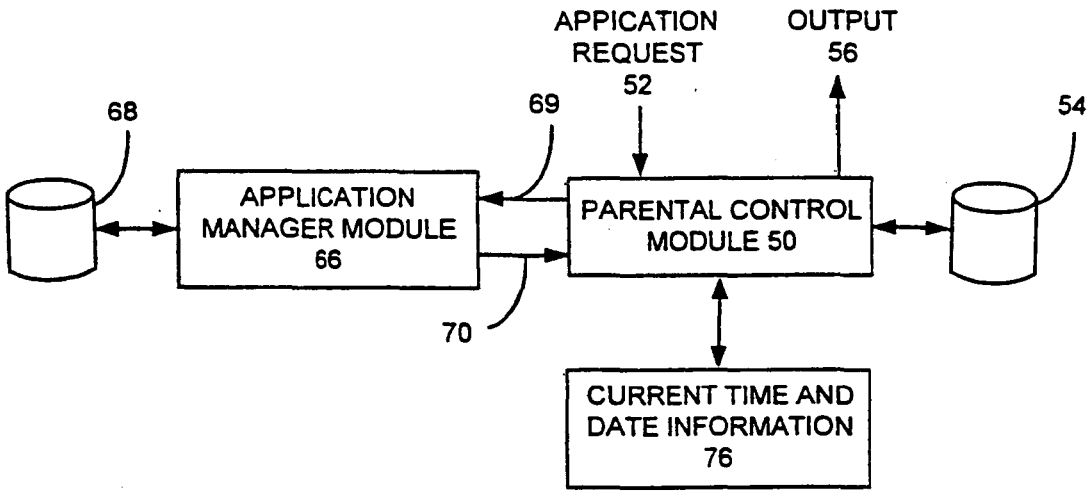


FIG. 2

BNSDOCID: <WO 0060434A1 I >

Nintendo Exhibit 1003
Part 2 of 2
Page 00879

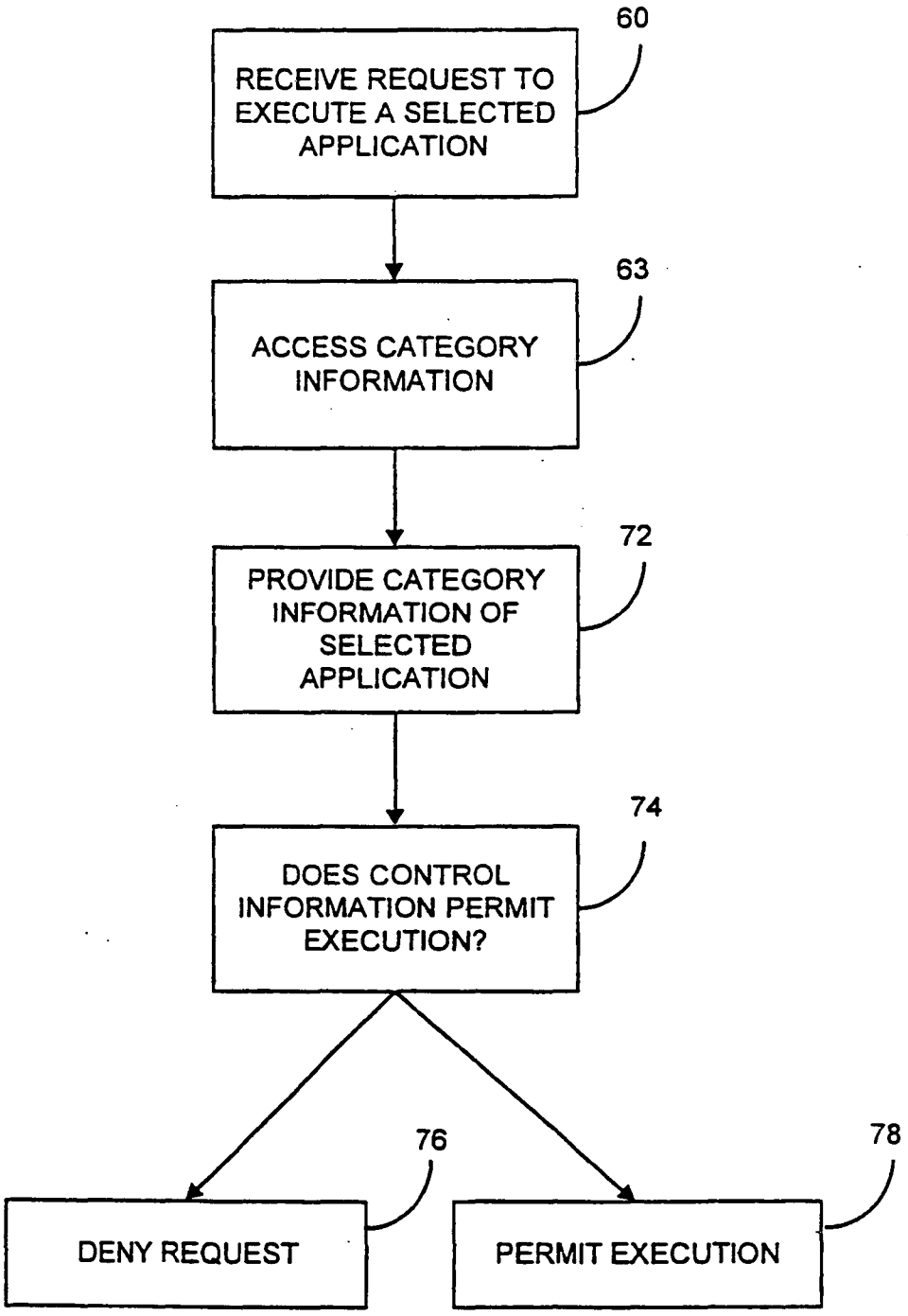


FIG. 3

Nintendo Exhibit 1003
Part 2 of 2
Page 00880

INTERNATIONAL SEARCH REPORT

Inter onal Application No
PCT/US 00/08868

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 5 809 230 A (PEREIRA J L A) 15 September 1998 (1998-09-15) abstract; figures 3-5 column 2, line 31 - line 41 column 3, line 59 -column 8, line 51 column 10, line 34 - line 47	1-6, 12-17, 23-28, 33,34, 38,39, 43,44
Y	-/—	7-10, 18-21, 29-32, 35-37, 40-42, 45-47

[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
31 July 2000	07/08/2000

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	Sigolo, A

1

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

BNSDOCID: <WO___ 0060434A1 I >

Nintendo Exhibit 1003
Part 2 of 2
Page 00881

INTERNATIONAL SEARCH REPORT

Inter ional Application No
PCT/US 00/08868

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 828 209 A (SAMSUNG ELECTRONICS CO LTD) 11 March 1998 (1998-03-11) abstract; figure 4 column 2, line 41 -column 3, line 23	1,11,12, 22,23, 33,38,43
Y	US 5 014 234 A (EDWARDS JR GORDON L) 7 May 1991 (1991-05-07) abstract; figures 3A,,3B,,6 column 1, line 44 -column 2, line 3 column 3, line 3 -column 5, line 48	7-10, 18-21, 29-32, 35-37, 40-42, 45-47

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

Nintendo Exhibit 1003
Part 2 of 2
Page 00882

INTERNATIONAL SEARCH REPORT

...formation on patent family members

Inter...nal Application No
PCT/US 00/08868

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5809230 A	15-09-1998	NONE	
EP 0828209 A	11-03-1998	JP 10133762 A US 5978920 A	22-05-1998 02-11-1999
US 5014234 A	07-05-1991	CA 1267443 A	03-04-1990

Form PCT/ISA/210 (patent family annex) (July 1992)

BNSDOCID: <WO_____0060434A1_I_>

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inter nal Application No
PCT/US 00/08868

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5809230 A	15-09-1998	NONE	
EP 0828209 A	11-03-1998	JP 10133762 A US 5978920 A	22-05-1998 02-11-1999
US 5014234 A	07-05-1991	CA 1267443 A	03-04-1990

Form PCT/ISA/210 (patent family annex) (July 1992)

Cited Ref 6

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-182983
(43)Date of publication of application : 28.06.2002

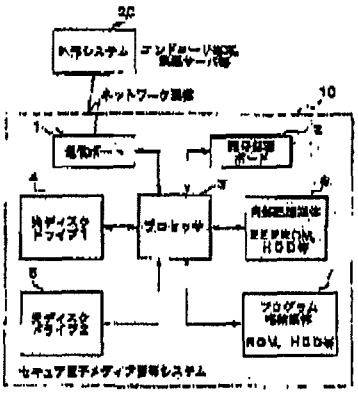
(51)Int.Cl. G06F 12/00
G06F 12/14
G09C 1/00

(21)Application number : 2000-385714
(22)Date of filing : 19.12.2000

(71)Applicant : RICOH CO LTD
(72)Inventor : KANAI YOICHI
YANAIDA MASUYOSHI

(54) SECURE ELECTRONIC MEDIUM MANAGEMENT METHOD

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a secure electronic medium management method for securing the truth of electronic data already recorded in an optical disk by another system.
SOLUTION: All the electronic data (files) recorded in a data medium mounted on an optical disk drive 1 (4) are read and a hash value is calculated. A hash list for which the pair of a file name and the hash value is one entry is prepared. An electronic signature is calculated in a cipher processing board 2 and imparted to the hash list, it is recorded in an internal storage medium 6 as a security file by the file name containing a data medium serial number and the data medium serial number is reported to the outside (request origin). A new optical disk (security medium) is mounted on an optical disk drive 2 (5) and the security file corresponding to the specified data medium serial number is moved to the security medium.



Nintendo Exhibit 1003
Part 2 of 2
Page 00885

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号
特開2002－182963
（P2002－182963A）
(43)公開日　平成14年６月28日(2002.6.28)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード(参考)
Ｇ０６Ｆ　12/00	５３７	Ｇ０６Ｆ　12/00	５３７Ｈ	５Ｂ０１７
12/14	３２０	12/14	３２０Ａ	５Ｂ０８２
Ｇ０９Ｃ　1/00	６４０	Ｇ０９Ｃ　1/00	６４０Ｂ	５Ｊ１０４

審査請求　未請求　請求項の数９　ＯＬ　（全 10 頁）

(21)出願番号　特願2000－385714(P2000－385714)

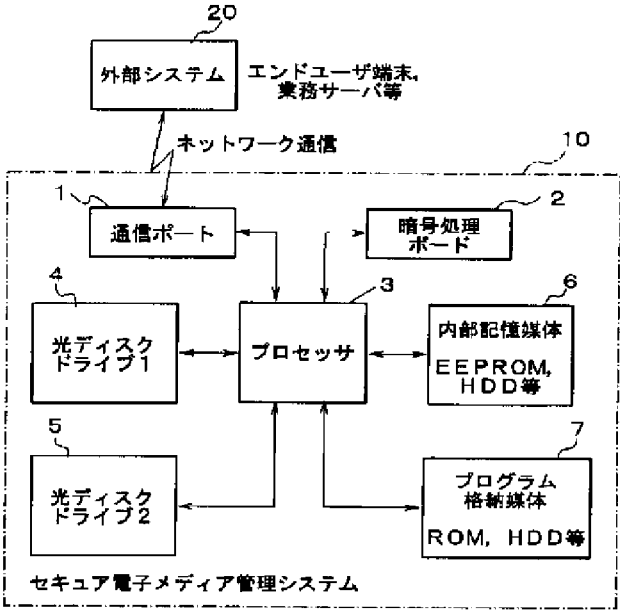
(22)出願日　平成12年12月19日(2000.12.19)

(71)出願人　000006747
株式会社リコー
東京都大田区中馬込１丁目３番６号
(72)発明者　金井　洋一
東京都大田区中馬込１丁目３番６号　株式会社リコー内
(72)発明者　谷内田　益義
東京都大田区中馬込１丁目３番６号　株式会社リコー内
(74)代理人　100079843
弁理士　高野　明近　（外２名）

最終頁に続く

(54)【発明の名称】　セキュア電子メディア管理方法

(57)【要約】
【課題】　既に他のシステムによって光ディスクに記録されている電子データの真正性を確保するセキュア電子メディア管理方法を提供する。
【解決手段】　光ディスクドライブ１（４）に装着されたデータメディアに記録されているすべての電子データ（ファイル）を読み出し、ハッシュ値を計算する。ファイル名とハッシュ値のペアを１エントリとするハッシュリストを作成する。ハッシュリストに対して暗号処理ボード２で電子署名を計算して付与し、セキュリティファイルとしてデータメディアシリアル番号を含むファイル名で内部記憶媒体６に記録し、データメディアシリアル番号を外部（要求元）に通知する。光ディスクドライブ２（５）に新しい光ディスク（セキュリティメディア）を装着し、指定されたデータメディアシリアル番号に対応するセキュリティファイルをセキュリティメディアに移動する。



Nintendo Exhibit 1003
Part 2 of 2
Page 00886

1

【特許請求の範囲】

【請求項１】　データメディアの真正性を維持するためのセキュリティメディアを作成する方法を有するセキュア電子メディア管理方法であって、データメディアに記録されているデータファイルを読み出し、該読み出したデータファイルに対してハッシュ値を計算し、前記データファイルのファイル名と計算したハッシュ値を合わせたハッシュエントリを１つまたは複数集めてハッシュリストを作成し、該作成したハッシュリストに対して当該セキュア電子メディア管理システム内部に保持している秘密鍵で電子署名を計算し、該計算した電子署名を前記ハッシュリストに付与してハッシュファイルとし、該作成したハッシュファイルをセキュリティファイルとしてセキュリティメディアに記録することにより、セキュリティメディアを作成することを特徴とするセキュア電子メディア管理方法。

【請求項２】　データメディアの真正性を維持するためにセキュリティメディアを更新する方法を有するセキュア電子メディア管理方法であって、セキュリティメディアからセキュリティファイルを読み出し、該読み出したセキュリティファイルに付与されている電子署名を検証し、該検証により前記電子署名が正当であると確認された場合に、前記セキュリティファイルに対して新しい暗号アルゴリズムで電子署名を計算し直し、該計算し直した電子署名を前記セキュリティファイルに付与し、該電子署名を付与し直したセキュリティファイルを前記セキュリティメディアに記録することにより、セキュリティメディアを更新することを特徴とするセキュア電子メディア管理方法。

【請求項３】　データメディアに記録されたデータファイルの真正性を検証する方法を有する、データメディアの真正性を維持するためのセキュア電子メディア管理方法であって、セキュリティメディアからデータメディアに対応するセキュリティファイルを読み出し、該読み出したセキュリティファイルに付与されている電子署名を検証し、該検証により前記電子署名が正当であると確認された場合に、該セキュリティファイルに記録されているファイル名及びハッシュ値（１）を取得し、該ファイル名に対応する前記データメディア内のデータファイルを読み出し、該読み出したデータファイルのハッシュ値（２）を計算し、前記ハッシュ値（１）とハッシュ値（２）を比較し、該比較においてハッシュ値が一致した場合に、前記データファイルが正当であると確認されたことを通知することにより、データメディアに記録されたデータファイルの真正性を検証することを特徴とするセキュア電子メディア管理方法。

【請求項４】　前記セキュリティファイルはシリアル番号を用いて表すことを特徴とする請求項１乃至３のいずれか１記載のセキュア電子メディア管理方法。

【請求項５】　前記セキュリティメディアはシリアル番

2

号を用いて表すことを特徴とする請求項１乃至４のいずれか１記載のセキュア電子メディア管理方法。

【請求項６】　前記電子署名の計算に使用する秘密鍵が物理的な耐タンパー性を持つデバイスに格納され、電子署名の計算が該デバイスの内部で行われることを特徴とする請求項１乃至５のいずれか１記載のセキュア電子メディア管理方法。

【請求項７】　請求項１乃至５のいずれか１記載のセキュア電子メディア管理方法における各工程を実行させるためのプログラム。

【請求項８】　請求項１乃至５のいずれか１記載のセキュア電子メディア管理方法における各工程を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項９】　請求項１乃至５のいずれか１記載のセキュア電子メディア管理方法における各工程を実行するための手段を備えたセキュア電子メディア管理装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、セキュア電子メディア管理方法に関し、より詳細には、光ディスクなどのオフラインメディアに記録された電子データを長期間に渡って電子的な証拠として扱えるよう、データメディアの真正性を維持するためのセキュリティメディアを管理するセキュア電子メディア管理方法、さらには該方法を実行させるためのプログラム、該プログラムを記録したコンピュータ読み取り可能な記録媒体、及び該方法を実行するための手段を備えたセキュア電子メディア管理装置に関する。

【０００２】

【従来の技術】情報システムが高度に発展する中、電子データのセキュリティについて関心が高まっている。セキュリティという言葉は広い意味で使用されるが、特に媒体に記録された電子データを証拠として扱えるよう真正性を確保する技術や、長期にデータを保存可能とする保存性確保の技術が電子政府の実現においても必要となってきている。電子データの長期保存の観点からすれば磁気ディスクよりも光ディスクが適していることは良く知られているため、様々な情報システムにおいて記録として残す電子データを最終的に光ディスクに記録するようにしたものが多くなってきている。

【０００３】電子データを証拠として記録する技術としては、特願平１１－０９０２１２号「原本性保証電子保存方法及び装置」、特願平１１－１４５３４０号「原本性保証電子保存方法およびその方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体」、特願平１１　１７３３７１号「原本性保証電子保存装置、原本性保証電子保存方法およびその方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体」、特願平１１－

Nintendo Exhibit 1003
Part 2 of 2
Page 00887

３２８８０２号「原本性保証電子保存方法、装置及びコンピュータ読み取り可能な記録媒体」、特開平１０－２８３２６２号公報「ファイルシステムおよびプログラム記憶媒体」、小尾他：原本性保証電子保存システムの開発―基本機能の実現―,Medical Imaging Technology,Vol.１６,No.４,Proceedings of JAMIT Annual Meeting'９８（１９９８）、金井他：原本性保証電子保存システムの開発―システムの構築―,Medical Imaging Technology,Vol.１６,No.４,Proceedings of JAMIT Annual Meeting'９８（１９９８）、国分他：原本性保証電子保存システムの開発,（特）情報処理振興事業協会発行　創造的ソフトウェア育成事業及びエレクトロニック・コマース推進事業　最終成果発表会論文集　創造的ソフトウェア育成事業編（１９９８）、金井：原本性保証電子保存システムについて,Vol.３４,No.８,行政＆ＡＤＰ（１９９８）、がある。

【０００４】上述した技術は、証拠として残す電子データを外部から受け取り、その電子データに対して改ざん検知コードを計算する等の処理を行い、その改ざん検知コードを、受け取った電子データとともに光ディスクなどに記録するというものである。

【０００５】

【発明が解決しようとする課題】これらの保存システムの場合、既に光ディスクに記録されている電子データを証拠として残すためには、一度光ディスクから電子データを読み出し、その電子データを保存システムに送り込み、保存システムが改ざん検知コードとともに電子データを新たな光ディスクに記録するという方法をとらざるを得なかった。この場合、既に光ディスクに電子データが記録されていたにもかかわらず、それを読み出してから再度光ディスクに記録するという無駄な処理を行わなければならなかった。

【０００６】また、長期に渡って電子データを保存しておくことを想定した場合、上記の保存システムによって光ディスクに記録された改ざん検知コードは、周囲のシステムの計算能力の向上に伴って相対的に強度が低下するという問題があった。もちろん、そのような場合でも改ざん検知コードを強度の高いものにつけかえれば良いが、光ディスク一枚一枚の電子データ一つ一つに対して改ざん検知コードを計算し直すのは現実的ではない。

【０００７】上述したように、従来技術では、既に他のシステムによって光ディスクに記録されている電子データの真正性を確保する際に無駄が生じるという問題や、長期に渡る安全性を確保するのに手間がかかるという問題があった。

【０００８】本発明は上述のごとき実情に鑑みてなされたものであり、上記問題に対して、既に他のシステムによって光ディスクに記録されている電子データの真正性を確保することが可能なセキュア電子メディア管理方法を提供することをその目的とする。また、本発明は、光ディスク等のオフラインメディアに記録された電子データを長期に渡って安全性を確保しながら電子的な証拠として扱えるようにするための、手間のかからないセキュア電子メディア管理方法を提供することを他の目的とする。

【０００９】

【課題を解決するための手段】請求項１の発明は、データメディアの真正性を維持するためのセキュリティメディアを作成する方法を有するセキュア電子メディア管理方法であって、データメディアに記録されているデータファイルを読み出し、該読み出したデータファイルに対してハッシュ値を計算し、前記データファイルのファイル名と計算したハッシュ値を合わせたハッシュエントリを１つまたは複数集めてハッシュリストを作成し、該作成したハッシュリストに対して当該セキュア電子メディア管理システム内部に保持している秘密鍵で電子署名を計算し、該計算した電子署名を前記ハッシュリストに付与してハッシュファイルとし、該作成したハッシュファイルをセキュリティファイルとしてセキュリティメディアに記録することにより、セキュリティメディアを作成することを特徴としたものである。

【００１０】請求項２の発明は、データメディアの真正性を維持するためにセキュリティメディアを更新する方法を有するセキュア電子メディア管理方法であって、セキュリティメディアからセキュリティファイルを読み出し、該読み出したセキュリティファイルに付与されている電子署名を検証し、該検証により前記電子署名が正当であると確認された場合に、前記セキュリティファイルに対して新しい暗号アルゴリズムで電子署名を計算し直し、該計算し直した電子署名を前記セキュリティファイルに付与し、該電子署名を付与し直したセキュリティファイルを前記セキュリティメディアに記録することにより、セキュリティメディアを更新することを特徴としたものである。

【００１１】請求項３の発明は、データメディアに記録されたデータファイルの真正性を検証する方法を有する、データメディアの真正性を維持するためのセキュア電子メディア管理方法であって、セキュリティメディアからデータメディアに対応するセキュリティファイルを読み出し、該読み出したセキュリティファイルに付与されている電子署名を検証し、該検証により前記電子署名が正当であると確認された場合に、該セキュリティファイルに記録されているファイル名及びハッシュ値（１）を取得し、該ファイル名に対応する前記データメディア内のデータファイルを読み出し、該読み出したデータファイルのハッシュ値（２）を計算し、前記ハッシュ値（１）とハッシュ値（２）を比較し、該比較においてハッシュ値が一致した場合に、前記データファイルが正当であると確認されたことを通知することにより、データメディアに記録されたデータファイルの真正性を検証す

Nintendo Exhibit 1003
Part 2 of 2
Page 00888

ることを特徴としたものである。
【００１２】請求項４の発明は、請求項１乃至３のいずれか１の発明において、前記セキュリティファイルはシリアル番号を用いて表すことを特徴としたものである。
【００１３】請求項５の発明は、請求項１乃至４のいずれか１の発明において、前記セキュリティメディアはシリアル番号を用いて表すことを特徴としたものである。
【００１４】請求項６の発明は、請求項１乃至５のいずれか１の発明において、前記電子署名の計算に使用する秘密鍵が物理的な耐タンパー性を持つデバイスに格納され、電子署名の計算が該デバイスの内部で行われることを特徴としたものである。
【００１５】請求項７の発明は、請求項１乃至５のいずれか１記載のセキュア電子メディア管理方法における各工程を実行させるためのプログラムである。
【００１６】請求項８の発明は、請求項１乃至５のいずれか１記載のセキュア電子メディア管理方法における各工程を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体である。
【００１７】請求項９の発明は、請求項１乃至５のいずれか１記載のセキュア電子メディア管理方法における各工程を実行するための手段を備えたセキュア電子メディア管理装置である。
【００１８】
【発明の実施の形態】図１は、本発明の一実施形態に係るセキュア電子メディア管理方法を実行させるためのセキュア電子メディア管理システムの構成を示す図で、図中、１０はセキュア電子メディア管理システム（本発明のセキュア電子管理装置の一実施形態でもある。以下、本システムともいう。）、２０は本システム１０とネットワーク（一般的な通信路で構わない）で接続されたエンドユーザ端末，業務サーバ装置等の外部システムである。セキュア電子メディア管理システム１０は、通信ネットワークを介して外部システム２０との通信を行うためのインターフェース部である通信ポート１と、セキュリティファイルの電子署名を検証する暗号処理ボード２と、持ち運び可能なデータ記憶媒体を駆動させる駆動装置の一例としての光ディスクドライブ１（４），光ディスクドライブ２（５）と、ＥＥＰＲＯＭ，ＨＤＤ等の内部記憶媒体６と、ＲＯＭ，ＨＤＤ等のプログラム格納媒体７とを具備する。
【００１９】通信ポート１は、後述するデータメディア登録処理，セキュリティメディア作成処理，セキュリティメディア更新処理，データファイル検証処理等の指示を外部システム２０から受信する。
【００２０】プログラム格納媒体７は、主制御プログラム、本発明に係る電子メディア管理プログラムなどの各種プログラムを格納したメモリであり、たとえば書換可能なＥＥＰＲＯＭや読み出し専用のＲＯＭなどからなる。プロセッサ３は、プログラム格納媒体７に格納された各種プログラムを読み出して実行する制御装置である。内部記憶媒体６は、各種プログラムの実行に必要となるパラメータを記憶し、さらにセキュリティファイル及び後述するシリアル番号等を記憶するＥＥＰＲＯＭなどからなるメモリである。
【００２１】暗号処理ボード２としては、三菱電機のＭＩＳＴＹＫＥＹＰＥＲ等が例として挙げられる。基本的に、ボード内部で公開鍵暗号方式の鍵ペアを生成することができ、その鍵ペアをボード内部で保持することができ、その鍵ペアに対して外部で発行された公開鍵認証書をボード内部で保持することができ、その鍵を使用してボード内部で暗号処理を実行することができ、さらに生成した秘密鍵は外部に読み出すことができず、ボード自身が物理的な耐タンパー性を持っているようなものを想定している。
【００２２】本実施形態のセキュア電子メディア管理システム１０では、暗号処理を行う暗号処理ボード２を組み込み、暗号処理ボード２で鍵生成，鍵保管，暗号化，復号化の処理を行っているが、プログラム格納媒体７に鍵生成プログラム，暗号化プログラム，復号化プログラムなどを格納し、プロセッサ３によってそれらプログラムを実行する方式としてもよい。その際、鍵生成プログラムにより生成された、或いは外部で生成した公開鍵ペアの秘密鍵を暗号処理ボード２のようなハードウェアを使用して、ハードウェア内部に安全に鍵を保管することが好ましい。
【００２３】本システムで外部からの要求を受けて様々な処理を行う前に、既に暗号処理ボード２は内部で公開鍵暗号方式の鍵ペアを生成してあり、生成した鍵ペアのうちの公開鍵に対して外部で発行された公開鍵認証書を内部に保持してあることを想定している。さらに、生成した鍵ペアのうち、秘密鍵を使用して電子署名を生成することができ、公開鍵を使用して電子署名を検証できる状態となっていることを想定している。
【００２４】図２は、本発明の一実施形態に係るセキュア電子メディア管理方法におけるセキュリティメディア作成処理を説明するためのフロー図、図３及び図４は、図２のセキュリティメディア作成処理を詳細に説明するための図である。セキュリティメディアを作成する際には、まず、その基となるデータメディアを本システムに登録し、その後、登録されたセキュリティファイルをセキュリティメディアに記録するようにする。その詳細を以下に説明する。
【００２５】まず、データメディア１１に記録されているデータファイル１２を読み出し（ステップＳ１）、データファイル１２に対してハッシュ値１３を計算する（ステップＳ２）。次に、データファイル１２のファイル名と計算したハッシュ値１３を合わせたハッシュエントリを１つ又は複数集めてハッシュリスト１４を作成し（ステップＳ３）、そのハッシュリスト１４に対して本

Nintendo Exhibit 1003
Part 2 of 2
Page 00889

システム内部に保持している秘密鍵で電子署名を計算する（ステップＳ４）。最後に電子署名をハッシュリスト１４に付与してハッシュファイルとし（ステップＳ５）、ハッシュファイルをセキュリティファイル１５としてセキュリティメディア１６に記録する（ステップＳ６）。なお、セキュリティファイル１５やセキュリティメディア１６にそれぞれシリアル番号を付与して表すようにしてもよい。

【００２６】データメディア登録処理について詳細に説明する。光ディスクドライブ１（４）に光ディスク（データメディア１１）を装着しておく。外部からデータメディア登録処理の要求を受け取ると、まず、データメディア１１に記録されているすべての電子データ（ファイル）１２を読み出す。次に読み出したファイル１２に対してそれぞれハッシュ値１３を計算する。読み出したファイル１２のファイル名と、計算したハッシュ値１３のペアを１エントリとするリスト（ハッシュリスト）１４を作成する。次に、内部記憶媒体６からデータメディアシリアル番号を読み出し、データメディアシリアル番号を１増加させて内部記憶媒体６に記録する。ハッシュリスト１４に対して暗号処理ボード２で電子署名を計算して付与し、セキュリティファイル１５としてデータメディアシリアル番号を含むファイル名で内部記憶媒体６に記録する。その後、データメディアシリアル番号を外部（要求元）に通知する。なお、要求元は、通知されたデータメディアシリアル番号をデータメディア１１に電子データとして記録するか、メディア１１の表面にラベルとして印刷しておくか、外部データベースで管理するか等して、データメディア１１とデータメディアシリアル番号の関係付けを行っておく。

【００２７】内部記憶媒体６に記録されたセキュリティファイル１５をセキュリティメディアに記録する処理について詳細に説明する。光ディスクドライブ２（５）に新しい光ディスク（セキュリティメディア）１６を装着しておく。まず、外部からセキュリティメディア作成要求と、データメディアシリアル番号のリストを受け取る。次に、指定されたデータメディアシリアル番号に対応するセキュリティファイル１５をセキュリティメディア１６に移動する。内部記憶媒体６からセキュリティメディアシリアル番号を読み出し、セキュリティメディアシリアル番号を１増加させて内部記憶媒体６に記録する。セキュリティメディア１６にセキュリティメディアシリアル番号を記録し、セキュリティメディアシリアル番号を外部（要求元）に通知する。

【００２８】図５は、本発明の一実施形態に係るセキュア電子メディア管理方法におけるセキュリティメディア更新処理を説明するためのフロー図である。まず、セキュリティメディアからセキュリティファイルを読み出し（ステップＳ１１）、そのセキュリティファイルに付与されている電子署名を検証する（ステップＳ１２）。ここで、電子署名が正当であるかの判断により（ステップＳ１３）、電子署名が正当であると確認された場合に、そのセキュリティファイルに対して新しい暗号アルゴリズムで電子署名を再計算し（ステップＳ１４）、再計算した電子署名をそのセキュリティファイルに付与する（ステップＳ１５）。最後に、電子署名を付与し直したセキュリティファイルをセキュリティメディアに記録して（ステップＳ１６）、処理を終了する。なお、本実施形態においてもセキュリティファイル１５やセキュリティメディア１６にそれぞれシリアル番号を付与して表すようにしてもよい。

【００２９】セキュリティメディア更新処理についてさらに詳細に説明する。長期に渡って真正性を確保するために、セキュリティメディアの更新を行うことがある。セキュリティメディアの更新処理を行うには、まず、光ディスクドライブ１（４）に更新するセキュリティメディアを、光ディスクドライブ２（５）に新しい光ディスクを、それぞれ装着しておく必要がある。この状態で、更新するセキュリティメディアからすべてのセキュリティファイルを読み出し、暗号処理ボード２を使用して各セキュリティファイルの電子署名を検証する。検証に成功したセキュリティファイルのみ、暗号処理ボード２で電子署名を計算し直してセキュリティファイルに付与する。新しい光ディスクに、電子署名を付け直したセキュリティファイルを記録する。さらに、内部記憶媒体６からセキュリティメディアシリアル番号を読み出し、セキュリティメディアシリアル番号を１増加させて内部記憶媒体６に記録し、セキュリティメディアシリアル番号を新しい光ディスクに記録し、新しいセキュリティメディアとする。

【００３０】この更新処理を行うのは、暗号アルゴリズムとして新しい強度の高いものに変更する場合が主になるため、この処理を行う前に、新しい暗号アルゴリズムに対応するため、新しい暗号処理ボードが本システムに装着されていることを想定している。新しい暗号処理ボードには、古い暗号処理ボードで使用していた秘密鍵に対応する公開鍵証明書を記録しておくことになる。

【００３１】この方法の場合、セキュリティファイルに含まれているハッシュ値の更新がなされないが、電子署名を計算するのに用いた秘密鍵を解読するのに比較して、同じハッシュ値を持つ「意味のある」電子データを作成することの方が遥かに困難であるため構わない。

【００３２】図６は、本発明の一実施形態に係るセキュア電子メディア管理方法におけるデータファイル検証処理を説明するためのフロー図である。まず、セキュリティメディアからデータメディアに対応するセキュリティファイルを読み出し（ステップＳ２１）、そのセキュリティファイルに付与されている電子署名を検証する（ステップＳ２２）。ここで、電子署名が正当であるかの判断により（ステップＳ２３）、電子署名が正当であると

Nintendo Exhibit 1003
Part 2 of 2
Page 00890

確認された場合に、そのセキュリティファイルに記録されているファイル名及びハッシュ値１を取得し（ステップＳ２４）、そのファイル名に対応するデータメディア内のデータファイルを読み出し（ステップＳ２５）、そのデータファイルのハッシュ値２を計算する（ステップＳ２６）。次に、ハッシュ値１とハッシュ値２を比較し（ステップＳ２７）、ハッシュ値が一致した場合（ステップＳ２８においてＹＥＳ）に、データファイルが正当であると確認されたことを通知（ステップＳ２９）し、処理を終了する。また、ステップＳ２３，Ｓ２８においてＮＯであった場合に、データファイルが正当であると確認されなかった旨を通知し（ステップＳ３０）、処理を終了する。なお、本実施形態においてもセキュリティファイル１５やセキュリティメディア１６にそれぞれシリアル番号を付与して表すようにしてもよい。

【００３３】データ検証処理についてさらに詳細に説明する。作成されたセキュリティメディアは本システムで利用することも可能であるが、外部で利用することも可能である。セキュリティメディアに記録されるセキュリティファイルの仕様を公開することにより、基本的には公開鍵暗号方式を使用して外部でもデータメディア内のデータの真正性を検証することができる。ここでは本システムでデータメディアの真正性検証を行う場合の処理を説明する。基本的には外部においても処理することは同じである。

【００３４】データ検証処理を行うには、まず、光ディスクドライブ１（４）に検証するデータメディアを、光ディスクドライブ２（５）にデータメディアに対応するセキュリティメディアを、それぞれ装着しておく必要がある。まず、外部からデータ検証要求とともにファイル名，データメディアシリアル番号を受け取る。セキュリティメディアから、データメディアシリアル番号に対応するセキュリティファイルを読み出し、セキュリティファイルに付与されている電子署名を暗号処理ボード２により検証する。検証に成功した場合、そのセキュリティファイルに記録されている、外部から指定されたファイル名に対応するハッシュ値を取り出す。逆に検証に失敗した場合には外部（要求元）にエラーを通知する。データメディアから、指定されたファイル名のファイルを読み出し、読み出したファイルのハッシュ値を計算する。次に、先に取り出したハッシュ値と計算したハッシュ値を比較し、ハッシュ値が一致すれば真正性の検証に成功したことを要求元に通知する。なお、ハッシュ値が異なれば、エラーを要求元に通知する。

【００３５】電子署名の計算方法、付与方法、検証方法はＰＫＣＳ　＃７（ＰｕｂｌｉｃＫｅｙ　Ｃｒｙｐｔｏｇｒａｐｈｙ　Ｓｔａｎｄａｒｄ）のような標準に従って行えばよい。また、電子署名データには上位の認証書がすべて記録されることを想定している。

【００３６】また、前述の説明で分かるように、光ディスクドライブ１（４）はデータメディアやセキュリティメディアの読み出しにしか使用していない。したがって、本システムに直接接続されている必然性はなく、例えばネットワークを介して他のシステム（例えばメディアジュークボックス等）に装着されているデータメディアからデータを取り出すようにしても構わない。その場合、ネットワークの向こうにあるデータメディアが光ディスクである必要もなく、ハードディスクであっても構わない。また、もちろんメディアの差し替えを行えば、光ディスクドライブ１が光ディスクドライブ２の役割も果たすようなシステム構成でも構わない。

【００３７】以上、本発明のセキュア電子メディア管理方法におけるセキュリティメディアの作成、更新、及び検証処理を中心に各実施形態を説明してきたが、本発明は、上記方法の各処理を実行するための手段を構成要素とするセキュア電子メディア管理装置しても実現可能である。また、本発明は、この各処理に係る各手段を実行可能に備えたセキュア電子メディア管理装置と同様に、コンピュータにセキュア電子メディア管理方法を実行させるためのプログラム、及び該プログラムを記録したコンピュータ読取り可能な記録媒体としての形態も可能である。

【００３８】本発明によるセキュア電子メディア管理の各処理を実現するためのプログラムやデータを記憶した記録媒体の実施形態を説明する。記録媒体としては、具体的には、ＣＤ－ＲＯＭ、光磁気ディスク、ＤＶＤ－ＲＯＭ、フロッピー（登録商標）ディスク、フラッシュメモリ、及びその他各種ＲＯＭやＲＡＭ等が想定でき、これら記録媒体に上述した本発明の各実施形態に係る各処理をコンピュータに実行させ、セキュア電子メディア管理の機能を実現するためのプログラムを記録して流通させることにより、当該機能の実現を容易にする。そしてコンピュータ等の情報処理装置に上記のごとくの記録媒体を装着して情報処理装置によりプログラムを読み出すか、若しくは情報処理装置が備えている記憶媒体に当該プログラムを記憶させておき、必要に応じて読み出すことにより、本発明に係わるセキュア電子メディア管理機能を実行することができる。

【００３９】

【発明の効果】本発明によれば、既に他のシステムによって光ディスクに記録されている電子データの真正性を確保する際に無駄なく実行可能となる。さらに、本発明によれば、手間をかけることなく長期に渡る安全性を確保することが可能である。

【図面の簡単な説明】

【図１】　本発明の一実施形態に係るセキュア電子メディア管理方法を実行させるためのセキュア電子メディア管理システムの構成を示す図である。

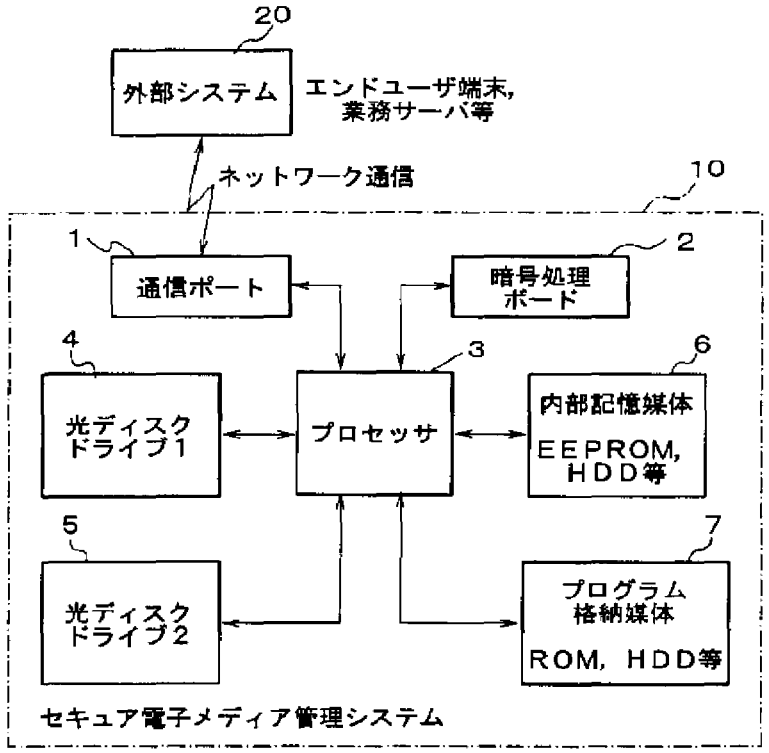
【図２】　本発明の一実施形態に係るセキュア電子メディア管理方法におけるセキュリティメディア作成処理を

Nintendo Exhibit 1003
Part 2 of 2
Page 00891

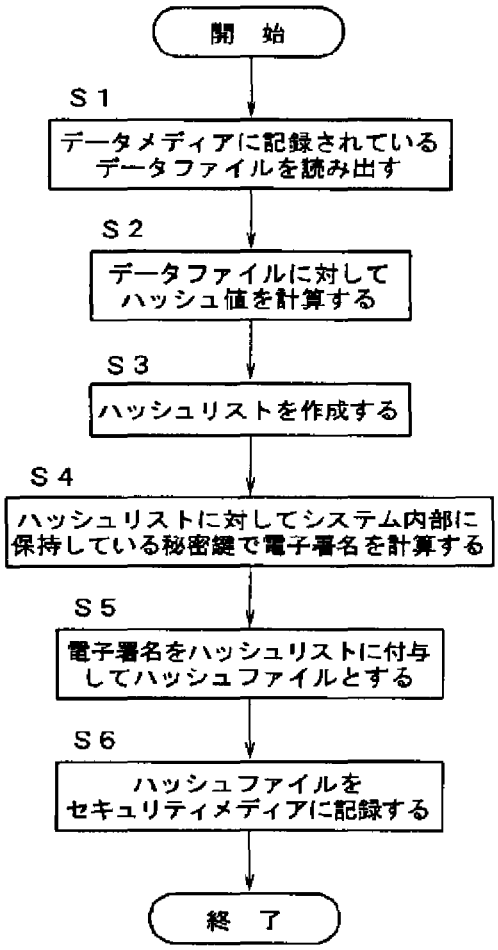
説明するためのフロー図である。
【図３】　図２のセキュリティメディア作成処理を詳細に説明するための図である。
【図４】　図２のセキュリティメディア作成処理を詳細に説明するための図である。
【図５】　本発明の一実施形態に係るセキュア電子メディア管理方法におけるセキュリティメディア更新処理を説明するためのフロー図である。
【図６】　本発明の一実施形態に係るセキュア電子メディア管理方法におけるデータファイル検証処理を説明す＊るためのフロー図である。

【符号の説明】
１…通信ポート、２…暗号処理ボード、３…プロセッサ、４…光ディスクドライブ１、５…光ディスクドライブ２、６…内部記憶媒体、７…プログラム格納媒体、１０…セキュア電子メディア管理システム、１１…データメディア、１２…データファイル、１３…ハッシュ値、１４…ハッシュリスト、１５…セキュリティファイル、１６…セキュリティメディア、２０…外部システム。

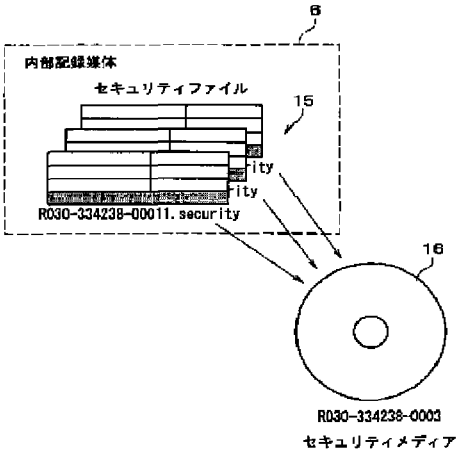
【図１】



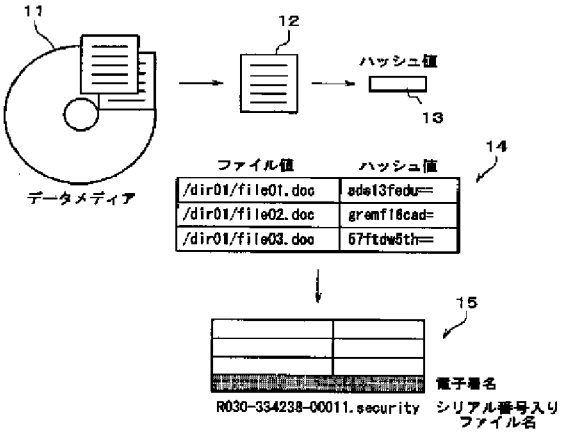
【図２】



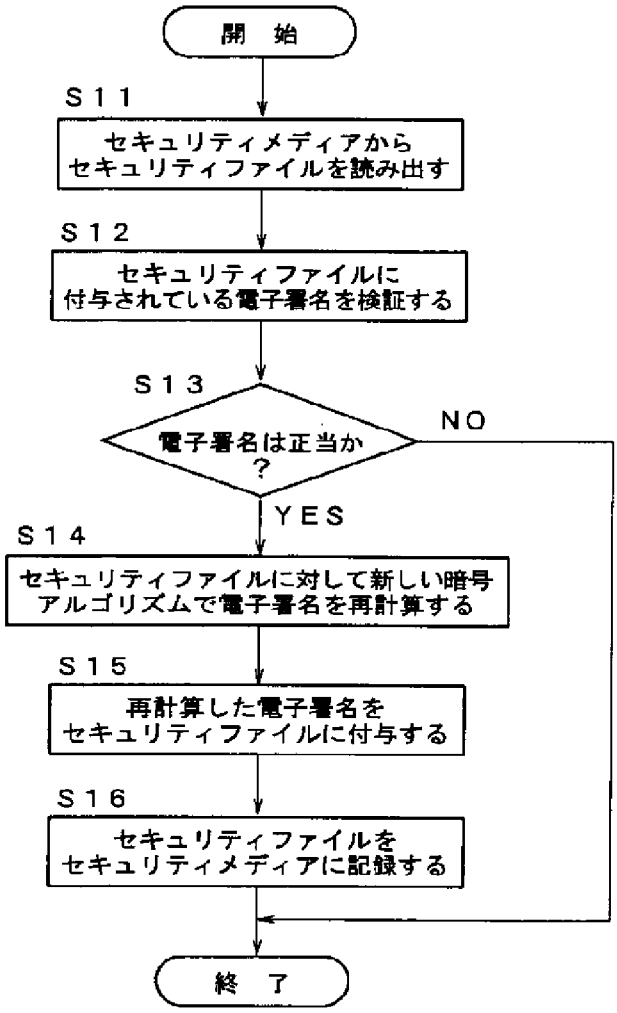
【図４】



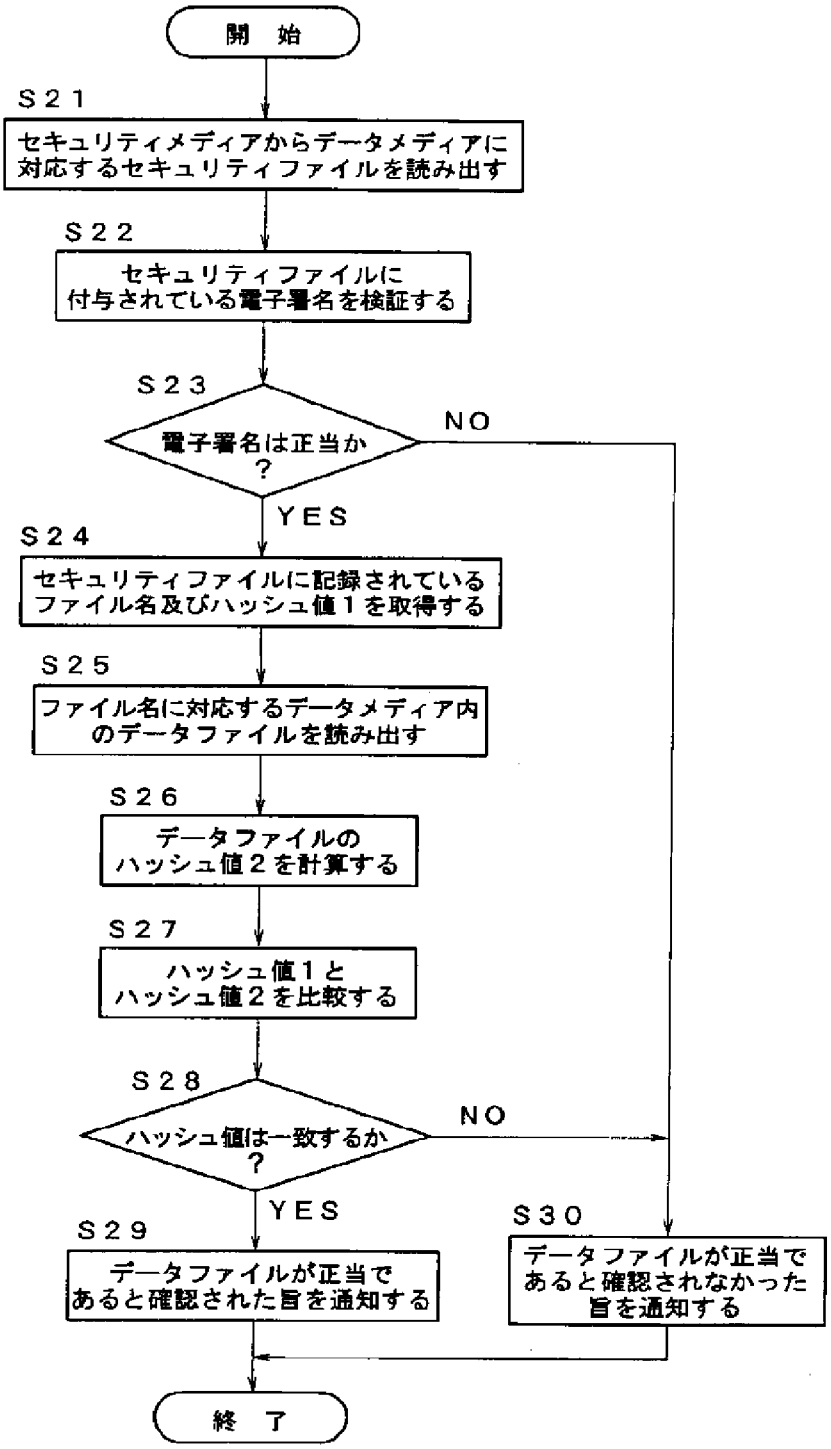
【図３】


11
12
ハッシュ値
13
14
ファイル値
ハッシュ値
/dir01/file01.doc
ade13fedu=
/dir01/file02.doc
gremf18cad=
/dir01/file03.doc
67ftdw8th=
データメディア
15
電子署名
R030-334238-00011.security
シリアル番号入り
ファイル名

【図５】


開　始
S11
セキュリティメディアから
セキュリティファイルを読み出す
S12
セキュリティファイルに
付与されている電子署名を検証する
S13
電子署名は正当か
？
NO
YES
S14
セキュリティファイルに対して新しい暗号
アルゴリズムで電子署名を再計算する
S15
再計算した電子署名を
セキュリティファイルに付与する
S16
セキュリティファイルを
セキュリティメディアに記録する
終　了

【図６】


開　始
S２１
セキュリティメディアからデータメディアに対応するセキュリティファイルを読み出す
S２２
セキュリティファイルに付与されている電子署名を検証する
S２３
電子署名は正当か？
NO
YES
S２４
セキュリティファイルに記録されているファイル名及びハッシュ値１を取得する
S２５
ファイル名に対応するデータメディア内のデータファイルを読み出す
S２６
データファイルのハッシュ値２を計算する
S２７
ハッシュ値１とハッシュ値２を比較する
S２８
ハッシュ値は一致するか？
NO
YES
S２９
データファイルが正当であると確認された旨を通知する
S３０
データファイルが正当であると確認されなかった旨を通知する
終　了

フロントページの続き

Ｆターム(参考) 5B017 AA03 BA07 BB06 CA15 CA16
5B082 EA11 GA11
5J104 AA09 LA05 LA06 NA02 NA12
NA32 NA42 PA14

(12)UK Patent (19)GB (11)2 333 865 (13)B

(45) Date of publication: **14.05.2003**

(54) Title of the invention: # A method and a mechanism for synchronised updating of interoperating software

(51) Int Cl[7]: **G06F** 9/44

(21) Application No: **9808657.2**

(22) Date of Filing: **24.04.1998**

(30) Priority Data:
(31) **9801661** (32) **28.01.1998** (33) **GB**

(43) Date A Publication: **04.08.1999**

(52) UK CL (Edition V):
G4A AFLA

(56) Documents Cited:
EP 0841615 A2 **WO 1994/025923 A1**
WO 1992/022870 A1 **US 5155847 A**
Tool pushes app distribution limits, Review of Marimba Castanet v3.0, LAN Times, 31 August 1998

(58) Field of Search:
As for published application 2333865 A *viz:*
UK CL (Edition P) **G4A** AFGDT AFGN AFL
INT CL[6] **G06F** 9/44
Other: **ONLINE: WPI, INTERNET**
updated as appropriate

Additional Fields
INT CL[7] **G06F** 9/445
Other: **ONLINE: EPODOC, JAPIO**

(72) Inventor(s):
Seamus Brendan Donohue

(73) Proprietor(s):
International Business Machines Corporation
(Incorporated in USA - New York)
Armonk, New York 10504,
United States of America

(74) Agent and/or Address for Service:
M J Jennings
IBM United Kingdom Limited,
Mail Point 110, Intellectual Property Law,
Hursley Park, WINCHESTER, Hampshire,
SO21 2JN, United Kingdom

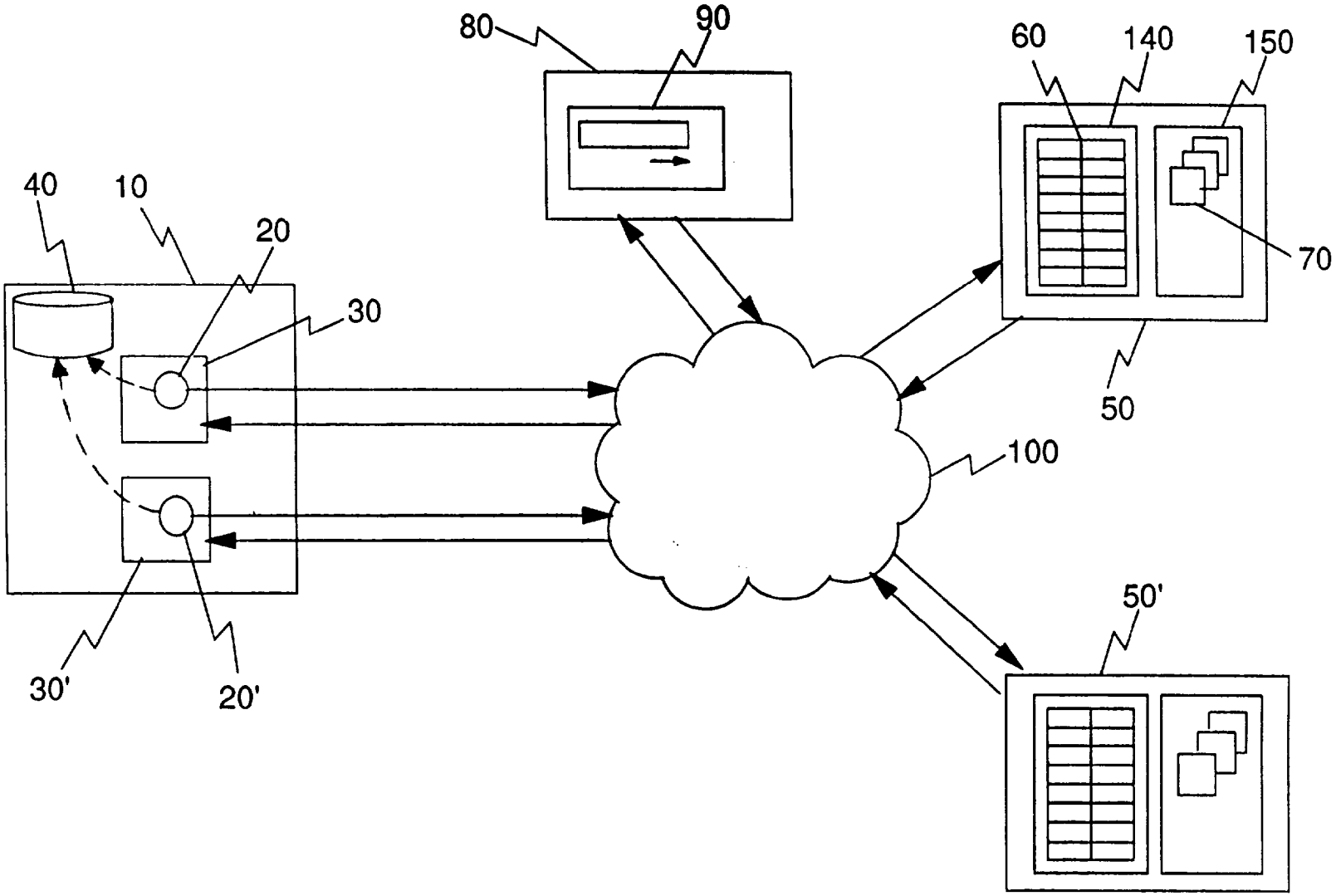
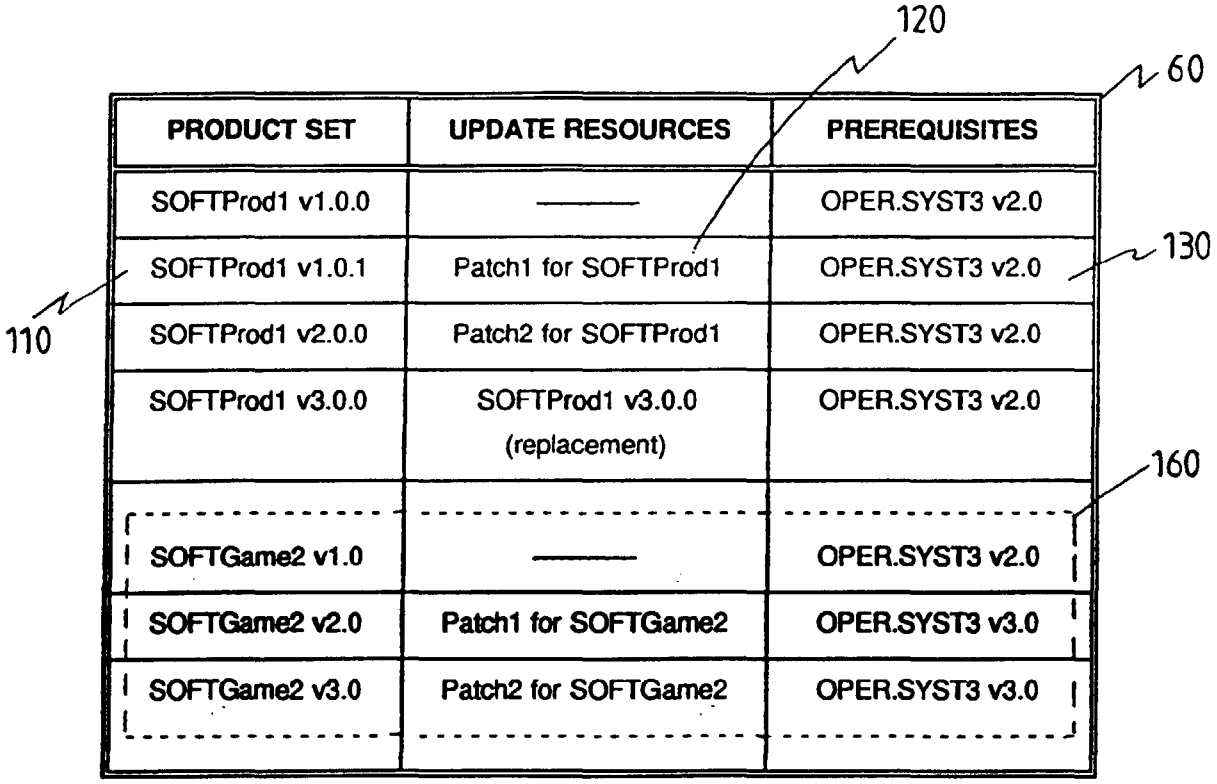


FIG. 1

2,333,865

Nintendo Exhibit 1003
Part 2 of 2
Page 00897



PRODUCT SET	UPDATE RESOURCES	PREREQUISITES
SOFTProd1 v1.0.0	———	OPER.SYST3 v2.0
SOFTProd1 v1.0.1	Patch1 for SOFTProd1	OPER.SYST3 v2.0
SOFTProd1 v2.0.0	Patch2 for SOFTProd1	OPER.SYST3 v2.0
SOFTProd1 v3.0.0	SOFTProd1 v3.0.0 (replacement)	OPER.SYST3 v2.0
SOFTGame2 v1.0	———	OPER.SYST3 v2.0
SOFTGame2 v2.0	Patch1 for SOFTGame2	OPER.SYST3 v3.0
SOFTGame2 v3.0	Patch2 for SOFTGame2	OPER.SYST3 v3.0

FIG. 2

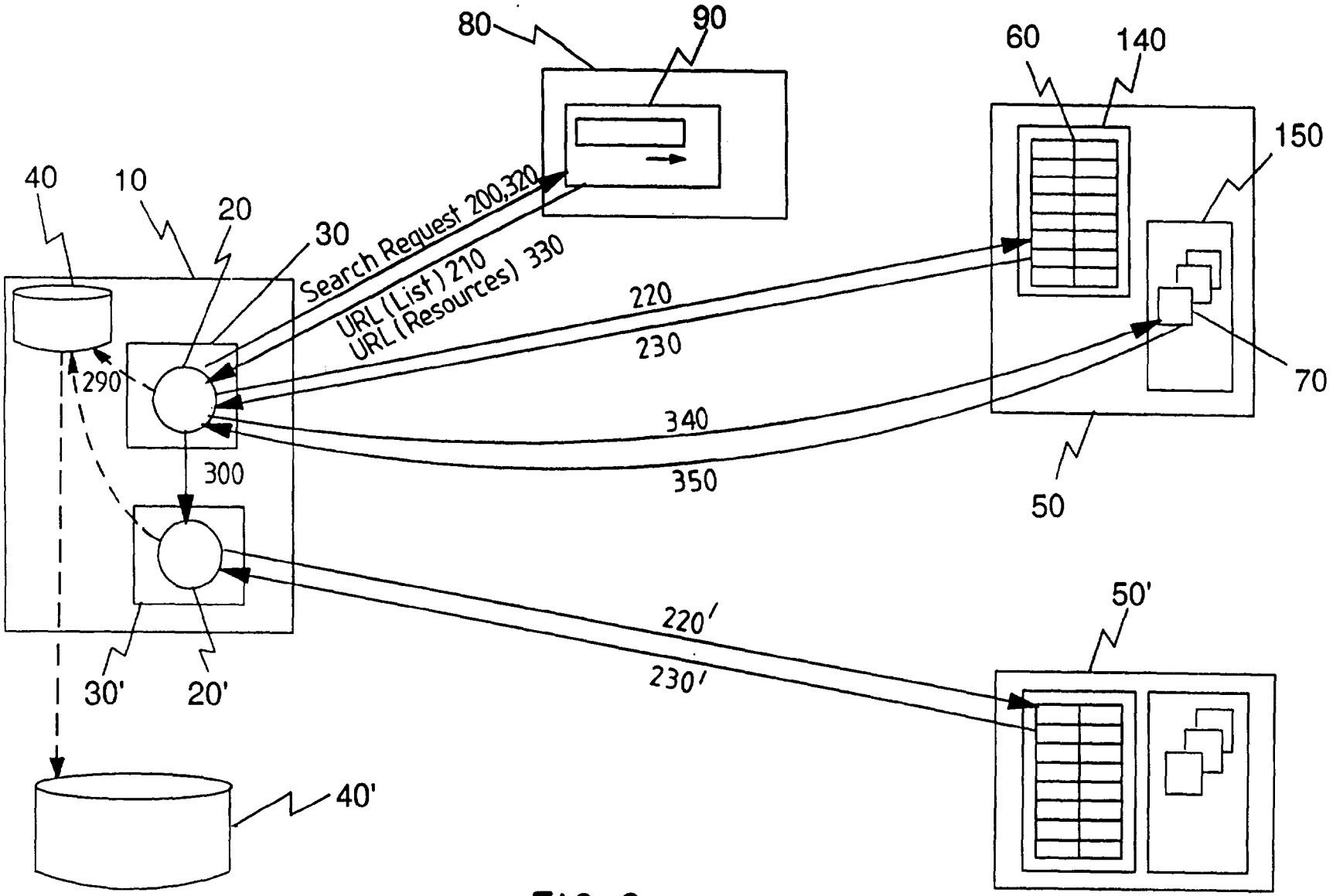


FIG. 3

3/5

Nintendo Exhibit 1003
Part 2 of 2
Page 00899

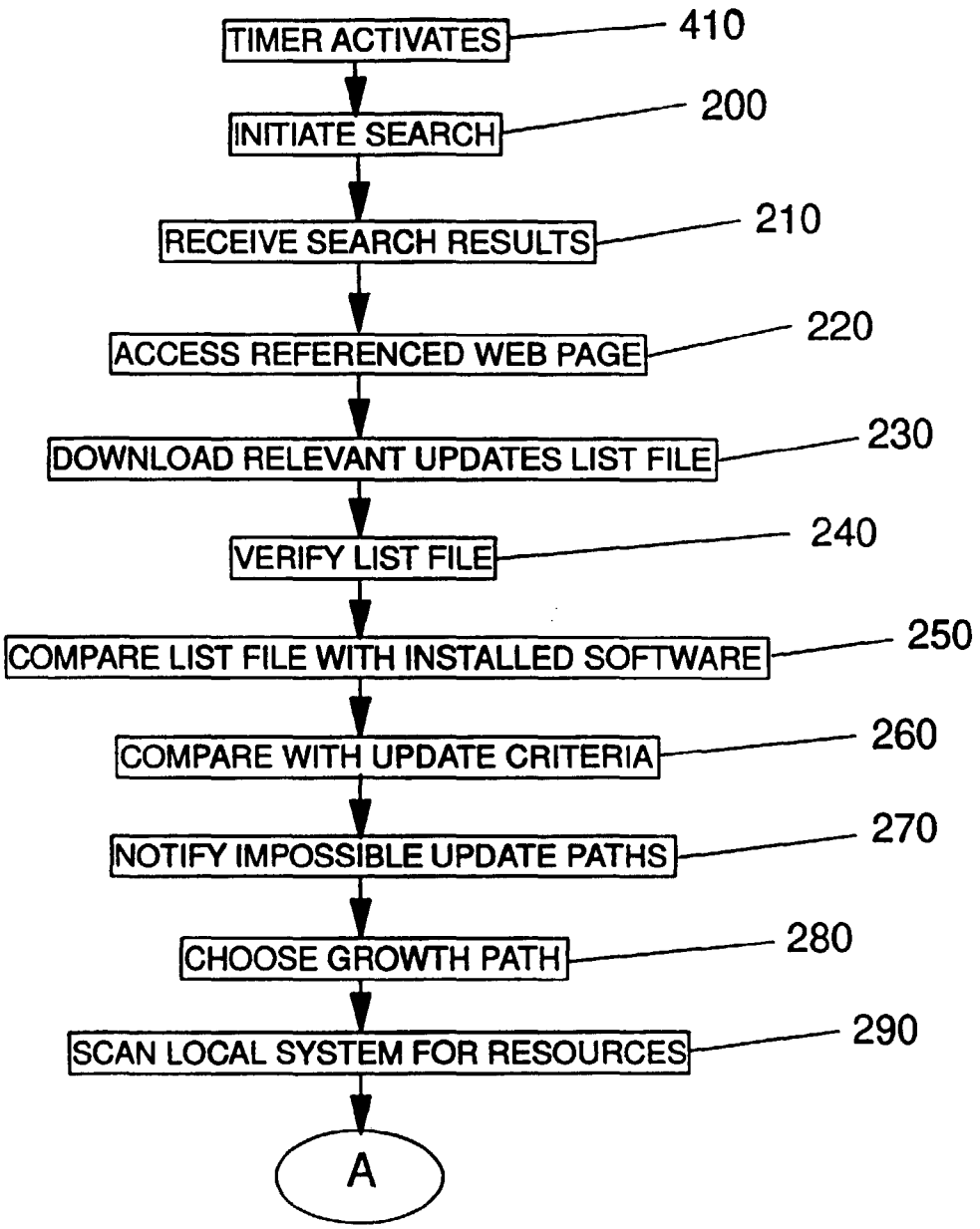

TIMER ACTIVATES
410
INITIATE SEARCH
200
RECEIVE SEARCH RESULTS
210
ACCESS REFERENCED WEB PAGE
220
DOWNLOAD RELEVANT UPDATES LIST FILE
230
VERIFY LIST FILE
240
COMPARE LIST FILE WITH INSTALLED SOFTWARE
250
COMPARE WITH UPDATE CRITERIA
260
NOTIFY IMPOSSIBLE UPDATE PATHS
270
CHOOSE GROWTH PATH
280
SCAN LOCAL SYSTEM FOR RESOURCES
290
A

FIG. 4A

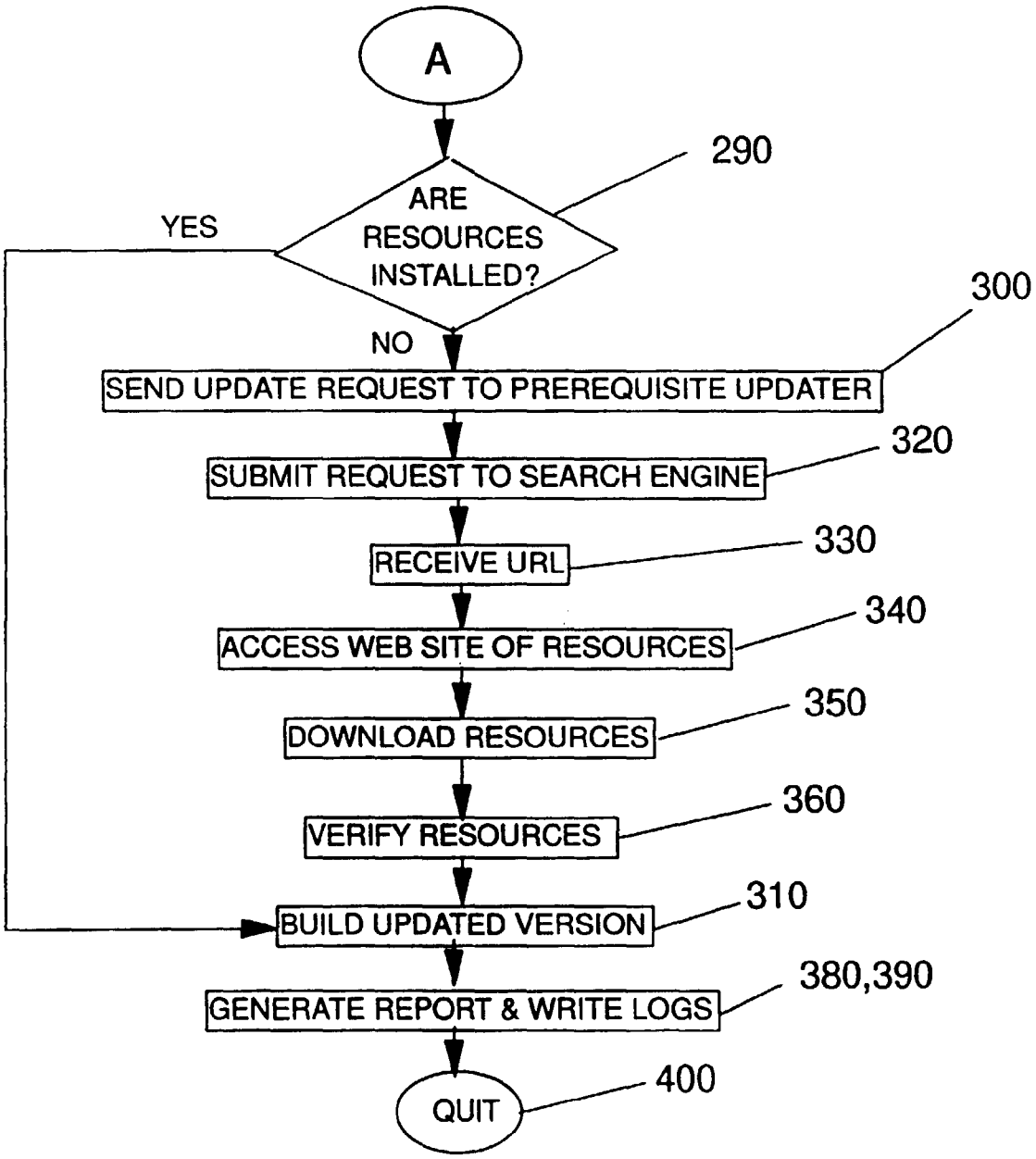

A
290
ARE RESOURCES INSTALLED?
YES
NO
300
SEND UPDATE REQUEST TO PREREQUISITE UPDATER
320
SUBMIT REQUEST TO SEARCH ENGINE
330
RECEIVE URL
340
ACCESS WEB SITE OF RESOURCES
350
DOWNLOAD RESOURCES
360
VERIFY RESOURCES
310
BUILD UPDATED VERSION
380,390
GENERATE REPORT & WRITE LOGS
400
QUIT

FIG. 4B

A METHOD AND A MECHANISM FOR SYNCHRONISED UPDATING OF INTEROPERATING SOFTWARE

Field of Invention

The present invention relates to synchronised updating of interoperating computer programs to maintain interoperability when the programs are updated.

Background

Software has conventionally been distributed in the form of programs recorded on a recording medium such as a diskette or compact disk. Customers buy the recording medium and a licence to use the software recorded on the medium, and then install the software onto their computers from the recording medium. The manufacture and distribution of the pre-recorded recording media are expensive, and this cost will be passed on to the customer. Also, the effort for customers of ordering or shopping for the software is undesirable.

The distribution cost is particularly problematic because most software products are frequently updated, both to correct bugs and to add new features, after the software has been delivered to the user. Some types of software products are updated many times each year. The cost of sending a new diskette or CD to all registered customers every time the software is upgraded or corrected is prohibitive and, although many customers want their software to be the most up-to-date, highest performance version and to be error free, not all customers want to receive every update. For example, the vendor may charge more for updates than the customer wants to spend, or new versions may require upgrading of other pre-requisite software products which the customer does not want to buy, or migrating to new versions may require migration of data which would disable the customer's system for a period of time.

Thus, software vendors tend to publicise the availability of new versions of their software and leave it for the customer to decide whether to purchase the latest upgraded version. For some software products, however, it is appropriate for the software vendor to proactively send out upgraded versions, or at least error correction and enhancement code (known as "patches") for their software products.

Whatever a particular company's policy, significant costs and effort are involved in releasing these various types of software updates.

Increasingly, software distributors are using the Internet as a mechanism for publicising the availability of updates to their software, and even for distributing some software. The Internet is a network of computer networks having no single owner or controller and including large and small, public and private networks, and in which any connected computer running Internet Protocol software is, subject to security controls, capable of exchanging information with any other computer which is also connected to the Internet. This composite collection of networks which have agreed to connect to one another relies on no single transmission medium (for example, bidirectional communication can occur via satellite links, fiber-optic trunk lines, telephone lines, cable television wires and local radio links).

The World Wide Web Internet service (hereafter 'the Web') is a wide area information retrieval facility which provides access to an enormous quantity of network-accessible information and which can provide low cost communications between Internet-connected computers. It is known for software vendors' customers who have Internet access to access the vendors' Web sites to manually check lists of the latest available versions of products and then to order the products on-line. This reduces the amount of paperwork involved in ordering software (and is equally applicable to other products). Some companies have also enabled their software to be downloaded directly from a Web site on a server computer to the customer's own computer (although this download capability is often restricted to bug fixing patches, low cost programs, and demonstration or evaluation copies of programs, for security reasons and because applying patches tends not to require any change to pre-requisite software or any data migration).

Information about the World Wide Web can be found in "Spinning the Web" by Andrew Ford (International Thomson Publishing, London 1995) and "The World Wide Web Unleashed" by John December and Neil Randall (SAMS Publishing, Indianapolis 1994). Use of the WWW is growing at an explosive rate because of its combination of flexibility, portability and ease-of-use, coupled with interactive multimedia presentation capabilities. The WWW allows any computer connected to the Internet and having the appropriate software and hardware configuration to retrieve any document that has been made available anywhere on the Internet.

Nintendo Exhibit 1003
Part 2 of 2
Page 00903

This increasing usage of the Internet for ordering and distribution of software has saved costs for software vendors, but for many software products the vendor cannot just rely on all customers to access his Web pages at appropriate times and so additional update mechanisms are desirable.

As well as the problem of manufacture and distribution cost associated with distributing media, there is the problem that customers typically need to make considerable proactive effort to find out whether they have the best and the latest version and release of a software product and to obtain and apply updates. Although this effort is reduced when Internet connections are available, even a requirement for proactive checking of Web sites is undesirable to many users since it involves setting up reminders to carry out checks, finding and accessing a software provider's Web site, navigating to the Web page on which latest software versions and patches are listed, and comparing version and release numbers within this list with the installed software to determine whether a relevant product update is available and to decide whether it should be ordered. There may be an annoying delay between ordering an update and it being available for use, and even if the update can be downloaded immediately the task of migrating to an upgraded version of a software product can be difficult. If these steps have to be repeated for every application, control panel, extension, utility, and system software program installed on the system then updating becomes very tedious and time consuming. Therefore, manual updating tends not to be performed thoroughly or regularly.

There is the related problem that software vendors do not know what version of their software is being used by each customer. Even if the latest version of their software has been diligently distributed to every registered customer (by sending out CDs or by server-controlled on-line distribution), there is still no guarantee that the customer has taken the trouble to correctly install the update. This takes away some of the freedom of software developers since they generally have to maintain backward compatibility with previous versions of their software or to make other concessions for users who do not upgrade.

It is known in a client-server computing environment for a system-administrator at the server end to impose new versions of software products on end users at client systems at the administrator's discretion. However, this has only been possible where the administrator

has access control for updating the client's system. This takes no account of users who do not want upgrades to be imposed on them.

Yet a further related problem is that software products often require other software products to enable them to work. For example, application programs are typically written to interoperate with a specific operating system. Since specific versions of one product often require specific versions of other products, upgrading a first product without upgrading others can result in the first product not working.

"Insider Updates 2.0" is a commercially available software updater utility from Insider Software Corporation which, when triggered by the user, creates an inventory of installed software on a user's Apple Macintosh computer and compares this with a database of available software update patches (but not upgraded product versions) and downloads relevant updates. "Insider Updates" shifts the responsibility for finding relevant updates from the user to the database maintainer, but the access to update patches is limited to a connection to an individual database and the tasks of scanning the Internet and on-line services to find updates and of maintaining the database of available updates require significant proactive effort. "Insider Updates" does not install the updates or modify the user's software in any way. "Insider Updates" does not address the problem of unsynchronised pre-requisite software products.

A similar product which scans selected volumes of a computer system to determine the installed software and which connects to a database of software titles for the Apple Macintosh®, but does not download updates, is Symmetry Software Corporation's "Version Master 1.5".

An alternative update approach is provided by "Shaman Update Server 1.1.6" from Shaman Corporation, which consists of: a CD-ROM (updated and distributed monthly) that users install on a PowerMac® file server; client software for each Macintosh computer to be inventoried and updated; and means for accessing an FTP site storing a library of current updates. "Shaman Update Server" creates an inventory of networked computers and downloads and distributes latest versions of software to each computer. Network administrators centrally control this inventory and updating process. The distribution of CD-ROMs has the expense problems described earlier.

Marimba Inc's software product "Castanet UpdateNow"(™) implements a combination of transmission of updates to client systems and support for the development of client applications following the instructions of administrator defined update command files to achieve either centralised or user controlled update management. Castanet UpdateNow does not provide for synchronized updating of collaborative software products.

Novadigm Inc's "Enterprise Desktop Manager"™ (EDM) provides for desktop software packages to be updated by means of an 'actual state' model and a 'desired state' model, which are administrator-created definitions of existing and desired desktop configurations. These object models are placed on a server computer and the server, at some predefined time, communicates with client computer systems and compares their actual state model and desired state model. The differences are translated into a list of required resources (software components, configuration files, etc) which are then downloaded and installed on the client system. EDM provides a centralised resource management which relies on server control and manual effort by an administrator to set and test configuration models and to install update resources at the server.

Summary of Invention

According to a first aspect of the invention there is provided method, implemented by an updater component installed on a computer system connected within a computer network, for determining whether the updating of computer programs installed on the computer system requires updating of prerequisite computer programs, the method comprising: in response to initiation of an update for a first one of said installed computer programs, the updater component initiating access to a list of computer programs which are prerequisites for the operation of said first computer program, comparing the listed prerequisite computer programs with computer programs installed on the computer system, and determining positively or negatively whether performing the requested update of the first computer program requires updating of prerequisite computer programs; and if the determination is positive, the updater component initiating an update of said prerequisite computer programs.

Controlling synchronisation of updates of computer programs and their pre-requisite programs by means of an updater component installed on the local system represents an improvement over prior art methods which implement server-controlled distribution of updates since a greater degree of automation and client control is achievable. With known server-controlled distribution, synchronisation of updates is only possible with manual setting of configuration models which define desired desktop

configurations (as well as installation and testing of update software at a distribution server), followed by server-controlled distribution. The predefinition and maintenance of desired state desktop configurations represents a significant overhead.

With the present invention, updating of computer programs and their pre-requisite programs can be initiated and controlled by a locally-installed updater component which accesses a list of the pre-requisite products for each enhanced version of the programs. Initiation of an update may be in response to a timer expiry for periodic update attempts, or may be part of the computer system's boot-up procedure, or may be at other times as defined by an update policy for the system.

The triggering of updating of pre-requisite computer programs is preferably implemented by the updater component of a first computer program communicating an update requirement to an updater component of the identified pre-requisite computer program. According to a preferred embodiment of the invention, updater components register themselves with a repository within the computer system when the updater components are installed, and installation of updater components is preferably an integral part of installation of the updater component's associated computer program. Since the updater components for all installed computer programs register with the repository, each updater component is identifiable and contactable by other updater components by reference to registration information in the repository.

A distinct advantage of the present invention over server-controlled update mechanisms which rely on administrator defined system configurations is that the present invention has the flexibility to automatically deal with any depth of inter-dependency in the relationship between pre-requisite products. That is, updating a first program may require updating of a second, which may require updating of a third, and this can be implemented by updater components communicating their update requirements to each other and consequent triggering of updates without any input from the administrator or user.

Furthermore, the invention according to the preferred embodiment deals with cyclic dependencies (such as where a first product A at version 2.1 requires product B at version 2.1, and B at 2.1 also requires A at 2.1). Indeed, the method of the present invention according to the preferred embodiment imposes no constraint over the topology of the update sequence of mutually dependent interoperating products.

An updater component according to a preferred embodiment of the invention thus controls upgrading of, and fixing of bugs within, an associated software product or products automatically without requiring any interaction by the user after an initial agreement of update criteria. The update criteria can be associated with the products'

licensing terms and conditions. This ensures that users who adopt a suitable update policy can always have the most up-to-date software available, with errors being corrected automatically from the viewpoint of the user. The user does not need to know where software updates come from, how to obtain them or how to install them since the update component takes care of this. The software vendor avoids having to ship special CDs or diskettes to correct errors or provide additional features; the vendor can easily release code on an incremental basis such that customers receive new product features sooner and with no effort.

An updater component according to a preferred embodiment of the invention performs a comparison between available software updates and installed software on the local computer system to identify which are relevant to the installed software, compares the available relevant updates with update criteria held on the local computer system (these update criteria are predefined for the current system or system user), and then automatically downloads and applies software updates which satisfy the predefined criteria.

This automatic applying of software updates preferably involves installing available software patches and/or upgraded versions in accordance with both the predefined update criteria and instructions for installation which are downloaded together with the program code required for the update. This feature of executing dynamically downloaded instructions provides flexibility in relation to the types of updates that can be handled by an updater component. It can also be used to enable a single generic updater component to be used with many different software products. Alternatively, the installation instructions for certain software updates may be pre-coded within the updater component. The "software resources" are typically a combination of program code, machine readable installation instructions and any required data changes such as address information.

A system according to the invention preferably retrieves update resources for implementing an update of an installed computer program by sending an update request to one or more computer systems at network locations at which the required software resources are provided (together with a list of update resources and pre-requisites). The information used to identify the network locations may be explicit network location information held by the updater component, or it may be a software vendor name or any other information which can be used as a search parameter for

identifying the location. In the preferred embodiment, the information is a product identifier which is provided by the updater component to a network search engine to initiate a search to identify the relevant network location at which are stored software resources for implementing an update to that product. This search may be performed by a conventional Internet (or other network) search engine which is called by the updater component. When the search engine returns an identification of the network location, the updater component retrieves from this location a list of available relevant updates, checks the list against the locally held software product version and against predefined update criteria, and retrieves the update resources onto the local computer system if those criteria are satisfied.

According to a preferred embodiment of the invention, a standardised naming convention is used for software resources from which to build software updates, and the updater component can search for these resources on a Network Operating System filesystem. This allows software resources to be stored at multiple locations to mitigate against network availability problems and makes it easier for developers and distributors to provide their error-fixing patches and upgraded versions of software products. For example, a developer can make new software updates available via a public network disk drive on their LAN using a known filename or via a published Uniform Resource Locator (URL) which can be searched for using known key words.

Updater components are preferably an integral part of the products they will serve to update. Hence, the updater component is distributed to software users together with an initial version of a software product, the updater component then automatically obtaining and applying software updates in accordance with preset criteria (such as a time period between successive searches for updates, and whether the particular user has selected to receive all updates or only certain updates - such as to receive updating patches but not replacement product versions for example).

The updater component's update capability preferably includes updating itself. Indeed, the update criteria may be set such that the updater component always accesses appropriate network locations to obtain updates to itself before it searches for software resources for updating its associated software products.

An updater component according to the invention preferably includes means for checking whether pre-requisite products are available, and are synchronised to the required version, as part of the process of selecting an update path for the current product. In a preferred embodiment, as well as checking their availability, the updater component is capable of instructing the updater components associated with pre-requisite software products to initiate updates to their software where this is the agreed update policy. If each software product's updater component is capable of triggering updates to pre-requisite products, then updates can ripple through the set of installed software products without the user having to be involved in or aware of the updates. This capability is a significant advantage over prior art updater agents which do not deal with the problem of unsynchronised software versions when one updates, and supports the increasing desire within the software industry for collaboration between distributed objects to perform tasks for the end user.

The updater component preferably also includes a mechanism for verifying the authenticity of downloaded software, using cryptographic algorithms. This avoids the need for dedicated, password-protected or otherwise protected software resource repository sites. The software resources can be anywhere on the network as long as they are correctly named and or posted to the network search engines.

Thus, the present invention provides an agent and a method for synchronising software updates which significantly reduces the cost and effort for software distributors of distributing and tracking software updates and significantly reduces the effort for system administrators and end users of applying updates to installed software.

Brief Description of Drawings

The present invention will now be described in more detail, by way of example only, with reference to the accompanying drawings in which:

Figure 1 is a schematic representation of a computer network including a local computer system having an installed updater component, server computers storing lists of available updates and storing software resources for applying updates, and a search engine for locating the servers;

Nintendo Exhibit 1003
Part 2 of 2
Page 00911

Figure 2 is an example of a software vendor's list of their software versions and the resources and prerequisites for building from one version to another;

Figure 3 represents the sequence of operations of an updater component according to an embodiment of the invention; and

Figure 4 is a further representation of the sequence of operations of an updater component.

Detailed Description of Preferred Embodiments

As shown in Figure 1, an updater component 20 is installed in system memory of a conventional network-connected computer system 10, together with an associated computer program 30. The updater component may have been delivered to the user of the local computer system on a storage medium (diskette or CD) for him to install, or it may have been explicitly downloaded from another computer system. In preferred embodiments of the invention, updater components are integrated within the computer program they are intended to maintain (or are otherwise delivered via the same mechanism and at the same time as their associated program). The updater component is then installed as part of the installation procedure of its associated program, such that the user is not required to take any special action to obtain or activate it. The installation of each updater component includes the updater component registering itself with the operating system or another repository 40 on the local computer system. Thus, at least the updater components on the local system are identifiable and contactable by address information, and/or their product identifier, within the register entry. In alternative embodiments of the invention, the repository 40 may be a central or distributed repository for the network, as will be discussed below.

It is a feature of the preferred embodiment of the invention that each updater component can locate, can be located by, and can communicate with other updater components which manage other software products. This capability is used when one updater component requires another one to update to a specific level before the former can execute its own update, as will be discussed below. This is enabled by the updater components registering within the operating system or other repository 40.

In the preferred embodiment, each registration entry contains two items: the updater path and the updater network address. The path is the location of the updater component binary file so that the updater component can be launched by the operating system during the boot up process. This ensures that the updater component is always active and ready to perform work or handle requests issued to it from other updater components. The network address is the address used by components on other computer systems in the network to locate it on the network and to communicate with it.

An example of such registration using a UNIX (TM) operating system and the TCP/IP protocol suite uses the following naming convention for updater components: SoftwareVendorName+_product_name+_updater.

Path registrations can be entered in the UNIX/etc/inittab file to store the path entry. When, for example, the updater component for IBM Corporation's DB2 (TM) database product is installed it will add an entry to the /etc/inittab file of the form:

ibm_db2pe_updater:2:respawn:/usr/abin/db2_updater_binary

Every time the computer system reboots it will read this file and launch the DB2 updater component. The "respawn" keyword in the updater entry ensures that, should the updater component process fail for some reason during general system operation, it will be restarted by the operating system automatically. This approach will ensure that all updater components for all installed applications are always active.

Network Location registrations can be entered in the UNIX/etc/services file. For example when the DB2 updater component is installed it will add an entry to the /etc/services file of the form:

ibm_db2pe_updater 5000/tcp #net location of DB2 updater component

When another updater component wishes to communicate with the DB2 updater component it will find it by searching this /etc/services file for the DB2 updater component name ibm_db2pe_updater (actually done indirectly by the UNIX call getservbyname() - the name is built by the caller according to the standard naming convention). When it is found it knows that the DB2 updater is listening for connections on port number 5000 and will use the TCP protocol. This allows the updater component in

question to establish a link to the DB2 updater component and start a conversation (described later).

For an updater component to find and talk to another updater component on another remote machine the above information would have to be augmented by having a repository 40' which is accessible from both machines (preferably a central or distributed database accessible from anywhere in the network, such as a Web Page or pan-network file) and is available to all updater components that require it. Entries would be of the form updater_name machine_ip_address (OR DNS entry), port number, protocol.

For example, the manufacturing department of an organisation may have three computer systems on which distributed software products collaborate with each other, the systems being called a, b and c. Typical entries in the Web page or file manufacturing_collaborators.html might be:

ibm_catia_updater	**a.manufacturing.com**	**5000**	**tcp**
ibm_db2pe_updater	**b.manufacturing.com**	**5100**	**tcp**
ibm_cics_updater	**c.manufacturing.com**	**4780**	**tcp**

An updater component can then connect and talk to any other updater component using the DNS name to create an IP address and the port number which the remote updater component is listening to at that address.

The steps of updater registration at installation are therefore:

1) Create entry in /etc/inittab file (register updater process code location)

2) Create entry in /etc/services file (register updater process local address)

3) Create entry in central database file (register updater process pan-network address).

The installation process may also involve providing to the updater component the local IP address of a Web proxy server. It will be clear to persons skilled in the art that many alternative registration implementations are possible.

Updater components include data fields for an identifier and version number for their associated software products. The updater components may be delivered to customers with these fields set to null values, and then the installation procedure includes an initial step of the updater component interrogating its software product to obtain an identifier and current program version and release number. Alternatively, the software vendor may pre-code the relevant product ID and version number into the updater component.

The system 10 of Figure 1 is shown connected within a network 100 of computers including a number of remote server systems (50,50') from which software resources are available for applying updates to programs installed on the local system 10. Each server system includes within storage a list 60 of the latest versions of, and patches for, software products which are available from that server. Each vendor is assumed here to make available via their Web sites such a list 60 of software updates (an example of which is shown in Figure 2) comprising their product release history, in a format which is readable by updater components, and to make available the software resources 70 required to build the releases from a given level to a new level (this transition from a software product release to a new level will be referred to hereafter as a 'growth' path). The entries in the software updates list 60 include for each software product version 110 an identification 120 of the software resources required for applying the update and an identification 130 of its prerequisite software products and their version numbers. In some cases, the required resources are complete replacement versions of software and associated installation instructions, and deletion instructions for the software being replaced. In other cases, the resources comprise patch code for modifying an existing program (e.g. for error correction) and the patch's installation instructions.

For the current example, we will assume that the network 100 is the Internet, although the invention may be implemented within an Intranet or any computer network. Also shown within the network 100 is a server system 80 on which a search engine 90 is installed for use in finding update source locations on the network. This is shown located remotely from the local system 10, although it need not be. In the Figure, each updater component 20 is shown associated with a single program 30, and it is a feature of this embodiment of the invention that all installed software products have associated updater components which manage them,

but neither of these features is essential to the invention as will be explained later.

The operation of an updater component will now be described, with reference to Figures 3 and 4. When an installed updater component executes, in response to completion of a cycle period or in response to a request from another software product's updater component, its first action is to initiate 200 a search for available updates to the particular software product. It provides to one or more search engines 90 search arguments comprising the product identifier and product version release number obtained by the updater component at install time. Software vendors wishing to benefit from the services of the updater component provide via their Web sites a list 60 of available product updates referenced by product identifier and release number 110 (or some other consistent naming convention is used). The search identifies the relevant Web site 140 on which the update information is available. A URL identifying the relevant Web site 140 for update information is returned 210 to the updater component as a result of the search. If the initial attempt to start a search engine is unsuccessful, then the updater component will attempt to start a different search engine (which may be in a different geographical location to the first), but in alternative embodiments could wait for a preset time period and then retry.

The updater component uses the URL to access 220 the list 60 and downloads 230 a file 160 comprising the portion of the list 60 of available updates which relates to the particular product. The updater component then performs steps 240-280 as shown in Figure 4. Each file 160 contains message digests (e.g. MD5) which are digitally signed. The retrieved file 160 is then analyzed 240 using a digital signature checking algorithm (such as the algorithm described in US patent 5231668). This verifies that the file 160 represents the correct software updates list for the particular software product, and that the file has not been tampered with since signing. Also, checking for the digital signature is a useful way of filtering the results of the search since these may include a plurality of Web page URLs other than the correct one (the search may find other pages which have a reference to the named product version, including pages not published by the software vendor). If an attempt to download and verify a file is not successful, then the updater component moves on to the next URL found in the search.

The updater component then performs on the local computer system a comparison 250 between the current installed software product's identifier and release number and the listed available updates in the retrieved file 160. This comparison determines possible growth paths from the current to updated versions, but these possible growth paths are then compared 260 with predefined update criteria, and any possible paths which do not satisfy the update criteria are discarded. Thus, the updater component determines whether it is possible to migrate from a current software product to the available new versions and whether it is possible to apply patches to the current version under the currently agreed licence terms and conditions.

For example, the software product licence may enable migration to any future version of the product and application of any available patches, or only migration up to a specified version, or it may only permit applying of available patches which modify or correct errors in the current version. Possible update paths which are unavailable due to current license limitations are notified 270 as a system generated message sent to the software asset manager (who may be an end user or IT procurement manager) of the currently installed version, to enable them to make decisions about whether the current licence is adequate.

As well as licence restrictions as to the updates that are possible, an updater component's update criteria or growth policy includes a cycle period (for example weekly or monthly) and criteria for determining which of a plurality of possible growth paths to select (such as always select latest version permitted by licence, or always select latest patch and only notify availability of new versions, or only select new versions if prerequisite software is already available on local system). The growth criteria may also include control information such as when to upgrade to new versions that are downloaded by the updater component - if data migration is required when migrating to a new software product version it may be essential for this to be done outside of office hours or only at a single scheduled time each month or each year and this can be controlled by the updater component.

The growth policy definition may also include a parameter determining that updating of pre-requisite software products should be requested when required to maintain synchronisation with the current product. This will be described in more detail below. Persons skilled in

the art will appreciate that there is great flexibility in the criteria that can be set and applied by the updater component.

The updater component then decides 280 on a particular growth path (i.e. which available version to upgrade to) from the set of possible growth paths using the update criteria. For example, the updater component may select the highest possible version or release number of the available updates which is permitted by the update criteria, if that is the update policy.

The updater component performs 290 (see Figures 3 and 4) a scan of the operating system file system to check whether the required software resources are already available on the local computer system. The required resources are the software update artifacts required to bring the current application software to the new level, and the software updates required for updating pre-requisite software to required levels. Each updater component associated with pre-requisite installed products is contacted 300 to ensure (a) that it is installed on the local system, and (b) that it is at or greater than the version level required for interoperability. If all required resources are available locally (or on another machine in the case of software relying on some pre-requisite software operating on a remote machine), and have been verified, then the updater component progresses to the step 310 (see Figure 4) of building the updated software version. If not, the update component must obtain the required resources.

As shown in Figures 3 and 4, if required software resources for building the updated version are not found on the local system, the updater component submits 320 a further request to one or more search engines to find the required resources. The search engine returns 330 one or more URLs and the updater component uses these to retrieve 340,350 the software resources into storage of the local computer system. At this stage, the updater component or the user need not have any knowledge of what corrections or enhancements may be included in the new version - the update criteria determine what type of updates are required such that the user is spared the effort of studying the content of every update. In practice, it is desirable for users to be able to determine the effects of updates and so the software resources for the update include a description of these effects which a user or administrator can read.

As examples, the software product to be updated may be a word processor application program. If the word processor as sold missed certain fonts or did not include a thesaurus, patches may subsequently be made available for adding these features. The updater component has the capability to add these to the word processor, subject to the update criteria.

In alternative embodiments of the invention, the search for required software resources is unnecessary following the initial search for the updates list (or is only necessary where there are pre-requisite software products as well as patches or new versions for the current product - see below). This is because the update software resources required directly by the current product are stored in association with the list of required resources. That is, the list includes a pointer to the network location of the required resources such that a selection of a growth path from the list involves a selection of a pointer to the network location of the required updates (and possibly also pointers to the locations of pre-requisite software products).

A second verification by digital signature checking is performed 360 (see Figure 4), this time on the downloaded resources. After verifying 360 the legitimacy of the downloaded resources, the updater component automatically builds 310 the installation in the target environment in accordance with the update policy. In practice, this may require information from the user such as an administration password, or a database usage parameter value, but in the preferred embodiment of the invention installing of the downloaded code is automatic in the sense that it does not require the user to know or obtain from elsewhere any installation information and in that it generally enables the user to be freed from making any decisions at run time if the predefined update criteria enable the updater component to automatically apply updates.

It is well known to include machine readable installation instructions encoded in a shell (for example as Script, or an interpretive language such as PERL, or an executable such as setup.exe in the case of applications on Microsoft's Windows (TM) operating system). Updater components according to the invention will download 350 the machine readable instructions together with the relevant software resources and will automatically execute them 310. The updater component thus automatically processes installation instructions, avoiding the input from a person which is conventionally required. The Scripts can be

adapted to reuse information gleaned from the first human installer who installed the first version of the updater component (for example, information such as user name and password of application administrator, installation directory, etc).

The method of updating according to the preferred embodiment of the invention requires software vendors to organise the software resources required to build from one product level to another. For example, a move from version 1.1.1 to 1.1.4 would typically include a series of patches to be applied, and the required order of installation if any would advantageously be encoded in machine processable installation instructions. The user is then spared the effort and the risk of human error which are inherent in methods which require the user to control the order of application of fixes and enhancements. The problem of how to migrate from one product level to another is thus dealt with by the software vendor instead of the customer, and updater components can only move to levels supported by the vendor (i.e. those growth paths published by the software vendor for a specific existing product level).

The updater generates 380 a report and writes 390 to log records, and then quits execution 400 (in the preferred embodiment the updater goes into a sleep or idle state) until activated again 410 upon expiry of a predetermined update cycle period (the repeat period parameter is configured when the updater component is installed).

Structure of updater component

The structure of an updater component comprises data, methods for operating on that data, and a public application programming interface (API) which allows other updater components to contact and communicate with it. This structure will now be described in detail.

UPDATER COMPONENT DATA:

The updater component includes the following persistent data:

Product_ID : an identifier of the software product which is managed by this updater component

Current_Installed_Version :
a version identifier for the installed software (e.g. version 3.1.0)

Current_License : a version identifier corresponding to the software product version up to which the current software license allows the user to upgrade (e.g. version 4.0.z). Alternatively, this may be a licence identifier (e.g. LIC1) for use when accessing machine readable licence terms.

Installation_Environment:
a list of attribute name/attribute value pairs. This is used by the updater component to store values entered by the user when the updater was used for the first time. For example, the updater

installation userid and password, possibly the root password, the installation directory, the web-proxy server address, search engine URLs, log file name, software asset manager e-mail address etc. This data will be re-used when subsequent automatic updates are required.

Growth policy parameters:

a. Growth_Cycle: data determining whether the updater component should attempt to update its software product every day, week or month, etc.

b. Growth_Type: data determining whether the updating is limited to bug fixing and enhancements (i.e. patches) only or requires upgrading to the latest release in each growth cycle.

c. Force_Growth: (YES/NO) a parameter determining whether to force other software resources to upgrade if that is a pre-requisite for this software to upgrade. (Some implementations will provide more flexible controls over

forcing other software to update than this simple YES/NO)

Last_Growth_Time: Date and time when updater component last executed

The updater component also includes the following non-persistent data:

Possible_Growth_Paths:
transient data representing the available upgrade paths (e.g. version numbers 3.1.d, 3.2.e, 4.0.a)

Update_Thread: string containing the update thread information. It has the following structure:
<Product_Id #1,version #1><Product_Id #2, version#2>.....<Product_id #n-1,version#n-1> <Product_id #n,version#n> .
The thread information lists the products (i.e. Product_Id #x) which are participating in the synchronisation transaction on this component and the versions they are attempting to grow to (i.e. version#y). It represents a thread of forced updates originating from component 1 (or root component) which is attempting to update unilaterally through component 2 which is being forced to update by component 1 and component 3 which is being forced to update by component 2 and so on to the nth component being forced by the $n-1^{th}$ component. It is the nth component in this Update thread which is forcing this component to update to a specific level. This string is empty if this is the first component attempting to update unilaterally (i.e. it is not being forced to update). This information is assembled to guard against cyclic dependencies in between products. See the "example of update synchronisation" section to find out how it is used to accomplish this.

PRIVATE UPDATER FUNCTIONS:

The updater component logic includes the following methods:

Discover_Possible_Growth_Paths()

Search for Growth_Path information for this software product on the Internet (or Intranet or other network). This search method initiates a search via a standard search engine server. The information returned is a list of newer versions and associated pre-requisite product information.

The Growth_Path information is then reduced in accordance with the Growth policy parameters. For all members in the Growth_Paths list, a check is performed of whether appropriate versions of pre-requisite products are available on the local and/or remote computer. The updater components managing these pre-requisite products are accessed and forced to grow if this is the policy.

If all pre-requisite products exist locally at the correct level, or are available remotely on the network and there is with a "force growth" policy, then i identifiers for newer versions of the software product are added to the Possible_Growth_Paths list.

Avoiding cyclic update dependencies :

If one of the pre-requisite software components is found in the Update_Threads string with a version which complies with the required version level then it is deleted from the pre-requisite list to ensure that a force update is not attempted on it.

If the prerequisite software is in the Update_thread list but with an insufficient version number then this is not a valid growth path. The final chosen growth path will have all pre-requisite software components elements which are either already available locally or already specified in the current Update_Thread or can be forced to update to the required version via an in place force update policy.

Nintendo Exhibit 1003
Part 2 of 2
Page 00923

```
Decide_Growth_Path()
        Interpret the growth policy and select a single growth
        path. Some implementations of the invention will
        involve user interaction to select the path, for
        example if there are considerations such as whether to
        force updates to other programs.

Get_Resources(Parameter: Chosen_Growth_Path)
        Given Chosen_Growth_Path (e.g.3.2.0), search for
        required resources (Parameters Product_ID,
        Current_Installed_Version, Chosen_Growth_Path),
        download all resources to local computer. This will
        include software required for the new version plus
        machine processable installation instructions.

Install_Resources()
        Process installation instructions including installing
        required files in correct locations, possibly
        compilation of the files and modifying the
        configuration of the existing system to accommodate the
        software, logging all actions to a file (and enabling
        an "uninstall" method to undo all actions).

Grow()
        Initiates methods:
        Discover_Possible_Growth_Paths()
              //uses Update_Thread information to avoid update
              //dependency cycles
        if no possible growth paths exist then
        updater component becomes idle
        else
              Decide_Growth_Path()
              Get_Resources(Parameter: Chosen_Growth_Path)
              Install_Resources().

        Then Grow() writes all completed actions to log and
        finishes execution of the updater component. The
        updater component becomes idle either until time to
        check again for new update requirements or until
        prompted by another updater component to do so.
```

PUBLIC UPDATER COMPONENT API:

The updater component includes the following public API. These functions would be callable using existing network communications software, such as remote procedure calls, message oriented middleware, ORB (Object request broker), etc.

Get_Release()

This function is called by other updater components and returns the release level of the product managed by this updater component.

Update(update_thread,new_level)

Other updater components call this function to move the product managed by this updater component to a new level indicated by the new_level parameter value. The update_thread parameter is a string containing the update thread information. It is of the form <product_Id,version><product_Id,version> ... . The thread information lists the products (i.e. Product_Id #x) which are participating in the synchronisation transaction and the versions they are attempting to grow to (i.e. version#y). When a software component calls this function on another software component it must append its own Product_Id and version to its copy of the string before passing it as a parameter (This string will only contain one Product_id and version if the calling component is the root component or first component in the update task). This will call the private function Grow().

Receive_Event(event details)

When an updater component receives a request to update, it must inform the calling updater component when it has completed the update or otherwise e.g. if it failed for some reason. The updater component performing the update on behalf of another updater component will call this function of the requesting updater component to communicate success of the update or otherwise. Event details can be a string like

Nintendo Exhibit 1003
Part 2 of 2
Page 00925

"product id, new release level, ok" or "product id, new release level, failure".

The automatic handling of the potential problem of unsynchronised pre-requisite products by enabling forcing of updates (or, if forcing of updates is not part of the update policy, sending of notifications to the software asset manager) is a significant advance over prior art update schemes.

Since the updates list file 160 returned to the updater component in response to an initial search includes an identification 130 of pre-requisite software, that information enables the aforementioned examination 290 of the updater component registration database 40,40' to check whether pre-requisite software is available locally or remotely. If it finds all the updater components located locally or remotely, it can be sure that the software pre-requisites are available and it next needs to contact each updater component for each software product to be sure all pre-requisites are at the correct level. If an updater component 20' having a required product identifier for pre-requisite software 30' but not having the required version number is found locally or remotely, and if forcing of updates is the update policy, then the updater component 20 of the first computer program contacts 300 this pre-requisite updater component 20' and requests that it attempt to update its associated pre-requisite software product 30'. This updater component 20' can, if necessary, request other updater components of its pre-requisite software to update their versions, and so on.

If at some stage no relevant updater component is found locally or remotely, then a message is sent to the asset manager to inform him/her of the requirement for a new product in order to grow the associated product further. If at some stage during the chain of updater requests to grow to a new level one updater component fails to move to the required level then this failure is reported back to its calling updater component, prompting failure of that component's update operation, and so on back to the updater component which initiated the whole transaction.

Thus, as well as their autonomous behaviour defined by their update criteria, updater components can react to external stimuli such as requests from other updater components.

Nintendo Exhibit 1003
Part 2 of 2
Page 00926

Example of update synchronisation

An example of the implementation of update synchronisation between two products will now be described. This example shows how one updater component can communicate with another to synchronise pre-requisite software so that all products are present and at compatible release levels.

A CORBA (Common Object Request Broker Architecture) ORB (Object Request Broker) is used for location of and communication between two updater components. Using the above public API it is a simple matter for those familiar with the art of CORBA programming to develop communication code so that one updater component can talk to another updater component anywhere on a network. In this example the component updater registration database 40 is a directory or folder available over the network (e.g. via NFS) which contains for each installed updater component a file called "updater_component_name.iop" (iop stands for interoperable object reference).

This file contains a sequence of bytes which can be converted into a reference to the updater component by any updater component which reads the file using for example the CORBA function:

CORBA::Object::_string_to_object() in C++

Furthermore this reference can be to an updater component anywhere on the network as it represents a unique address for the corresponding updater component. When updater component A has manufactured a reference to updater component B then updater component A can call a public API function simply by using, for example, a C++ mapping A->Get_Release() which will then return the value of the release level of the software managed by the A updater component.

In this example we will consider two products - IBM Corporation's DB2 database product and a Query Tool called "Query Builder", on different machines M and N respectively. (Machines M and N could be the same machine; the present example merely shows that they may also be separate). Both products have updater components which use a CORBA ORB architecture as briefly outlined above. An ORB communication daemon is active on participating systems M and N.

Step 1) Registration phase:

The DB2 Updater Component starts when the operating system starts on system M and immediately creates a file called ibm_db2_updater.iop (according to some naming standard used to aid subsequent searches for the file) in the network file system folder or directory. This directory could be hosted on any machine and not necessarily M or N. The file contains a series of bytes which can be used to manufacture a reference to the updater component.

[pseudocode]

```
Filehandle = open("/network/filesystem/directory","ibm_db2_updater.iop");
ReferenceBytes = CORBA::Object::_object_to_string();
Write(FileHandle, ReferenceBytes);
close(Filehandle);
```

QueryBuilder Updater component starts and writes its registration to the same directory or folder, again in this case calling the file ibm_querybuilder_updater.iop.

At this stage both updater components are active and have registered their presence and location in the network directory.

Step 2).

QueryBuilder attempts to grow from version 1 to version 2 but a prerequisite is DB2 version 2.1 or higher. The following sequence of actions will occur. QueryBuilder is denoted QB and DB2 as DB2.

QB : searches for file ibm_db2_updater.iop (file name manufactured according to standard) in network directory. It finds the file, reads it and converts it to a usable reference.

[pseudocode]

```
if (dbref =
CORBA::Object::_string_to_object(readfile(ibm_db2_updater.iop)))
                then SUCCESS we have connected to the updater
          else
                FAIL : Prerequisite software does not exist in set of
                collaborating systems - send e-mail to software asset
                manager to notify situation.
```

Nintendo Exhibit 1003
Part 2 of 2
Page 00928

Give up on trying to grow to new version.
endif.

Step 3).

At this stage we know that DB2 exists somewhere in our set of networked computers. Now we need to know if it is at the right level. We simply do this by executing its public API function Get_Release() defined above, from within the QB updater, the QB updater is therefore a client requesting the DB2 updater to do something for it, i.e. tell it what release it is.

[pseudocode]
db2_release = dbref->Get_Release();

Let us say this returns the value "2.0".

Step 4)

Client Side:

The QB Updater Component knows that this is not sufficient , it requires version 2.1. It examines its Force_Growth parameter which is, for example, "YES" meaning it should force pre-requisite software to grow to the level required before it can perform its own update procedure. Therefore the QB updater tells the DB2 updater to grow to the new release, and then waits until the pre-requisite has grown to the new release or failed in doing so.

[pseudocode]
dbref->Update("QB,2.0","2.1", QBref);
//the first parameter is the update thread parameter, so far only
//one product is participating in the update chain, i.e. Query
//Builder which is attempting to grow to version 2.1. The DB2
//updater component will use this information in this parameter to
//avoid the problem of update cycles. See server side for how this
//information is processed. QBref is a ready made reference to the
//QB Updater. It is passed to the DB2 updater so that it can
//quickly send the results, success or failure, when the DB2
//Updater has finished trying to update itself.
EVENT= null;

While (EVENT equals null)
{ do nothing; }
if (EVENT equals "SUCCESS")
then attempt to grow software managed by this updater component
i.e. Query Builder.
else
Write failure to log;
do not attempt to grow;
go to sleep and try later;
endif.

Server Side:

The DB2 Updater component receives the request to grow. Which it attempts to do.

It reports the result to the calling client (it knows how to contact the calling client as it is receives a reference to the caller in the function call.)

[pseudocode]
DB2 updater component Update_Thread is set to equal update_thread passed to it via the update call. i.e. Update_Thread = "QB,2.0"
//see private grow function for details on how this Update_Thread
//information is processed.

DB2 attempts to grow.
if Growth Successful then
QBRef->Receive_Event("SUCCESS"); // Note the implementation of the
// function Receive_Event simply sets the variable
// called EVENT in the QB Updater component to the
// value of the parameter passed in the API call , i.e.
// "SUCCESS" if in this section of the IF statement.
else
QBREF->Receive_Event("FAILURE");
end if

As noted previously, predefined update criteria may determine which of an available set of updates should be applied and which should be disregarded. The update criteria may include an instruction to the updater component to send a notification to the end user or system

administrator when a software update is identified as being available but applying this update is not within the update policy or is impossible. One of the examples given previously is that the update policy may be not to install full replacement versions of software products since that may require upgrading of pre-requisite software products or migration of data (for example if the software product is a database product), whereas it may be intended policy to install any error-correction patches. Notification rather than automatic installation of updates may also be implemented where to upgrade one product to a new version would require upgrading of other pre-requisite complementary products.

The update policy can also determine the degree of automation of the updating process, by defining the circumstances in which the updater requests input from the user or administrator.

The execution of a particular example updater component will now be described in more detail by way of example. This updater component's function is to keep an installed product called "Test" totally up-to-date with all released patches, but not to install replacement versions of Test. Firstly, the updater component is configured with the following data instantiations:

```
Product_ID: Test
Current_Installed_Version: 1.0.a
Current_License: LIC1
Installation_Environment:"USERID:TestOwner, USERPASSWORD:easy"
                         "INSTALLPATH: /usr/bin/testapp/"
Growth_Cycle: weekly
Growth_Type: patches, latest, automatically
Force_Growth: no
Last_Growth_Time: Monday 08/10/97.
```

The updater then executes weekly, for example each Monday night at 3am (it is the system administrator who decides the timing).

The following represents a possible execution trace for this example updater component.

Example execution trace

Step 1) The growth cycle starts:

>>>> START : Discover_possible_Growth_Paths()
* Execute search on remote search engine (e.g. Internet Search Engine) using Phrase ("IBM Test 1.0.a Growth Paths")
Search returns URL published by software vendor outlining current growth paths for product;
* Download URL:
File contents are :
"1.0.b,none; 2.0, other_required product_product_id 1.0.c;"
* Authenticate URL file using hashing algorithm and digital signature.
If not authentic, return to search for another URL matching criteria
* Build growth_path_list : growth_path list = "1.0.b,none; 2.0,other_required_product_id 1.0.c;"
* Remove all but patch level increases (according to Growth_Policy) from Growth_path list (i.e. only those with the first version and second release number matching 1.0).
* growth_path list = "1.0.b,none;"
* For all members in list, ensure prerequisites exist. In this example, all members of list meet this criteria trivially.
(Note : If this was a forced update then the Update_Thread would be examined to see if any pre-requisite component for any path existed already in the Update_Thread, if it does and at a level greater than or equal to the level required by the growth path then that prerequisite is assumed to exist (as will be in place once the update transaction is complete).)
* Place candidate growth_paths into Possible_Growth_Paths list = 1.0.b
<<<< END : Discover_possible_Growth_Paths()

Step 2) Next the updater component decides on the Growth Path to pursue:

>>>> START Decide_Growth_Path()
* The growth policy dictates that we should grow to latest patched revision. (In this example, determining the latest revision is trivial i.e. it is 1.0.b)
* chosen_growth_path = 1.0.b
(NOTE : For all pre-requisites which are not already in place at the required revision level, an update is forced on their associated updater agents to make them migrate to the required level. Note that, because items in the Update_Thread

are considered to already exist, no software component which is already on the Update_Thread is forced to update - thus preventing cyclic dependencies between products.)

<<<< END : Decide_Growth_Path()

Step 3) The updater component then obtains the required resources to revise the current software level to the new one.

>>>> Get_Resources()

* Execute search on remote search engine (e.g. Internet Search Engine) using Phrase ("IBM Test REVISION 1.0.a to 1.0.b RESOURCES").

* Search returns URL say ftp://ftp.vendor-site/pub/test/resources/1.0.a-b"

* Updater downloads file pointed to by URL and places in secure holding area where it verifies authenticity.

* Updater verifies authenticity (using, for example, digital signatures based on RSA algorithm, or any method)

If files not authentic, then return to search (see Note 1 below)

* Updater unpacks resources into a temporary directory (see Note 2 below). These resources include machine processible installation instructions (for example, instructions written in a script language such as a UNIX shell script or MVS REXX) and files (either binary or requiring compilation) which actually contain the software fix.

<<<< END : Get_Resources()

Notes on above tasks

Note 1 - To save time the updater looks for a standard file before downloading the URL ,called "signature", which contains the URL ftp://ftp.vendor-site/pub/test/resources/1.0.a-b and a listing of its contents. This is hashed and signed. Using this signature, the Updater component can quickly establish authenticity of the URL (to some extent) before downloading it and use the information i.e. file listings to corroborate the final downloaded resources after they have been unpacked into the temporary directory. When the final URL is downloaded it is also checked again for authenticity (to guard against

someone placing a bogus artefact in an authentic URL location).

Note 2 - Part of the unpacking is that the updater component will examine the installation scripts and modify them based on the contents of its installation environment data where required. For example if the installation instructions were coded in a shell script it will replace all instances of INSTALLPATH with the token "/usr/bin/testapp/". Again Naming conventions of attributes are standardised as it the method of token substitution in installation instructions. This makes totally automatic installation possible.

Step 4) The updater component then implements the actual software upgrade:

```
>>>> START Install_Resources()
* execute the installation instructions.
* update the values of
Current_Installed_Version = 1.0.b
Last_Growth_Time = Date+Time.
* send an e-mail to software asset manager informing of
installation and whether or not a reboot of the Operating System
or restart of the application is required before the upgrade
takes affect.
<<<< END Install_Resources()
```

This is the end of this current growth cycle. The seed updates the Last_Growth_Time value the current time and then exits. The time taken for this cycle could be anything from a few seconds where the updater component found no upgrade paths for the currently installed version to several hours if a totally new release from the current one is to be downloaded and installed together with new pre-requisite software.

An alternative to the embodiment described above in detail does not require an independent updater component for every different software product, but uses a single generic updater component installed on a system together with product-specific plug-in objects and instructions which are downloaded with each product. These objects interoperate with the generic code to provide the same functions of the product-specific

updater components described above. It will be clear to persons skilled in the art that the present invention could be implemented within systems in which some but not all application programs and other software products installed on the system have associated updater components, and that other changes to the above-described embodiments are possible within the scope of the present invention.

Nintendo Exhibit 1003
Part 2 of 2
Page 00935

CLAIMS

1. A method, implemented by an updater component installed on a computer system connected within a computer network, for determining whether the updating of computer programs installed on the computer system requires updating of prerequisite computer programs, the method comprising:

in response to initiation of an update for a first one of said installed computer programs, the updater component initiating access to a list of computer programs which are prerequisites for the operation of said first computer program, comparing the listed prerequisite computer programs with computer programs installed on the computer system, and determining positively or negatively whether performing the requested update of the first computer program requires updating of prerequisite computer programs; and

if the determination is positive, the updater component initiating an update of said prerequisite computer programs.

2. A method according to claim 1, wherein the updater component initiates the update for the first installed computer program in accordance with predefined update criteria corresponding to software licence terms and conditions.

3. A method according to claim 2, wherein the updater component initiates update attempts periodically according to a set cycle period defined for the updater component.

4. A method according to any one of claims 1 to 3, wherein the step of initiating access to a list of prerequisites includes the step of the updater component providing to a network search engine a search parameter for identifying one or more network locations at which are provided a list of software resources required for applying updates and a list of prerequisite computer programs.

5. A method according to claim 4, including the steps of:

the updater component initiating access to one or more network locations at which are provided the software resources required for applying an update;

downloading the required software resources from said one or more locations onto the computer system; and

using the retrieved software resources and the first computer program to generate a new version of the first computer program.

6. A method according to any one of the preceding claims, wherein the step of determining whether updating of prerequisite computer programs is required includes comparing said listed prerequisite computer programs with predefined update criteria stored on said computer system.

7. A method according to any one of the preceding claims, wherein the step of initiating an update of prerequisite computer programs comprises communicating an update requirement to one or more updater components associated with the prerequisite computer programs.

8. A method according to claim 7, including the following steps implemented by an updater component in response to an update request from another updater component:

initiating access to a list of the updater component's respective computer program's prerequisite computer programs,

comparing the listed prerequisite computer programs with computer programs installed on the computer system,

determining positively or negatively whether performing the requested update of the updater component's respective computer program requires updating of prerequisite computer programs; and

if the determination is positive, initiating an update of said prerequisite computer programs.

9. An updater component for installation on a computer system connected within a computer network, for determining whether the updating of computer programs installed on the computer system requires updating of prerequisite computer programs, the updater component including:

means responsive to initiation of an update for a first one of said installed computer programs for initiating access to a list of said first computer program's prerequisite computer programs;

means for comparing the listed prerequisite computer programs with computer programs installed on the computer system and determining positively or negatively whether performing the requested update of the first computer program requires updating of prerequisite computer programs; and

means for initiating an update of said prerequisite computer programs if the determination is positive.

10. A computer system including an updater component for determining whether the updating of computer programs installed on the computer system requires updating of prerequisite computer programs, the updater component including:

means responsive to initiation of an update for a first one of said installed computer programs for initiating access to a list of said first computer program's prerequisite computer programs;

means for comparing the listed prerequisite computer programs with computer programs installed on the computer system and determining positively or negatively whether performing the requested update of the first computer program requires updating of prerequisite computer programs; and

means for initiating an update of said prerequisite computer programs if the determination is positive.

cited Reference 2 (Abstract)

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-091598
(43)Date of publication of application : 29.03.2002

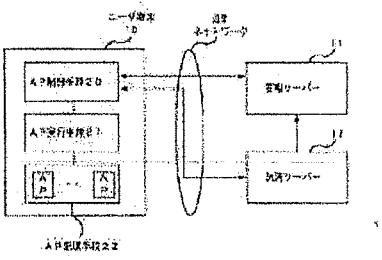
(51)Int.Cl. G06F 1/00
G06F 13/00

(21)Application number : 2000-284392
(22)Date of filing : 19.09.2000

(71)Applicant : NIPPON COMPUTER CO LTD
(72)Inventor : FUJIWARA HIDEJI

(54) SYSTEM AND METHOD FOR PROCESSING INFORMATION AND RECORDING MEDIUM RECORDED WITH INFORMATION PROCESSING PROGRAM OPERATING ON COMPUTER

(57)Abstract:
PROBLEM TO BE SOLVED: To manage performance authority of a function mounted on a user terminal while separating the performance authority from the user without inviting an increase in connection cost or a decrease in a performance speed.
SOLUTION: An information processing system is provided with a managing means for managing information (hereafter, called 'permission information') showing whether or not the function mounted on the user terminal can be performed and a controlling means for receiving the performance request for the function, inquiring of the managing means to acquire the permission information of the function and controlling the acceptance/rejection of the function performance on the basis of the acquired permission information. The controlling means is arranged in the user terminal, and the managing means is arranged in a host computer capable of communicating with the user terminal via a communication network.



Nintendo Exhibit 1003
Part 2 of 2
Page 00939

Cited Reference 2

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－91598
（P2002－91598A）
(43)公開日　平成14年３月29日(2002.3.29)

(51)Int.Cl.[7]	識別記号	ＦＩ	テーマコード(参考)
Ｇ０６Ｆ　1/00		Ｇ０６Ｆ　13/00	５３０Ｓ　５Ｂ０７６
13/00	５３０	9/06	６６０Ｃ

審査請求　未請求　請求項の数12　ＯＬ　（全８頁）

(21)出願番号　特願2000－284392(P2000－284392)

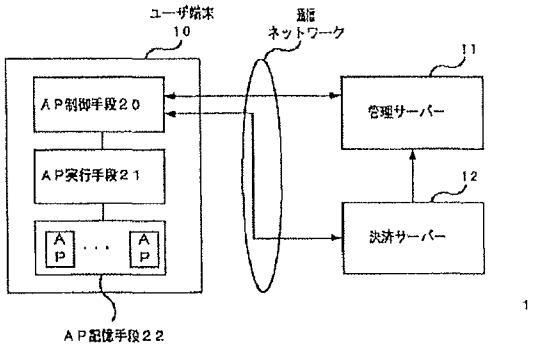
(22)出願日　平成12年９月19日(2000.9.19)

(71)出願人　500245972
日本コンピューター株式会社
福岡県北九州市小倉北区堅町二丁目４番１号
(72)発明者　藤原　秀司
福岡県北九州市小倉北区堅町二丁目４番１号　日本コンピューター株式会社内
(74)代理人　100079108
弁理士　稲葉　良幸　（外２名）

Ｆターム(参考)　5B076 FB05

(54)【発明の名称】　情報処理システム及びその方法、並びにコンピュータ上で動作する情報処理プログラムを記録した記録媒体

(57)【要約】
【課題】　接続コストの増大や実行速度の低下を招くことなく、ユーザ端末に実装される機能の実行権限をユーザから切り離して管理することを目的とする。
【解決手段】　ユーザ端末に実装される機能を実行してよいかどうかを示す情報（以下、「許可情報」と呼ぶ。）を管理する管理手段と、前記機能の実行要求を受けつけ、前記管理手段に問い合わせて前記機能の許可情報を取得し、前記取得した許可情報に基づいて前記機能の実行可否を制御する制御手段とを備える。前記制御手段は前記ユーザ端末内に配置され、前記管理手段は前記ユーザ端末と通信ネットワークを介して通信可能に構成されたホストコンピュータ内に配置される。



Nintendo Exhibit 1003
Part 2 of 2
Page 00940

【特許請求の範囲】
【請求項１】　ユーザ端末に実装される機能の実行を制御する情報処理システムであって、
前記機能を実行してよいかどうかを示す情報（以下、「許可情報」と呼ぶ。）を管理する管理手段と、
前記機能の実行要求を受けつけ、前記管理手段に問い合わせて前記機能の許可情報を取得し、前記取得した許可情報に基づいて前記機能の実行可否を制御する制御手段とを備えており、
前記制御手段は前記ユーザ端末内に配置され、前記管理手段は前記ユーザ端末と通信ネットワークを介して通信可能に構成されたホストコンピュータ内に配置されていることを特徴とする情報処理システム。
【請求項２】　前記機能は、ユーザ端末に記憶されるアプリケーションソフトウェアによって実現される機能であることを特徴とする請求項１記載の情報処理システム。
【請求項３】　前記許可情報は、前記アプリケーションソフトウェアの代金に関して決済が終了している場合に実行可を示し、終了していない場合に実行不可を示すように構成されていることを特徴とする請求項２記載の情報処理システム。
【請求項４】　前記制御手段は、取得した許可情報が実行不可を示す場合、前記ユーザ端末と通信ネットワークを介して通信可能に構成された決済装置に決済処理を行うように通信を行い、
前記管理手段は、前記決済装置より決済処理が終了したことを示す情報を受け取った場合に許可情報について実行可を示すように変更することを特徴とする請求項３記載の情報処理システム。
【請求項５】　前記管理手段は、更に、前記アプリケーションソフトウェアを実行できる期間、及び／又は実行できる回数、及び／又は実行できる機能の範囲に関する条件を記憶しており、
前記許可情報は、前記アプリケーションソフトウェアに関して決済が終了しており、かつ前記条件が満たされている場合に、実行可を示すように構成されていることを特徴とする請求項４記載の情報処理システム。
【請求項６】　ユーザ端末に実装される機能の実行を制御する情報処理方法であって、
前記ユーザ端末と通信ネットワークを介して通信可能に構成された管理手段において、前記機能を実行してよいかどうかを示す情報（以下、「許可情報」と呼ぶ。）を記憶する工程と、
前記ユーザ端末において、前記機能の実行要求を受け付ける工程と、前記管理手段に問い合わせて前記機能の許可情報を取得する工程と、前記取得した許可情報に基づいて前記機能の実行可否を制御する工程とを備えることを特徴とする情報処理方法。
【請求項７】　前記機能は、ユーザ端末に記憶されるアプリケーションソフトウェアによって実現される機能であることを特徴とする請求項６記載の情報処理方法。
【請求項８】　前記許可情報は、前記アプリケーションソフトウェアの代金に関して決済が終了している場合に実行可を示し、終了していない場合に実行不可を示すように構成されていることを特徴とする請求項７記載の情報処理方法。
【請求項９】　更に、前記ユーザ端末において、取得した許可情報が実行不可を示す場合、前記ユーザ端末と通信ネットワークを介して通信可能に構成された決済装置に決済処理を行うように通信を行う工程と、
前記管理手段において、前記決済装置より決済処理が終了したことを示す情報を受け取った場合に、許可情報について実行可を示すように変更する工程とを備えることを特徴とする請求項８記載の情報処理方法。
【請求項１０】　更に、前記管理手段において、前記アプリケーションソフトウェアを実行できる期間、及び／又は実行できる回数、及び／又は実行できる機能の範囲を記憶する工程を備えており、
前記許可情報は、前記アプリケーションソフトウェアに関して決済が終了しており、かつ前記条件が満たされている場合に、実行可を示すように構成されていることを特徴とする請求項９記載の情報処理方法。
【請求項１１】　ユーザ端末に実装される機能の実行要求を受け付ける工程と、前記機能を実行してよいかどうかを示す情報（以下、「許可情報」と呼ぶ。）を記憶する管理サーバーに問い合わせて前記機能の許可情報を取得する工程と、前記取得した許可情報に基づいて前記機能の実行可否を制御する工程とを備えることを特徴とする情報処理方法。
【請求項１２】　請求項６乃至１１のいずれか一項に記載の情報処理方法をコンピュータで実行させるためのプログラムを格納したコンピュータ読み取り可能な記録媒体。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、ユーザ端末に実装される機能、特に、ユーザ端末に実装されたソフトウェアによって実現される機能の実行を管理するための情報処理技術に関する。
【０００２】
【従来の技術】ソフトウェアの販売形態としては、ソフトウェア自体をインターネット、CD-ROMなどを通じてユーザに提供し、ユーザはそのソフトウェアを自己のユーザ端末に実装して利用する、いわゆる売り切り型の販売形態が一般的である。かかる販売形態には、販売時に代金を課金する店頭販売タイプと、ダウンロードしたりインストールした段階では代金を課金せず、継続使用する場合に代金を課金するシェアウェアタイプが存在する。
【０００３】シェアウェアタイプでは、ユーザにいった

Nintendo Exhibit 1003
Part 2 of 2
Page 00941

んソフトウェアを渡した後に代金を課金する必要があるため、例えば、無償で使用できる試用期間や、無償で使用できる機能に制限を設けておき、かかる制限を解除するパスワード等を代金と交換に渡すといった課金方法が取られる。

【０００４】一方、近年では、ソフトウェアをユーザに「売り切り」するのではなく、ソフトウェア自体はホストコンピュータに実装しておき、ユーザが通信ネットワークを介してソフトウェアの機能のみを利用するという機能提供型の販売形態も広がってきている。このような機能提供型としては、ＡＳＰ（アプリケーション・サービス・プロバイダー）と呼ばれる形態が広く知られている。

【０００５】

【発明が解決しようとする課題】ソフトウェアは、デジタルコンテンツという特性上、消耗劣化することなく容易に複製することが可能であるため、正当購入者以外に不正に実行されてしまうおそれがある。このような不正を防止するためには、ソフトウェアが実行されるたびに、実行者がソフトウェアを正当に購入した者であるかどうかを確認できることが望ましい。

【０００６】しかし、売り切り型では、販売された後やパスワード等が交付された後は、ソフトウェア自体はユーザの管理下に置かれ、いつ誰がソフトウェアを実行するかはユーザに委ねられることになる。別言すれば、ソフトウェアの実行をコントロールする権限が販売者の手を離れてしまうため、販売者が、ソフトウェアを実行する者が正当購入者であるかどうかを確認する術はない。

【０００７】更に、売りきり型では、ソフトウェアの実行を販売者がコントロールできない点で、事実上、販売＝ソフトウェアを将来にわたって実行する権利の譲渡となってしまう。そのため、実行回数や実行期間に制限を設けてソフトウェアを販売する、いわば「切り売り」といった販売形態を取ることは難しい。

【０００８】一方、機能提供型では、ソフトウェア自体はホストコンピュータに置かれていることから、販売者がソフトウェアの実行をコントロールすることができる。そのため、ソフトウェアが実行されるたびにユーザの正当性を確認することも可能である。

【０００９】しかし、機能提供型には以下のような問題がある。一つは、ソフトウェア実行時に常にユーザ端末とサーバーとの間で通信路を確保しておく必要があるため、接続にかかるコストも利用量に応じて増大してしまうという問題である。もう一つは、ソフトウェアの実行速度が通信ネットワークの混雑状況やホストコンピュータの処理能力によって影響を受けてしまい、常に一定の実行速度を保つことが困難であるという問題である。

【００１０】そこで、本発明は、接続コストの増大や実行速度の低下を招くことなく、ソフトウェアを実行する者が正当購入者であるかどうかを確認でき、更に、何回分、何日分というように、実行回数や実行期間に応じて柔軟に細かくソフトウェアを「切り売り」することができるソフトウェアの管理技術を提供することを目的とする。

【００１１】

【課題を解決するための手段】本発明の情報処理方法は、ユーザ端末に実装される機能の実行を制御する情報処理方法であって、前記ユーザ端末と通信ネットワークを介して通信可能に構成された管理サーバーにおいて、前記機能を実行してよいかどうかを示す情報（以下、「許可情報」と呼ぶ。）を記憶する工程と、前記ユーザ端末において、前記機能の実行要求を受け付ける工程と、前記管理手段に問い合わせて前記機能の許可情報を取得する工程と、前記取得した許可情報に基づいて前記機能の実行可否を制御する工程とを備えることを特徴とする。

【００１２】前記機能は、ユーザ端末に記憶されるアプリケーションソフトウェアによって実現される機能であることが望ましい。

【００１３】また、前記許可情報は、前記アプリケーションソフトウェアの代金に関して決済が終了している場合に実行可を示し、終了していない場合に実行不可を示すように構成されていることが望ましい。

【００１４】好適には、更に、前記ユーザ端末において、取得した許可情報が実行不可を示す場合、前記ユーザ端末と通信ネットワークを介して通信可能に構成された決済装置に決済処理を行うように通信を行う工程と、前記管理手段において、前記決済装置より決済処理が終了したことを示す情報を受け取った場合に、許可情報について実行可を示すように変更する工程とを備える。

【００１５】また好適には、更に、前記管理手段において、前記アプリケーションソフトウェアを実行できる期間、及び／又は実行できる回数、及び／又は実行できる機能の範囲を記憶する工程を備えており、前記許可情報は、前記アプリケーションソフトウェアに関して決済が終了しており、かつ前記条件が満たされている場合に、実行可を示すように構成されていることが望ましい。

【００１６】本発明の情報処理方法は、ユーザ端末に実装される機能の実行要求を受け付ける工程と、前記機能を実行してよいかどうかを示す情報（以下、「許可情報」と呼ぶ。）を記憶する管理サーバーに問い合わせて前記機能の許可情報を取得する工程と、前記取得した許可情報に基づいて前記機能の実行可否を制御する工程とを備えることを特徴とする。

【００１７】本発明の情報処理方法はコンピュータにより実施することができるが、そのためのコンピュータプログラムは、ＣＤ－ＲＯＭ、磁気ディスク、半導体メモリといった各種の記録媒体に記憶しておき、コンピュータにインストールまたはロードすることができる。

【００１８】本発明の情報処理システムは、ユーザ端末

Nintendo Exhibit 1003
Part 2 of 2
Page 00942

に実装される機能の実行を制御する情報処理システムであって、前記機能を実行してよいかどうかを示す情報（以下、「許可情報」と呼ぶ。）を管理する管理手段と、前記機能の実行要求を受けつけ、前記管理手段に問い合わせて前記機能の許可情報を取得し、前記取得した許可情報に基づいて前記機能の実行可否を制御する制御手段とを備えており、前記制御手段は前記ユーザ端末内に配置され、前記管理手段は前記ユーザ端末と通信ネットワークを介して通信可能に構成されたホストコンピュータ内に配置されていることを特徴とする。

【００１９】前記機能は、ユーザ端末に記憶されるアプリケーションソフトウェアによって実現される機能であることが望ましい。

【００２０】前記許可情報は、前記アプリケーションソフトウェアの代金に関して決済が終了している場合に実行可を示し、終了していない場合に実行不可を示すように構成されていることが望ましい。

【００２１】好適には、前記制御手段は、取得した許可情報が実行不可を示す場合、前記ユーザ端末と通信ネットワークを介して通信可能に構成された決済装置に決済処理を行うように通信を行い、前記管理手段は、前記決済装置より決済処理が終了したことを示す情報を受け取った場合に許可情報について実行可を示すように変更する。

【００２２】また好適には、前記管理手段は、更に、前記アプリケーションソフトウェアを実行できる期間、及び／又は実行できる回数、及び／又は実行できる機能の範囲に関する条件を記憶しており、前記許可情報は、前記アプリケーションソフトウェアに関して決済が終了しており、かつ前記条件が満たされている場合に、実行可を示すように構成されていることが望ましい。

【００２３】なお、本発明において、アプリケーションソフトウェアとは、ユーザ端末に実装されることで所定の機能を実現することができるソフトウェアを指し、シェアウェアなどを含む概念である。

【００２４】

【発明の実施の形態】（第１実施形態）以下に本発明の実施の形態について図面を用いて説明する。図１は、本発明の第１の実施形態に係る情報処理システムの全体構成をあらわすブロック図である。本システムは、ユーザ端末１０、管理サーバー１１、決済サーバー１２を含んで構成される。ユーザ端末１０、管理サーバー１１、決済サーバー１２は、互いに通信ネットワークを介して接続されており、それぞれ通信するために必要な通常の構成を備えている。例えば、インターネットを介して接続する場合であれば、ＰＰＰドライバ、ＴＣＰ／ＩＰドライバを有する通信モジュールを備える。なお、図１ではユーザ端末１を一つのみ図示しているが、実際には、ユーザ端末はユーザの数に応じて複数用意されることになる。

【００２５】管理サーバー１１は、アプリケーションソフトウェア（以下、「ＡＰ」と呼ぶ。）を実行してよいかどうかを示す情報を管理する。具体的には、各ユーザＩＤに対応づけて、パスワード、ＡＰ識別情報、決済情報、有効期間、有効残回数を記憶している。

【００２６】ＡＰ識別情報とは、ＡＰを識別するための情報であり、例えばアプリケーション名やＡＰに付された商品コード等を用いることができる。

【００２７】決済情報は、ユーザＩＤで特定されるユーザについて、ＡＰ識別情報で特定されるＡＰに関して決済が終了しているか否かを示す情報である。かかる情報は、決済サーバー１２からの入力によって更新される（後述）。

【００２８】有効期間はユーザがＡＰを実行できる期間を、有効残回数はユーザがＡＰを実行できる回数を示す情報である。これらの情報の初期値はＡＰの販売者によって決定され、管理サーバー１１に登録される。かかる初期値は、ＡＰの販売者が管理サーバー１１に直接登録できるように構成することが望ましいが、管理サーバー１１のの管理者等が代わりに登録するように構成してもよい。なお、有効回数は、ユーザがＡＰを実行するごとに－１カウント更新されることになる。

【００２９】管理サーバー１１に記憶されるデータの構造は、概念的には図２に示すような表形式となる。データの管理や検索には、リレーショナルデータベース等の従来のデータベース技術を用いることができる。

【００３０】決済サーバー１２は、ユーザに対し課金を行う決済処理を実行する。決済処理には、外部の決済サービスを利用することができる。この場合、各決済サービスにおける決済サーバーが、決済サーバー１２となる。

【００３１】外部の決済サービスとしては、いわゆるクレジットカード会社の決済サービス、通信料金の一部として課金し決済するサービス（ＮＴＴのダイヤルＱ２サービス、ＮＴＴコミュニケーションズの電子決済サービスであるＣａｌｌｅ（カルレ）等）、インターネットを利用したプリペイド課金サービス（ビットキャッシュ社のＢｉｔＣａｓｈ、ウェッブマネー社のＷｅｂＭｏｎｅｙ等）などが知られている。

【００３２】例えば、インターネットを利用したプリペイド課金サービスを利用する場合、決済サーバー１２は、決済対象となるＡＰについて、ＡＰ識別情報、代金情報を記憶する商品データベースと、プリペイドカード番号と残金情報を対応づけて記憶する残金データベースを備えている。ユーザ端末よりユーザＩＤ、ＡＰ識別情報、プリペイドカード番号を受け取った場合、プリペイドカード番号で特定される残金から、ＡＰ識別情報で特定されるＡＰの代金を減算して、決済処理を実行する。そして、ユーザＩＤで特定されるユーザに関し、決済処理が行われたことを示す情報を管理サーバー１１に出力

Nintendo Exhibit 1003
Part 2 of 2
Page 00943

する。

【００３３】ユーザ端末１０は、図１に示すように、ＡＰ記憶手段２０、ＡＰ制御手段２１、ＡＰ実行手段２２を備えている。本実施形態では、一つのＡＰ制御手段が複数のＡＰに対応可能に構成されている。以下、各手段の構成・動作について説明する。

【００３４】ＡＰ記憶手段２０は、ＡＰ識別情報に対応づけて各ＡＰを記憶している。ＡＰは、例えばＣＤ－ＲＯＭやフロッピディスク等の記録媒体に記録された状態でユーザに提供され、図示しない記録媒体読取手段によりユーザ端末１０に読み込まれ、ＡＰ記憶手段２０に記憶される。また、例えば、インターネットや衛星通信等の通信ネットワークを介してユーザに提供され、図示しない通信モジュールを通じてユーザ端末にダウンロードされ、ＡＰ記憶手段２０に記憶される。

【００３５】ＡＰ制御手段２１は、ユーザからＡＰの実行要求を受けつけ、管理サーバー１１に問い合わせて実行要求対象ＡＰの許可情報を取得し、前記取得した許可情報に基づいてＡＰの実行可否を制御する。詳細については後述する。

【００３６】ＡＰ実行手段２２は、ＡＰ制御手段２１から実行が許可されたＡＰをＡＰ記憶手段１０から読み出して実行する。

【００３７】以下、第１実施形態において、ＡＰの実行がどのようにして制御されるかを図３、図４、図５に示すフローチャートに沿って説明する。なお、以下の各ステップは処理内容に矛盾を生じない範囲で任意に順番を変更して実行することができる。

【００３８】（ＡＰ制御手段の処理フロー：図３）ＡＰ制御手段２１は、ユーザ端末の表示装置（図示せず）に、ＡＰ選択メニュー画面を出力する（ステップＳ１００）。ユーザは、かかるメニュー画面に基づいて実行を要求するＡＰを指定する。

【００３９】ＡＰ制御手段２１は、ユーザからの指定に基づいてＡＰを選択する（ステップＳ１０１）。そして、ユーザＩＤ入力画面を前記表示装置に出力し、ユーザからユーザＩＤ及びパスワードを受け付ける（ステップＳ１０２）。なお、ユーザＩＤがユーザ端末内に記憶されている場合は、その記憶されているユーザＩＤを読み出して用いるようにしてもよい。

【００４０】ＡＰ制御手段２１は、通信モジュールを介して管理サーバー１１との間に通信路を確保し、前記選択したＡＰのＡＰ識別情報、ユーザから入力されたユーザＩＤ及びパスワードを、管理サーバー１１に送信する（ステップＳ１０３）。これを受けて、管理サーバー１１では所定の処理に従って許可情報を、実行可、実行不可、パスワード不一致のいずれかを示すように構成し、ＡＰ制御手段２１に送信する。管理サーバー１１の処理内容については後述する。

【００４１】ＡＰ制御手段２１は、管理サーバー１１より許可情報を受信した場合（ステップＳ１０４）、許可情報の内容に応じて以下の処理を行う（ステップＳ１０５）。

【００４２】許可情報がパスワード不一致を示す場合、「パスワードが一致しません」等のメッセージをユーザ端末の表示装置に表示し、処理を終了するか、又はＳ１０２に戻る（ステップＳ１０６）。

【００４３】許可情報が実行可を示す場合、ＡＰ実行手段２２に前記選択したＡＰのＡＰ識別情報を出力する（ステップＳ１０７）。ＡＰ実行手段２２は、受け付けたＡＰ識別情報に対応するＡＰ、すなわち実行要求がだされたＡＰをＡＰ記憶手段１０から読み出し、実行する。これにより、実行要求がだされたＡＰによって達成される機能がユーザ端末において実現されることになる。

【００４４】許可情報が実行不可を示す場合、プリペイド番号入力画面を前記表示装置に出力し、ユーザからプリペイド番号を受け付ける（ステップＳ１０８）。そして、通信モジュールを介して決済サーバー１２との間に通信路を確保し、前記選択したＡＰのＡＰ識別情報、ユーザＩＤ、パスワード及びプリペイド識別番号を、決済サーバー１２に送信する（ステップＳ１０９）。これを受けて、決済サーバー１２では所定の処理に従って決済終了を示す情報を管理サーバー１１に、又は決済不可を示す情報をＡＰ制御手段２１に送信する。決済サーバー１２の処理内容については後述する。

【００４５】ＡＰ制御手段２１は、決済サーバー１２より決済不可を示す情報を受信した場合（ステップＳ１１０）、「残金が足りません」等のメッセージを前記表示装置に出力し、処理を終了するか、又はＳ１０８に戻る（ステップＳ１１１）。

【００４６】（管理サーバー１１の処理フロー：図４）管理サーバー１１は、ＡＰ制御手段２１より、ＡＰ識別情報、ユーザＩＤ及びパスワードを受け付けた場合（ステップＳ２００）、ユーザＩＤに対応づけて記憶されているパスワードを読み出し、受信したパスワードと比較する（ステップＳ２０１）。

【００４７】パスワードが一致する場合は、受信したＡＰ識別情報及びユーザＩＤに対応づけて記憶されている決済情報、有効期間、有効残回数を読み出し、以下の条件判断を行う（ステップＳ２０２）。

【００４８】決済情報が決済終了を示し、かつ実行要求時が有効期間内であり、かつ有効回数が１以上である場合、許可情報を実行可を示すように構成する。また、有効回数から１を減算し、記憶内容を更新する（ステップＳ２０３）。

【００４９】決済情報、有効期間、有効回数に関し、上記の条件のいずれかが満たされていない場合、許可情報を実行不可を示すように構成する（ステップＳ２０４）。

Nintendo Exhibit 1003
Part 2 of 2
Page 00944

【００５０】また、パスワードが一致しない場合は、許可情報をパスワード不一致を示すように構成する（ステップＳ２０５）。

【００５１】そして、構成した許可情報をＡＰ制御手段２１に送信する（ステップＳ２０６）。

【００５２】（決済サーバー１２の処理フロー：図５）決済サーバー１２は、ＡＰ制御手段２１より、ＡＰ識別情報、ユーザＩＤ、パスワード及びプリペイド識別番号を受信した場合（ステップＳ３００）、プリペイドカード番号に対応づけて記憶される残金と、ＡＰ識別情報に対応づけて記憶されるアプリケーション代金を読み出し、比較する（ステップＳ３０１）。

【００５３】残金の方が多い場合、残金から代金を減算することにより残金を更新し、ユーザＩＤ及びＡＰ識別情報と、決済が終了したことを示す情報を管理サーバー１１に送信する（ステップＳ３０２）。これを受けて、管理サーバー１１は、ユーザＩＤ及びＡＰ識別情報に対応づけて記憶される決済情報を、決済終了を示すように変更する。

【００５４】アプリケーション代金の方が多い場合、決済不可を示す情報をＡＰ制御手段２１に送信する（ステップＳ３０３）。

【００５５】本実施形態の構成によれば、管理サーバーにおいてアプリケーションの実行可否を示す情報（許可情報）を管理しておくとともに、ユーザ端末に管理サーバーに問い合わせてアプリケーションの実行を制御するＡＰ制御手段を設けたため、アプリケーション自体はユーザの管理下に置いておきながら、アプリケーションの実行をコントロールする権限についてはユーザから切り離して管理することが可能となる。

【００５６】かかる構成では、管理サーバーから許可情報を取得した後は、アプリケーション自体はユーザ端末において実行され、実行中はユーザ端末と管理サーバーを接続しておく必要はない。そのため、通信ネットワークや管理サーバー等のパフォーマンスの影響を受けずにユーザ端末においてアプリケーションを実行することができる。

【００５７】また、管理サーバにおいて、アプリケーションの決済が終了している場合に実行可を示すように許可情報を構成することで、アプリケーションを正当に購入した者のみが実行できるように管理することが可能となる。

【００５８】更に、管理サーバにおいて、アプリケーションの有効期間や有効残回数を記憶しておき、これに基づいて許可情報を構成することで、実行期間や実行回数に制限を設けてアプリケーションを販売する「切り売り」を行うことが可能となる。

【００５９】（変形例）図６は、本実施形態の変形例におけるユーザ端末の構成を示すブロック図である。第１実施形態では、一つのＡＰ制御手段が複数のＡＰに対応していたのに対し、本変形例では、ＡＰ制御手段が各ＡＰと１対１に対応している点で異なっている。すなわち、ＡＰ記憶手段２０に記憶される各ＡＰは、それぞれＡＰ制御プログラムとＡＰ本体プログラムから構成されており、ＡＰ制御手段はＡＰ制御プログラムによって機能的に実現される構成となっている。以下、第１実施形態と異なる構成・動作について説明する。

【００６０】制御手段２３は、ユーザからＡＰの実行要求を受けつけ、実行要求対象のＡＰをＡＰ記憶手段２０から読み出して実行する。ＡＰが起動されると、最初にＡＰ制御プログラムが実行され、ユーザ端末にＡＰ制御手段２１が機能的に実現される。

【００６１】ＡＰ制御手段２１は、第１実施形態と同様に、管理サーバー１１に問い合わせて実行要求対象ＡＰの許可情報を取得し、前記取得した許可情報に基づいてＡＰ本体プログラムの実行可否を制御する。

【００６２】本変形例における処理の流れは、原則として第１実施例と同様である。ただし、本変形例では、ＡＰ制御手段２１はＡＰと１対１に対応していることから、ＡＰ選択メニュー画面の出力は行われず、ＡＰ制御手段２１の処理はステップＳ１０２から実行されることになる。

【００６３】また、受信した許可情報が実行可を示す場合は（ステップＳ１０７）、制御手段２３によってＡＰ本体プログラムが続いて実行されることになる。これにより、実行要求がだされたＡＰ（ＡＰ本体）によって達成される機能がユーザ端末において実現される。

【００６４】かかる構成は、特にシェアウェアの実行制御に適している。シェアウェアの開発者は、自己が開発したシェアウェア本体のプログラムに、初期実行部分として本発明に係るＡＰ制御手段の機能を実現するプログラムを予め組み込んでシェアウェアを構成することにより、本発明によるソフトウェアの実行制御の枠組みを利用することができる。すなわち、パスワード等の交付後においてもシェアウェアを実行する権利をユーザから切り離してホストコンピュータにおいて管理することができ、また、実行期間や実行回数に制限を設けてシェアウェアを販売する「切り売り」を行うことが可能となる。

【００６５】（その他）本発明の他の実施形態として、情報処理プログラムを記録した記録媒体が考えられる。記録媒体としてはＣＤ－ＲＯＭ、磁気ディスク、半導体メモリその他の記録媒体を用いることができる。

【００６６】更に、本発明は、上記実施形態に限定されることなく種々に変形して適用することが可能である。

【００６７】例えば、管理サーバー１１において、更に、有効機能範囲を記憶しておくように構成してもよい。有効機能範囲は、ＡＰが複数の機能を有している場合に、それら複数の機能のうちユーザに対し実行が許可されている機能の範囲を示す情報である。このように構成することで、ＡＰ単位ではなくＡＰが有する機能単位

Nintendo Exhibit 1003
Part 2 of 2
Page 00945

で実行を制御することが可能となる。

【００６８】また、例えば、ユーザ端末に、所定のサーバー上に構築された仮想世界にアクセスするためのアクセス手段を備えるように構成してもよい。図７に２次元仮想世界の例を示す。かかる２次元仮想世界は一つの町を形成しており、町にはユーザが操作するキャラクター１００、他のキャラクター１０２、部屋（家）１０１、掲示板１０３などが配置されている。ユーザが操作するキャラクター１００は、ユーザがアクセス手段を介して行う入力に基づいて、町内で移動や会話等のアクションを取ることができる。

【００６９】ＡＰ記憶手段２０には、ユーザが操作するキャラクター１００が取ることのできるアクションのうち、課金を伴う（有料の）アクションを実現するためのＡＰが記憶されている。課金を伴うアクションとしては、例えば、特定の掲示板１０３を閲覧するアクション、他のキャラクター１０２とチャットをするアクション、又は仮想世界に配置された所定の部屋（家）に入るためにドア１０４を開けるアクション、などが考えられる。

【００７０】ユーザが、キャラクター１００に課金を伴うアクションを取らせようとした場合、アクセス手段を通じて、かかるアクションに対応するＡＰのＡＰ識別情報がＡＰ制御手段２１に渡される。そして、第１実施例のステップＳ１０２以降の処理が実行される。

【００７１】このように構成することで、仮想世界においてユーザが取ることができるアクション、特に課金を伴うアクションに対し、そのアクションの実行可否を本発明によって制御することが可能となる。

【００７２】

＊【発明の効果】本発明によれば、接続コストの増大や実行速度の低下を招くことなく、ソフトウェアを実行する者が正当購入者であるかどうかを確認でき、更に、何回分、何日分というように、実行回数や実行期間に応じて柔軟に細かくソフトウェアを「切り売り」することができるソフトウェアの管理技術を提供することができる。

【００７３】

【図面の簡単な説明】

【図１】　本発明の第１の実施形態である情報処理システムの構成を示すブロック図である。

【図２】　管理サーバーにおけるデータ構造を説明するための図である。

【図３】　ＡＰ制御手段における処理の流れを示すフローチャートである。

【図４】　管理サーバーにおける処理の流れを示すフローチャートである。

【図５】　決済サーバーにおける処理の流れを示すフローチャートである。

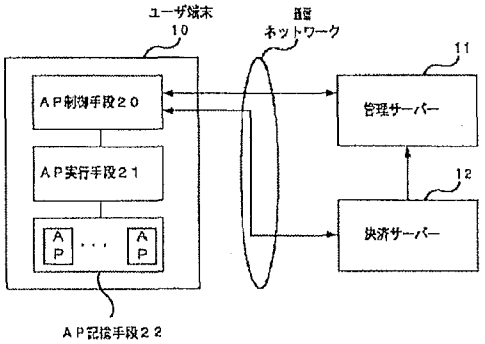
【図６】　第１の実施形態の変形例の構成を示すブロック図である。

【図７】　２次元仮想世界を説明するための図である。

【符号の説明】

１０　ユーザ端末
１１　管理サーバー
１２　決済サーバー
２０　ＡＰ記憶手段
２１　ＡＰ制御手段
２２　ＡＰ実行手段
２３　制御手段

【図１】

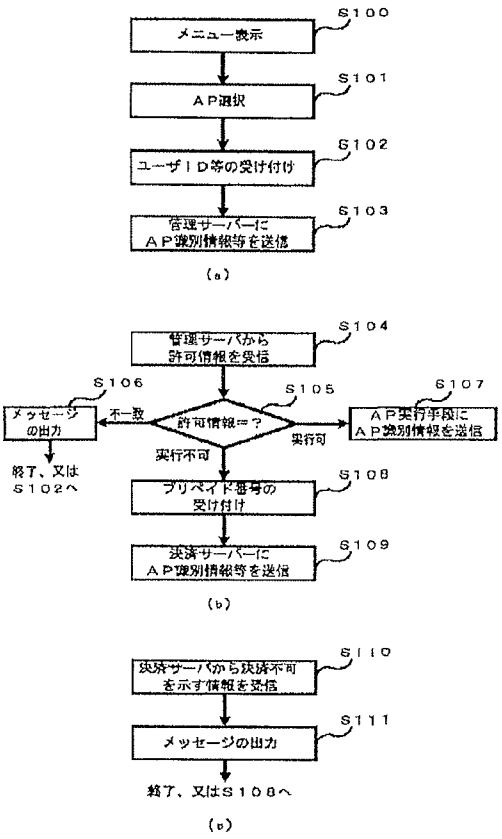


【図２】

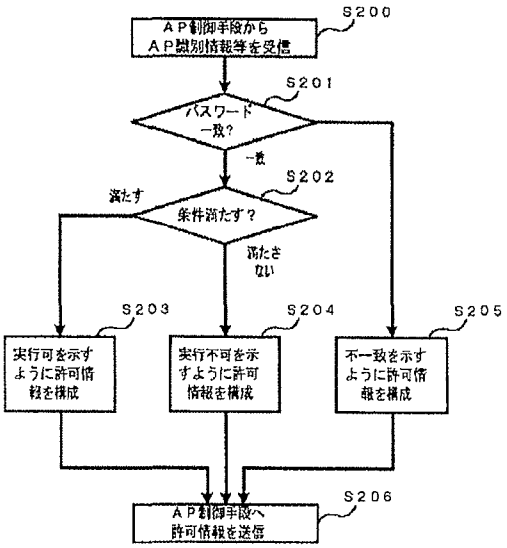
ユーザID	ＡＰ識別情報	決済情報	有効期間	有効残回数
111	表計算ＡＢＣ	有り	3/1～6/1	100
111	家計簿ＸＹＺ	無し	4/1～9/1	365
222	表計算ＡＢＣ	無し	4/1～7/1	150
・	・	・	・	・
・	・	・	・	・
・	・	・	・	・

Nintendo Exhibit 1003
Part 2 of 2
Page 00946

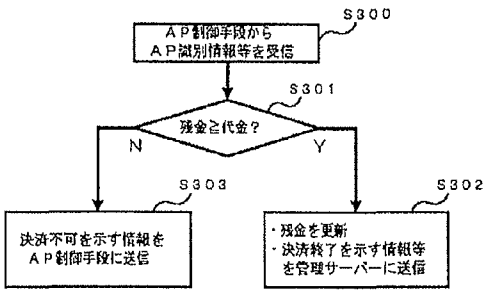
【図３】


S100
メニュー表示
S101
ＡＰ選択
S102
ユーザＩＤ等の受け付け
S103
管理サーバーに
ＡＰ識別情報等を送信
(a)
S104
管理サーバから
許可情報を受信
S105
許可情報＝？
S106
メッセージ
の出力
不一致
S107
ＡＰ実行手段に
ＡＰ識別情報を送信
実行不可
実行可
終了、又は
Ｓ１０２へ
S108
プリペイド番号の
受け付け
S109
決済サーバーに
ＡＰ識別情報等を送信
(b)
S110
決済サーバから決済不可
を示す情報を受信
S111
メッセージの出力
終了、又はＳ１０８へ
(c)

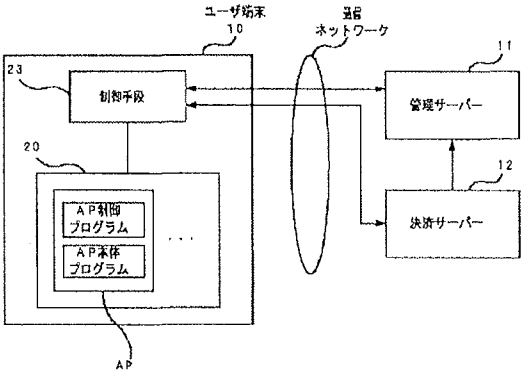
【図４】


S200
ＡＰ制御手段から
ＡＰ識別情報等を受信
S201
パスワード
一致?
一致
S202
条件満たす？
満たす
満たさ
ない
S203
実行可を示す
ように許可情
報を構成
S204
実行不可を示
すように許可
情報を構成
S205
不一致を示す
ように許可情
報を構成
S206
ＡＰ制御手段へ
許可情報を送信

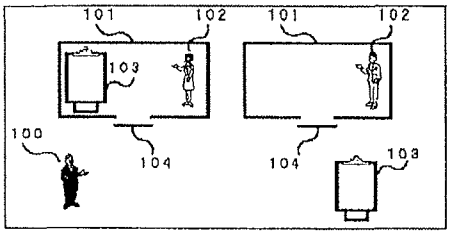
【図５】


S300
ＡＰ制御手段から
ＡＰ識別情報等を受信
S301
残金≧代金？
N
Y
S303
決済不可を示す情報を
ＡＰ制御手段に送信
S302
・残金を更新
・決済終了を示す情報等
を管理サーバーに送信

【図６】


ユーザ端末
10
通信
ネットワーク
11
23
制御手段
管理サーバー
20
12
ＡＰ制御
プログラム
ＡＰ本体
プログラム
決済サーバー
AP

【図７】


101
102
101
102
103
100
104
104
103

(12) UK Patent Application (19) GB (11) 2 355 819 (13) A

(43) Date of A Publication 02.05.2001

(21) Application No 9925320.5

(22) Date of Filing 26.10.1999

(71) Applicant(s)
Marconi Communications Limited
(Incorporated in the United Kingdom)
New Century Park, PO Box 53, COVENTRY, CV3 1HJ,
United Kingdom

(72) Inventor(s)
Icarus William John Sparry
Stuart Charles Wray

(74) Agent and/or Address for Service
Marconi Intellectual Property
Waterhouse Lane, CHELMSFORD, Essex, CM1 2QX,
United Kingdom

(51) INT CL[7]
G06F 1/00 12/14 , G07F 7/08 19/00 , H04L 9/00

(52) UK CL (Edition S)
G4A AAP
H4P PDCSP
U1S S2120 S2124 S2204 S2205 S2207 S2212 S2215

(56) Documents Cited
GB 2079109 A **EP 0387599 A2** **EP 0268141 A2**
WO 97/31450 A1

(58) Field of Search
UK CL (Edition R) **G4A AAP , H4P PDCSA PDCSC PDCSP**
INT CL[7] **G06F , G07F , H04L**

(54) Abstract Title
Authentication of data and software

(57) The invention provides a method of authenticating software and data in a computer network comprising a number of devices and servers (12, 18, 20) mutually communicating through a communication system. To improve security of the network, private keys and corresponding public keys are used, the public keys for decoding data and software encoded by corresponding private keys, thereby authenticating the data and software and distinguishing it from corrupt data and software provided by a hostile attacker. On initial supply of the device (12) by a supplying party to an operator of the network, the device (12) includes operating software (130) capable of interpreting security policy change instructions for revoking and substituting public keys initially programmed into the device (12), the instructions capable of being authenticated using public keys present in the device (12) prior to their revocation and substitution. Key revocation and substitution prevents the network from accepting software and data subsequently loaded thereinto by the supplying party if it becomes hostile. Alternatively, it relieves the party of suspicion if the network is subsequently disrupted by the hostile attacker.

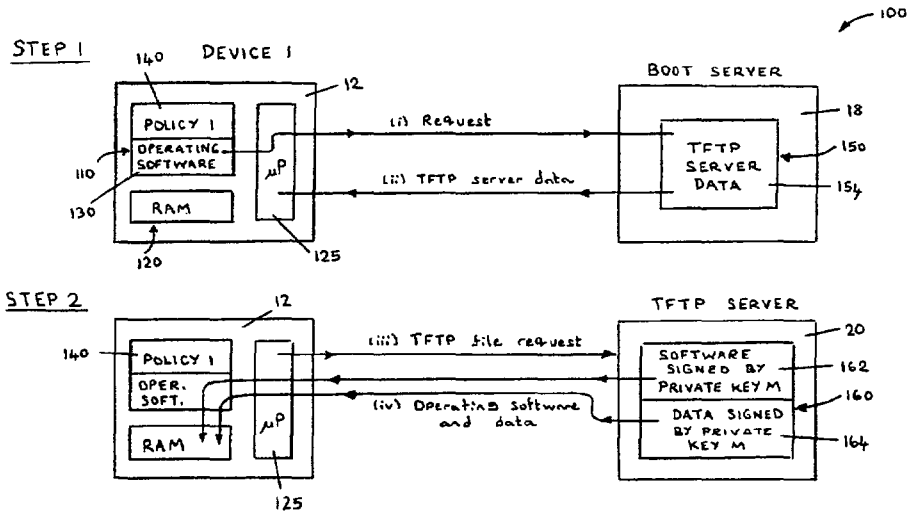
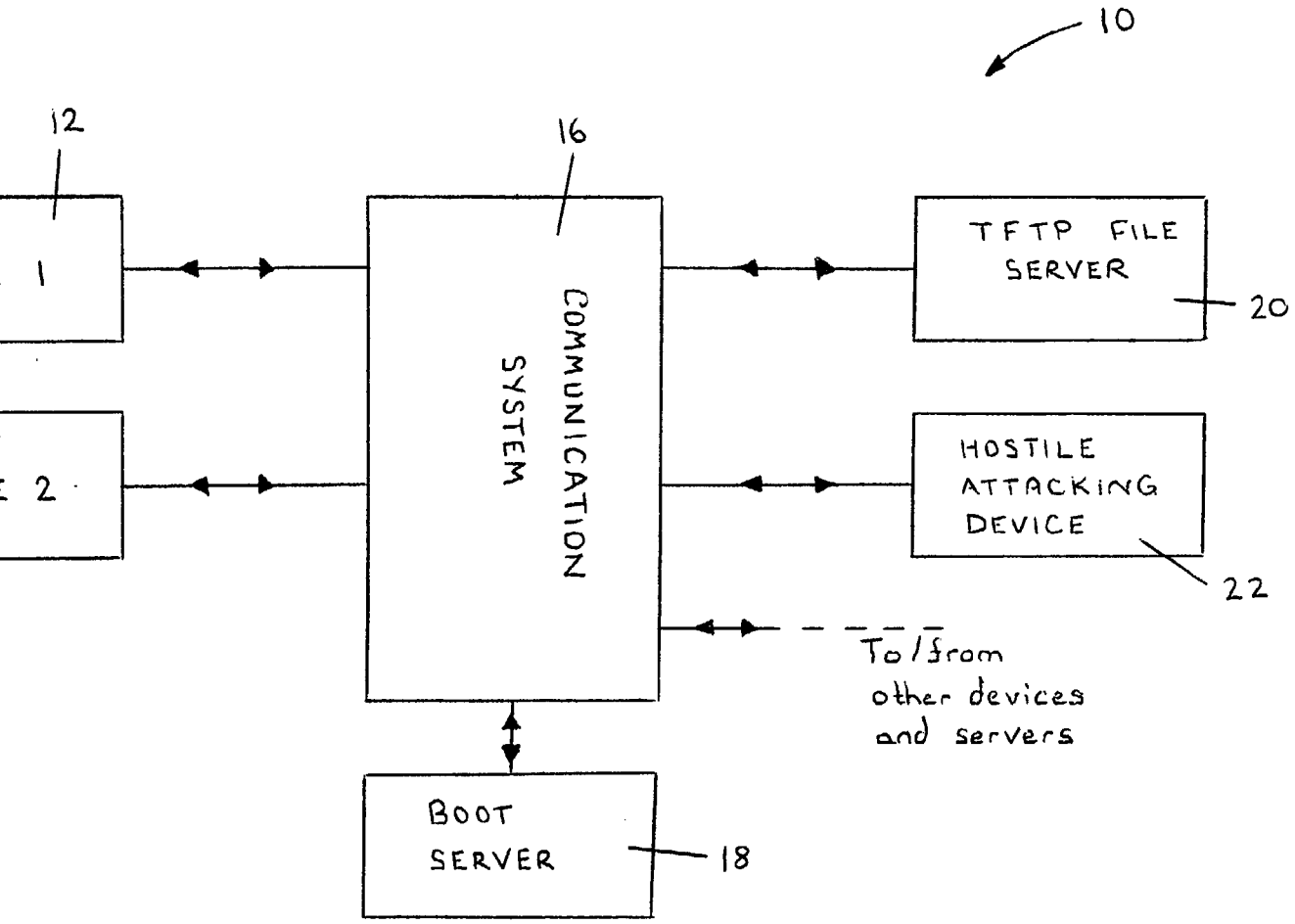


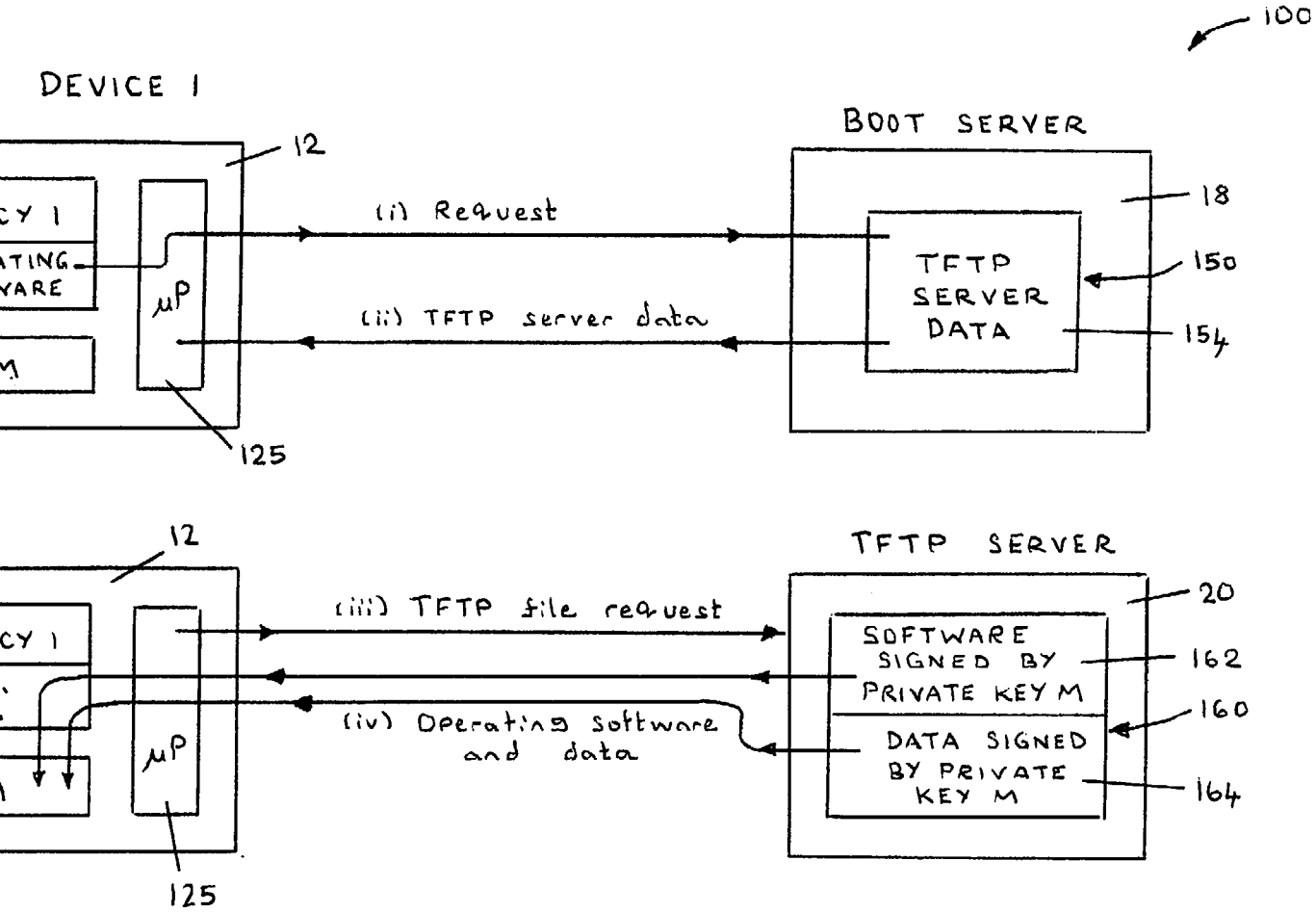
Fig. 2

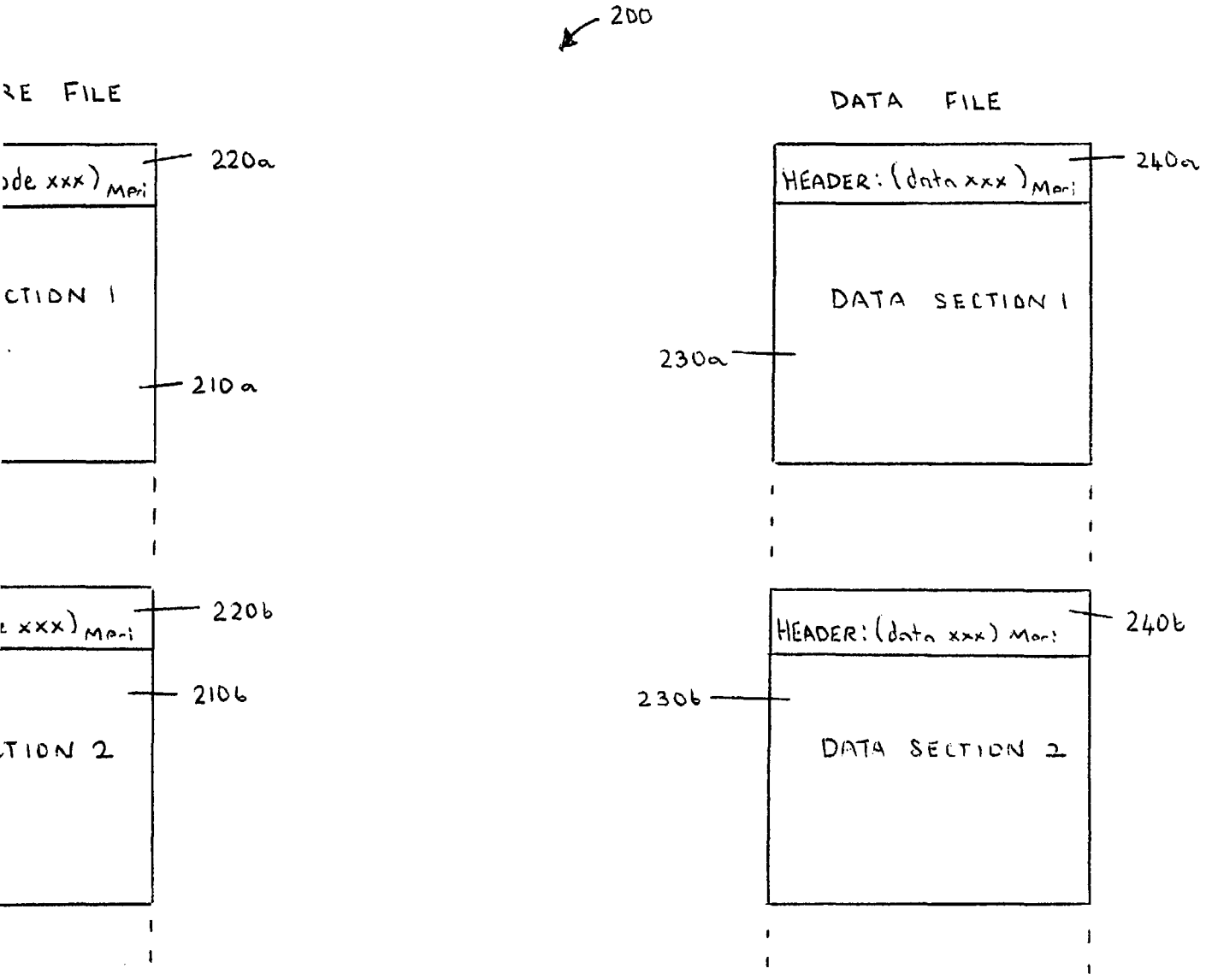
GB 2 355 819 A

Nintendo Exhibit 1003
Part 2 of 2
Page 00948


10
12
16
1
2
COMMUNICATION SYSTEM
TFTP FILE SERVER
20
HOSTILE ATTACKING DEVICE
22
To/from other devices and servers
BOOT SERVER
18

R ART)


100
DEVICE 1
12
μP
125
BOOT SERVER
18
TFTP SERVER DATA
150
154
(i) Request
(ii) TFTP server data
12
μP
125
TFTP SERVER
20
SOFTWARE SIGNED BY PRIVATE KEY M
162
160
DATA SIGNED BY PRIVATE KEY M
164
(iii) TFTP file request
(iv) Operating software and data


200
RE FILE
ode xxx) Mem
220a
CTION 1
210a
xxx) Mem
220b
2106
TION 2
DATA FILE
HEADER: (data xxx) Mem
240a
DATA SECTION 1
230a
HEADER: (data xxx) Mem
240b
2306
DATA SECTION 2

METHOD OF DATA AND SOFTWARE AUTHENTICATION

The present invention relates to a method of data and software authentication, in particular, but not exclusively, to a method of data and software authentication in a computer network. The invention also relates to a network operating according to the method.

A conventional computer network comprises a number of hardware devices, for example computers and associated devices such as printers and file servers, linked together by a communication system. The system may, for example, comprise optical fibre communication channels and telephone lines. Such a network is potentially vulnerable to attack from hostile parties attempting to gain access thereto. Reasons for attacking the network include an intention to cause disruption, to eavesdrop or to gain unpaid access to facilities available on the network, for example access to databases.

One simple conventional approach to resist attack from hostile parties is for supplying parties to provide only hardware to an operator responsible for a network. The operator can install his own software onto the hardware, the software being unfamiliar to both the supplying parties and the hostile parties. As a result of this unfamiliarity, the supplying parties and the hostile parties are thereby hindered from attacking the network.

However, this approach is unsatisfactory where the supplying parties not only provide hardware to the network operator but also associated software for operating the hardware. Contemporary computers and other network devices are frequently of such a complex nature that the operator must rely upon the supplying parties to supply software for

Nintendo Exhibit 1003
Part 2 of 2
Page 00952

operating their proprietary hardware. The simple approach described above is even more unsatisfactory where one supplying party supplies hardware to the network operator and a number of other supplying parties provide corresponding software and parameterising data for operating the hardware. There is a risk that defective software supplied by hostile parties becomes unintentionally loaded into the hardware thereby compromising its security and reliability.

A more elaborate conventional approach to improving security of the network is for each supplying party to generate a pair of complementary codes referred to as "keys". The codes are such that one code of the pair cannot reasonably be generated merely from knowledge of the other code of the pair. When each supplying party supplies software and associated data to the network operator, the supplying party "signs" the software and data using one of its keys, the supplying party retaining this key secret as its "private key" which it does not divulge it to the network operator. However, the supplying party divulges the other of its keys of the pair to the network operator, this being known as the supplying party's "public key". Thus, public keys provided by the supplying parties can be used by the network operator for verifying authenticity of software signed by private keys belonging to the supplying parties.

Generation of private and public keys is known from a US patent number 4 200 770 which is hereby incorporated by reference with regard to the generation of complementary keys. The private and public keys are usually of about 200 bytes in length and can be generated using a mathematical transformation described in the US patent.

Conventional "signing" of software and data by using private keys will now be described

Nintendo Exhibit 1003
Part 2 of 2
Page 00953

in further detail.

In a first approach, "signing" involves encoding software and data in its entirety using a private key to generate corresponding encoded software and data. The encoded software and data can be decoded using a corresponding public key to the private key. Other unrelated public keys will not be able to successfully decode the encoded software and hardware to generate viable software and data for use in the network. If the network operator discovers such unviable software and data when decoding, the operator will thereby determine that the software and data is not authentic and should not be loaded onto the network.

In a second approach, "signing" involves calculating a parity checksum for software and data using a checksum generation program known to both supplying parties and the network operator; known proprietary checksum generating programs include "Message Digest 5" (MD5) and "Secure Hash Algorithm" (SHA). The checksum is relatively short, usually in a range of 30 to 40 bytes in length. Each supplying party can use its private key for encoding a checksum generated using a public checksum generation program for its software and data and then supply the software and data unencoded together the encoded checksum, sometimes referred to as a Method Authentication Code (MAC), to the network operator. When the network operator receives the software and data from the supplying party, the operator can apply the public checksum generation program to generate an operator checksum for the unencoded software and data. The operator can then decode the encoded checksum provided by the supplying party using the supplying party's public key to generate a decoded checksum which the operator then compares with its operator checksum. If the decoded checksum is identical to the operator checksum,

Nintendo Exhibit 1003
Part 2 of 2
Page 00954

the network operator will thereby have determined that the software and data are authentic. If the decoded checksum and the operator checksum are not identical, the network operator will have identified that the software and data are suspect.

The second approach to "signing" software and data is faster and involves less computation than the first approach because checksum generation is a computationally quick operation to perform and encoding a checksum of 30 to 40 bytes is easier than encoding software and data in its entirety which can be, for example, several tens of Megabytes in size.

When the term "signing" is used by the inventors to describe their invention, this is intended to refer to either signing by the first approach or signing according to the second approach as described above.

A hostile party attempting to supply suspect software and data to the network operator will not have access to the private keys of bona fide supplying parties. The network operator will be able to detect that the suspect software and data are unauthentic because none of its public keys will be capable of authenticating the suspect software and data. This improves security of the network by preventing suspect software manipulating operation of the network in favour of a hostile party supplying the suspect software.

The more elaborate approach described above suffers a number of problems which can compromise security of the network.

A first problem arises when a supplying party supplies software and data to the network

Nintendo Exhibit 1003
Part 2 of 2
Page 00955

operator together with a public key for authenticating the software and data. The supplying party subsequently changes to become hostile but the network operator becomes immediately aware of this change. Such a change can arise, for example, by an employee of the supplying party deciding to cause malicious damage to his employer's interests. The network operator will need to take action to prevent the hostile supplying party from being able to load software and data onto the network which could compromise the network or cause damage thereto.

A second problem arises when a hostile party succeeds in obtaining illegally a private key from a bona fide supplying party and the network operator becomes immediately aware that security has been compromised by divulgation of the private key. The bona fide party wants to be free from suspicion and blame if the hostile party attempts to load software and data onto the network which wreaks costly damage to the network, for example corrupting a database of crucial financial records.

These problems are further complicated when new hardware is being added to an existing network and later upgrades of corresponding software from a number of suppliers are being installed onto the network.

The inventors have appreciated that conventional simple approaches to ensuring network security using complementary private and public keys are inadequate for addressing the problems described above because they do not provide a mechanism for updating public keys in the network for use in authenticating signed software and data. Thus, the inventors have devised a method of data and software authentication which addresses the aforesaid problems.

Nintendo Exhibit 1003
Part 2 of 2
Page 00956

According to a first aspect of the present invention, there is provided a method of data and software authentication in a network incorporating at least one device provided with at least one associated present security key for use in verifying data and software authenticity thereat, and key modifying means for issuing instructions to the at least one device for revoking its associated at least one present key and substituting it with at least one corresponding alternative security key, the method including the steps of:

(a) receiving instructions from the modifying means at the at least one device interpretable thereat for revoking its at least one present key and substituting it with at least one corresponding alternative key, the instructions being authenticable using one or more of the at least one present key; and

(b) designating the at least one alternative key as included in the at least one present key for authenticating software and data and repeating step (a) as appropriate for ensuring network security.

The invention provides the advantage that an operator of the network is capable of revoking earlier keys and replacing them with later keys using the earlier keys to authenticate revocation, thereby improving network security.

Advantageously in the method, each device is operable to authenticate signed software and data loaded thereinto using its at least one present security key. Such authentication reduces an opportunity for a hostile party to download corrupted or suspect software and associated parameterising data into the device. Headers in the signed software and data are preferably included for identifying keys that can be used for authenticating the software and data.

Nintendo Exhibit 1003
Part 2 of 2
Page 00957

Beneficially to improve security, complementary private and public key pairs are used in the method. Thus, preferably, the at least one security key and at least one alternative key are public keys, the public keys having associated therewith corresponding private security keys, each public key having a complementary private key, each private key substantially incapable of being derived merely from knowledge of its corresponding public key, the data and software in the network signable using one or more private key and authenticable at the at least one device using one or more corresponding public key. Use of private and public keys enables suppliers to sign their software and data using private keys, and for the at least one device in the network to verify authenticity of the software and data and decode them using corresponding public keys.

Preferably in the method, a subset of the at least one present key in each device is reserved for authenticating key revocation and substitution instructions only. The subset is not used for authenticating network software and data loaded into the device for execution therein. Such a subset, referred to as "master keys" in the following description, is capable of improving network security because its one or more keys are used infrequently on account of being reserved solely for authenticating key revocation and substitution instructions.

Conveniently, the instructions to revoke and substitute the at least one present key in the at least one device are issued from a file server included in the network, the file server operable to issue the instructions at initial boot-up of the at least one device. As a consequence, key change can be implemented promptly when new devices are connected to the network, thereby giving hostile parties less opportunity to interfere with the at least one devices.

Nintendo Exhibit 1003
Part 2 of 2
Page 00958

Preferably, the method is adapted such that a supplier of each device installs associated at least one present key thereinto prior to delivery of the device for incorporation into the network, and also supplies corresponding instructions for inclusion in the modifying means for revoking and substituting the at least one present key with at least one corresponding alternative key supplied by an operator of the network to the supplier for use in generating the instructions. Supplying each device with an associated at least one present key and instructions for its revocation and substitution with corresponding at least one alternative key provides the advantage that the network operator after revocation and substitution can sign software and data which the at least one device is capable of accepting, the supplier being relieved of suspicion if corrupt software becomes loaded and executed on the network.

Advantageously, the supplier is provided with at least one public key from the operator for generating the instructions, the supplier encoding the instructions using a private key generated by the supplier, the private key having a corresponding public key included by the supplier in the at least one device prior to delivery thereof for inclusion in the network. Such use of private and public keys provides the advantage that the operator can load the instructions into the modifying means so that the operator's public key is substituted into the at least one device thereby allowing software and data signed by the operator's private key to be executed on the at least one device.

Conveniently in the method, initial keys supplied in the at least one device when initially included in the network are stored in non-volatile memory, and subsequently substituted corresponding alternative keys are stored in volatile memory, the at least one device capable of reverting back to initial keys in an event of the at least one device being

Nintendo Exhibit 1003
Part 2 of 2
Page 00959

powered down and subsequently powered up again. Such operation provides the advantage that the at least one device need not to be returned to its supplier in an event of an erroneous key substitution being made.

In a second aspect, the invention provides a broadcast network operable according to a method of the first aspect of the invention.

In a third aspect, the invention provides a computer network operable according to a method of the first aspect of the invention.

In a fourth aspect, the invention provides a multiple application smart card operable according to a method of the first aspect of the invention.

Embodiments of the invention will now be described with reference to the following diagrams in which:

Figure 1 is a schematic diagram of a prior art computer network;

Figure 2 is a diagram illustrating software and data exchange in the computer network in Figure 1 operating to a method according to the invention; and

Figure 3 is an illustration of TFTP file structures including headers for interpretation according to the method of the invention.

Referring now to Figure 1, there is a shown a prior art computer network indicated by 10.

Nintendo Exhibit 1003
Part 2 of 2
Page 00960

The network 10 comprises a first network device 12, a second network device 14, a communication system 16, a boot file server 18 and a TFTP file server 20; TFTP is an abbreviation for "Trivial File Transfer Protocol." The devices 12, 14 and the servers 18, 20 are mutually linked through the system 16 which comprises a mesh of communication channels linking the network 10 together. Other devices and servers (not shown) are also connected to the system 16; connection to them is indicated by dotted lines from the system 16.

The first device 12 incorporates non-volatile read only memory, for example in the form of fusible ROM or EPROM, which is programmed by its supplying party at manufacturer to include simple operating software and one or more public keys. The device 12 also incorporates volatile random access memory (RAM) into which programs and data for execution in the device 12 can be downloaded through the communication system 16.

Operation of the prior art network 10 when the first device 12 is booted up will now be described with reference to Figure 1. Initially, the network 10 is active and the first device 12 is then powered up. The first device 12 initializes itself from its operating software contained within its non-volatile memory. The operating software causes the device 12 to emit a "boot-me" request and an identification code indicative of the device 12 type onto the system 16. The request is subsequently received by the boot server 18 which recognises the identification code. The boot server 18 then sends a response onto the system 16 which the device 12 subsequently receives, the response providing the address of the TFTP file server 20 which can supply the device 12 with network operating software and associated parameterising data. The device 12 then proceeds to download the network software and data from the TFTP server 20 into its volatile memory. The

Nintendo Exhibit 1003
Part 2 of 2
Page 00961

network software and data are provided to the device 12 in packets of digits. Finally, the device 12 proceeds to authenticate the network software and data as being bona fide using its public key. If the software and data are found by the device 12 to be corrupted, the device 12 requests retransmission of packets from the TFTP server 20, or, alternatively, re-boots itself. If the software and data are authentic with regard to the one or more public keys, the device 12 proceeds to execute the network software and thereby becomes a participating device in the network 10.

Network software is provided from the TFTP server 20 so that software upgrades can be more easily entered into the network 10; in practice, many devices in the network 10 can access the TFTP and boot servers 18, 20 hence it is more efficient to upgrade the software at these servers 18, 20 rather than at each individual device connected to the network 10.

In the network 10, servers and devices connected thereto can be geographically widely separated, for example in different countries or continents, hence data transmission errors can occasionally arise in practice, especially when the communication system uses error prone commercial telephone lines for example.

A hostile device 22 can interrupt operation of the device 12 in a number of ways. In a first way, when the device 12 emits its "boot-me" request, the hostile device 22 can bombard the device 12 with fake boot server responses before the boot server 18 has an opportunity to respond. The fake responses may, for example, instruct the device 12 to seek its network software from another server connected to the system 16 other than the server 20. In a second way, when the device 12 is receiving data in packets from the TFTP server 20, the hostile device 22 can present packets of software just prior to the server 20

Nintendo Exhibit 1003
Part 2 of 2
Page 00962

such that the device 12 is fooled into receiving software from the hostile device 22 and ignoring at least part of that provided from the server 20. It is part of permitted TFTP protocol that duplicated packets of information can be transmitted hence this represents a security weakness of the network 10.

If the device 12 is modified to comprise operating software contained in a non-volatile memory which is capable of being overwritten, for example in EEPROM, the attacking device 22 can attempt to disable the device 12 by inserting instructions into the non-volatile memory. This is serious because merely switching off the device 12 and then allowing it to reboot itself on reapplication of power will not correct such disruption to its operating software. In this scenario, the device 12 would need to be returned to its manufacturer for reprogramming which can incur disruption and expense.

The attacking device 22 can comprise a computer of the network 10 into which has been loaded corrupted software, for example software incorporating software viruses.

It is thus a crucial issue to the operator of the network 10 that he can take effective action once network security is known to have been compromised to prevent unauthorised devices and corrupt software from being used in the network 10. The invention is concerned with addressing this issue.

The inventors have appreciated that it is essential for the network operator to have a method by which public keys in te system can be revoked and substituted with new public keys in the event that network security is known to have been compromised.

Nintendo Exhibit 1003
Part 2 of 2
Page 00963

Referring now to Figure 2, there is illustrated steps of a method of data and software authentication according to the invention; steps of the method are indicated by 100. In Figure 2, the device 12 incorporates a non-volatile memory indicated by 110 and a random access memory (RAM) indicated by 120. The device 12 further includes a processor 125. The non-volatile memory 110 is implemented as an EEPROM, namely its contents can be altered under software control but are retained in an event of power being interrupted to the device 12.

In brief overview, the device 12 in Figure 2 is modified at manufacture to include a security policy represented as data 140 in the memory 110 together with simple operating software 130 capable of authenticating software and data downloaded from the TFTP server 20 into the RAM 120 of the device 12. The downloaded software and data are each partitioned into one or more sections, each section including an associated header declaring which public keys can be used for authenticating the section; such partitioning is illustrated in Figure 3 where software and data files are indicated by 200, the files 200 including code sections 210a, 210b each having corresponding headers 220a, 220b respectively and data sections 230a, 230b each having corresponding headers 240a, 240b respectively. The simple operating software 130 is operable to interpret the headers, for example the headers 220a, 220b, check that they correspond to the security policy 140 stored in the memory 110 and then act to authenticate sections 210a, 210b associated with the headers. If the headers 220a, 220b, 240a, 240b do not match with the security policy 140 or the sections fail to authenticate correctly using public keys declared in the headers, the simple operating software 130 determines there to be an error and raises an alarm or causes the device 12 to re-boot itself.

Nintendo Exhibit 1003
Part 2 of 2
Page 00964

Operation of the device 12 when booting-up in the network 10 will now be described in more detail with reference to Figure 2. The security policy 140 (Policy 1) incorporated into the memory 110 is used to authenticate software and data downloaded from the TFTP server 20. The policy 140 can be represented by a listing as follows:

Policy 1:

Public keys M_{pub}, MM_{pub}

$[code\ xxx]_{Mpri}$

$[data\ xxx]_{Mpri}$

$[change\ M_{pub}\ to\ <any\ key>]_{MMpri}$

In the policy 140, symbol "xxx" corresponds to any digit sequence, symbol "code" corresponds to executable code, "data" corresponds to data associated with executable software and "change" corresponds to revoking a present public key M_{pub} stored in the memory 110 and substituting it with an alternative specified public key, namely <any key>. A public key MM_{pub} is used for authenticating instructions invoking a change of the public key M_{pub} in the memory 110. Thus, "public keys M_{pub}, MM_{pub}" means that the manufacturer of the device 12 has programmed the public keys M_{pub} and MM_{pub} into the memory 110, the manufacturer retaining the corresponding private keys M_{pri} and MM_{pri} secret. Moreover, $[code\ xxx]_{Mpri}$ means that the device 12 is prepared to accept executable code from the TFTP server 20 provided that it can be authenticated against the public key M_{pub} or any key substituted for M_{pub} by virtue of the "change" option in the policy 140. Furthermore, $[data\ xxx]_{Mpub}$ means that the device 12 is prepared to accept data from the TFTP server 20 provided that it can be authenticated against the public key M_{pub} or any key substituted for M_{pub} by virtue of the "change" option in the policy 140. The private

Nintendo Exhibit 1003
Part 2 of 2
Page 00965

and public keys M_{pub} , Mpri and MM_{pub}, MM_{pri} are generated in a manner as described in the aforementioned US patent.

The boot server 18 incorporates a non-volatile memory 150, for example a read/write optical disc memory, onto which is stored file data 154 specifying the TFTP server 20 capable of supplying executable software and data for controlling the device 12 when fully interacting within the network 10.

The TFTP server 20 incorporates a non-volatile memory 160, for example a read/write hard magnetic disc memory, into which is stored executable software 162 together with associated network configuration data 164. The executable software 162 and the data 164 are supplied by the manufacturer of the device 12 and signed using the manufacturer's private key M_{pri}.

In Step 1 of the method 100 in Figure 2, power is applied to the device 12 to bring it into function. The processor 125 commences by executing the operating software 130 that firstly performs a self diagnosis of the device 12 to ensure its correct operation. The processor 125 then proceeds under control of the operating software 130 to emit a "boot-me" request (i) onto the system 16 which propagates therethrough and is subsequently received by the boot server 18. The boot server 18 responds to the request (i) by emitting TFTP server data (ii) specifying the server 20 which is received by the device 12.

The processor 125 under direction of the operating software 130 then proceeds to Step 2 of the method where it emits a TFTP file request (iii) to the system 16 which is subsequently received at the TFTP server 20. The server 20 responds by supplying the

Nintendo Exhibit 1003
Part 2 of 2
Page 00966

executable software 162 and the data 164 through the system 16 to the device 12. The processor 125 under control of the operating software 130 then isolates headers in the software 162 and the data 164, namely:

Headers 1:

$(\text{code xxx})_{Mpri}$

$(\text{data xxx})_{Mpri}$

and checks that they are consistent with the policy 140. If they are not consistent, the device 12 raises an alarm or re-boots itself. If the headers are consistent with the policy 140, the device proceeds to authenticate sections of code and data associated with the headers to ensure that they authenticate against the public key specified in the headers, namely public key M_{pub}. If the sections cannot be authenticated correctly, the device 12 again raises an alarm or re-boots itself.

If a hostile party gains access to the manufacturer's private key M_{pri} illegally, the hostile party can potentially generate and sign corrupt software and data which the network operator will find indistinguishable from bona fide software and data supplied by the manufacturer signed by its private key M_{pri}. However, if the network operator becomes aware of the hostile party illegally obtaining the private key M_{pri}, the network operator can take corrective action by approaching the manufacturer to generate a pair of new complementary keys, namely private and public keys N_{pri}, N_{pub} where the manufacturer retains the private key N_{pri} secret, and also to generate a token instruction which is capable

Nintendo Exhibit 1003
Part 2 of 2
Page 00967

of revoking the public key M_{pub} in the device 12 and substituting it with the new public key N_{pub}. The token instruction is signed using the manufacturer's private key MM_{pri} and is of the following form:

Token instruction 1:

(change M_{pub} N_{pub})$_{MMpri}$

The hostile party cannot generate such a token instruction without having knowledge of the manufacturer's private key MM_{pri} which the manufacturer keeps as a carefully guarded secret which is only invoked in the event of network security being compromised.

The network operator then receives the token instruction from the manufacturer and loads it via the boot server 18 or the TFTP server 20 onto the device 12. The operating software 130 of the device 12 proceeds to check the token against its policy and finds that it is an instruction corresponding to the policy and authenticable using the public key MM_{pub} initially programmed by the manufacturer in the device 12.

The software 130 then interprets the token instruction and substitutes its public key M_{pub} stored in the memory 110 with the new public key N_{pub} as provided by the token instruction. When this substitution is performed, the device 12 will thereafter not accept executable code and data signed by the private key M_{pri}, but only that signed by the private key N_{pri}. This substitution prevents the device 12 from accepting corrupt software that the corrupt party may attempt to load into the network 10 after the substitution has taken place, the hostile party only capable of providing software signed by the private key M_{pri}.

Nintendo Exhibit 1003
Part 2 of 2
Page 00968

The method 100 depicted and described with reference to Figure 2 provides a number of advantages compared to prior art methods of software and data authentication. The method 100 provides a mechanism whereby public keys can be revoked and substituted in the event of network security being compromised, instructions for revocation and substitution being signed using special private master keys, for example MM_{pri}, and being authenticable using corresponding special public master keys, for example MM_{pub}, already programmed into the device 12 during its manufacture.

If the hostile party gains knowledge of both the private keys M_{pri} and MM_{pri}, the network 10 becomes extremely vulnerable to hostile attack from the hostile party. Thus, it is important that the manufacturer keeps the private key MM_{pri} as a closely guarded secret and only uses the key when security of the network 10 has been compromised. The private key M_{pri} is likely to be used more frequently as it is applied when signing software and data, for example when the manufacturer provides regular upgrades of software and data for the network operator.

The policy 140 (Policy 1) can be further enhanced to revoke and substitute more than once a present public key used by the device 12 to authenticate software and data; each time a breach of security has been suspected or identified, token instructions can be interpreted repetitively by the operating software 130 to revoke and substitute public keys in the policy 140. To achieve this, the policy 140 (Policy 1) includes a policy statement:

[change <present key> to <any key>]$_{MMpri}$

The method of the invention can, for example, use a more complex new policy (Policy 2)

Nintendo Exhibit 1003
Part 2 of 2
Page 00969

instead of the policy 140 (Policy 1) to obtain different security characteristics, namely:

Policy 2:

Public keys Q_{pub}, C_{pub}, MQ_{pub}, MC_{pub}

[code xxx]$_{Qpri}$

[data xxx]$_{Cpri}$

[change Q_{pub} to <any key>]$_{MQpri}$

[change C to <any key>]$_{MCpri}$

In this more complex new policy (Policy 2), the network operator generates two set of complementary private and public keys C_{pri}, C_{pub} and MC_{pri}, MC_{pub} according to the aforementioned US patent, the operator retaining the private keys MC_{pri}, C_{pri} secret and divulging the public keys MC_{pub}, C_{pub} to the manufacturer of the device 12. Likewise, the manufacturer generates two sets of complementary private and public keys Q_{pri}, Q_{pub} and MQ_{pri}, MQ_{pub} according to the aforementioned US patent, the manufacturer retaining the private keys Q_{pri} , MQ_{pri} secret and divulging the public keys Q_{pub} , MQ_{pub} to the operator of the device 12, the manufacturer requiring the operator's public keys for generating the new policy for loading into the memory 110 of the device 12 and the manufacturer including its public keys Q_{pub}, MQ_{pub} so that the operator can authenticate software provided by the manufacturer and also authenticate key change tokens generated by the manufacturer or the network operator.

The manufacturer generates the new policy (Policy 2) and installs it into the device 12 during manufacturer. The new policy enables the device 12 to authenticate and execute software signed by the manufacturer using its private key Q_{pri}. The policy does not enable

Nintendo Exhibit 1003
Part 2 of 2
Page 00970

the device 12 to accept data provided by the manufacturer; the operator retains its private key C_{pri} for signing data which is accepted by the device 12. The new policy (Policy 2) thereby relieves the manufacturer of any responsibility for data used by the device 12. In the event of the operator's private key C_{pri} being accidentally divulged or illegally copied, the operator can use its private master key MC_{pri} to generate its own token instruction which the operating software 130 can authenticate and interpret to revoke the public key C_{pub} in the new policy (Policy 2) and substitute it with an alternative public key generated by the operator. Likewise, if the manufacturer's private key Q_{pri} is accidentally divulged or illegally copied, the manufacturer can generate for the operator a token instruction which the operating software 130 can authenticate and interpret to revoke the public key Q_{pub} and substitute it with an alterative public key generated by the manufacturer.

The new policy (Policy 2) provides the advantage that software and data from different sources can be authenticated by the policy.

Yet more complex alternative policies are feasible for authenticating software supplied by a number of different supplying parties, for example an advanced policy (Policy 3) is as follows:

Policy 3:

Public keys $R1_{pub}$, $R2_{pub}$, $R3_{pub}$, C_{pub}

$[\text{code xxx1}]_{R1pri}$

$[[\text{code xxx2}]_{R2pri}]_{Cpri}$

$[\text{code xxx3}]_{R3pri}$

$[\text{data xxx}]_{Cpri}$

Nintendo Exhibit 1003
Part 2 of 2
Page 00971

[change $R1_{pub}$ to <any key>]$_{R1pri}$

[change $R2_{pub}$ to <any key>]$_{R2pri}$

[change $R3_{pub}$ to <any key>]$_{R3pri}$

[change C to <any key>]$_{Cpri}$

This advanced policy enables the device 12 to authenticate software supplied and signed by any of three supply parties corresponding to R1, R2, R3, although for party R2, the operator must also sign the executable code [code xx2]$_{R2pri}$ for it to be acceptable to the device 12. Moreover, the alternative policy also enables revocation and substitution of public keys $R1_{pub}$, $R2_{pub}$, $R3_{pub}$ by tokens signed by their corresponding private keys. Furthermore, the alternatively policy (Policy 3) also enables the network operator to revoke and substitute its public key C_{pub} with an alternative public key. None of the operator and the parties corresponding to R1, R2, R3 need to divulge their private keys to one another, thereby ensuring security. In the alternative policy (Policy 3), there are no master keys included which are specifically reserved for key revocation and substitution as in the policy 140 (Policy 1).

More complex policies are feasible which enable the device 12 to be very selective regarding software and data which it is prepared to accept. Token instructions can include what is known mathematically as Regular Expressions, for example alternatives, repetitions and optional elements which only revoke and substitute keys when certain conditions are fulfilled.

In the network 10, it is possible to have identical policies in each of the devices 12, 14 or, alternatively, a number of different security policies customised for their respective

Nintendo Exhibit 1003
Part 2 of 2
Page 00972

devices 12, 14.

The method of the invention is distinguished from prior art in that key revocation is performed by token instructions which are interpreted by the operating software 130 loaded at manufacture into the device 12. An alternative approach when key security has been compromised is to load executable revocation software into the device 12 which simply overwrites the policy 140. Such an approach is inferior to the method of the invention described above because no authentication is carried out on the revocation software by the device. If the revocation software is supplied by a hostile party, the devices 12 is not provided with an opportunity of authenticating the software against a reliable key, for example MM_{pub}. In the device 12, the operating software 130 is configured to prevent the policy 140 being directly overwritten and only modifiable by way of token instructions which the operating software 130 must first authenticate and interpret before acting to modify the policy 140.

In some cases, manufacturers of devices such as the device 12 are not in a position to provide corresponding policy files for its customers; the customers can, for example, be situated in a different continents and their security requirements will not be known in advance at time of manufacture. To cope with this situation, devices similar to the device 12 can be manufactured with a very simple security policy loaded into non-volatile memory 110 thereof, namely (Policy 4):

Policy 4:

Public key X_{pub}

Nintendo Exhibit 1003
Part 2 of 2
Page 00973

$[code\ zzz]_{Xpri}$

The operating software 130 can be configured to interpret a public key X_{pub} as a special unique key such that any executable code signed by the private key X_{pri} and authenticable using the public key X_{pub} is allowed by the operating software 130 to overwrite the policy (Policy 4) in the memory 110 and substitute it with a replacement policy. Such a unique option enables the manufacturer to produce standard devices with the initial security policy (Policy 4) installed into memory which is subsequently reconfigured by software signed by the manufacturer to meet customers individual security requirements.

The key X_{pri} is thus a particularly valuable key which has to be guarded carefully because a hostile party gaining knowledge of the private key X_{pri} can corrupt security policies in devices such as the device 12. However, once the initial policy (Policy 4) is overwritten and replaced by a corresponding customized security policy, new keywords will be installed in the memory 110 which will prevent subsequent overwriting of the customized policy using code signed by the private key X_{pri}.

It will be appreciated by one skilled in the art that modifications can be made to the method of the invention without departing from the scope of the invention. Revocation and substitution of keys can be, for example, recorded in the random access memory 120. Substituted keys can override keys stored in the non-volatile memory 110 but operable not to delete them. If the device 12 malfunctions on account of corrupt software being downloaded into it or an incorrect key revocation and substitution occurring, the device 12 can be switched off and then power reapplied causing the operating software 130 which interprets policy changes to commence operation using its original security policy

Nintendo Exhibit 1003
Part 2 of 2
Page 00974

as recorded in the non-volatile memory 110. Such a method counteracts a need for the network operator to return the device 12 to its manufacturer in the event of a revocation and substitution error occurring.

Although the method of the invention is described in the context of the computer network 10, it can be applied to other types of system where unauthenticable software and data provided thereto are not to be accepted and executed, for example in set top boxes for receiving cable broadcasts from a cable network operator. The method of the invention enables software and data loaded into set-top boxes to be authenticated therein, thereby reassuring owners of the boxes that execution of the software and data supplied will not potentially cause malfunction and failure of their boxes. Likewise, the method also enables vendors supplying software and data to the set-top boxes to update public keys in the boxes in case their private keys are compromised and there is a risk of inferior fake software being loaded into customer's set-top boxes by hostile third parties.

Although application of the method of the invention to a set top box linked to a cable network is described above, the method is equally applicable to other types of network, for example radio networks and networks linked through satellites. The method is, for example, especially applicable to the Internet when loading new software and data from the Internet into a personal computer connected thereto.

The method of the invention can be further applied to multiple application smart cards, such cards giving access to multiple services from a plurality of vendors. Vendors conventionally supply their own proprietary cards to customers providing them with access to the vendors' services. Such a practice results in customers often acquiring

Nintendo Exhibit 1003
Part 2 of 2
Page 00975

numerous cards which can be inconvenient, namely one card for each vendor. It is therefore desirable for customers to have fewer cards, each card providing access to more than one vendor's services.

Such multiple application smart cards provide access to multiple facilities, for example financial credit facilities. Each smart card incorporates a random access memory, a read only memory and a processor. Moreover, the smart cards need to be updated from time to time as new vendor services become available. New software and data loaded into the smart cards needs to be checked to guarantee its authenticity otherwise, if defective software and data are loaded into the cards, there is a risk that the defective software causes an unwanted transaction to occur, for example transferring funds from a card owners bank account into a criminal bank account. Each smart card can therefore advantageously apply the method of the invention to authenticate new software and data being loaded into it and only run the software or use the data if it authenticates correctly against a security policy recorded within the card.

In the event of a vendor's private key being copied by a hostile third party and it being discovered that the hostile party is about to download defective software or data into customers' smart cards, the vendor can use the method of the invention to revoke and substitute public keys in the smart cards corresponding to the vendor so as to prevent the hostile third party from successfully downloading defective software or data into the smart cards.

The method of the invention is also applicable to other types of computer-based communication apparatus, for example mobile telephones and computer-based radios,

Nintendo Exhibit 1003
Part 2 of 2
Page 00976

where software upgrades to improve functionality of such apparatus are provided by associated vendors from time to time. Owners of the apparatus desire reassurance that software and data downloaded to their apparatus will not cause malfunction thereof even if the vendors have had their private key security comprised and have taken corrective action according to the invention by revoking and substituting public keys in the apparatus.

Nintendo Exhibit 1003
Part 2 of 2
Page 00977

CLAIMS

1. A method of data and software authentication in a network incorporating at least one device provided with at least one associated present security key for use in verifying data and software authenticity thereat, and key modifying means for issuing instructions to the at least one device for revoking its associated at least one present key and substituting it with at least one corresponding alternative security key, the method including the steps of:

(a) receiving instructions from the modifying means at the at least one device interpretable thereat for revoking its at least one present key and substituting it with at least one corresponding alternative key, the instructions being authenticable using one or more of the at least one present key; and

(b) designating the at least one alternative key as included in the at least one present key for authenticating software and data and repeating step (a) as appropriate for ensuring network security.

2. A method according to Claim 1 wherein the at least one security key and at least one alternative key are public keys, the public keys having associated therewith corresponding private security keys, each public key having a complementary private key, each private key substantially incapable of being derived merely from knowledge of its corresponding public key, the data and software in the network signable using one or more private keys and authenticable at the at least one device using one or more corresponding public keys.

Nintendo Exhibit 1003
Part 2 of 2
Page 00978

3. A method according to Claim 2 wherein the data and software include headers for identifying public keys which can be used for authenticating them.

4. A method according to Claim 1, 2 or 3 wherein a subset of the at least one present key in each device is reserved for authenticating key revocation and substitution instructions only.

5. A method according to any one of Claims 1 to 4 wherein the instructions to revoke and substitute the at least one present key in the at least one device are issued from a file server included in the network.

6. A method according to any one of Claims 1 to 5 wherein a supplier of each device installs associated at least one present key thereinto prior to delivery of the device for incorporation into the network, and also supplies corresponding instructions for inclusion in the modifying means for revoking and substituting the at least one present key with at least one corresponding alternative key supplied by an operator of the network to the supplier for use in generating the instructions.

7. A method according to Claim 6 wherein the supplier is provided with at least one public key from the operator for generating the instructions, the supplier encoding the instructions using a private key generated by the supplier, the private key having a corresponding public key included by the supplier in the at least one device prior to delivery thereof for inclusion in the network.

8. A method according to Claim 7 wherein interpretation of the instructions at the at

Nintendo Exhibit 1003
Part 2 of 2
Page 00979

least one device revokes at least one public key supplied by the supplier with at least one corresponding public key provided by the operator.

9. A method according to any preceding claims, wherein different keys are used in the at least one device for authenticating software and data provided by corresponding different suppliers to the network.

10. A method according to any preceding claim wherein the keys in the at least one device are stored in non-volatile memory therein.

11. A method according to any preceding claim wherein initial keys supplied in the at least one device when initially included in the network are stored in non-volatile memory, and subsequently substituted corresponding alternative keys are stored in volatile memory, the at least one device capable of reverting back to initial keys in an event of the at least one device being powered down and subsequently powered up again.

12. A broadcast network operable according to a method of any preceding claim for managing access to the network.

13. An multiple application smart card operable according to a method of any one or more of Claims 1 to 11.

14. A computer network operating according to a method of any one of Claims 1 to 10.

Nintendo Exhibit 1003
Part 2 of 2
Page 00980

15. A method substantially as herein before described with reference to one or more of Figures 1 to 3.

Nintendo Exhibit 1003
Part 2 of 2
Page 00981

INVESTOR IN PEOPLE

Application No: GB 9925320.5 **Examiner:** Mike Davis
Claims searched: 1-15 **Date of search:** 18 May 2000

Patents Act 1977
Search Report under Section 17

Databases searched:

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

UK Cl (Ed.R): G4A (AAP), H4P (PDCSP, PDCSA, PDCSC)

Int Cl (Ed.7): G06F, G07F, H04L

Other:

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
X	GB 2079109 A	(OAK) eg abstract	1 at least
X	EP 0387599 A2	(TANDEM COMPUTERS) eg abstract	"
X	EP 0268141 A2	(IBM) eg abstract and column 23 lines 37-48	"
X	WO 97/31450 A1	(VISA...) eg abstract	"

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

Nintendo Exhibit 1003
Part 2 of 2
Page 00982

PTO/SB/08a (01-10)
Approved for use through 07/31/2012. 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 contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	14715227
Filing Date	2015-05-18
First Named Inventor	Neil P. Adams
Art Unit	2143
Examiner Name	Unknown
Attorney Docket Number	11913-US-CNT3 4214-25714

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	5560008		1996-09-24	Johnson, et al.	
	2	5619710		1997-04-08	Travis, Jr., et al.	
	3	5778348		1998-07-07	Manduley, et al.	
	4	5944821		1999-08-31	Angelo	
	5	6061794		2000-05-09	Angelo, et al.	
	6	6101607		2000-08-08	Bachand, et al.	
	7	6134659		2000-10-17	Sprong, et al.	
	8	6134593		2000-10-17	Alexander, et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	14715227
	Filing Date	2015-05-18
	First Named Inventor	Neil P. Adams
	Art Unit	2143
	Examiner Name	Unknown
	Attorney Docket Number	11913-US-CNT3 4214-25714

	9	6141754		2000-10-31	Choy	
	10	6167521		2000-12-26	Smith, et al.	
	11	6192476	B1	2001-02-20	Gong	
	12	6202207	B1	2011-03-13	Donohue	
	13	6212635	B1	2001-04-03	Reardon	
	14	6216116	B1	2001-04-10	Barkan, et al.	
	15	6233608	B1	2001-05-15	Laursen, et al.	
	16	6480961	B2	2002-11-12	Rajasekharan, et al.	
	17	6487665	B1	2002-11-26	Andrew, et al.	
	18	6513121	B1	2003-01-28	Serkowski	
	19	6922782	B1	2005-07-26	Spyker, et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	14715227
	Filing Date	2015-05-18
	First Named Inventor	Neil P. Adams
	Art Unit	2143
	Examiner Name	Unknown
	Attorney Docket Number	11913-US-CNT3 4214-25714

	20	6931379	B1	2005-08-16	Sato, et al.	
	21	7546956	B2	2009-06-16	Adams, et al.	
	22	7793355	B2	2010-09-07	Little, et al.	
	23	7815100	B2	2010-10-19	Adams, et al.	
	24	8302185	B2	2012-10-30	Little, et al.	
	25	8429410	B2	2013-04-23	Little, et al.	
	26	8887988	B2	2014-11-18	Adams, et al.	
	27	9033216	B2	2015-05-19	Adams, et al.	

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20010002485	A1	2001-05-31	Bisbee, et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	14715227
	Filing Date	2015-05-18
	First Named Inventor	Neil P. Adams
	Art Unit	2143
	Examiner Name	Unknown
	Attorney Docket Number	11913-US-CNT3 4214-25714

	2	20010032188	A1	2001-10-18	Miyabe, et al.	
	3	20020007454	A1	2002-01-17	Tarpenning, et al.	
	4	20020152376	A1	2002-10-17	Eigeles	
	5	20030172034	A1	2003-09-11	Schneck, et al.	
	6	20030191719	A1	2003-10-09	Ginter, et al.	
	7	20040039911	A1	2004-02-26	Oka, et al.	
	8	20040078565	A1	2004-22-04	Hofmeister, et al.	
	9	20050050319	A1	2005-03-03	Suraski	
	10	20060272028	A1	2006-11-30	Maes	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	14715227
	Filing Date	2015-05-18
	First Named Inventor	Neil P. Adams
	Art Unit	2143
	Examiner Name	Unknown
	Attorney Docket Number	11913-US-CNT3 4214-25714

	1	0384610	EP	A2	1992-02-06	International Business Machines		☐
	2	2312767	GB	A	1997-11-05	Mitel Corporation		☐
	3	0828209	EP	A2	1998-03-11	Samsung Electronics, Ltd.		☐
	4	1998044404	WO	A1	1998-10-08	Sun Mircrosystems, Inc.		☐
	5	0060434	WO	A1	2000-10-12	Microsoft Corporation		☐
	6	2355819	GB	A	2001-05-02	Marconi Communications United		☐
	7	2001171590	WO	A1	2001-09-27	Vasara Research, Inc.		☐
	8	2001077811	WO	A1	2001-10-18	Everdream, Inc.		☐
	9	1168141	EP	A2	2002-01-02	Franklin Electronic Publishers, Inc.		☐
	10	2002056360	JP	A	2002-02-20	Hitachi, Ltd.		☒
	11	2002091598	JP	A	2002-03-29	Nippon Computer Co., Ltd.		☒

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	14715227
	Filing Date	2015-05-18
	First Named Inventor	Neil P. Adams
	Art Unit	2143
	Examiner Name	Unknown
	Attorney Docket Number	11913-US-CNT3 4214-25714

	12	2002170063	JP	A	2002-06-14	NTT Communications		☒
	13	2002182963	JP	A	2002-06-28	Ricoh Co., Ltd.		☒
	14	2002097620	WO	A2	2002-12-05	Qualcomm Incorporated		☐
	15	2825565	FR	A1	2002-12-06	Sagem		☒
	16	2378780	GB	A	2003-02-19	Elan Digital Systems Limited		☐
	17	2333865	GB	B	2003-05-14	International Business Machines Corp.		☐
	18	2003058411	WO	A1	2003-07-17	NTT Docomo, Inc.		☐
	19	2003063524	WO	A1	2003-07-31	Yahoo, Inc.		☐
	20	1185923	EP	B1	2004-03-31	Elata Limited		☐
	21	1427166	EP	A2	2004-06-09	Canon Kabushiki Kaisha		☐
	22	1465041	EP	A2	2004-10-06	NTT Docomo, Inc.		☐

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	14715227
	Filing Date	2015-05-18
	First Named Inventor	Neil P. Adams
	Art Unit	2143
	Examiner Name	Unknown
	Attorney Docket Number	11913-US-CNT3 4214-25714

	23	0813132	EP	B1	2005-01-26	International Business Machines Corporation		☐
	24	2429148	EP	A2	2012-03-14	NTT DoCoMo, Inc.		☐

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	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, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1	TAKARAGI; "Angou Houshiki to Ouyou [Encryption system and Application]"; Journal of Information Processing; Volume 32, No. 6; Japan: Information Processing Society of Japan; June 1991; 11 pages. (as referenced in Japanese Office Action dated February 10, 2009 for Application No. 2004-557718)	☐
	2	BOS, BERT; "Re: CSS: Extended Tiling. Proposal"; www-style@w3.org, http://lists.w3.org/Archives/Public/www-style/2004May/0020.html; May 5, 2004; 2 pages.	☐
	3	W3C; "CSS3 Module: Border"; http://www.w3.org/TR/2002/WD-css3-border-20021107/ November 7, 2002; 14 pages.	☐
	4	W3C; "CSS3 Backgrounds and Borders Module"; www.w3.org/TR/2005/WD-css3-background-20050216/; February 16, 2005; 30 pages.	☐
	5	SAMARATI, Pierangela, et al.; "Access Control: Policies, Models, and Mechanisms"; Foundations of Security Analysis and Design; Lecture Notes in Computer Science Volume 2171; 2001; 60 pages.	☐
	6	Microsoft; "Managing Authorization and Access Control"; http://technet.microsoft.com/en-us/library/bb457115.aspx; November 3, 2005; 84 pages.	☐
	7	CLARKSON, Michael; "Access Control"; http://www.cs.cornell.edu/courses/cs513/2007fa/NL.accessControl.html; 2007; 8 pages.	☐

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	14715227
	Filing Date	2015-05-18
	First Named Inventor	Neil P. Adams
	Art Unit	2143
	Examiner Name	Unknown
	Attorney Docket Number	11913-US-CNT3 4214-25714

	8	Wikipedia; "Access Control List"; http://en.wikipedia.org/wiki/Access_control_list; retrieved July 2, 2013; 3 pages.	☐
	9	Wikipedia; "Access Control List"; http://en.wikipedia.org/wiki/Access_control_list; retrieved June 5, 2013; 1 page.	☐
	10	Office Action dated August 21, 2007; U.S. Application No. 10/784,024 filed February 20, 2004; 21 pages. 10802-US-PAT	☐
	11	Office Action dated June 11, 2007; U.S. Application No. 10/732,132 filed December 10, 2003; 13 pages. (4214-27101)	☐
	12	Office Action dated February 22, 2008; U.S. Application No. 10/732,132 filed December 10, 2003; 11 pages. (4214-27101)	☐
	13	Final Office Action dated September 23, 2008; U.S. Application No. 10/732,132 filed December 10, 2003; 9 pages. (4214-27101)	☐
	14	Advisory Action dated December 8, 2008; U.S. Application No. 10/732,132 filed December 10, 2003; 3 pages. (4214-27101)	☐
	15	Office Action dated April 2, 2009; U.S. Application No. 10/732,132 filed December 10, 2003; 10 pages. (4214-27101)	☐
	16	Final Office Action dated October 28, 2009; U.S. Application No. 10/732,132 filed December 10, 2003; 10 pages. (4214-27101)	☐
	17	Advisory Action dated January 19, 2010; U.S. Application No. 10/732,132 filed December 10, 2003; 3 pages. (4214-27101)	☐
	18	Notice of Allowance dated May 4, 2010; U.S. Application No. 10/732,132 filed December 10, 2003; 7 pages. (4214-27101)	☐

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	14715227
	Filing Date	2015-05-18
	First Named Inventor	Neil P. Adams
	Art Unit	2143
	Examiner Name	Unknown
	Attorney Docket Number	11913-US-CNT3 4214-25714

	19	Office Action dated August 3, 2011; U.S. Application No. 12/869,589 filed August 26, 2010; 20 pages. (4214-27115)	☐
	20	Office Action dated January 19, 2012; U.S. Application No. 12/869,589 filed August 26, 2010; 11 pages. (4214-27115)	☐
	21	Notice of Allowance dated June 13, 2012; U.S. Application No. 12/869,589 filed August 26, 2010; 9 pages. (4214-27115)	☐
	22	Office Action dated June 17, 2013; U.S. Application No. 13/606,814 filed September 7, 2012; 19 pages. (4214-27116)	☐
	23	Final Office Action dated October 10, 2013; U.S. Application No. 13/606,814 filed September 7, 2012; 6 pages. (4214-27116)	☐
	24	Office Action dated May 29, 2008; U.S. Application No. 11/118,844 filed April 29, 2005; 4 pages. (4214-25701)	☐
	25	Office Action dated March 10, 2009; U.S. Application No. 11/118,844 filed April 29, 2005; 6 pages. (4214-25701)	☐
	26	Office Action dated August 20, 2009; U.S. Application No. 11/118,844 filed April 29, 2005; 15 pages. (4214-25701)	☐
	27	Notice of Allowance dated February 23, 2010; U.S. Application No. 11/118,844 filed April 29, 2005; 7 pages. (4214-25701)	☐
	28	Notice of Allowance dated June 10, 2010; U.S. Application No. 11/118,844 filed April 29, 2005; 8 pages. (4214-25701)	☐
	29	Office Action dated April 7, 2008; U.S. Application No. 11/118,748 filed April 29, 2005; 11 pages. (4214-25702)	☐

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	14715227
	Filing Date	2015-05-18
	First Named Inventor	Neil P. Adams
	Art Unit	2143
	Examiner Name	Unknown
	Attorney Docket Number	11913-US-CNT3 4214-25714

	30	Final Office Action dated October 27, 2008; U.S. Application No. 11/118,748 filed April 29, 2005; 10 pages. (4214-25702)	☐
	31	Notice of Allowance dated February 9, 2009; U.S. Application No. 11/118,748 filed April 29, 2005; 6 pages. (4214-25702)	☐
	32	Office Action dated March 22, 2010; U.S. Application No. 12/468,441 filed May 19, 2009; 12 pages. (4214-25703)	☐
	33	Notice of Allowance dated August 11, 2010; U.S. Application No. 12/468,441 filed May 19, 2009; 12 pages. (4214-25703)	☐
	34	Office Action dated November 12, 2013; U.S. Application No. 12/885,281 filed September 17, 2010; 32 pages. (4214-25712)	☐
	35	Office Action dated April 3, 2014; U.S. Application No. 12/885,281 filed September 17, 2010; 9 pages. (4214-25712)	☐

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

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

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

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	14715227
Filing Date	2015-05-18
First Named Inventor	Neil P. Adams
Art Unit	2143
Examiner Name	Unknown
Attorney Docket Number	11913-US-CNT3 4214-25714

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☐ The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/J. Robert Brown, Jr./	Date (YYYY-MM-DD)	2015-06-04
Name/Print	J. Robert Brown, Jr.	Registration Number	45438

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Espacenet

Bibliographic data: FR2825565 (A1) — 2002-12-06

Method for activating a function in a terminal subscribed to a network

No documents available for this priority number.

Inventor(s): CHABANNE HERVE ± (CHABANNE HERVE)

Applicant(s): SAGEM [FR] ± (SAGEM SA)

Classification: **- international:** ***H04W8/24;*** *H04W88/02;* (IPC1-7): H04Q7/32
- cooperative: **H04W8/245;** H04W88/02

Application number: FR20020002207 20020221

Priority number(s): FR20020002207 20020221 ; FR20010007255 20010601

Also published as: FR2825565 (B1) EP1263248 (A1)

Abstract not available for FR2825565 (A1)
Abstract of corresponding document: EP1263248 (A1)

The system includes transmission of a special message to a subscriber terminal, such as a mobile telephone, in order to activate a specific function within the terminal. This may be the locking or unlocking of the SIM card within a telephone, by the transmission of a message which is coded using a code known only to the network operator, and verified using a public code. In order to activate a logic function in the subscriber terminal , the network operator transmits to the terminal a special message containing information necessary for activation of the logic function, the message being individualised for the terminal. The logic function is only activated when the terminal has verified that the special message has been individualised for the specific terminal. The special message is signed under a key which is private to the operator of the network, in order to verify the operation using a public key. The special message includes the following elements: a unique public identifier; a unique private identifier encoded in a way only known to the operator of the network; and a unique physical private identifier held in memory in the terminal. The message may also carry information regarding the state of locking of the terminal.

Last updated: 13.03.2013 Worldwide Database 5.8.11.1; 92p

Nintendo Exhibit 1003
Part 2 of 2
Page 00995

(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) **N° de publication :** **2 825 565**

(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **02 02207**

(51) Int Cl[7] : **H 04 Q 7/32**

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 21.02.02.

(30) **Priorité :** 01.06.01 FR 00107255.

(43) **Date de mise à la disposition du public de la demande :** 06.12.02 Bulletin 02/49.

(56) **Liste des documents cités dans le rapport de recherche préliminaire :** *Se reporter à la fin du présent fascicule*

(60) **Références à d'autres documents nationaux apparentés :**

(71) **Demandeur(s) :** *SAGEM SA Société anonyme* — FR.

(72) **Inventeur(s) :** CHABANNE HERVE.

(73) **Titulaire(s) :**

(74) **Mandataire(s) :** CABINET PLASSERAUD.

(54) PROCEDE D'ACTIVATION D'UNE FONCTION LOGIQUE DANS UN TERMINAL ABONNE A UN RESEAU.

(57) Procédé pour activer une fonction logique dans un terminal abonné à un réseau, caractérisé en ce que l'opérateur du réseau émet, à destination du terminal, au moins un message spécial contenant les informations utiles à l'activation de la fonction logique dans le terminal, ledit message étant individualisé pour ledit terminal.

FR 2 825 565 - A1

2825565

PROCEDE D'ACTIVATION D'UNE FONCTION LOGIQUE DANS UN TERMINAL ABONNE A UN RESEAU

La présente invention concerne des perfectionnements apportés au processus d'activation d'une fonction logique dans un terminal abonné à un réseau et notamment, bien que non limitativement, elle concerne des perfectionnements apportés, dans un réseau de communication sans fil, à l'activation d'une fonction logique dans un terminal mobile détenu par un abonné dudit réseau.

Actuellement, un terminal mobile peut être livré à l'abonné avec une ou plusieurs fonctions verrouillées (c'est-à-dire inutilisables) en relation par exemple avec des clauses contractuelles de l'abonnement liant l'abonné à l'opérateur du réseau ; mais ces fonctions initialement verrouillées peuvent devoir être déverrouillées ultérieurement (par exemple clause d'abonnement à action limitée dans le temps ; modification de l'abonnement ; etc…).

Toutefois, il convient que le déverrouillage de la fonction souhaitée ne puisse intervenir que pour un abonné déterminé et qu'avec l'accord de l'opérateur du réseau, et il convient d'empêcher des déverrouillages frauduleux.

C'est le cas en particulier pour le verrouillage dit "simlocking" liant un abonné à un opérateur de réseau donné.

Les opérateurs de réseaux de téléphonie mobile investissent dans l'équipement en terminaux en subventionnant une partie de leur coût. En contrepartie, au début de son utilisation, le terminal mobile est verrouillé ("simlocking") sur le réseau de l'opérateur spécifique, c'est-à-dire qu'il ne peut fonctionner que sur le réseau de l'opérateur qui l'a subventionné pour l'abonnement dans lequel il était inclus. Toutefois, ce

Nintendo Exhibit 1003
Part 2 of 2
Page 00997

verrouillage n'est prévu que pour une durée déterminée (par exemple 6 mois ou 1 an) et, à la fin de cette période, l'abonné peut demander à l'opérateur que son terminal mobile soit déverrouillé.

Aujourd'hui le verrouillage de sécurité ("simlocking") est mis en œuvre dans le terminal mobile et, pour des raisons de coût, il s'agit en général d'un verrouillage essentiellement logique, c'est-à-dire faisant intervenir un logiciel ou agissant sur un logiciel, et de façon simplifiée, on peut considérer que le verrouillage du terminal mobile est obtenu par l'introduction, par l'opérateur du réseau, d'une information dérivée d'un code dans le terminal mobile lors de la mise en service de celui-ci.

Le moment venu, le déverrouillage est alors effectué par l'abonné lui-même qui introduit dans son terminal le code qui lui a été communiqué par l'opérateur du réseau lorsque l'abonné a fait part de son désir de déverrouillage. Le terminal mobile compare le code introduit par l'abonné avec l'information qu'il détient de façon secrète depuis sa mise en service et commande le déverrouillage donnant accès à d'autres réseaux.

Ce processus présente l'inconvénient de laisser à des fraudeurs la possibilité de remonter ("reverse engineering") jusqu'à l'information tenue en mémoire dans le terminal mobile et parfois d'en déduire le code devant être introduit pour provoquer le déverrouillage, ou parfois aussi de remplacer/modifier l'information mémorisée pour la faire correspondre avec un code choisi par le fraudeur. Le terminal mobile se trouve alors déverrouillé sans l'autorisation de l'opérateur du réseau.

Inversement, il peut être souhaité de verrouiller tout ou partie des fonctions d'un terminal mobile

jusqu'alors actives, par exemple afin de neutraliser un terminal volé. Par analogie avec ce qui précède, cette neutralisation peut être faite en verrouillant ("simlocking") le terminal volé sur un réseau inexistant.

Plus généralement, dans le domaine informatique, il peut être souhaité d'activer une fonction logique dans un terminal informatique, par exemple pour autoriser ou interdire une connexion à un serveur ou sur un site Internet dans le cadre d'un abonnement.

L'invention a donc pour but de proposer un procédé d'activation d'une fonction logique perfectionné de manière à empêcher une activation frauduleuse de la fonction logique.

A cette fin, l'invention propose un procédé pour activer une fonction logique dans un terminal abonné à un réseau, lequel procédé se caractérise, conformément à l'invention, en ce que l'opérateur du réseau émet, à destination du terminal, au moins un message spécial contenant les informations utiles à l'activation de la fonction logique dans le terminal, ledit message spécial étant individualisé pour ledit terminal.

Ainsi, conformément à l'invention, toutes les informations nécessaires à l'activation de la fonction logique ne se trouvent plus présentes dans le terminal mobile dès la mise en service de celui-ci et il ne suffit plus de rentrer le code communiqué par l'opérateur du réseau pour commander l'activation de cette fonction : on accroît la sécurité en prévoyant l'introduction, dans le terminal, des informations - données et/ou programme - nécessaires à l'activation de la fonction uniquement au moment où celle-ci est souhaitée. Cette introduction s'effectue à distance par l'envoi d'un message spécial approprié au terminal.

Notamment, on peut prévoir que le susdit code soit communiqué au préalable à l'utilisateur du terminal par l'opérateur du réseau et que l'utilisateur l'introduise dans son terminal afin que ce dernier puisse lire le susdit message spécial lorsqu'il le reçoit de l'opérateur.

Mais, on peut aussi prévoir que le code ou une information équivalente soit contenu dans ledit message spécial : l'abonné n'a alors aucune autre démarche à effectuer que de demander à l'opérateur du réseau l'activation de la fonction logique dans son terminal ou bien, dans le cas d'un terminal devant être neutralisé (cas d'un terminal volé par exemple), le message spécial est émis par l'opérateur à sa propre initiative pour neutraliser ledit terminal.

Dans les deux cas, c'est la double introduction, quasi simultanée ou simultanée, de deux types d'informations dans le terminal - d'un côté, le code qui est introduit par l'abonné qui le reçoit de l'opérateur du réseau ou bien qui est contenu dans le message spécial et, de l'autre côté, les informations contenues dans le message spécial envoyé par l'opérateur du réseau directement au terminal - qui valide l'activation de la fonction logique. Cette activation résulte ainsi de la volonté clairement exprimée de l'opérateur du réseau d'activer une fonction dans un terminal donné. Par exemple, dans le cas du verrouillage "simlocking" d'un terminal mobile, l'opérateur envoie dans le message spécial le nouvel état de verrouillage, cet état pouvant être par exemple "terminal déverrouillé" ou, par exemple encore, "terminal verrouillé sur un réseau inexistant" pour neutraliser un terminal volé.

D'une façon intéressante, les informations utiles à l'activation de la fonction logique peuvent comprendre

Nintendo Exhibit 1003
Part 2 of 2
Page 001000

des données et/ou, de façon encore plus sécuritaire, au moins un programme d'activation de ladite fonction. On est ainsi assuré que, quelles que soient les tentatives faites par un fraudeur avant que ces informations soient transmises au terminal par le message spécial, la fonction logique ne pourra pas être activée puisque les éléments logiciels nécessaires à cette activation ne sont pas présents dans le terminal.

Pour éviter qu'un message spécial destiné à un terminal donné soit intercepté par un fraudeur et réutilisé avec succès ultérieurement et/ou sur un autre terminal, on prévoit que le message spécial soit individualisé en outre en relation avec le terminal auquel il est spécifiquement destiné et sous au moins une donnée contenue dans le terminal et connue de l'opérateur du réseau.

De façon préférée dans ce cas, le message spécial est signé sous une clé privée de l'opérateur du réseau de manière à pouvoir être vérifié, une fois reçu par le terminal, sous une clé publique de l'opérateur du réseau qui est tenue en mémoire dans ledit terminal, l'activation de ladite fonction logique étant tributaire de la réussite de ladite vérification. On peut également envisager avantageusement que ladite fonction logique n'est activée que lorsque le terminal a vérifié que le message spécial était bien individualisé à son égard.

De façon également préférée, la donnée sous laquelle le message spécial est individualisé entre en compte dans le calcul de la signature. En particulier, lorsque le code est le seul élément d'individualisation et qu'il est communiqué par ailleurs à l'abonné, la signature porte à la fois sur le code et le message spécial.

Nintendo Exhibit 1003
Part 2 of 2
Page 001001

Toutefois, il est souhaitable, pour améliorer la sécurité, que la donnée sous laquelle le message spécial est individualisé et/ou chiffré comprenne d'autres informations, et en particulier des informations caractéristiques du terminal et/ou tenues en mémoire dans celui-ci. Ainsi, le message spécial pourra en outre être individualisé par un des éléments suivants :

- un identifiant unique public et/ou
- un identifiant unique privé logiciel, c'est-à-dire connu seulement de l'opérateur du réseau (par exemple un numéro de série mis par le fabricant, qui n'est jamais communiqué à l'extérieur du terminal hormis vers l'opérateur), et/ou
- un identifiant unique privé matériel (par exemple peut être mis en œuvre en plaçant dans le terminal un composant spécial prévu à cet effet).

Tous ces identifiants sont en pratique tenus en mémoire dans le terminal.

Dans un mode de mise en œuvre préféré du procédé de l'invention, les messages spéciaux utiles à l'activation de la fonction logique sont envoyés chiffrés sous les identifiants précités et le code.

Enfin, toujours dans un mode de mise en œuvre préféré du procédé de l'invention, l'activation de la fonction logique n'est réalisable que pendant un laps de temps prédéterminé, déclenché par exemple par la première occurrence de l'introduction du code dans le terminal par l'abonné et par la réception par le terminal du message spécial, le terminal étant de construction prévue à cet effet et calculant lui-même la fenêtre temporelle de validation de l'activation de la fonction logique.

Finalement, le processus d'activation de la fonction logique peut, conformément à l'invention, se

Nintendo Exhibit 1003
Part 2 of 2
Page 001002

dérouler comme suit en pratique. Les messages spéciaux utiles à l'activation de la fonction souhaitée sont envoyés, par l'opérateur du réseau à destination du terminal, chiffrés sous les susdits identifiants et le code. Après réception de ce message par le terminal, et éventuellement après l'introduction du code par l'abonné à l'aide du clavier du terminal, le terminal déchiffre le message spécial en mémoire volatile RAM et l'efface de sa mémoire non volatile (il en résulte que, si le terminal est recalé à zéro, l'effet du message spécial sera perdu). Le terminal disposant alors des données et/ou du programme nécessaire, l'exécution du programme d'activation de la fonction s'engage si le terminal a réussi la vérification de la signature grâce à la clé publique de l'opérateur qui est contenue dans sa mémoire.

Comme on l'a indiqué plus haut, une application tout particulièrement intéressante des dispositions de l'invention concerne le domaine de la téléphonie mobile et notamment le déverrouillage du terminal mobile préalablement verrouillé ("simlocking") comme exposé précédemment dans le préambule.

Dans ce contexte, pour, dans un réseau de communication sans fil, désactiver une fonction de verrouillage de type logique dans un terminal mobile détenu par un abonné dudit réseau, on peut procéder comme il suit :

- l'opérateur du réseau communique à l'abonné un code pour que celui-ci l'introduise dans son terminal mobile

et dans le même temps

- l'opérateur du réseau émet, à destination du terminal mobile, au moins un message spécial, tel qu'un message SMS, contenant les informations utiles à la

désactivation de la fonction de verrouillage dudit terminal mobile, le message spécial, notamment SMS, étant individualisé.

Mais dans un autre mode opératoire, on peut prévoir que l'opérateur du réseau émet, uniquement à destination du terminal mobile, au moins un message spécial, tel qu'un message SMS, contenant les informations utiles à la désactivation de la fonction de verrouillage dudit terminal mobile, le message SMS étant individualisé pour ledit terminal mobile.

En effet, pour éviter qu'un message SMS destiné à un terminal mobile donné soit intercepté par un fraudeur et réutilisé avec succès ultérieurement et/ou sur un autre terminal, on prévoit que le message SMS soit individualisé en outre en relation avec le terminal mobile auquel il est spécifiquement destiné et sous au moins une donnée contenue dans le terminal.

De façon préférée, on peut faire en sorte que le message SMS inclut la signature sous une clé privée de l'opérateur du réseau téléphonique, tandis qu'une clé publique de l'opérateur est tenue en mémoire dans le terminal mobile de manière que le message SMS puisse être vérifié une fois reçu.

Grâce à un tel message signé, il devient aisé, pour l'opérateur, de rendre non fonctionnel un terminal mobile qui doit être éliminé du réseau (par exemple terminal volé).

Le message SMS pourra être individualisé avec :

- un identifiant unique public IMEI (International Mobile Equipment Identity ; norme ETSI GSM 02.16), et/ou
- un identifiant unique privé logiciel (c'est-à-dire connu seulement de l'opérateur du réseau) IDSOFT (par

exemple un numéro de série mis par le fabricant, qui n'est jamais communiqué à l'extérieur du terminal mobile hormis vers l'opérateur), et/ou

- un identifiant unique privé matériel IDHARD (par exemple peut être mis en œuvre en plaçant dans le terminal mobile un composant spécial tel que le composant DS 2401 Silicon Serial Number chip fabriqué par DALLAS Semi-Conductor).

Tous ces identifiants sont en pratique tenus en mémoire dans le terminal mobile.

Dans un mode de mise en œuvre préféré du procédé de l'invention, les messages spéciaux SMS utiles à la désactivation de la fonction de verrouillage sont envoyés chiffrés sous IMEI, IDSOFT, IDHARD et CODE.

Enfin, toujours dans un mode de mise en œuvre préféré du procédé de l'invention, la désactivation de la fonction de verrouillage n'est réalisable que pendant un laps de temps prédéterminé, déclenché par exemple par la première occurrence de l'introduction du code dans le terminal mobile par l'abonné et par la réception par le terminal mobile du message spécial SMS, le terminal mobile étant de construction prévue à cet effet et calculant lui-même la fenêtre temporelle de validation de la désactivation de la fonction de verrouillage .

Finalement, le processus de désactivation de la fonction de verrouillage peut, conformément à l'invention, se dérouler comme suit en pratique. Les messages spéciaux SMS utiles à la désactivation recherchée sont envoyés, par l'opérateur du réseau à destination du terminal mobile, chiffrés sous IMEI, IDSOFT, IDHARD et CODE. Après réception de ces messages SMS par le terminal mobile et après l'introduction du code CODE par l'abonné à l'aide du clavier du terminal mobile, celui-ci déchiffre les

messages SMS en mémoire volatile RAM et les efface de sa mémoire non volatile (il en résulte que, si le terminal mobile est recalé à zéro, l'effet des messages SMS sera perdu). Le terminal disposant alors des données et/ou du programme nécessaire, l'exécution du programme de désactivation de la fonction de verrouillage s'engage si le terminal mobile a réussi la vérification de la signature grâce à la clé publique de l'opérateur contenue dans sa mémoire.

Comme cela a été précisé plus haut, les moyens de l'invention propres à commander une désactivation sécurisée du verrouillage d'une fonction du terminal mobile ont une application qui n'est pas limitée à la désactivation du "simlocking", mais peuvent trouver application en relation avec d'autres fonctions optionnelles prévues de construction, mais non initialement mises en service (par exemple fonction mains libres ; fonction de renvoi sur un autre terminal ; fonction d'affichage d'un numéro appelant ; ...).

Au surplus, les dispositions de l'invention ne se limitent pas à l'envoi, par l'opérateur vers le terminal mobile, de seuls messages SMS qui sont mis en œuvre actuellement dans le réseau GSM ; elles pourront tout aussi bien être exploitées avec toutes sortes de messages pouvant être transmis dans le réseau WAP, ou encore d'autres réseaux futurs.

Enfin, on soulignera que, dans les explications qui précèdent, l'opérateur du réseau peut, dans certains cas, déléguer sa signature à une autorité compétente.

REVENDICATIONS

1. Procédé pour activer une fonction logique dans un terminal abonné à un réseau, caractérisé en ce que l'opérateur du réseau émet, à destination du terminal, au moins un message spécial contenant les informations utiles à l'activation de la fonction logique dans le terminal, ledit message étant individualisé pour ledit terminal.

2. Procédé selon la revendication 1, caractérisé en ce que ladite fonction logique n'est activée que lorsque le terminal a vérifié que le message spécial était bien individualisé à son égard.

3. Procédé selon la revendication 1 ou 2, caractérisé en ce que le message spécial est signé sous une clé privée de l'opérateur du réseau de manière à pouvoir être vérifié, une fois reçu par le terminal, sous une clé publique de l'opérateur du réseau qui est tenue en mémoire dans ledit terminal, l'activation de ladite fonction logique étant tributaire de la réussite de ladite vérification.

4. Procédé selon l'une quelconque des revendications 1 à 3, caractérisé en ce que le message spécial est en outre individualisé par un des éléments suivants :

- un identifiant unique public, et/ou
- un identifiant unique privé logiciel connu uniquement de l'opérateur du réseau, et/ou
- un identifiant unique privé matériel,

tous tenus en mémoire dans le terminal.

5. Procédé selon l'une quelconque des revendications 1 à 4, caractérisé en ce que le message spécial comprend l'état d'un verrouillage dans le terminal.

6. Procédé selon la revendication 5, caractérisé en ce que l'état de verrouillage dans le terminal est propre à bloquer ledit terminal.

7. Procédé selon l'une quelconque des revendications 1 à 6, caractérisé en ce qu'en parallèle à la transmission du message spécial au terminal, un code est remis à l'utilisateur dudit terminal et en ce que l'utilisateur introduit ce code dans le terminal.

8. Procédé selon la revendication 7, caractérisé en ce que la signature sous une clé privée de l'opérateur du réseau est appliquée au susdit message spécial et au code.

9. Procédé selon l'une quelconque des revendications 1 à 8, caractérisé en ce que l'activation de la fonction logique est rendue possible par le terminal seulement pendant un laps de temps prédéterminé à compter de l'introduction du code par l'abonné ou de la réception du message spécial émis par l'opérateur du réseau.

10. Procédé selon l'une quelconque des revendications 1 à 9, caractérisé en ce que, après réception du message spécial émis par l'opérateur du réseau, le terminal déchiffre le message spécial en mémoire volatile RAM et l'efface de sa mémoire non volatile, ensuite de quoi le programme d'activation de la fonction logique s'exécute.

11. Procédé selon l'une quelconque des revendications 1 à 10, caractérisé en ce que le terminal est un terminal téléphonique mobile et l'opérateur du réseau est un opérateur de téléphonie mobile.

RÉPUBLIQUE FRANÇAISE

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

2825565

N° d'enregistrement national

FA 614782
FR 0202207

RAPPORT DE RECHERCHE PRÉLIMINAIRE

établi sur la base des dernières revendications déposées avant le commencement de la recherche

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
X	WO 01 17297 A (QUALCOMM INC) 8 mars 2001 (2001-03-08) * page 5, ligne 25 - ligne 28 * * page 8, ligne 10 - ligne 18 * * page 8, ligne 26 - ligne 31 * * page 10, ligne 1 - page 11, ligne 8 *	1,2,4, 10,11	
X	US 5 864 757 A (PARKER JOHN PATRICK) 26 janvier 1999 (1999-01-26) * colonne 4, ligne 5 - colonne 5, ligne 17 * * colonne 6, ligne 29 - ligne 67 *	1,2,4-8, 11	
X A	EP 0 796 023 A (NOKIA MOBILE PHONES LTD) 17 septembre 1997 (1997-09-17) * colonne 1, ligne 46 - ligne 57 * * colonne 4, ligne 35 - colonne 5, ligne 3 *	1-4,7,9, 11 5,6	
X	US 5 386 468 A (MOTOHASHI KAZUTOSHI ET AL) 31 janvier 1995 (1995-01-31) * colonne 7, ligne 29 - ligne 68 *	1-8,11	DOMAINES TECHNIQUES RECHERCHÉS (Int.CL.7) H04Q G06F

Date d'achèvement de la recherche	Examinateur
17 juin 2002	Kampouris, A

CATÉGORIE DES DOCUMENTS CITÉS

X : particulièrement pertinent à lui seul
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie
A : arrière-plan technologique
O : divulgation non-écrite
P : document intercalaire

T : théorie ou principe à la base de l'invention
E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.
D : cité dans la demande
L : cité pour d'autres raisons

& : membre de la même famille, document correspondant

1

EPO FORM 1503 12.99 (P04C14)

ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 0202207 FA 614782

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche préliminaire visé ci-dessus.
Les dits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du 17-06-2002
Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets, ni de l'Administration française

Document brevet cité au rapport de recherche		Date de publication	Membre(s) de la famille de brevet(s)		Date de publication
WO 0117297	A	08-03-2001	US	6301484 B1	09-10-2001
			AU	7096200 A	26-03-2001
			EP	1208708 A1	29-05-2002
			WO	0117297 A1	08-03-2001
US 5864757	A	26-01-1999	AU	715488 B2	03-02-2000
			AU	1409997 A	03-07-1997
			CA	2239550 A1	19-06-1997
			EP	0867099 A2	30-09-1998
			JP	11501182 T	26-01-1999
			JP	3080409 B2	28-08-2000
			WO	9722221 A2	19-06-1997
			US	6124799 A	26-09-2000
EP 0796023	A	17-09-1997	FI	961154 A	14-09-1997
			EP	0796023 A2	17-09-1997
			US	6032038 A	29-02-2000
US 5386468	A	31-01-1995	JP	6097931 A	08-04-1994

EPO FORM P0465

Pour tout renseignement concernant cette annexe : voir Journal Officiel de l'Office européen des brevets, No.12/82

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 December 2002 (05.12.2002)

PCT

(10) International Publication Number
WO 02/097620 A2

(51) International Patent Classification[7]: G06F 9/445

(21) International Application Number: PCT/US02/16485

(22) International Filing Date: 23 May 2002 (23.05.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/872,418 31 May 2001 (31.05.2001) US

(71) Applicant: **QUALCOMM INCORPORATED** [US/US]; 5775 Morhouse Drive, San Diego, CA 92121-1714 (US).

(72) Inventors: **LUNDBLADE, Laurence**; 3062 Navgatuck, San Diego, CA 92117 (US). **PHILLIPS, Marc, S.**; 4008 Gros Ventre Avenue, San Diego, CA 92117 (US). **MINEAR, Brian**; 13704 Fontanelle Place, San Diego, CA 92128 (US). **ZHUANG, Yan**; 8223 Jade Coast Road, No. 120, San Diego, CA 92126 (AU). **KRISHNAN, Anand**; 13775 Sorbonne Court, San Diego, CA 92128 (US). **SPRIGG, Stephen, A.**; 12124 Travertine Court, Poway, CA 92064 (US). **CHMAYTELLI, Mazen**; 1752 Linwood Street #G, San Diego, CA 92110 (US). **OLIVER, Mitchell**; 9737 Caminito Suelto, San Diego, CA 92131 (US). **HOREL, Gerald**; 6500 Torin Road, Brentwood Bay, British Columbia V8M2H5 (CA). **CROSSLAND, Karen**; 5044 Maynard Street, San Diego, CA 92122 (US).

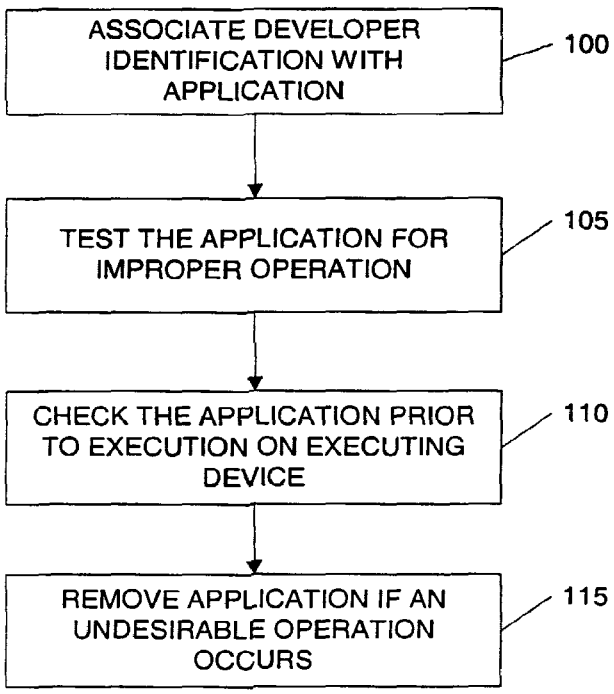
(74) Agents: **WADSWORTH, Philip, R.** et al.; Qualcomm Incorporated, 5775 Morehouse Drive, San Diego, CA 92121-1714 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: SAFE APPLICATION DISTRIBUTION AND EXECUTION IN A WIRELESS ENVIRONMENT



(57) Abstract: The present invention provides safe and secure application distribution and execution by providing systems and methods that test an application to ensure that it satisfies predetermined criteria associated with the environment in which it will execute. Furthermore, by using rules and permission lists, application removal, and a modification detection technique, such as digital signatures, the present invention provides mechanisms to safely distribute and execute tested, or untested, applications by determining whether the application has been modified, determining if it has permission to execute in a given wireless device environment, and removing the application.

WO 02/097620 A2

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SAFE APPLICATION DISTRIBUTION AND EXECUTION IN A WIRELESS ENVIRONMENT

Field of the Invention

[0001] The present invention relates to processing of applications for use in a wireless device, and more particularly, to increasing the security, safety and integrity of applications executed on a wireless device.

Background

[0002] Wireless communication has experienced explosive growth in recent years. As consumers and businesses rely more on their wireless devices, such as mobile phones and personal digital assistants (PDAs), wireless service providers, i.e., carriers, strive to provide additional functionality on these wireless devices. This additional functionality would not only increase the demand for wireless devices but also increase the usage among current users. Increasing functionality, specifically by increasing the applications accessible by the wireless device, however, is costly and complicated thereby discouraging carriers from providing this functionality.

[0003] Furthermore, there is little to no assurance that an application, once placed on a wireless device, will execute properly. Currently, reliance on the application's ability to execute on a wireless device rest on the developer, the wireless device maker and/or the carrier. As more applications are developed and the number of applications on a wireless device increases, the wireless device environment becomes more dynamic. For example, a wireless device may choose to retrieve or execute a number of different applications from large pool of available applications at any given time. Thus, ensuring that any given application will be distributed to the wireless device and execute safely becomes much more difficult to control.

[0004] This is of particular concern because improper execution of an application may not only detrimentally affect the wireless device, but it may also be harmful to the carrier network and other network components, including other wireless devices. For example, one application, if not restricted, could take control of a wireless device's power control and cause interference among other wireless devices and decrease the overall capacity in the cell servicing the wireless device.

Nintendo Exhibit 1003
Part 2 of 2
Page 001013

[0005] Currently, neither wireless device manufacturers nor carriers are equipped to support the testing and safe distribution of applications in a dynamic application distribution and execution environment. Thus, there is a concern that applications will be distributed and executed on wireless devices that may cause harm to the wireless device, carrier network, or other network components.

[0006] In addition, other safety issues arise as more applications are developed and the environment by which applications are transmitted to a wireless device becomes more dynamic. As the number of applications and the number of developers creating these applications increases, the desire to know the source of any given application, i.e., the developer, also increases. A carrier or a handset manufacturer will want to know, with some degree of reliability, that they can determine the source of an application should the application cause harm.

[0007] Consequently, what is needed in the art is a system and method for providing a more safe environment for the distribution and execution of applications on a wireless device.

SUMMARY OF THE INVENTION

[0008] Systems and methods consistent with the present invention overcome the shortcomings of existing systems by creating a safer environment for application distribution and execution that test applications with predetermined standards, provide traceability to the developer for nonrepudiation, check for unintended modifications to the application, allow the removal of the application from the wireless device, and/or use rules and permissions that define the environment on which an application may execute.

[0009] Certifying that an application meets predetermined standards provides the advantage of catching possible errors that could occur during execution ahead of time. This helps to prevent the detrimental effect of an application's execution.

[0010] Traceability provides the advantage of non-repudiation. If there is any problem with the application, it is beneficial to trace back to the source of the application, i.e., the developer, to correct the problem. In addition, having traceability discourages developers from creating applications that have harmful results, either whether intended or unintended.

[0011] Furthermore, the ability to determine if an application is modified prior to receiving it at the wireless device provides the advantage of increased safety by ensuring that the application received is the same one that was transmitted. As applications are distributed more freely in the wireless environment, the ability to determine if an application was modified increases the confidence that an application received by the wireless device was not modified, either accidentally or intentionally.

[0012] Providing a set of rules and permissions that define when applications may execute also increases the safety of an application distribution and execution system by preventing the unauthorized execution of an application on platforms, e.g., systems or environments in which it is not authorized.

[0013] The ability to remove applications from a wireless device also increases the safety of an application distribution system. If an application is installed on a handset, either by the manufacturer or through an application download, having a mechanism to remove the application because of unforeseen negative consequences increases the safety of an application distribution and execution system by removing harmful and undesirable code that could be detrimental.

[0014] Systems and methods consistent with the present invention may invoke one or more of the techniques disclosed herein. By invoking all of the techniques disclosed and referenced herein, however, systems and methods consistent with the present invention provide for high quality and safe distribution and execution of applications.

[0015] In one embodiment of the present invention, a method for distributing and processing an application, comprises the steps of receiving the application and identification information, certifying the application satisfies a predetermined criterion, assigning a permission to the application, transmitting the application, the permission and the identification information to the a device using a modification detection technique, determining whether the application was modified during transmission, storing a rule on the device, determining if the application may be process using the permission and the rule, and removing the application from the device.

[0016] In another embodiment of the present invention, a method for executing an application on a wireless device, comprises the steps of storing a rule to evaluate a permission, receiving information comprising the application, the permission and an

identification using a modification detection technique, receiving a request to execute the application on the wireless device, evaluating the received information to determine if the received information was modified, in the event the received information was not modified, evaluating the permission associated with the application, and in the event the permission is granted, executing the application.

[0017] In yet another embodiment of the present invention, a method for executing an application on a wireless device comprises the steps of storing a rule to evaluate a permission, receiving information comprising the application, the permission and an identification using a modification detection technique, receiving a request to execute the application on the wireless device, evaluating the received information to determine if the received information was modified, in the event the received information was not modified, evaluating the permission associated with the application, and in the event the permission is granted, executing the application.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] 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. In the drawings:

[0019] Figure 1 is a flowchart depicting the high-level process of safe application distribution and execution in an exemplary embodiment of the present invention;

[0020] Figure 2 is a block diagram depicting a system architecture in which an exemplary embodiment of the present invention may be practice;

[0021] Figure 3 is a block diagram depicting a wireless network architecture in which a safe application distribution processing system may be practiced in an exemplary embodiment of the present invention;

[0022] Figure 4 is a block diagram depicting a wireless device and some internal components in an exemplary embodiment of the present invention;

[0023] Figure 5 is a block diagram depicting information used to create a digital signature and transmitted to a wireless device in an exemplary embodiment of the present invention;

[0024] Figure 6 is a flowchart depicting the steps used to by a server or severs in distributing an application in an exemplary embodiment of the present invention; and

[0025] Figure 7 is a flowchart depicting the steps used by a wireless device when executing an application in an exemplary embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0026] Reference will now be made in detail to the exemplary and preferred embodiments of the invention as illustrated in the accompanying drawings, in which like reference characters designate like or corresponding parts throughout the several drawings. The nature, objectives and advantages of the present invention will become more apparent to those skilled in the art after considering the following detailed description in connection with the accompanying drawings.

[0027] The present invention provides safe and secure application distribution and execution by providing systems and methods that test an application to ensure that it satisfies the predetermined criteria associated with the environment in which it will execute. Furthermore, by using rules and permission lists, application removal, and a modification detection technique, such as digital signatures, the present invention provides mechanisms to safely distribute and execute a tested, or untested, application by determining whether the application has been modified, determining if it has permission to execute in a given wireless device environment, and removing the application should it be desirable to do so.

[0028] It will be recognized to those skilled in the art that the forgoing describes an application file type being distributed and executed for simplicity of description. An "application" may also include files having executable content, such as: object code, scripts, java file, a bookmark file (or PQA files), WML scripts, byte code, and perl scripts. In addition, an "application" referred to herein, may also include files that are not executable in nature, such as documents that may need to be opened or other data files that need to be accessed.

[0029] Figure 1 is a flowchart depicting the high level process of safe application distribution and execution in a manner consistent with an exemplary embodiment of the present invention. An embodiment of the present invention enables a developer identification to be associated with the application, performs testing on the application for the environment where the application is intended to execute, assigns permissions that can be used to dictate which devices or systems may execute the application, and provides for application removal should an application perform illegal or undesirable actions.

[0030] It is preferable that systems and methods employ all these techniques to increase the safe distribution and execution of an application. It will be recognized, however, that even employing one or more of these techniques will increase the safe distribution and execution of an application.

[0031] The high level process begins by associating the developer identification with an application (Step 100). This process may be performed by binding the developer identification with the application as it is distributed. Alternatively, the associated developer identification may be stored along with the corresponding application on a server in the system. It is also preferable that the developer identification information be stored and associated with the application information so that it cannot be easily modified.

[0032] The application is then tested for improper operation (Step 105). The application may be used in an environment where improper operation may not only affect the device on which the application is running, but also those other devices that are connected or networked with that device. It is preferable to test the application so that it does not make improper system calls or negatively affect the device or other connected devices during its operation. In one embodiment, this testing is performed by a certification process where the application is tested to determine if it meets predetermined criteria. It is preferable also, to have a certification process independent of the developer, to test the application. The independence of the certification process encourages more accurate and reliable testing.

[0033] Prior to executing the application, the application is checked to determine if it is "allowed" to execute on the device (Step 110). This check may be performed by the use of permissions and rules, described below, or by other permission mechanisms

known to those skilled in the art. Furthermore, it is preferable that the application be checked prior to every attempt to execute the application. This persistent checking process increases the safety of executing the application. For example, it guards against an application having a Trojan horse that may have been inserted into that application on the executing device via another application.

[0034] An application that performs an improper or undesirable operation is then removed from the device (Step 115). This prevents the application from doing any further damage and also frees up the memory in the device for other uses. Alternatively, the application does not need to be removed from the application. Removing an application may refer to the disabling of the application and leaving the application on the device.

[0035] Figure 2 depicts a system architecture in which an exemplary embodiment of the present invention may be practiced. A developer 200 creates an application for use on the wireless device 230. As described above, it will be recognized by those skilled in the art that, while the foregoing description includes an application file type, other file types may be used. Furthermore, it will be recognized by those skilled in the art that the present invention may be used with other wireless or non-wireless devices and may employ wireless networks, non-wireless networks, or a combination thereof.

[0036] Typically, the developer 200 will have a set of development specifications for which to develop the application to execute on the wireless device 230. In one embodiment, the wireless device includes a software platform to assist the application's interface with the wireless device, such as the BREW™ software developed by QUALCOMM, Incorporated, headquartered in San Diego, California. The developer may create the application satisfying the software platform, or BREW™ software, specification standards, and conventions.

[0037] The developer 200 is connected to a central server 205, in one embodiment, so that it may electronically transmit the application to the central server 205. In one embodiment, the central server is an Application Control Center Headquarters (ACCHQ) server used in the distribution of applications to wireless devices. The developer 200 may digitally sign the application (discussed further below) to determine if the application was modified. It will be recognized that a physical

connection to the central server is not necessary. For example the developer 200 may send, such as via first class mail, the application to the central server 205 stored on a CD-ROM.

[0038] In addition, the developer sends various source identification information to the central server 205. This source identification information may include any type of information that may be associated with the application that identifies the developer, such as a company name, tax identification of the company, or other identifying information.

[0039] The central server 205, either by itself or using a certification server 210, is used in the analysis and certification of applications. In one embodiment, an Application Control Center (ACC) may be used as a certification server. The certification server 210 may be used to analyze the application to determine whether the application satisfies predetermined certification criteria. This criteria may include whether the application satisfied the development specification for execution on a wireless device or platform. The certification criteria, however, can be any criteria that an application must satisfy prior to execution on a wireless device or platform. Such criteria may include verifying: (a) that the application functions as claimed by the developer, so that the application does not harm the wireless device's operation (e.g., it does not crash the phone); (b) that the application does not access data or memory that it should not (e.g., it doesn't access data or files owned by other applications, the operating system or platform software); and (c) that it does not negatively impact the wireless devices resources, such as detrimentally monopolizing the input and output of the wireless device.

[0040] The central server 205 may also assign a set of permissions in a list associated with the application. This permission list is determined by various factors, including an analysis of whether the application passed the certification process, what networks 220 the application is approved to execute on, and whether the wireless device supports the application. There may be many factors used to determine a permission list and is left to those skilled in the art when implementing the present invention.

[0041] The central server 205 receives the developer identification information and correlates it with the application created by the developer 200. Should any problems exist with the application, the central server will be able to identify the source of the

application. In one embodiment, the developer information is passed to the wireless device 230 so that the correlation may be performed by the wireless device or other systems connected to the wireless device.

[0042] In one embodiment, the central server is also connected to an application download server (ADS) 215. The application download server 215 is used to interface with a wireless device via a wireless network 220 to download an application. The central server may also send the permissions list and developer identification associated with the application to the ADS where it can be stored until transmission to a wireless device. It is preferred that the application, permission list and the developer identification be digitally signed by the central server to increase security from modification.

[0043] It will be recognized by those skilled in the art that an ADS may be used to connect to multiple networks 220 for the distribution of applications, files and other information to the various wireless devices 230. Furthermore, wireless and non-wireless networks may be employed to transmit the application's permission list and developer identification to the wireless device.

[0044] In response to a request for an application, the ADS 215 will send the application, permission list, developer identification, and digital signature(s) to the wireless device 230 via the network 220. In one embodiment, the wireless device 230 will contain a key to check the digital signature in order to determine if the application, permission list and/or developer information is modified.

[0045] It is preferred, if digital signatures are employed in the present invention, that the central server use a secure key to create the digital signature and install a key on a wireless device to evaluate the digital signature. By using a secure key, the wireless device will have a higher degree of reliability that the digital signature was created by the central server and not an imposter.

[0046] Should the application cause any errors on the wireless device, or for any other desired reason, the wireless device may initiate the removal of the application. Furthermore, an application may be removed from the wireless device based on a request from the ADS or central server. This request from the server may be initiated for any desired reason. For example, the server may initiate the removal of applications from a wireless device because the application performed improperly on

another device, a new version of the application is distributed, or even business reasons dictating that the application should be removed. This application removal process further protects the wireless device environment from repeated execution of corrupted and/or destructive applications.

[0047] Figure 3 depicts a wireless network architecture in which an application distribution system may be practiced in an exemplary embodiment of the present invention. A central server 302 is an entity that certifies, either by itself or in combination with a certification server, the application programs as compatible with a defined set of programming standards or conventions. As described earlier, these programming standards may be established so that the application will execute on a software platform, such as the BREW™ platform.

[0048] In one embodiment, the central server database 304 consists of a record of the identifications for each application program downloaded at any time onto each wireless device 330 in the network 300, an Electronic Service Number ("ESN") for the individual who downloaded the application program, and a Mobile Identification Number ("MIN") unique to the wireless device 330 carrying that application program. Alternatively, the central server database 304 contains records for each wireless device 330 in the network 300 of the wireless device model, wireless network carrier, the region where the wireless device 330 is used, and any other information useful to identify which wireless device 330 are carrying which application programs. In addition, the central server database may also store this developer identifying information associated with an application.

[0049] In one embodiment, the central server 302 may also include a remove command source 322. The remove command source 322 is the person(s) or entity(ies) that may make the decision to remove one or more targeted application programs. The remove command source 322 also is the entity that constructs a remove command 316 (discussed below) that is broadcast to identified wireless device 330 carrying the targeted application program(s). Alternatively, and without limitation, the remove command source 322 may be one or more persons or entities involved with the development and issuance of the targeted application program, persons or entities involved with the manufacturing of the wireless device 330, and/or persons or entities involved with the function of any part of the network 300.

[0050] The central server 302 communicates with one or more computer servers 306, e.g., an ADS, over a network 308, such as the Internet, preferably secured. The servers 306 also communicate with a carrier network 310 via a network 308. The carrier network 310 communicates with the MSC 312 by both the Internet and Plain Ordinary Telephone System (POTS) (collectively identified in Figure 3 as 311). The Internet connection 311 between the carrier network 310 and the MSC 312 transfers data, and the POTS 311 transfers voice information. The MSC 312, in turn, is connected to multiple base stations ("BTS") 314. The MSC 312 is connected to the BTS by both the Internet 311 (for data transfer) and POTS 311 (for voice information). The BTS 314 sends messages wirelessly to the wireless devices 330 by short messaging service ("SMS"), or any other over-the-air method.

[0051] One example of a message sent by the BTS 314 in the present invention is a remove command 316. As further discussed herein, the wireless device 330, in response to receiving a remove command 316, responds by uninstalling a targeted application program stored on the wireless device 330. In one embodiment, the remove program may additionally or alternatively be programmed to disable the targeted application program or reprogram it to perform differently. The wireless device may also delete the application and any related information, such as a permission list.

[0052] The remove command 316 is constructed by the remove command source 322 (which may or may not be the same person(s) or entity(ies) that made the decision to initiate a remove of the targeted application program). The remove command 316 is sent by the remove command source 322 over the network 300 for broadcasting to the wireless devices 330.

[0053] By using the remove command as described in the above embodiment, the safety of application distribution and execution is increased by providing a mechanism to uninstall corrupted or undesirable applications. It will be recognized by those skilled in the art that, while the preceding described a remove command initiated by the central server, the wireless device may also initiate the removal or uninstallation of the application and its related information.

[0054] Similarly, the above network may be used to send the application, permission list and associated digital signatures from the central server to various servers 306 (e.g., ADS') through the MSC and BTS to the wireless devices 330.

[0055] Figure 4 depicts a wireless device and some internal components in an exemplary embodiment of the present invention. While this embodiment is directed to a wireless device 400, it is used as an example without any intended limitation. The invention could alternatively be performed on any form of remote module capable of communicating through a network, including, without limitation, wireless and non-wireless devices, such as personal digital assistants ("PDAs"), wireless modems, PCMCIA cards, access terminals, personal computers, devices without a display or keypad, or any combination or sub-combination thereof. These examples of remote modules also may have a user interface, such as a keypad, visual display or sound display.

[0056] The wireless device 400 shown in Figure 4 has an application-specific integrated circuit ("ASIC") 415 installed at the time the wireless device 400 is manufactured. The ASIC is a hardware component that is driven by software included in the ASIC. An application programming interface ("API") 410 is also installed in the wireless device 400 at the time of manufacture. In one embodiment, the API represents a BREW API or software platform. The API 410 is a software program configured to interact with the ASIC. The API 410 serves as an interface between the ASIC 415 hardware and application programs (discussed below) installed on the wireless device 400. Alternatively, the wireless device 400 may contain any other form of circuitry that would allow programs to be operated in a manner that is compatible with the hardware configuration of the wireless device 400. The wireless device 400 also has storage 405. The storage 405 consists of RAM and ROM, but, alternatively, could be any form of memory, such as EPROM, EEPROM or flash card inserts.

[0057] The storage area 405 of the wireless device may be used to store received applications and permission lists 425. In addition, the storage area 405 may be used to store one or more "keys" 405. These keys can be applied to a digital signature using a signature algorithm to determine whether the signed information was modified.

[0058] Rules 435 may also be installed on the wireless device 400. These rules may be used in conjunction with the permission list to determine if an application is allowed to execute. For example, a rule may state that an application is allowed to execute if a certification flag is set in the permission list (i.e., indicating the application passed certification). The permission list will have the certification flag set or not, depending on whether it passed certification. By applying the rule to the information contained in the permission list, permission to execute the application is either granted or denied.

[0059] The manufacturer (not shown) of the wireless device 400 may download application programs onto the storage 405 of the wireless device 400 at the time the wireless device 400 is manufactured. These application programs may be any program potentially useful or entertaining to the user of the wireless device, such as games, book, or any other type of data or software programs. The application programs also may be downloaded onto the wireless device 400 over the air after the wireless device is manufactured.

[0060] The remove program, when executed by the wireless device 400, uninstalls one or more targeted application programs from one of the applications stored on the wireless device 400. The targeted application program is an application program that needs to be uninstalled from the wireless device 400 for various reasons discussed below.

[0061] The wireless device 400 has a local database 420 installed by the manufacturer. The API of the wireless device is programmed to automatically update the local database 420 with a record of identifying information about each of the application programs stored on the wireless device 400. The local database 420 contains a record of the signature identifications unique to each application program stored on the wireless device 402. Additionally, the local database 420 may contain a record of the location of the application programs within the storage 405 on the wireless device 400 and any other information useful for keeping track of which application programs are downloaded on the wireless device 400, and where they are located.

[0062] Figure 5 is a block diagram depicting information used to create a digital signature and transmitted to a wireless device in an exemplary embodiment of the

present invention. As is known by those skilled in the art, digital signatures may be used to track whether a digital file has been modified. As described, digital signatures can be applied to any digital file, including documents, applications, databases, etc. In general, a digital signature is created by applying a key to a file using a signature algorithm. This digital signature is created using the information contained in the file. Typically, the digital signature is sent along with the file to a recipient. The recipient of the file and digital signature can then apply a key to the received file and digital signature to determine if the file has been modified during the transmission to the recipient.

[0063] Keys used to create and evaluate a digital signature can be used to determine the identity of the signer. For example, a key may be generated to create a digital signature by an entity and kept securely. This entity can distribute a corresponding key that can be used to evaluate the digital signature. If the key is kept securely and not compromised, the recipient evaluating the digital signature can determine not only whether the information was modified, but also the identity of the signer.

[0064] Alternatively, third-party entities can create keys for specific entities in a secure fashion. Therefore, a recipient having a key associated with a specific identity will be able to determine if that entity was the signer.

[0065] In one embodiment of the present invention, a digital signature 515 is generated by using the signer's key 525, e.g., a key of the central server (see Figure 2), the application 500, the permission list 505 and the developer identity information 510 as input to a digital signature algorithm 530. The result is a digital signature 515 that is dependent on the information contained in the inputs.

[0066] After creating the digital signature 515, the application 500, permission list 505, developer identity information 510 and digital signature 515 are transmitted to the wireless device 520. The wireless device can then use the digital signature to determine if any of the application or related information (i.e., the permission list and developer identity information) was modified. In addition, using one of the techniques described above, such as a secure key, the wireless device may also have confidence in the identity of the signer who transmitted this information to the wireless device.

[0067] Figure 6 is a flowchart depicting the steps used to by a server or servers in distributing an application in a manner consistent with an exemplary embodiment of the present invention. In this exemplary embodiment, the process begins by receiving an application and a digital signature (Step 600). The digital signature is information related to the application such that it can be determined if the application was modified prior to its reception. Furthermore, it is preferable that the key used to sign the digital signature be assigned by a third party in order validate that the entity or developer signing the application is the developer that received the assigned key.

[0068] After receiving the application and digital signature, the digital signature is evaluated to determine if the develop who sent the application is the same as the one who signed the application (Step 605). If a third party assigned the key to the developer to create the digital signature, then the third party may also assign the key to evaluate the digital signature to the receiving party, such as to the central server described with respect to Figure 2.

[0069] The identification of the developer, or whichever entity signed and/or created the application, is then stored and associated with the application (Step 610). The storage may be in a table, database or in some other manner such that it can be later retrieved in the event the identity of the developer needs to be determined. In one embodiment, the storage of the developer's identification is stored in a wireless device and not in a server.

[0070] The received application is then certified to determine if it meets specified criteria (Step 615). In one embodiment, an application may be written to execute on a specific platform, such as the BREW™ platform developed by QUALCOMM, Incorporated, headquartered in San Diego, California used in wireless devices. A specific platform, or device, may have specific requirements that an application must meet prior to having it executed on the device. For example, a platform or device may require that an application not access specific memory locations in the device so that the integrity of the device or other applications located in memory are not compromised. These criteria can be specified and the application can be tested to determine if these criteria are met. Preferably, these criteria are predetermined and provided to the developer to incorporate into the application's development.

[0071] After certification, the permissions associated with the application for a given environment are assigned (Step 620). Permission may be assigned based on many factors, depending on the environment in which the present invention is implemented. In one embodiment, the applications are intended for a wireless device. In this embodiment, assigning permissions may depend on the carrier network, a wireless device's requirements, results of certification testing, and developer, carrier or other testing environments, for example. Therefore, an example of a permission list is an indication that the application passed certification testing and that it may execute on a specific carrier's network.

[0072] The server then digitally signs the application, permission list, and developer identification (Step 625). In one embodiment, this signature is performed using a secure key so that the identity of the server can be determined by those receiving this digitally signed information. It is not required that the developer's signature that was received by the server also be signed or that the developer's signature be sent to the wireless device.

[0073] The application, permission list, developer identification and the signature created in step 625 is then transmitted to a wireless device (Step 630).

[0074] Figure 7 is a flowchart depicting the steps used by a wireless device when executing an application in a manner consistent with an exemplary embodiment of the present invention. In this embodiment, the wireless device stores rules to evaluate permissions associated with an application (Step 700). It will be recognized by those skilled in the art that, while the present invention describes a rules/permissions paradigm, there are many paradigms that may be used to grant permission to an application for a specific device or platform and these are considered within the scope of the present invention.

[0075] The wireless device then receives the application, permission list, developer identification and digital signature (Step 705). In one embodiment, the wireless device may evaluate the received digital signature to determine the identity of the signer. The digital signature may also be used to determine if the application, permission list or developer identification was modified after having been signed.

[0076] The wireless device then receives a request to execute the application (Step 710). This request may come from the user of the wireless device wanting to execute

a program. Alternatively, the request may be made by the wireless device itself or from some request transmitted to the wireless device, either through a network or direct connection to the wireless device.

[0077] After receiving the request, the wireless device evaluates the digital signature and the permission list associated with the application prior to its execution (Step 720). As described, the wireless device, in one embodiment, may use rules to evaluate the permissions list. If by evaluating the digital signature it is determined that the application, permission list or developer identification was not modified, then the wireless device evaluates the permission list using the stored rules. If there was no modification and the evaluation of the rules against the permission list indicates that the application is granted permission to execute in the wireless device, the processing proceeds to execute the application on the device (Step 730).

[0078] If the evaluation in Step 720 indicates that either the application, permission list or developer identification was modified after being signed, or that the application is denied permission to execute on the wireless device, then the application is not executed (Step 725). Processing proceeds to remove the application from the wireless device (Step 750). It is also preferred that the permission list and developer identification also be removed from the wireless device.

[0079] Following step 730, the application's execution is monitored to determine if it performs an illegal or improper operation (Step 735). The wireless device or the platform the wireless device is using may define certain operations to be illegal or improper. These operations may include those that access restricted areas of memory or memory locations used by other programs or files. In addition, these operations may involve harmful uses of the wireless device's resources such that they may not only affect the wireless device, but other devices on the network the wireless device is attached.

[0080] If such an illegal or improper operation is attempted, then the application's execution is stopped (Step 745) and removed from the wireless device (Step 750) along with, preferably, the developer identification and permission list. As stated above, alternatively, the remove process may involved the disabling of the application, thereby preventing its execution, and keeping the application on the wireless device.

[0081] If no illegal, improper, or undesirable operation is performed in step 735, then the application is allowed to continue execution (Step 740).

CONCLUSION

[0082] Using mechanisms to certify, detect modifications, determine source identity, assign permissions, and incorporate the ability to remove the application, systems and methods consistent with the present invention increase safe and secure application distribution and execution. Systems and methods may implement as few or all of these mechanisms. The more mechanisms implemented, the higher the degree of safety that is achieved.

[0083] In one embodiment, a developer sends an application to a server. The developer may sign the application to protect against unauthorized modification. A server checks the identity of the developer, and performs certification testing on the application. The server also assigns permissions to the application, creating a permission list. The application, permission list, developer identification are digitally signed by the server and sent to a wireless device along with the digital signature. A wireless device checks the digital signature for modification and the permission list against stored rules prior to executing the application. In one embodiment, these checks are performed prior to each attempt to execute the application on the wireless device. If the checks indicate the application has been modified or denied permission to execute, the application does not execute and is removed from the wireless device. Furthermore, if during execution, the application attempts an illegal or improper operation, the application is terminated and then removed from the wireless device.

[0084] The foregoing description of an implementation of the invention has been presented for purposes of illustration and description. It is not exhaustive and does not limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing of the invention. For example, the described implementation includes software but one embodiment of the present invention may be implemented as a combination of hardware and software or in hardware alone. The invention may be implemented with both object-oriented and non-object-oriented programming systems. Additionally, although aspects of the present invention are described as being stored in memory,

those skilled in the art will appreciate that these aspects can also be stored on other types of computer-readable media, such as secondary storage devices, like hard disks, floppy disks, or CD-ROM; a carrier wave from the Internet or other propagation medium; or other forms of RAM or ROM. The scope of the invention is defined by the claims and their equivalents.

What is claimed is:

CLAIMS

[c1] 1. A method for distributing and processing an application, comprising the steps of:

receiving the application and identification information;

certifying the application satisfies a predetermined criterion;

assigning a permission to the application;

transmitting the application, the permission and the identification information to the a device using a modification detection technique;

determining whether the application was modified during transmission;

storing a rule on the device;

determining if the application may be process using the permission and the rule; and

removing the application from the device.

[c2] 2. The method of claim 1 further comprising the steps of:

initiating the execution of the application on the device;

monitoring the execution of the application; and

detecting the application attempting to make an improper operation.

[c3] 3. The method of claim 1, wherein the application is denied permission to be processed on the device.

[c4] 4. The method of claim 1, wherein the identification information is retrieved by the device.

[c5] 5. The method of claim 1 further comprising the step of detecting a modification in the application.

[c6] 6. The method of claim 1 further comprising the step of detecting a modification in the permission.

[c7] 7. The method of claim 1, wherein the modification detection technique uses a digital signature.

[c8] 8. A method for distributing and processing an application, comprising the steps of:

receiving the application and identification information;

certifying the application satisfies a predetermined criterion;

assigning a permission to the application;

transmitting the application, the permission and the identification information to the a wireless device;

storing a rule on the device; and

determining if the application may be processed on the device using the permission and the rule.

[c9] 9. The method of claim 8 further comprising the steps of:

initiating the execution of the application on the wireless device;

monitoring the execution of the application;

detecting the application attempting to make an improper operation; and

removing the application from the wireless device.

[c10] 10. The method of claim 8, wherein the application was denied permission to be processed on the wireless device and further comprising the step of removing the application from the device.

[c11] 11. The method of claim 8, wherein the identification information is determined by the wireless device.

[c12] 12. The method of claim 8, wherein a modification detection technique is used to transmit the application, the permission and the identification information.

[c13] 13. The method of claim 12, wherein the modification detection technique uses a digital signature.

[c14] 14. The method of claim 12 further comprising the steps of:
detecting a modification in the application transmitted to the device; and
removing the application from the device.

[c15] 15. The method of claim 12 further comprising the steps of:
detecting a modification in the permission transmitted to the device; and
removing the application from the device.

[c16] 16. A method for distributing and processing an application, comprising the steps of:
receiving the application and identification information;
assigning a permission to the application;
transmitting the application, the permission and the identification information to the a device using a modification detection technique;
determining whether the application was modified during transmission;
storing a rule on the device;
determining if the application may be process using the permission and the rule; and
removing the application from the device.

[c17] 17. The method of claim 16 further comprising the steps of:
initiating the execution of the application on the wireless device;
monitoring the execution of the application;
detecting the application attempting to make an improper operation; and
removing the application from the wireless device.

[c18] 18. The method of claim 16, wherein the application was denied permission to be processed on the wireless device and further comprising the step of removing the application from the device.

[c19] 19. The method of claim 16, wherein the identification information is determined by the wireless device.

[c20] 20. The method of claim 16, wherein a modification detection technique is used to transmit the application, the permission and the identification information.

[c21] 21. The method of claim 20, wherein the modification detection technique uses a digital signature.

[c22] 22. The method of claim 20 further comprising the steps of:
detecting a modification in the application transmitted to the device; and
removing the application from the device;

[c23] 23. The method of claim 12 further comprising the steps of:
detecting a modification in the permission transmitted to the device; and
removing the application from the device.

[c24] 24. A system for distributing an executing an application on a wireless device, comprising:
a central server operable to receive the application and developer identification, to assign a permission associated with the application and to transmit the application information to the wireless device;
a server operable to certify the application satisfies a predefined criterion; and
the wireless device operable to receive the application and developer identification, and to evaluate the assigned permission.

[c25] 25. The system of claim 24, wherein the central server is further operable to transmit the application to the wireless device using a modification detection technique.

[c26] 26. The system of claim 24, wherein the central server is further operable to transmit the permission to the wireless device using a modification detection technique.

[c27] 27. The system of claim 24, wherein the server is operable to certify the application is the central server.

[c28] 28. The system of claim 24, wherein the wireless device is operable to evaluate the assigned permission using a rule stored in the wireless device.

[c29] 29. A system for distributing and executing an application on a wireless device, comprising:

means for receiving the application and identification information;

means for assigning a permission to the application;

means for transmitting the application, the permission and the identification information to the device using a modification detection technique;

means for determining whether the application was modified during transmission;

means for storing a rule on the device;

means for determining if the application may be process using the permission and the rule; and

means for removing the application from the device.

[c30] 30. A method of processing application distribution, comprising the steps of:

receiving the application and identification information;

certifying the application satisfies a predetermined criterion;

assigning a permission to the application;

transmitting the application, the permission and the identification information to the device using a modification detection technique; and

initiating a request to remove the application from a device.

[c31] 31. The method of claim 30, further comprising the step of:

evaluating the received application and identification information to determine the identity of a developer of the application.

[c32] 32. The method of claim 30, wherein the modification detection technique uses a digital signature.

[c33] 33. A method of processing application distribution, comprising the steps of:

receiving the application and identification information;

assigning a permission to the application; and

transmitting the application, the permission and the identification information to the device using a modification detection technique.

[c34] 34. The method of claim 33 further comprising the step of initiating a request to remove the application from a device.

[c35] 35 The method of claim 33 further comprising the step of certifying the application satisfies a predetermined criterion.

[c36] 36. The method of claim 33, further comprising the step of:

evaluating the received application and identification information to determine the identity of a developer of the application.

[c37] 37. The method of claim 33, wherein the modification detection technique uses a digital signature.

[c38] 38. A system for application distribution, comprising:

a central server operable to receive application and identity information, assign a permission, and transmit the application using a modification detection technique; and

a network connected to the central server to receive the application.

[c39] 39. The system of claim 38, wherein the central server is further operable to certify the application satisfies a predefined criterion.

[c40] 40. A system for application distribution, comprising:

means for receiving the application and identification information;

means for certifying the application satisfies a predetermined criterion;

means for assigning a permission to the application; and

means for transmitting the application, the permission and the identification information to the device using a modification detection technique.

[c41] 41. A computer-readable medium containing computer-executable instructions for distributing applications, comprising the steps of:

receiving the application and identification information;

certifying the application satisfies a predetermined criterion;

assigning a permission to the application; and

transmitting the application, the permission and the identification information to the device using a modification detection technique.

[c42] 42. A method for executing an application on a wireless device, comprising the steps of:

storing a rule to evaluate a permission;

receiving information comprising the application, the permission and an identification using a modification detection technique;

receiving a request to execute the application on the wireless device;

evaluating the received information to determine if the received information was modified;

in the event the received information was not modified, evaluating the permission associated with the application; and

in the event the permission is granted, executing the application.

[c43] 43. The method of claim 42, wherein the modification detection technique uses a digital signature.

[c44] 44. The method of claim 42 further comprising the step of monitoring the execution of the application to determine if an improper operation is attempted.

[c45] 45. The method of claim 42 further comprising the step of removing the application from the wireless device.

[c46] 46. A method for executing an application on a wireless device, comprising the steps of:

storing a rule to evaluate a permission;

receiving information comprising the application, the permission and an identification using a modification detection technique;

receiving a request to execute the application on the wireless device; and

evaluating the permission associated with an application; and

in the event the information was modified, removing the application from the wireless device.

[c47] 47. The method of claim 46, further comprising the steps of:

in the event the received information was not modified, evaluating the permission associated with the application; and

in the event the permission is granted, executing the application.

[c48] 48. The method of claim 46, wherein the modification detection technique uses a digital signature.

[c49] 49. The method of claim 46 further comprising the step of monitoring the execution of the application to determine if an improper operation is attempted.

[c50] 50. The method of claim 46 further comprising the step of removing the application from the wireless device in the event the improper operation is attempted.

[c51] 51. A wireless device for executing an application, comprising:

a input to receive an application and a permission;

a rule contained in storage to evaluate the permission; and

a key contained in storage to evaluate a digital signature,

wherein the wireless device is operable to execute the application upon evaluation of the permission and digital signature.

[c52] 52. A wireless device for executing an application, comprising:

means for storing a rule to evaluate a permission;

means for receiving information comprising the application, the permission and an identification using a modification detection technique;

means for receiving a request to execute the application on the wireless device;

means for evaluating the received information to determine if the received information was modified;

means for, in the event the received information was not modified, evaluating the permission associated with the application; and

means for, in the event the permission is granted, executing the application.

[c53] 53. A computer-readable medium containing computer-executable instructions for distributing applications, comprising the steps of:

storing a rule to evaluate a permission;

receiving information comprising the application, the permission and an identification using a modification detection technique;

receiving a request to execute the application on the wireless device;

evaluating the received information to determine if the received information was modified;

in the event the received information was not modified, evaluating the permission associated with the application; and

in the event the permission is granted, executing the application.

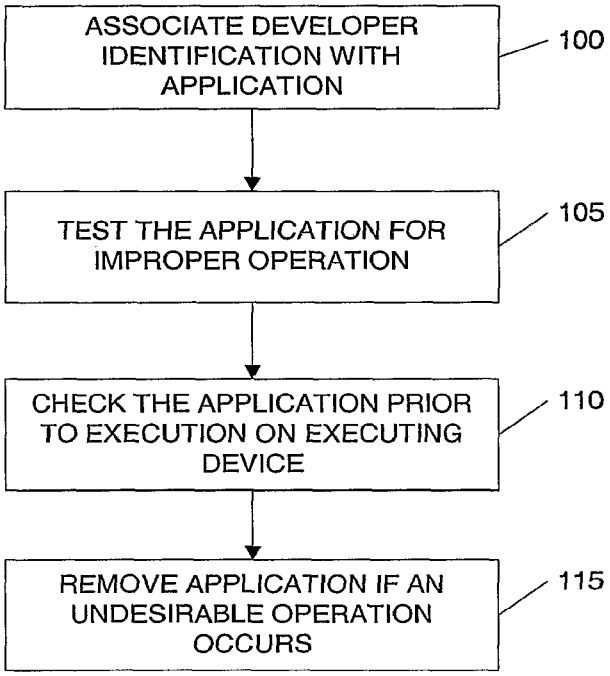

ASSOCIATE DEVELOPER IDENTIFICATION WITH APPLICATION
100
TEST THE APPLICATION FOR IMPROPER OPERATION
105
CHECK THE APPLICATION PRIOR TO EXECUTION ON EXECUTING DEVICE
110
REMOVE APPLICATION IF AN UNDESIRABLE OPERATION OCCURS
115

FIG. 1

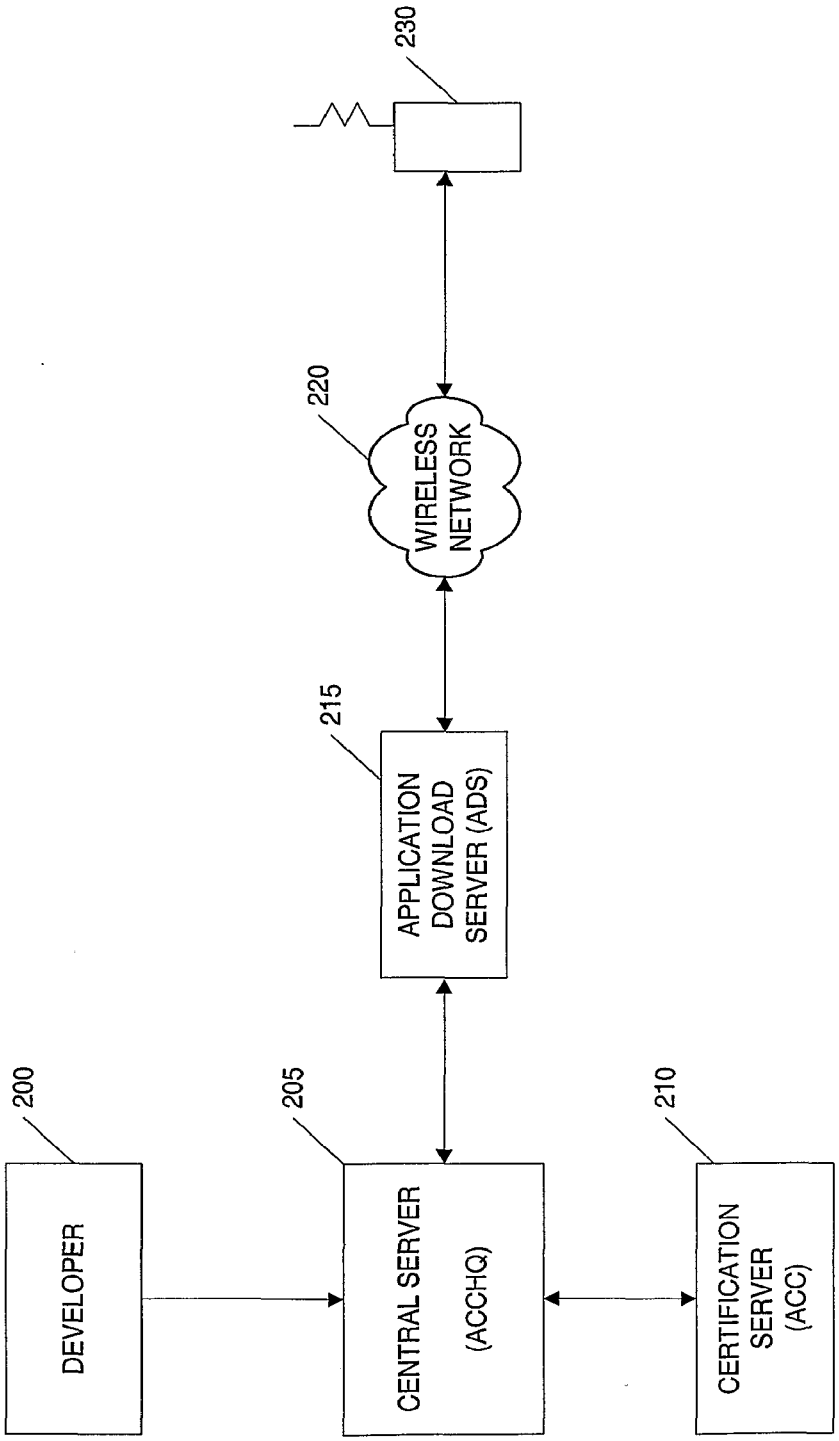

230
220
WIRELESS NETWORK
215
APPLICATION DOWNLOAD SERVER (ADS)
200
205
210
DEVELOPER
CENTRAL SERVER (ACCHQ)
CERTIFICATION SERVER (ACC)

FIG. 2

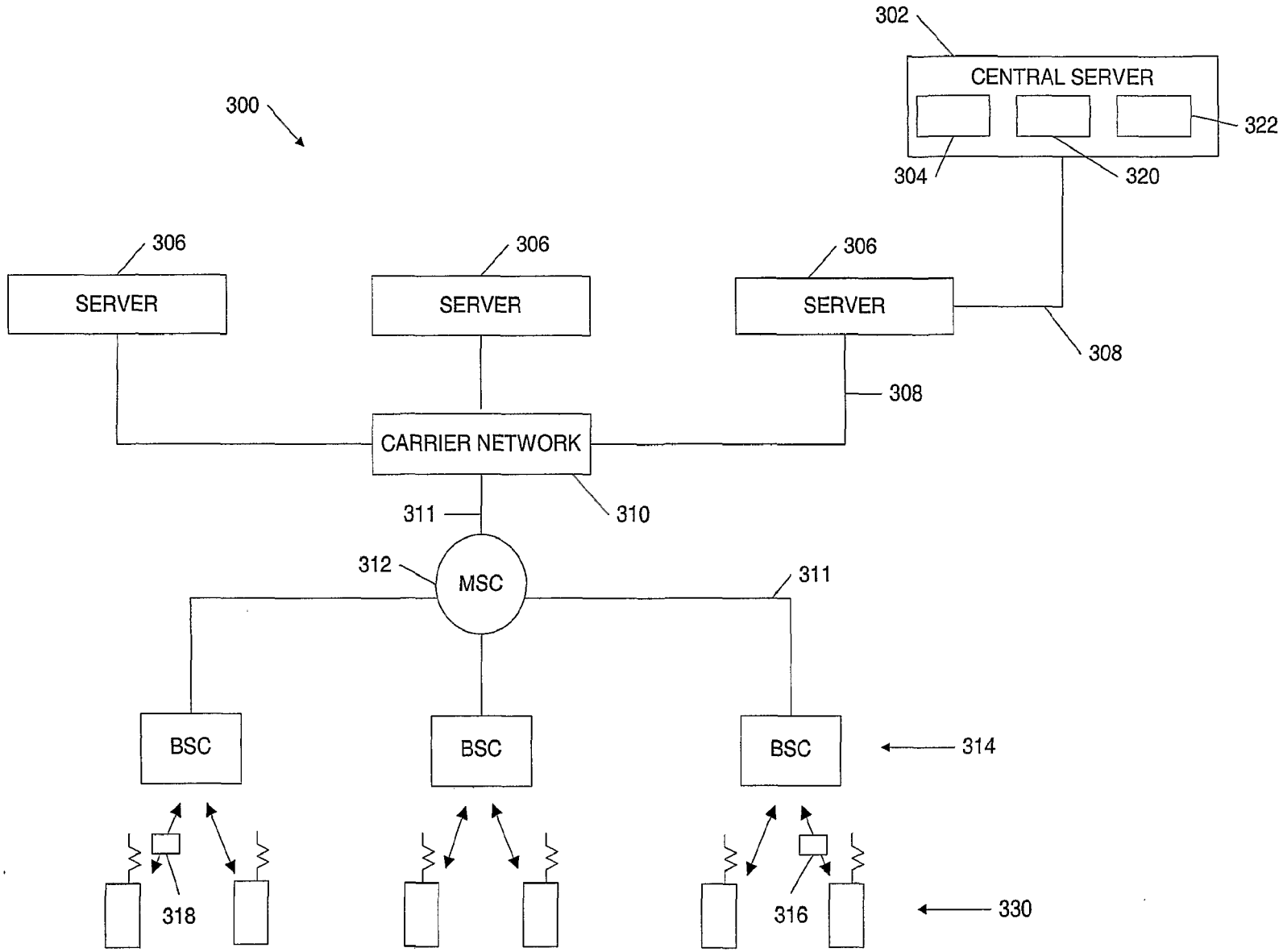


FIG. 3

Nintendo Exhibit 1003
Part 2 of 2
Page 001043

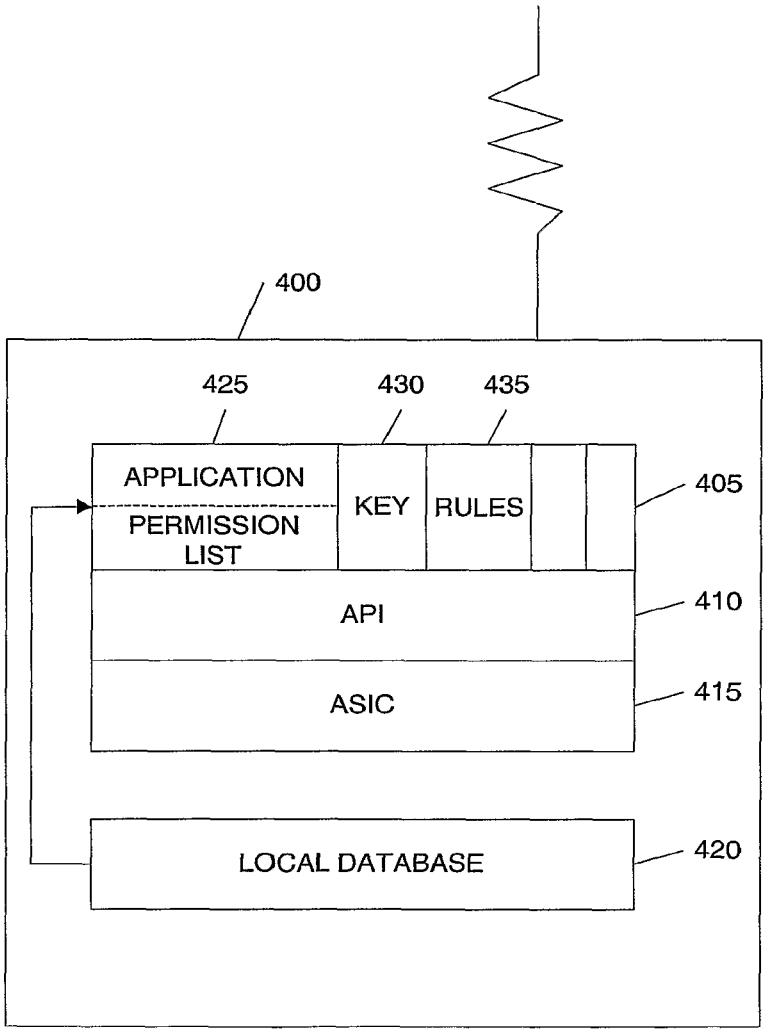

400
425
430
435
APPLICATION
PERMISSION LIST
KEY
RULES
405
API
410
ASIC
415
LOCAL DATABASE
420

FIG. 4

Nintendo Exhibit 1003
Part 2 of 2
Page 001045

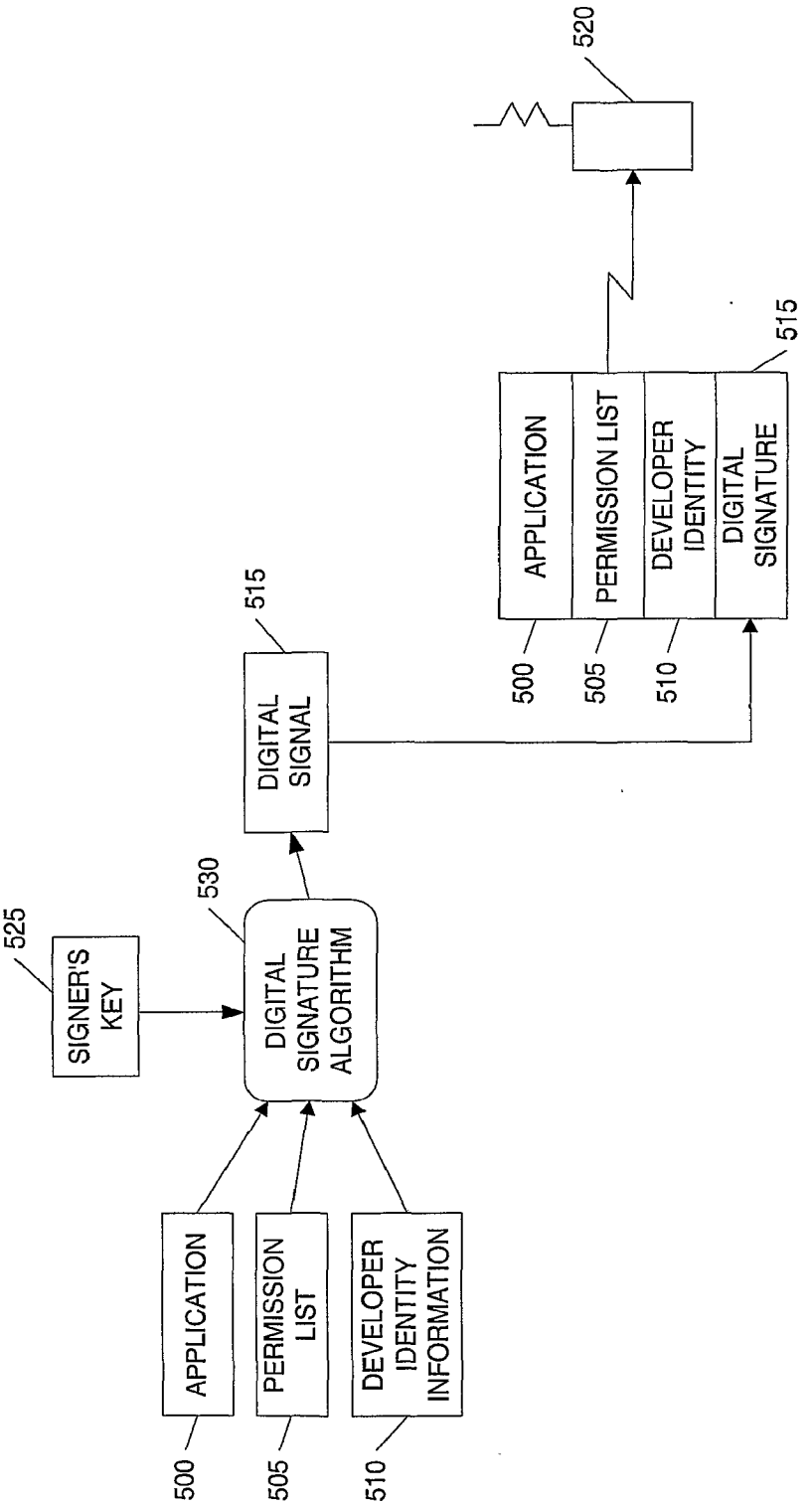


FIG. 5

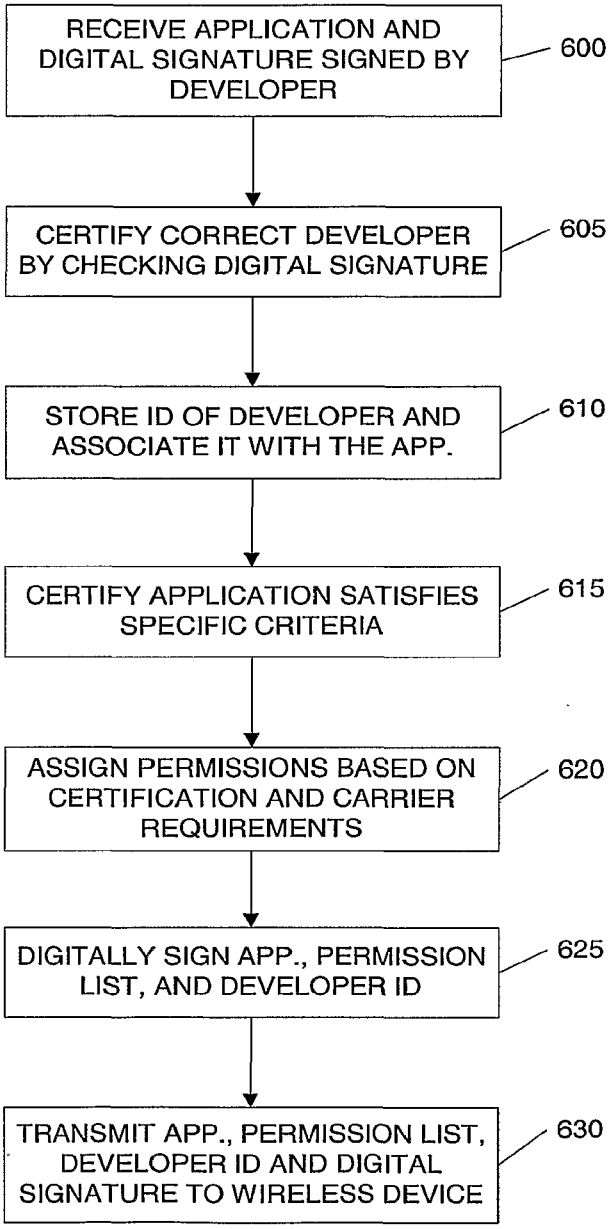

RECEIVE APPLICATION AND DIGITAL SIGNATURE SIGNED BY DEVELOPER
600
CERTIFY CORRECT DEVELOPER BY CHECKING DIGITAL SIGNATURE
605
STORE ID OF DEVELOPER AND ASSOCIATE IT WITH THE APP.
610
CERTIFY APPLICATION SATISFIES SPECIFIC CRITERIA
615
ASSIGN PERMISSIONS BASED ON CERTIFICATION AND CARRIER REQUIREMENTS
620
DIGITALLY SIGN APP., PERMISSION LIST, AND DEVELOPER ID
625
TRANSMIT APP., PERMISSION LIST, DEVELOPER ID AND DIGITAL SIGNATURE TO WIRELESS DEVICE
630

FIG. 6

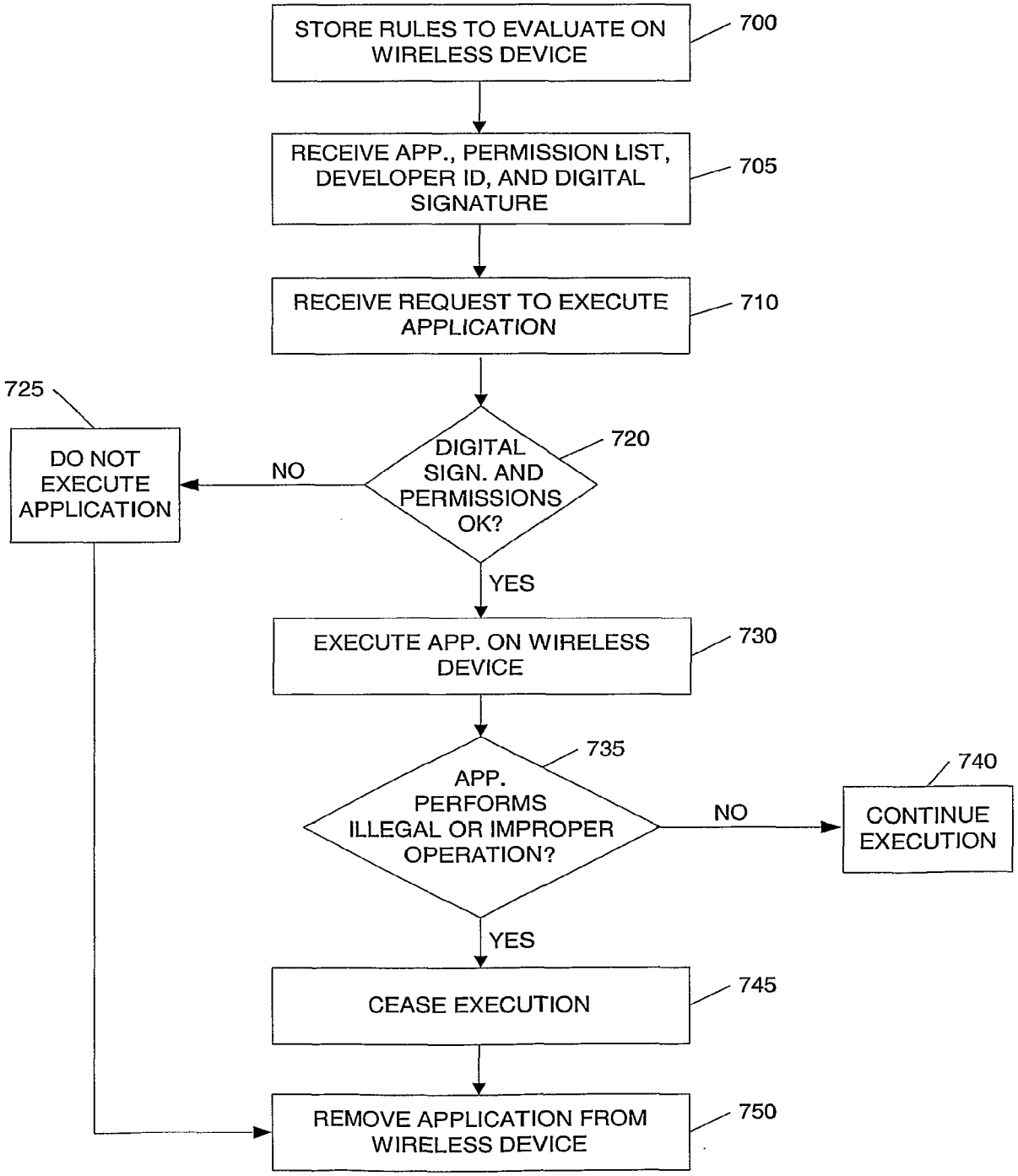

STORE RULES TO EVALUATE ON WIRELESS DEVICE
700
RECEIVE APP., PERMISSION LIST, DEVELOPER ID, AND DIGITAL SIGNATURE
705
RECEIVE REQUEST TO EXECUTE APPLICATION
710
725
DO NOT EXECUTE APPLICATION
NO
DIGITAL SIGN. AND PERMISSIONS OK?
720
YES
EXECUTE APP. ON WIRELESS DEVICE
730
735
APP. PERFORMS ILLEGAL OR IMPROPER OPERATION?
NO
740
CONTINUE EXECUTION
YES
CEASE EXECUTION
745
REMOVE APPLICATION FROM WIRELESS DEVICE
750

FIG. 7

Cited Ref 4

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-170063
(43)Date of publication of application : 14.06.2002

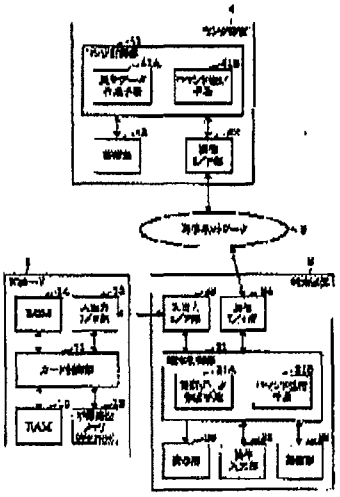
(51)Int.Cl. G06F 17/60
G06K 17/00
G06K 19/00
G07F 19/00

(21)Application number : 2000-367387
(22)Date of filing : 01.12.2000

(71)Applicant : NTT COMMUNICATIONS KK
(72)Inventor : OMA YASUYUKI
SUGANUMA TOMOHISA
MASUTSUKA KOUJI
SUZUKI KENICHI

(54) ELECTRONIC VALUE TRANSACTION SYSTEM, METHOD, TERMINAL DEVICE, AND CENTER DEVICE

(57)Abstract:
PROBLEM TO BE SOLVED: To prevent infraudulence for individual processing, to appropriately correspond to the abnormality generation, and to provide quick value transaction.
SOLUTION: When the balance of the value recorded by an IC(integrated circuit) card 1 is renewed by a terminal control part 21 of a terminal device 2, a renewal command with renewal data added thereto used for the renewal processing is transmitted to the IC card 1, the card control part 11 of the IC cad 1 receives a renewal command from the terminal device 2 and executes a prescribed renewal application so that a plurality of processing for renewing the value balance stored in a memory is collectively executed using the renewal data added to the renewal command.



Nintendo Exhibit 1003
Part 2 of 2
Page 001048

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－170063
（P2002－170063A）
(43)公開日　平成14年６月14日(2002.6.14)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード(参考)
Ｇ０６Ｆ　17/60	４１０	Ｇ０６Ｆ　17/60	４１０Ｅ	３Ｅ０４０
Ｇ０６Ｋ　17/00		Ｇ０６Ｋ　17/00	Ｒ	５Ｂ０３５
			Ｔ	５Ｂ０５８
			Ｌ	
19/00		19/00	Ｕ	

審査請求　未請求　請求項の数18　ＯＬ　（全 14 頁）　最終頁に続く

(21)出願番号　特願2000－367387(P2000－367387)

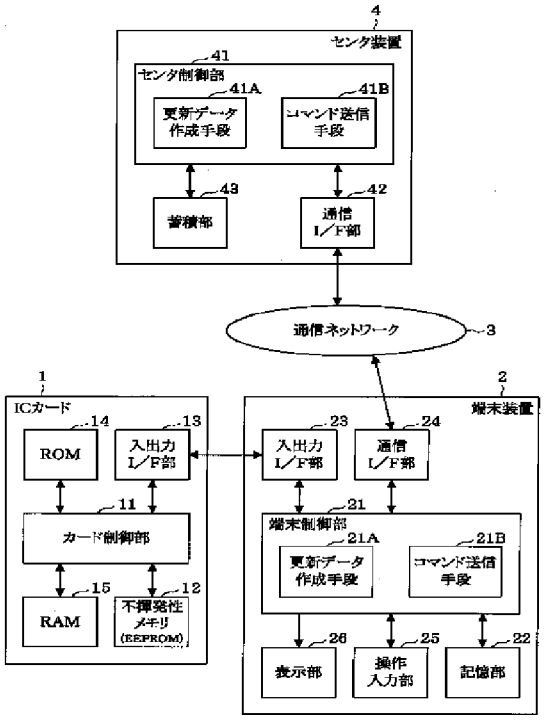
(22)出願日　平成12年12月１日(2000.12.1)

(71)出願人　399035766
エヌ・ティ・ティ・コミュニケーションズ株式会社
東京都千代田区内幸町一丁目１番６号
(72)発明者　大間　康之
東京都千代田区内幸町一丁目１番６号　エヌ・ティ・ティ・コミュニケーションズ株式会社内
(74)代理人　100064621
弁理士　山川　政樹

最終頁に続く

(54)【発明の名称】　電子価値取引システムおよび方法、電子価値取引端末装置、電子価値取引センタ装置

(57)【要約】
【課題】　個別処理に対する不正を防止できるとともに異常発生にも適切に対応でき、また迅速なバリュー取引を実現できるようにする。
【解決手段】　端末装置２の端末制御部２１で、ＩＣカード１で記憶するバリューの残高を更新する際、その更新処理に用いる更新データが付加された更新コマンドをＩＣカード１へ送信し、ＩＣカード１のカード制御部１１で、端末装置２からの更新コマンドを受信して所定の更新アプリケーションを実行することにより、その更新コマンドに付加されている更新データを用いて、前記メモリに記憶しているバリュー残高を更新するための複数の処理を一括して実行する。



Nintendo Exhibit 1003
Part 2 of 2
Page 001049

1

【特許請求の範囲】

【請求項１】　電子価値残高を内部のメモリで記憶し前記電子価値残高に対する更新コマンドに応じて更新アプリケーション・プログラムを実行することにより、前記電子価値残高を更新するＩＣカードと、

接続された前記ＩＣカードに対して前記更新コマンドを送信することにより、そのＩＣカードに記憶されている前記電子価値残高の更新を指示する端末装置とを備え、

前記端末装置は、電子価値の更新に用いる更新データが付加された更新コマンドを前記ＩＣカードへ送信する端末制御部を有し、

前記ＩＣカードは、前記端末装置からの更新コマンドを受信して対応する更新アプリケーション・プログラムを実行することにより、その更新コマンドに付加されている更新データを用いて、前記メモリに記憶している電子価値残高を更新するための複数の処理を一括して実行するカード制御部を有することを特徴とする電子価値取引システム。

【請求項２】　請求項１記載の電子価値取引システムにおいて、

前記ＩＣカードのカード制御部は、前記更新コマンドが電子価値残高の増額を指示する場合、電子価値残高を更新するための複数の処理として、前記更新データの正当性を検証するデータ正当性検証処理と、その正当性が検証された場合に前記メモリに記憶している電子価値残高を前記更新データに基づき増額する残高増額処理とからなる少なくとも２つの処理を一括して実行することを特徴とする電子価値取引システム。

【請求項３】　請求項２記載の電子価値取引システムにおいて、

前記ＩＣカードのカード制御部は、前記データ正当性検証処理として、前記更新データの再利用を防止するための一回性保証確認処理を含めて実行することを特徴とする電子価値取引システム。

【請求項４】　請求項１記載の電子価値取引システムにおいて、

前記ＩＣカードのカード制御部は、前記更新コマンドが電子価値残高の減額を指示する場合、電子価値残高を更新するための複数の処理として、前記メモリに記憶している電子価値残高を前記更新データに基づき減額する残高減額処理と、その減額処理を前記更新データの発行元に対して証明するための取引証明情報を作成する取引証明情報作成処理とからなる少なくとも２つの処理を一括して実行することを特徴とする電子価値取引システム。

【請求項５】　請求項１記載の電子価値取引システムにおいて、

通信ネットワークを介して前記端末装置と接続し、その端末装置に接続されている前記ＩＣカードに対する電子価値残高の更新を指示するセンタ制御部を有するセンタ装置をさらに備え、

2

このセンタ装置のセンタ制御部は、電子価値残高の更新に用いる更新データを作成する更新データ作成手段と、この更新データ作成部で作成された更新データを前記ＩＣカードに対する電子価値残高の更新を指示する更新コマンドに付加して前記端末装置へ送信するコマンド送信手段とを有し、

前記端末装置の端末制御部は、前記センタ装置から受信した更新コマンドを前記ＩＣカードへ転送することを特徴とする電子価値取引システム。

【請求項６】　請求項１記載の電子価値取引システムにおいて、

前記端末装置の端末制御部は、電子価値残高の更新に用いる更新データを作成する更新データ作成手段と、この更新データ作成部で作成された更新データを前記ＩＣカードに対する電子価値残高の更新を指示する更新コマンドに付加して前記ＩＣカードへ送信するコマンド送信手段とを有することを特徴とする電子価値取引システム。

【請求項７】　請求項１記載の電子価値取引システムにおいて、

通信ネットワークを介して前記端末装置と接続し、その端末装置に接続されている前記ＩＣカードに対する電子価値残高の更新を指示するセンタ制御部を有するセンタ装置をさらに備え、

このセンタ装置のセンタ制御部は、電子価値残高の更新に用いる更新データを作成して前記端末装置へ送信する更新データ作成手段を有し、

前記端末装置の端末制御部は、前記センタ装置から受信した更新データを、前記ＩＣカードに対する電子価値残高の更新を指示する更新コマンドに付加して前記ＩＣカードへ送信するコマンド送信手段とを有することを特徴とする電子価値取引システム。

【請求項８】　電子価値残高を内部のメモリで記憶し前記電子価値残高に対する更新コマンドに応じて更新アプリケーション・プログラムを実行することにより、前記電子価値残高を更新するＩＣカードと、接続された前記ＩＣカードに対して前記更新コマンドを送信することにより、そのＩＣカードに記憶されている前記電子価値残高の更新を指示する端末装置とを用いて、電子価値の取引を行う電子価値取引方法であって、

前記端末装置は、電子価値の更新に用いる更新データが付加された更新コマンドを前記ＩＣカードへ送信し、

前記ＩＣカードは、前記端末装置からの更新コマンドを受信して対応する更新アプリケーション・プログラムを実行することにより、その更新コマンドに付加されている更新データを用いて、前記メモリに記憶している電子価値残高を更新するための複数の処理を一括して実行することを特徴とする電子価値取引方法。

【請求項９】　請求項８記載の電子価値取引方法において、

前記ＩＣカードは、、前記更新コマンドが電子価値残高

Nintendo Exhibit 1003
Part 2 of 2
Page 001050

の増額を指示する場合、電子価値残高を更新するための複数の処理として、前記更新データの正当性を検証するデータ正当性検証処理と、その正当性が検証された場合に前記メモリに記憶している電子価値残高を前記更新データに基づき増額する残高増額処理とからなる少なくとも２つの処理を一括して実行することを特徴とする電子価値取引方法。

【請求項１０】　請求項９記載の電子価値取引方法において、
前記ＩＣカードは、前記データ正当性検証処理として、前記更新データの再利用を防止するための一回性保証確認処理を含めて実行することを特徴とする電子価値取引方法。

【請求項１１】　請求項８記載の電子価値取引方法において、
前記ＩＣカードは、前記更新コマンドが電子価値残高の減額を指示する場合、電子価値残高を更新するための複数の処理として、前記メモリに記憶している電子価値残高を前記更新データに基づき減額する残高減額処理と、その減額処理を前記更新データの発行元に対して証明するための取引証明情報を作成する取引証明情報作成処理とからなる少なくとも２つの処理を一括して実行することを特徴とする電子価値取引方法。

【請求項１２】　請求項８記載の電子価値取引方法において、
通信ネットワークを介して前記端末装置と接続されたセンタ装置は、電子価値残高の更新に用いる更新データを作成し、その更新データを前記ＩＣカードに対する電子価値残高の更新を指示する更新コマンドに付加して前記端末装置へ送信し、
前記端末装置は、前記センタ装置から受信した更新コマンドを前記ＩＣカードへ送信することを特徴とする電子価値取引方法。

【請求項１３】　請求項８記載の電子価値取引方法において、
前記端末装置は、電子価値残高の更新に用いる更新データを作成し、その更新データを前記ＩＣカードに対する電子価値残高の更新を指示する更新コマンドに付加して前記端末装置へ送信することを特徴とする電子価値取引方法。

【請求項１４】　請求項８記載の電子価値取引方法において、
通信ネットワークを介して前記端末装置と接続されたセンタ装置は、電子価値残高の更新に用いる更新データを作成して前記端末装置へ送信し、
前記端末装置は、前記センタ装置から受信した更新データを、前記ＩＣカードに対する電子価値残高の更新を指示する更新コマンドに付加して前記ＩＣカードへ送信するコマンド送信手段とを有することを特徴とする電子価値取引方法。

【請求項１５】　電子価値残高を内部のメモリで記憶し前記電子価値残高に対する更新コマンドに応じて更新アプリケーション・プログラムを実行することにより、前記電子価値残高を更新するＩＣカードと、接続された前記ＩＣカードに対して前記更新コマンドを送信することにより、そのＩＣカードに記憶されている前記電子価値残高の更新を指示する電子価値取引端末装置とを有する電子価値取引システムにおいて、
電子価値の更新に用いる更新データが付加された更新コマンドを前記ＩＣカードへ送信して、前記ＩＣカードで対応する更新アプリケーション・プログラムを実行させることにより、その更新コマンドに付加されている更新データを用いた、前記メモリに記憶している電子価値残高を更新するための複数の処理の一括実行を指示する端末制御部を備えることを特徴とする電子価値取引端末装置。

【請求項１６】　請求項１５記載の電子価値取引端末装置において、
前記端末制御部は、電子価値残高の更新に用いる更新データを作成する更新データ作成手段と、この更新データ作成部で作成された更新データを前記ＩＣカードに対する電子価値残高の更新を指示する更新コマンドに付加して前記ＩＣカードへ送信するコマンド送信手段とを有することを特徴とする電子価値取引端末装置。

【請求項１７】　電子価値残高を内部のメモリで記憶し前記電子価値残高に対する更新コマンドに応じて更新アプリケーション・プログラムを実行することにより、前記電子価値残高を更新するＩＣカードと、接続された前記ＩＣカードに対して前記更新コマンドを送信することにより、そのＩＣカードに記憶されている前記電子価値残高の更新を指示する端末装置とを有する電子価値取引システムにおいて、
通信ネットワークを介して前記端末装置と接続され、電子価値の更新に用いる更新データが付加された更新コマンドを前記端末装置を介して前記ＩＣカードへ送信して、前記ＩＣカードで対応する更新アプリケーション・プログラムを実行させることにより、その更新コマンドに付加されている更新データを用いた、前記メモリに記憶している電子価値残高を更新するための複数の処理の一括実行を指示するセンタ制御部を備えることを特徴とする電子価値取引センタ装置。

【請求項１８】　請求項１７記載の電子価値取引センタ装置において、
前記センタ制御部は、電子価値残高の更新に用いる更新データを作成する更新データ作成手段と、この更新データ作成部で作成された更新データを前記ＩＣカードに対する電子価値残高の更新を指示する更新コマンドに付加し前記端末装置を介して前記ＩＣカードへ送信するコマンド送信手段とを有することを特徴とする電子価値取引センタ装置。

Nintendo Exhibit 1003
Part 2 of 2
Page 001051

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、電子価値取引システムおよび方法、電子価値取引端末装置、電子価値取引センタ装置に関し、特にバリュー（電子価値）を内部のメモリで記憶するＩＣカードを端末装置に接続し、端末装置からの更新指示に応じてＩＣカードで所定アプリケーション・プログラムを実行することによりそのバリューを増減する電子価値取引システムおよび方法、電子価値取引端末装置、電子価値取引センタ装置に関するものである。

【０００２】

【従来の技術】インターネットをはじめとする通信ネットワークの普及に伴って、ネットワーク上での電子商取引が導入されつつある。このようなネットワーク上での電子商取引で決済を行う場合、クレジットカードなどを用いた口座引き落とし決済のほか、貨幣価値を有する電子マネーなどのバリュー（電子価値）を移動させて決済する方法がある。また、貨幣価値に準じるものとして、商品券、回数券、食券、サービスポイントなどのバリューを取り引きするサービスもある。このようなバリューを利用者が容易に取り引きするシステムとして、名刺サイズのカードにマイクロプロセッサを埋め込んだＩＣカードにバリューを記録して利用者が携帯し、銀行や店舗に設置された端末装置にそのＩＣカードを接続してバリューを取り引きする電子価値取引システムが提案されている。

【０００３】このような電子価値取引システムでは、ＩＣカードに記録されているバリューを更新する場合、端末装置とＩＣカードとの間でデータ通信を行うことにより、バリューの増額または減額を指示する更新コマンドや、そのバリューの更新に用いる更新データがやり取りされる。そして、ＩＣカード内で所定のアプリケーション・プログラムが実行され、ＩＣカード内部のメモリに記録されているバリューが更新される。この際、バリューの不正な更新を防止するため、バリューの更新処理に加えて、複数の処理を実行する必要がある。ここで、従来の電子価値取引システムでは、ＩＣカードへのデータ書き込みを指示する書き込みコマンド（UPDATE RECORD）やＩＣカードからのデータ読み込みを指示する読み込みコマンド（READ RECORD）などを組み合わせることにより、バリューの更新に必要な処理の指示、その処理に用いる更新データ、さらには処理結果を示す結果データを、端末装置とＩＣカードとの間で順にやり取りするものとなっていた。

【０００４】

【発明が解決しようとする課題】しかしながら、このような従来の電子価値取引システムでは、端末装置とＩＣカードとの間で、複数回にわたって、バリューの更新に必要な処理の指示コマンド、その処理に用いる更新データ、さらには処理結果を示す結果データをやり取りするものとなっていたため、バリュー更新時の各処理の連携による不正防止が機能しなくなる可能性があるという問題点があった。例えば、バリューを増額する際、ＩＣカードで更新指示や更新データの正当性が検証された場合だけバリューを更新することにより不正を防止できる。しかし、書き込み／読み込みコマンドなどを用いて複数回に分けて処理を行う場合は、各処理の連携が損なわれる可能性もあり、正当性の検証を行うことなくバリュー増額処理を行うなどの不正が行われる場合も考えられる。

【０００５】また、バリューを減額する際には、ＩＣカードでバリュー残高を減額した後、取引証明情報を作成して送信し、この取引証明情報の正当性を端末装置で検証することにより不正を防止できる。しかし、書き込み／読み込みコマンドなどを用いて複数回に分けて処理を行う場合は、各処理の連携が損なわれる可能性もあり、バリュー残高が減額されていないにもかかわらず、バリューの更新が正常終了したことを示す取引証明情報がＩＣカードから端末装置へ送信される可能性もある。さらに、端末装置とＩＣカードとの間でデータをやり取りする回数が増加して取引に要する処理時間が長くなり、バリューの迅速な取引を実現できないという問題点があった。本発明はこのような課題を解決するためのものであり、個別処理に対する不正を防止できるとともに異常発生にも適切に対応でき、また迅速なバリュー取引を実現できる電子価値取引システムおよび方法、電子価値取引端末装置、電子価値取引センタ装置を提供することを目的としている。

【０００６】

【課題を解決するための手段】このような目的を達成するために、本発明にかかる電子価値取引システムは、電子価値残高をメモリに記憶し外部からの電子価値残高に対する更新コマンドに応じて更新アプリケーションを実行することにより、電子価値残高を更新するＩＣカードと、接続されたＩＣカードに対して更新コマンドを送信することにより、そのＩＣカードに記憶されている電子価値残高の更新を指示する端末装置とを備え、端末装置は、電子価値の更新に用いる更新データが付加された更新コマンドをＩＣカードへ送信する端末制御部を有し、ＩＣカードは、端末装置からの更新コマンドを受信して対応する更新アプリケーションを実行することにより、その更新コマンドに付加されている更新データを用いて、メモリに記憶している電子価値残高を更新するための複数の処理を一括して実行するカード制御部を有するものである。

【０００７】電子価値残高を更新する際に一括して実行する複数の処理の具体例としては、更新コマンドが電子価値残高の増額を指示する場合、電子価値残高を更新するための複数の処理として、更新データの正当性を検証

Nintendo Exhibit 1003
Part 2 of 2
Page 001052

するデータ正当性検証処理と、その正当性が検証された場合にメモリに記憶している電子価値残高を更新データに基づき増額する残高増額処理とからなる少なくとも２つの処理を一括して実行するようにしてもよい。さらに、データ正当性検証処理として、更新データの再利用を防止するための一回性保証確認処理を含めて実行するようにしてもよい。

【０００８】このほか、電子価値残高を更新する際に一括して実行する複数の処理の具体例としては、更新コマンドが電子価値残高の減額を指示する場合、電子価値残高を更新するための複数の処理として、メモリに記憶している電子価値残高を更新データに基づき減額する残高減額処理と、その減額処理を更新データの発行元に対して証明するための取引証明情報を作成する取引証明情報作成処理とからなる少なくとも２つの処理を一括して実行するようにしてもよい。

【０００９】ＩＣカードへの電子価値更新を指示する形態としては、通信ネットワークを介して端末装置と接続し、その端末装置に接続されているＩＣカードに対する電子価値残高の更新を指示するセンタ制御部を有するセンタ装置をさらに備えて、このセンタ装置のセンタ制御部に、電子価値残高の更新に用いる更新データを作成する更新データ作成手段と、この更新データ作成部で作成された更新データをＩＣカードに対する電子価値残高の更新を指示する更新コマンドに付加して端末装置へ送信するコマンド送信手段とを設け、端末装置の端末制御部で、センタ装置から受信した更新コマンドをＩＣカードへ送信するようにしてもよい。

【００１０】ＩＣカードへの電子価値更新を指示する他の形態としては、端末装置の端末制御部に、電子価値残高の更新に用いる更新データを作成する更新データ作成手段と、この更新データ作成部で作成された更新データをＩＣカードに対する電子価値残高の更新を指示する更新コマンドに付加してＩＣカードへ送信するコマンド送信手段とを設け、端末装置から更新コマンドをＩＣカードへ送信するようにしてもよい。

【００１１】ＩＣカードへの電子価値更新を指示する他の形態としては、通信ネットワークを介して端末装置と接続し、その端末装置に接続されているＩＣカードに対する電子価値残高の更新を指示するセンタ制御部を有するセンタ装置をさらに備えて、このセンタ装置のセンタ制御部に、電子価値残高の更新に用いる更新データを作成して端末装置へ送信する更新データ作成手段を設けるとともに、端末装置の端末制御部に、センタ装置から受信した更新データを、ＩＣカードに対する電子価値残高の更新を指示する更新コマンドに付加してＩＣカードへ送信するコマンド送信手段とを設け、センタ装置で作成した更新データを端末装置で更新コマンドに付加してＩＣカードへ送信するようにしてもよい。

【００１２】また、本発明にかかる電子価値取引方法は、電子価値残高をメモリに記憶し外部からの電子価値残高に対する更新コマンドに応じて更新アプリケーションを実行することにより、電子価値残高を更新するＩＣカードと、接続されたＩＣカードに対して更新コマンドを送信することにより、そのＩＣカードに記憶されている電子価値残高の更新を指示する端末装置とを用いて、電子価値の取引を行う電子価値取引方法であって、端末装置で、電子価値の更新に用いる更新データが付加された更新コマンドをＩＣカードへ送信し、ＩＣカードで、端末装置からの更新コマンドを受信して対応する更新アプリケーションを実行することにより、その更新コマンドに付加されている更新データを用いて、メモリに記憶している電子価値残高を更新するための複数の処理を一括して実行するようにしたものである。

【００１３】電子価値残高を更新する際に一括して実行する複数の処理の具体例としては、ＩＣカードで、更新コマンドが電子価値残高の増額を指示する場合、電子価値残高を更新するための複数の処理として、更新データの正当性を検証するデータ正当性検証処理と、その正当性が検証された場合にメモリに記憶している電子価値残高を更新データに基づき増額する残高増額処理とからなる少なくとも２つの処理を一括して実行するようにしてもよい。さらに、ＩＣカードで、データ正当性検証処理として、更新データの再利用を防止するための一回性保証確認処理を含めて実行するようにしてもよい。

【００１４】このほか、電子価値残高を更新する際に一括して実行する複数の処理の具体例としては、ＩＣカードで、更新コマンドが電子価値残高の減額を指示する場合、電子価値残高を更新するための複数の処理として、メモリに記憶している電子価値残高を更新データに基づき減額する残高減額処理と、その減額処理を更新データの発行元に対して証明するための取引証明情報を作成する取引証明情報作成処理とからなる少なくとも２つの処理を一括して実行するようにしてもよい。

【００１５】ＩＣカードへの電子価値更新を指示する方法としては、通信ネットワークを介して端末装置と接続されたセンタ装置で、電子価値残高の更新に用いる更新データを作成し、その更新データをＩＣカードに対する電子価値残高の更新を指示する更新コマンドに付加して端末装置へ送信し、端末装置で、センタ装置から受信した更新コマンドをＩＣカードへ送信するようにしてもよい。また、端末装置で、電子価値残高の更新に用いる更新データを作成し、その更新データをＩＣカードに対する電子価値残高の更新を指示する更新コマンドに付加して端末装置へ送信するようにしてもよい。

【００１６】ＩＣカードへの電子価値更新を指示する他の方法としては、通信ネットワークを介して端末装置と接続されたセンタ装置で、電子価値残高の更新に用いる更新データを作成して端末装置へ送信し、端末装置で、センタ装置から受信した更新データを、ＩＣカードに対

Nintendo Exhibit 1003
Part 2 of 2
Page 001053

する電子価値残高の更新を指示する更新コマンドに付加してＩＣカードへ送信するようにしてもよい。

【００１７】また、本発明にかかる電子価値取引端末装置は、電子価値残高を内部のメモリで記憶し電子価値残高に対する更新コマンドに応じて更新アプリケーション・プログラムを実行することにより、電子価値残高を更新するＩＣカードと、接続されたＩＣカードに対して更新コマンドを送信することにより、そのＩＣカードに記憶されている電子価値残高の更新を指示する電子価値端末装置とを有する電子価値取引システムの電子価値端末装置に、電子価値の更新に用いる更新データが付加された更新コマンドをＩＣカードへ送信して、ＩＣカードで対応する更新アプリケーション・プログラムを実行させることにより、その更新コマンドに付加されている更新データを用いた、メモリに記憶している電子価値残高を更新するための複数の処理の一括実行を指示する端末制御部を設けたものである。

【００１８】電子価値取引端末装置の端末制御部の具体例として、電子価値残高の更新に用いる更新データを作成する更新データ作成手段と、この更新データ作成部で作成された更新データをＩＣカードに対する電子価値残高の更新を指示する更新コマンドに付加してＩＣカードへ送信するコマンド送信手段とを設けてもよい。

【００１９】また、本発明にかかる電子価値取引センタ装置は、電子価値残高を内部のメモリで記憶し電子価値残高に対する更新コマンドに応じて更新アプリケーション・プログラムを実行することにより、電子価値残高を更新するＩＣカードと、接続されたＩＣカードに対して更新コマンドを送信することにより、そのＩＣカードに記憶されている電子価値残高の更新を指示する端末装置とを有する電子価値取引システムにおいて、通信ネットワークを介して端末装置と接続された電子価値取引センタ装置であって、その電子価値取引センタ装置に、電子価値の更新に用いる更新データが付加された更新コマンドを端末装置を介してＩＣカードへ送信して、ＩＣカードで対応する更新アプリケーション・プログラムを実行させることにより、その更新コマンドに付加されている更新データを用いた、メモリに記憶している電子価値残高を更新するための複数の処理の一括実行を指示するセンタ制御部を設けたものである。

【００２０】電子価値取引センタ装置のセンタ制御部の具体例として、電子価値残高の更新に用いる更新データを作成する更新データ作成手段と、この更新データ作成部で作成された更新データをＩＣカードに対する電子価値残高の更新を指示する更新コマンドに付加し端末装置を介してＩＣカードへ送信するコマンド送信手段とを設けてもよい。

【００２１】

【発明の実施の形態】次に、本発明の実施の形態について図面を参照して説明する。図１は本発明の一実施の形態にかかる電子価値取引システムを示すブロック図である。この電子価値取引システムには、バリュー残高を記録するためのＩＣカード１、このＩＣカード１を接続してバリュー残高の更新を指示する端末装置２、および通信ネットワーク３を介して端末装置２と接続されたセンタ装置４が設けられている。ＩＣカード１は利用者により携帯され、店舗、サービス提供者や金融機関の支店に配置された端末装置１に接続してバリューの取引を行う。またセンタ装置４はサービス提供者や金融機関が運営し、ＩＣカード１の利用者の情報を管理するとともに、必要に応じてＩＣカード１に対するバリュー残高の更新を端末装置２へ指示する。

【００２２】ＩＣカード１には、端末装置２と接続してデータ通信を行うための入出力インターフェース部（以下、入出力Ｉ／Ｆ部という）１３、バリュー残高のほか、利用者の個人情報、デジタル署名や暗号化に用いる鍵情報、一回性保証の検証に用いる一回性保証情報、支払元特定情報などを記憶する不揮発性メモリ（例えば、ＥＥＰＲＯＭ）１２、ＩＣカード１内部で行う処理が記述されているアプリケーション・プログラムを格納するＲＯＭ１４、ＩＣカード１の各部を制御するとともに、入出力Ｉ／Ｆ部１３を介して端末装置１から受信した更新コマンドに応じて、ＲＯＭ１４からアプリケーション・プログラムを読み込んで実行することによりバリュー残高の更新処理や正当性検証処理などの各種処理を一括して行うカード制御部１１、各種データを一時的に記憶するＲＡＭ１５が設けられている。以下では、ＲＯＭ１４にアプリケーション・プログラムを格納しておく場合を例として説明するが、不揮発性メモリ（ＥＥＰＲＯＭ）にアプリケーション・プログラムを格納しておいてもよい。

【００２３】端末装置２には、ＩＣカード１と接続してデータ通信を行うための入出力インターフェース部（以下、入出力Ｉ／Ｆ部という）２３、通信ネットワーク３を介してセンタ装置４とデータ通信を行うための通信インターフェース部（以下、通信Ｉ／Ｆ部という）２４、各種情報や指示を入力するための操作入力部２５、各種情報を画面表示するための表示部２６、端末装置２の各部を制御するとともに、通信Ｉ／Ｆ部２４を介してセンタ装置４から受信した更新コマンドや内部で作成した更新コマンドを、この入出力Ｉ／Ｆ部２３を介してＩＣカード１へ更新コマンドを転送することにより、ＩＣカード１に対してバリュー残高の更新を指示する端末制御部２１、デジタル署名や暗号化に用いる鍵情報、支払先特定情報など、各種情報を記憶するための記憶部２２が設けられている。

【００２４】端末制御装部２１には、操作入力部２５からの操作入力に応じてＩＣカード１のバリューを更新するのに用いる更新データを作成する更新データ作成手段２１Ａと、この更新データ作成手段２１Ａで作成した更

Nintendo Exhibit 1003
Part 2 of 2
Page 001054

新データを更新コマンドに付加し、入出力Ｉ／Ｆ部２３を介して一括してＩＣカード１へ送信するコマンド送信手段２１Ｂとが設けられている。なお、これら更新データ作成手段２１Ａやコマンド送信手段２１Ｂでは、更新データや更新コマンドの正当性を保証するため、必要に応じて記憶部２２で記憶している鍵情報を用いて、これら更新データや更新コマンドに対するデジタル署名や暗号化処理が行われる。

【００２５】センタ装置４には、通信ネットワーク３を介して端末装置２と通信を行うための通信インターフェース部（以下、通信Ｉ／Ｆ部という）４２、ＩＣカード１の利用者の利用者情報やデジタル署名や暗号化に用いる鍵情報など、各種情報を蓄積するための蓄積部４３、センタ装置４の各部を制御するとともに、必要に応じて内部で作成した更新コマンドを、通信Ｉ／Ｆ部４２から通信ネットワーク３を介して端末装置２へ送信することにより、ＩＣカード１に対するバリュー残高の更新を指示するセンタ制御部４１が設けられている。

【００２６】センタ制御部４１には、端末装置２側からの要求に応じてＩＣカード１のバリューを更新するのに用いる更新データを作成する更新データ作成手段４１Ａと、この更新データ作成手段４１Ａで作成した更新データを更新コマンドに付加し、通信Ｉ／Ｆ部４２から通信ネットワーク３を介して端末装置２へ一括して送信するコマンド送信手段４１Ｂとが設けられている。なお、これら更新データ作成手段４１Ａやコマンド送信手段４１Ｂでは、更新データや更新コマンドの正当性を保証するため、必要に応じて蓄積部４３で記憶している鍵情報を用いて、これら更新データや更新コマンドに対するデジタル署名や暗号化処理が行われる。

【００２７】ＩＣカード１は、名刺サイズのカードにＣＰＵや数値演算チップなどのマイクロプロセッサを埋め込んだ装置であり、カード制御部１１はこのマイクロプロセッサからなるハードウェア資源と、予めマイクロプロセッサに格納されているプログラムやＲＯＭ１４に格納されているアプリケーション・プログラムなどのソフトウェア資源とを協動させることにより各種機能を提供している。また、端末装置２の端末制御部２１やセンタ装置４のセンタ制御部４１の各手段は、同じくＣＰＵや数値演算チップなどのマイクロプロセッサからなるハードウェア資源と、予めマイクロプロセッサに格納されているプログラムや記憶部２２あるいは蓄積部４３に格納されているアプリケーション・プログラムなどのソフトウェア資源とを協動させることにより各種機能を提供している。

【００２８】次に、図２を参照して、本発明の第１の実施の形態について説明する。図２は第１の実施の形態にかかる電子価値取引方法を示すシーケンス図である。以下では、センタ装置３が銀行に配置され、端末装置２が銀行窓口の自動現金支払装置（ＡＴＭ）に相当し、貨幣価値を有するバリューを銀行の利用者口座からＩＣカード１へ転送する場合、すなわちＩＣカード１で記憶するバリュー残高を増額する場合を例として説明する。

【００２９】利用者は、自己の利用者口座からＩＣカード１のバリュー残高を増額（チャージ）する場合、まず端末装置２に接続し、端末装置２の操作入力部２５を操作して増額指示やその金額、さらには暗証番号などを入力する。これにより、センタ装置４（または端末装置２）でＩＣカード１の認証が行われ（ステップ１１０）、端末装置２からセンタ装置４へ増額要求が行われる（ステップ１１１）。このとき、ＩＣカード１の不揮発性メモリ１２には、バリュー残高のほか、センタ装置４からの更新データや更新コマンドの正当性を検証するために用いるバリュー共通鍵、更新コマンドが再使用されていないという一回性保証を検証するための一回性保証情報などが予め格納されているものとする（ステップ１０１）。

【００３０】また、センタ装置４の蓄積部４３には、センタ装置４で作成した更新データや更新コマンドの正当性を検証するために用いるバリュー共通鍵、更新コマンドが再使用されていないという一回性保証を検証するための一回性保証情報などが予め格納されているものとする（ステップ１０２）。バリュー共通鍵とは、共通暗号化方式に基づき作成されたデジタル署名や暗号化のための暗号鍵である。このバリュー共通鍵は、ＩＣカード１とセンタ装置４とで予め共有して保管されており、このバリュー共通鍵でデジタル署名や暗号化した更新データや更新コマンドについては、同一バリュー共通鍵を用いてその発行元の正当性を検証できる。

【００３１】一回性保証情報とは、更新コマンドの再使用を防止するための情報であり、例えばＩＣカード１とセンタ装置４とで、そのＩＣカード１で実行したバリュー更新回数を一回性保証情報として管理しておく。そして、センタ装置４からの更新コマンド内にその更新コマンドが適用される場合のバリュー更新回数を指定し、ＩＣカード１で自己が管理するバリュー更新回数と更新コマンドで指定されたバリュー更新回数との一致／不一致を検証することにより、その更新コマンドの一回性保証を検証できる。なお、一回性保証情報については、公知のデジタル署名や暗号化方式を用いて、ＩＣカード１を端末装置２に接続した際に読み出し、センタ装置４へ通知するようにしてもよい。これにより、センタ装置４におけるバリュー共通鍵の管理処理を軽減できる。

【００３２】センタ装置４のセンタ制御部４１では、ＩＣカード１に対してバリューを移動させるため、その担保となる利用者口座からＩＣカード１のバリューを管理するための発行体口座へ、増額要求で利用者から指定されたバリュー増額分だけ資金移動（振り替え）を行う（ステップ１１２）。この資金移動については、蓄積部４３で利用者口座や発行体口座を管理している場合に

Nintendo Exhibit 1003
Part 2 of 2
Page 001055

は、それぞれの口座情報を更新すればよい。また、別途、口座管理装置を設けている場合には、センタ装置４からその口座管理装置に対して資金移動を指示すればよい。

【００３３】次に、センタ制御部４１では、更新データ作成手段４１Ａを用いて、ＩＣカード１のバリューを増額するための更新データ、すなわちバリュー増額データを作成する（ステップ１１２）。このバリュー増額データには、バリュー増額分を示すチャージ額や一回性保証情報のほか、バリューの有効期限などが含まれている。続いて、蓄積部４３に格納されている当該ＩＣカード１に格納されているバリュー共通鍵を用いて、バリュー増額データにデジタル署名や暗号化処理を行う（ステップ１１４）。コマンド送信手段４１Ｂでは、このようにして作成されたバリュー増額データを、バリュー増額を指示するチャージコマンドに付加し、これらチャージコマンドとバリュー増額データとを一括してＩＣカード１へ送信する（ステップ１１５）。

【００３４】これにより、チャージコマンドおよびバリュー増額データは、センタ装置４の通信Ｉ／Ｆ部４２から通信ネットワーク３へ送信され、端末装置２の通信Ｉ／Ｆ部２４を介して端末制御部２１で受信される。端末制御部２１のコマンド送信手段２１Ｂでは、センタ装置４からＩＣカード１向けのチャージコマンドおよびバリュー増額データを、そのまま一括して入出力Ｉ／Ｆ部２３からＩＣカード１へ転送する。ＩＣカード１のカード制御部１１では、端末装置２で転送されたセンタ装置４からのチャージコマンドおよびバリュー増額データを一括して受信し、そのチャージコマンドに応じたアプリケーション・プログラムをＲＯＭ１４から読み出して実行することにより、チャージコマンド処理を開始する（ステップ１１６）。

【００３５】チャージコマンド処理では、まず、不揮発性メモリ１２で保持しているバリュー共通鍵を用いて、そのチャージコマンドと一括して受信したバリュー増額データのデジタル署名を検証し、またはバリュー増額データの復号化処理を行うことにより、そのバリュー増額データの正当性を検証する（①増額バリューの正当性検証）。また、不揮発性メモリ１２で保持している一回性保証情報と、受信したバリュー増額データに含まれる一回性保証情報とを比較して、そのバリュー増額データの一回性保証を確認する（②増額バリューの一回性保証確認）。

【００３６】そして、これら正当性の検証や一回性保証の確認で正常な結果が得られた後、受信したバリュー増額データに含まれるチャージ額に基づき、不揮発性メモリ１２に格納されているバリュー残高を増額する（③バリュー増額）。また、受信したバリュー増額データに有効期限情報が含まれる場合は、不揮発性メモリ１２に格納されているバリューの有効期間を更新する（④有効期限更新）。カード制御部１１では、このような複数の処理からなるチャージコマンド処理を一括して実行する。そして、これら処理が正常に終了したか否かを処理結果として、センタ装置４へ送信する（ステップ１１７）。

【００３７】この処理結果は、チャージコマンドおよびバリュー増額データとは、逆の経路で端末装置２から通信ネットワーク３を介してセンタ装置４へ転送される。そして、センタ制御部４１で、その処理結果を受信し、ＩＣカード１に対するチャージコマンドが正常に処理されたことが確認される（ステップ１１８）。また、チャージコマンド処理中に正当性検証や一回性保証確認で、正常な結果が得られなかった場合には、端末装置２での判断を待つことなくカード制御部１１で直ちに処理を中止でき、不正なバリュー増額指示に対応できる。また、チャージコマンド処理中に端末装置２との通信異常が生じた場合でも、チャージコマンド処理を最後まで継続して実行することができ、その後の通信正常化に応じてその処理結果をセンタ装置４へ通知できる。

【００３８】このように、本実施の形態では、端末装置２の端末制御部２１で、ＩＣカード１で記憶するバリューの残高を更新する際、その更新処理に用いる更新データが付加された更新コマンドをＩＣカード１へ送信し、ＩＣカード１のカード制御部１１で、端末装置２からの更新コマンドを受信して所定の更新アプリケーションを実行することにより、その更新コマンドに付加されている更新データを用いて、メモリに記憶しているバリュー残高を更新するための複数の処理を一括して実行するようにしたので、更新コマンドを１回送信するだけで一連の処理が実行される。したがって、従来のように、バリュー更新処理を構成する複数の処理をそれぞれ個別に実行する場合と比較して、個別処理に対する不正を防止できるとともに異常発生にも適切に対応できる。また、ＩＣカード１と端末装置２との間でのデータ通信回数を最低限に抑えることができ、迅速なバリュー取引を実現できる。

【００３９】また、ＩＣカード１に対してバリュー増額を指示する更新コマンドすなわちチャージコマンドに応じて、その更新処理に用いるバリュー増額データの正当性を確認する正当性確認処理と、実際にＩＣカードで記憶するバリュー残高を増額するバリュー増額処理との、少なくとも２つの処理をＩＣカード１で一括して行うようにしたので、バリューを増額する際に重要なこれら処理の連携が確保され、個々の処理に対する不正を確実に防止できる。また、この正当性処理として、一回性保証の確認処理を含むようにしたので、バリュー増額データやチャージコマンドの再使用という不正についても確実に防止できる。さらに、共通鍵暗号方式に基づくセンタ装置４とのバリュー共通鍵を用いて、ＩＣカード１においてバリュー増額データの正当性を確認するようにしたので、１回の検証処理で、バリュー増額データの内容だ

Nintendo Exhibit 1003
Part 2 of 2
Page 001056

けでなくその発行元についても検証でき、短時間で高いセキュリティレベルでの検証処理を実現できる。

【００４０】第１の実施の形態では、センタ装置４からの更新コマンドを端末装置２を介してＩＣカード１へ転送するようにした場合を例として説明したが、これに限定されるものではなく、端末装置２の端末制御部２１で、更新コマンドや更新データを作成してＩＣカード１へ送信するようにしてもよい。例えば、貨幣価値に準じるものとして、商品券、回数券、食券、サービスポイントなどのバリューを取り引きするサービスに本発明を適用する場合、サービス規模が限定されてセンタ装置４を配置する必要がなくなるため、これらバリューを端末装置２で発行するようにしてもよい。

【００４１】上記第１の実施の形態では、ＩＣカード１とセンタ装置４との間で共通鍵暗号方式を用いた場合を例として説明したがこれに限定されるものではなく、公開鍵暗号方式を用いてもよい。公開鍵暗号方式を用いる場合、共通鍵暗号化方式のように、予めＩＣカード１でセンタ装置４の公開鍵を保有しておいてもよいが、ＩＣカード１における更新コマンド実行までに、センタ装置４からその公開鍵を入手しておけばよく、例えば更新データと一括して入手するようにしてもよい。これにより、端末装置２で各ＩＣカードごとに公開鍵を保有管理する必要がなくなる。

【００４２】次に、図３を参照して、本発明の第２の実施の形態について説明する。図３は第２の実施の形態にかかる電子価値取引方法を示すシーケンス図である。以下では、店舗に配置された端末装置２を用いて、貨幣価値を有するバリューをＩＣカード１から端末装置２へ転送して、代金の支払いを行う場合、すなわちＩＣカード１で記憶するバリュー残高を減額する場合を例として説明する。

【００４３】利用者は、ＩＣカード１のバリュー残高を減額（ペイ）する場合、まず端末装置２に接続する。続いて、利用者は、端末装置２の操作入力部２５を操作して支払金額の確認、さらには暗証番号などを入力する。そして、端末装置２を管理している店舗の店員は、操作入力部２５を操作して、ＩＣカード１のバリューをその支払金額分だけ端末装置２側へ転送するよう、すなわちＩＣカード１のバリューを支払金額分だけ減額するよう指示する。これにより、端末装置２への減額要求が行われる（ステップ１３１）。

【００４４】このとき、ＩＣカード１の不揮発性メモリ１２には、バリュー残高のほか、端末装置２側でＩＣカード１からの取引証明情報の正当性を検証するために用いる公開鍵証明書、この取引証明情報に対してデジタル署名や暗号化処理を行うために用いるＩＣカード秘密鍵、およびＩＣカード１を他のＩＣカードと識別するための支払元特定情報などが予め格納されているものとする（ステップ１２１）。また、端末装置２の記憶部２２には、端末装置２を他の端末装置と識別するための支払先特定情報が予め格納されている（ステップ１２２）。

【００４５】ＩＣカード１のカード制御部１１では、端末装置２との間で相互認証が得られた後、不揮発性メモリ１２に格納されている公開鍵証明書、およびＩＣカード１を特定する支払元特定情報を読み出し、入出力Ｉ／Ｆ部１３を介して端末装置２へ送信する（ステップ１３２）。公開鍵証明書とは、ＩＣカード１（利用者）の公開鍵の正当性を証明する電子証明書であり、一般には第三者の認証機関からデジタル署名付きで発行される。ここでは、その公開鍵証明書をＩＣカード１に予め格納しておき、端末装置２へ配布される。このとき、ＩＣカード１の公開鍵も一緒に配布される。

【００４６】端末装置２の端末制御部２１では、ＩＣカード１からの公開鍵および公開鍵証明書を入出力Ｉ／Ｆ部２３を介して受信し、公開鍵の正当性を確認して記憶部２２へ格納しておく。これにより、端末装置２で各ＩＣカードごとに公開鍵を保有管理する必要がなくなる。また、この公開鍵証明書とともに受信したＩＣカード１の支払元特定情報や、操作入力部２５から入力された支払額を記憶部２２へ格納しておく。

【００４７】続いて、端末装置２の端末制御部２１では、更新データ作成手段２１Ａを用いて、ＩＣカード１のバリュー残高を減額するのに用いる更新データとして、バリュー減額データを作成する（ステップ１３３）。ここでは、動的データ認証用情報、ＩＣカード１のバリュー残高を減額する金額すなわち支払額、および端末装置２を特定するための支払先特定情報から、バリュー減額データを作成する。

【００４８】動的データ認証用情報とは、端末装置２が作成したバリュー減額データに基づきＩＣカード側でバリューの減額が行われたことを示す取引証明情報の正当性を、端末装置２で検証するために用いる情報であり、例えばバリュー減額データを作成した時刻や乱数などが用いられる。このような、端末装置２以外で不正に作成するのが困難な情報をバリュー減額データに含め、これをＩＣカード１側で取引証明情報に格納して返送するように取り決めておくことにより、取引証明情報の正当性を認証できる。したがって端末制御部２では、作成した動的データ認証用情報を記憶部２２へ格納しておく。

【００４９】コマンド送信手段２１Ｂでは、このようにして作成されたバリュー減額データを、バリュー減額を指示するペイコマンドに付加し、これらペイコマンドとバリュー減額データとを一括してＩＣカード１へ送信する（ステップ１３４）。これにより、ペイコマンドおよびバリュー減額データは、一括して入出力Ｉ／Ｆ部２３からＩＣカード１へ転送される。ＩＣカード１のカード制御部１１では、端末装置２からのペイコマンドおよびバリュー減額データを一括して受信し、そのペイコマンドに応じたアプリケーション・プログラムをＲＯＭ１４

Nintendo Exhibit 1003
Part 2 of 2
Page 001057

から読み出して実行することにより、ペイコマンド処理を開始する（ステップ１３５）。

【００５０】ペイコマンド処理では、まず、チャージコマンドと一括して受信したバリュー減額データの動的データ認証用情報、支払額および支払先特定情報と、不揮発性メモリ１２に格納されている支払元特定情報とから取引証明情報を作成する。そして、不揮発性メモリ１２に格納されているＩＣカード秘密鍵を用いて、その取引証明情報に対してデジタル署名あるいは暗号化処理を行う（❶取引証明情報の作成）。また、チャージコマンドと一括して受信したバリュー減額データの支払額に基づき、不揮発性メモリ１２に格納されているバリュー残高を減額する（❷バリュー減額）。カード制御部１１では、このような複数の処理からなるペイコマンド処理を一括して実行する。そして、その処理で作成された取引証明情報を処理結果として、端末装置２へ送信する（ステップ１３６）。

【００５１】端末装置２の端末制御部２１では、入出力Ｉ／Ｆ部２３を介してＩＣカード１からの取引証明情報を受信し、記憶部２２に格納してあるＩＣカード公開鍵を用いて、その取引証明情報のデジタル署名を検証し、または取引証明情報の復号化処理を行うことにより、その取引証明情報発行元の正当性を検証する。さらに、その取引証明情報に含まれている動的データ認証用情報と記憶部２２に格納しておいた動的データ認証用情報とを照合し、支払額、支払先特定情報および支払元特定情報についても、同様にして記憶部２２に格納されているものと照合することにより、取引証明情報の内容についても検証する。これにより、ＩＣカード１に対するペイコマンドが正常に処理されたことが確認される（ステップ１３７）。

【００５２】このように、本実施の形態では、端末装置２の端末制御部２１で、ＩＣカード１で記憶するバリューの残高を更新する際、その更新処理に用いる更新データが付加された更新コマンドをＩＣカード１へ送信し、ＩＣカード１のカード制御部１１で、端末装置２からの更新コマンドを受信して所定の更新アプリケーションを実行することにより、その更新コマンドに付加されている更新データを用いて、メモリに記憶しているバリュー残高を更新するための複数の処理を一括して実行するようにしたので、更新コマンドを１回送信するだけで一連の処理が実行される。したがって、従来のように、バリュー更新処理を構成する複数の処理をそれぞれ個別に実行する場合と比較して、個別処理に対する不正を防止できるとともに異常発生にも適切に対応できる。また、ＩＣカード１と端末装置２との間でのデータ通信回数を最低限に抑えることができ、バリューの迅速な取引を実現できる。

【００５３】また、ＩＣカード１に対してバリュー減額を指示する更新コマンドすなわちペイコマンドに応じて、その更新処理を証明する取引証明情報を作成するための取引証明情報作成処理と、実際にＩＣカードで記憶するバリュー残高を減額するバリュー減額処理との、少なくとも２つの処理をＩＣカード１で一括して行うようにしたので、バリューを減額する際に重要なこれら処理の連携が確保され、個々の処理に対する不正を確実に防止できる。さらに、公開鍵暗号方式に基づくＩＣカード１のＩＣカード公開鍵を用いて、端末装置４において取引証明情報の正当性を確認するようにしたので、１回の検証処理で、取引証明情報の内容だけでなくその発行元についても検証でき、短時間で高いセキュリティレベルでの検証処理を実現できる。

【００５４】上記第２の実施の形態では、ＩＣカード１と端末装置２との間で公開鍵暗号方式を用いた場合を例として説明したがこれに限定されるものではなく、共通鍵暗号方式を用いてもよい。共通鍵暗号方式を用いる場合には、ＩＣカード１および端末装置２の双方で共通鍵を共有しておけばよい。したがって、図３のステップ１３２のように、ＩＣカード１から端末装置２へ公開鍵および公開鍵証明書を送信する必要がなくなる。

【００５５】また、上記の第１および第２の実施の形態では、端末装置２またはセンタ装置４のいずれか一方で、更新データを作成するとともに、その更新データを更新コマンドに付加してＩＣカード１へ送信する場合について説明したが、センタ装置２の更新データ作成手段４１Ａで作成した更新データを、通信ネットワーク３を介して端末装置２へ送信し、この更新データを端末装置２のコマンド送信部２１Ｂで受信し、更新コマンドに付加してＩＣカード１へ送信するようにしてもよい。この際、上記のように、センタ装置４で更新データにデジタル署名または暗号化処理を施しておけば、端末装置２で更新データの改竄を行うことができないため、セキュリティ上の問題はない。

【００５６】

【実施例】次に、図４および図５を参照して、本発明にかかる実施例について説明する。図４は、バリューチャージ処理を示すシーケンス図、図５はバリューペイ処理を示すシーケンス図である。以下では、上記のＩＣカード１で貨幣価値を有するバリューを取り扱うものとし、そのバリューでの決済時に、銀行に設置されるセンタ装置４とバリュー取引を行わず、ＩＣカード１と端末装置２との間でバリュー取引を行うオフライン取引の場合を例として説明する。以下の処理は、従来のクレジットカード用ＩＣカードの標準仕様であるＥＭＶ（Europay MasterCard International VisaInternational）の規定処理を一部拡張した処理となっている。なお、本発明はＥＭＶの規定処理に準拠しないシステムに対しても適用可能である。

【００５７】図４に示したバリューチャージ処理では、まず端末装置２の端末制御部２１においてＩＣカード１

Nintendo Exhibit 1003
Part 2 of 2
Page 001058

の実装する銀行アプリケーション・プログラム（以下、ＡＰという）選択処理を行う（ステップ２００）。ここでは、ＩＣカード１のＲＯＭ１４に格納されているＡＰが端末制御部２１へ通知され、その中からバリューチャージ処理に必要なＡＰが選択される。次に、選択した銀行ＡＰを起動するＡＰ起動処理を行う（ステップ２０１）。これにより、ＩＣカード１のカード制御部１１は、指定されたＡＰをＲＯＭ１４から読み出して起動する。そして、起動した各ＡＰの実行に必要なＡＰデータをＡＰデータ読込処理により、端末装置２側へ読み込んで記憶部２２へ格納する（ステップ２０２）。

【００５８】次に、端末制御部２１では、端末アクション分析を行う（ステップ２０３）。この処理は、これから行う取引をセンタ装置４を含むオンライン処理か、センタ装置４を含まないオフライン処理かを分析し決定する。ここでは、オフライン処理が選択される。続いて、オンライン処理により、取引内容を示す取引データをセンタ装置４へ送信して取引内容の確認を促す（ステップ２０４）。これにより、センタ装置４からＩＣカードとの認証が得られた後に実行すべきコマンドがイシュア・スクリプト（Issure Script）により指定される。これに応じて、端末制御部２１では、ＩＣカード１へセンタ装置４とのオンライン処理が終了したことを通知し、ＩＣカード１におけるバリュー発行元すなわち銀行側の発行者認証処理を行う（ステップ２０５）。そして、ＥＭＶ規定処理の完了を確認する完了処理を行う（ステップ２０６）。

【００５９】その後、先のイシュア・スクリプトでセンタ装置４から指示されたコマンド、すなわちチャージコマンドを実行するためのスクリプト処理を行う（ステップ２０７）。これにより、端末制御部２１の更新データ作成手段２１Ａで更新データが作成され、この更新データが付加されたチャージコマンドがコマンド送信手段２１ＢによりＩＣカード１へ一括して送信される。ＩＣカード１のカード制御部１１では、このチャージコマンドに応じて、第１の実施の形態で説明したチャージコマンド処理を実行することにより、バリュー増額のための複数の処理が一括して行われ（ステップ２０８）、一連の処理を終了する。

【００６０】また図５で示したバリューペイ処理では、図４のステップ２００～２０２まで同様の処理を行い、ＡＰデータ読込処理（ステップ２０２）でＩＣカード１から読み込んだ公開鍵証明書の検証を行う（ステップ２１３）。この後、ＩＣカード利用者がバリューペイ処理の正規の対象であることを確認するオフラインＰＩＮ検証を行う（ステップ２１４）。続いて、図４のステップ２０３と同様の端末アクション分析を行う。このバリューペイ処理では、センタ装置４を含まないオフライン処理と決定される。そして、ＥＭＶ規定処理の完了を確認する完了処理を行う（ステップ２０６）。

【００６１】その後、端末制御部２１では、ＩＣカード１のバリューを減額するためのペイコマンドを実行する（ステップ２１７）。これにより、端末制御部２１の更新データ作成手段２１Ａで更新データが作成され、この更新データが付加されたペイコマンドがコマンド送信手段２１ＢによりＩＣカード１へ一括して送信される。ＩＣカード１のカード制御部１１では、このペイコマンドに応じて、第２の実施の形態で説明したペイコマンド処理を実行することにより、バリュー減額のための複数の処理が一括して行われることになる（ステップ２１８）。端末制御部２１では、ＩＣカード１からの応答として受信したデジタル署名付きの取引証明情報に対する検証を行うことにより、取引内容およびＩＣカードの正当性を確認し（ステップ２１９）、一連の処理を終了する。

【００６２】このように、ＩＣカード１で貨幣価値を有するバリューを取り扱うとともに、ＩＣカード１と端末装置２との間でオフラインのバリュー取引を行う電子価値取引システムに適用したので、このようなのバリュー取引においても、従来のように、バリュー更新処理を構成する複数の処理をそれぞれ個別に実行する場合と比較して、個別処理に対する不正を防止できるとともに異常発生にも適切に対応できる。また、ＩＣカード１と端末装置２との間でのデータ通信回数を最低限に抑えることができ、迅速なバリュー取引を実現できる。

【００６３】

【発明の効果】以上説明したように、本発明は、電子価値の更新に用いる更新データが付加された更新コマンドを端末装置からＩＣカードへ送信し、ＩＣカードで、その更新コマンドを受信して更新アプリケーション・プログラムを実行することにより、その更新コマンドに付加されている更新データを用いて、メモリに記憶している電子価値残高を更新するための複数の処理を一括して実行するようにしたので、従来のように、バリュー更新処理を構成する複数の処理をそれぞれ個別に実行する場合と比較して、個別処理に対する不正を防止できるとともに異常発生にも適切に対応できる。また、ＩＣカードと端末装置との間でのデータ通信回数を最低限に抑えることができ、迅速なバリュー取引を実現できる。

【図面の簡単な説明】

【図１】　本発明の一実施の形態による電子価値取引システムを示すブロック図である。

【図２】　第１の実施の形態にかかる電子価値取引方法を示すシーケンス図である。

【図３】　第２の実施の形態にかかる電子価値取引方法を示すシーケンス図である。

【図４】　バリューチャージ処理の一例を示すシーケンス図である。

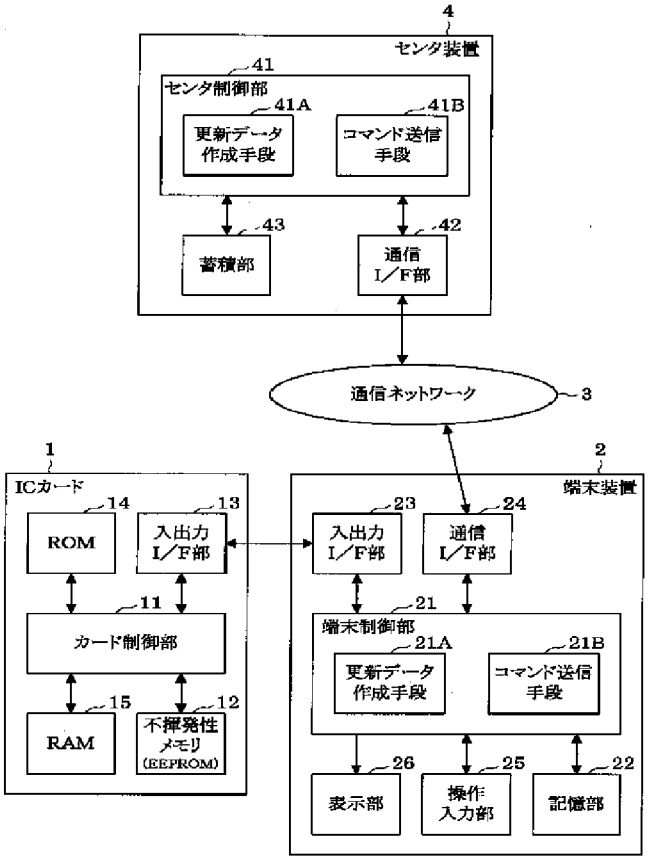
【図５】　バリューペイ処理の一例を示すシーケンス図である。

Nintendo Exhibit 1003
Part 2 of 2
Page 001059

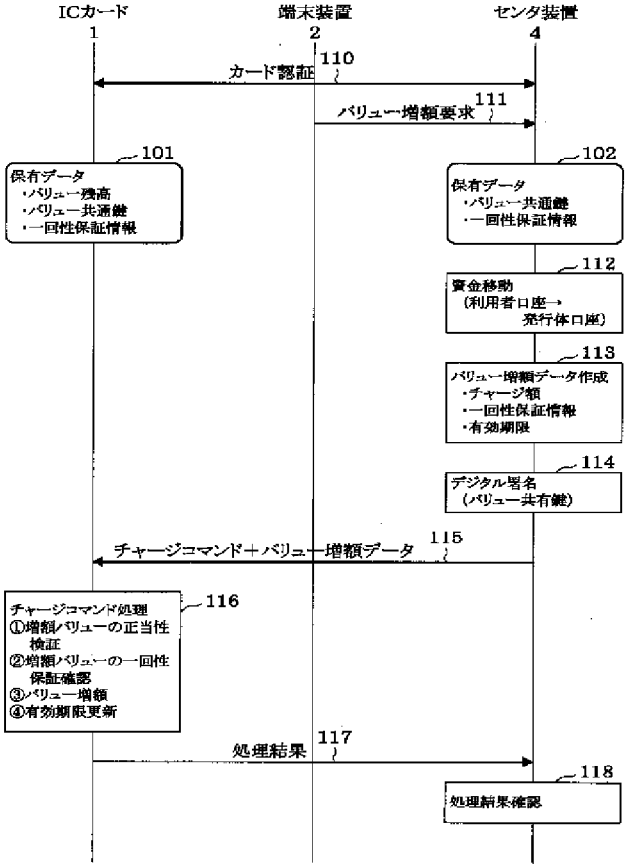
【符号の説明】

１…ＩＣカード、２…端末装置、３…通信ネットワーク、４…センタ装置、１１…カード制御部、１２…不揮発性メモリ（ＥＥＰＲＯＭ）、１３，２３…入出力Ｉ／Ｆ部、１４…ＲＯＭ、１５…ＲＡＭ、２１…端末制御＊部、２１Ａ，４１Ａ…更新データ作成手段、２１Ｂ，４１Ｂ…コマンド送信手段、２２…記憶部、２４，４２…通信Ｉ／Ｆ部、２５…操作入力部、２６…表示部、４１…センタ制御部、４３…蓄積部。

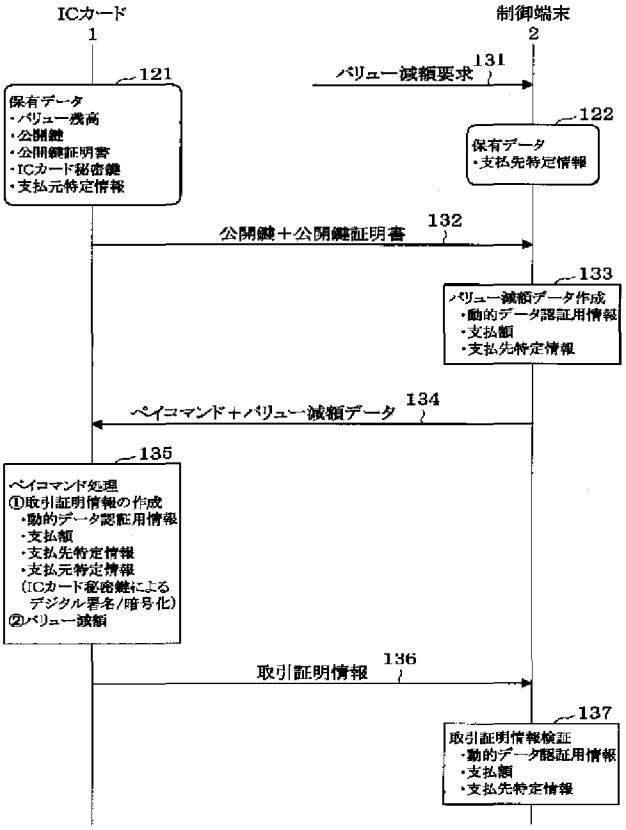
【図１】



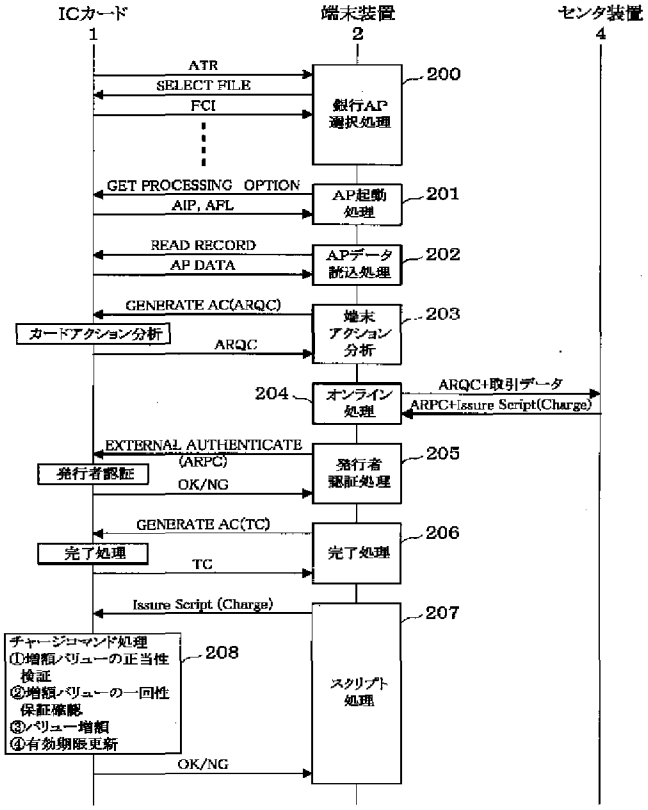
【図２】



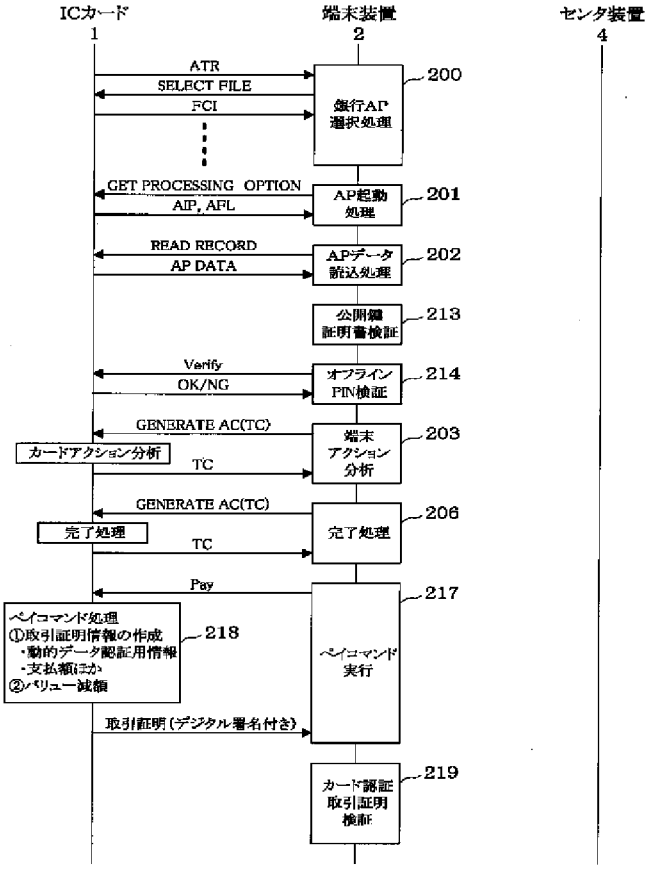
【図３】


ICカード
1
制御端末
2
121
保有データ
・バリュー残高
・公開鍵
・公開鍵証明書
・ICカード秘密鍵
・支払元特定情報
131
バリュー減額要求
122
保有データ
・支払先特定情報
132
公開鍵＋公開鍵証明書
133
バリュー減額データ作成
・動的データ認証用情報
・支払額
・支払先特定情報
134
ペイコマンド＋バリュー減額データ
135
ペイコマンド処理
①取引証明情報の作成
・動的データ認証用情報
・支払額
・支払先特定情報
・支払元特定情報
（ICカード秘密鍵による
デジタル署名/暗号化）
②バリュー減額
136
取引証明情報
137
取引証明情報検証
・動的データ認証用情報
・支払額
・支払先特定情報

【図４】


ICカード
1
端末装置
2
センタ装置
4
ATR
SELECT FILE
FCI
銀行AP
選択処理
200
GET PROCESSING OPTION
AIP, AFL
AP起動
処理
201
READ RECORD
AP DATA
APデータ
読込処理
202
GENERATE AC(ARQC)
カードアクション分析
ARQC
端末
アクション
分析
203
204
オンライン
処理
ARQC+取引データ
ARPC+Issure Script(Charge)
EXTERNAL AUTHENTICATE
(ARPC)
発行者認証
OK/NG
発行者
認証処理
205
GENERATE AC(TC)
完了処理
TC
完了処理
206
Issure Script (Charge)
207
チャージコマンド処理
①増額バリューの正当性
検証
②増額バリューの一回性
保証/確認
③バリュー増額
④有効期限更新
208
スクリプト
処理
OK/NG

【図５】



フロントページの続き

(51)Int.Cl.[7]　　識別記号　　F I　　ﾃｰﾏｺｰﾄﾞ(参考)
Ｇ０７Ｆ　19/00　　Ｇ０６Ｋ　19/00　　Q
Ｇ０７Ｄ　9/00　　４７６

(72)発明者　菅沼　知久
東京都千代田区内幸町一丁目１番６号　エヌ・ティ・ティ・コミュニケーションズ株式会社内
(72)発明者　益塚　幸児
東京都千代田区内幸町一丁目１番６号　エヌ・ティ・ティ・コミュニケーションズ株式会社内
(72)発明者　鈴木　健一
東京都千代田区内幸町一丁目１番６号　エヌ・ティ・ティ・コミュニケーションズ株式会社内

Ｆターム(参考)　3E040 AA03 BA18 CB01 FK09
5B035 AA13 BB09 CA38
5B058 KA01 KA06 KA08 KA31 KA35

Nintendo Exhibit 1003
Part 2 of 2
Page 001062

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

(11) EP 2 429 148 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
14.03.2012 Bulletin 2012/11

(21) Application number: **11192774.5**

(22) Date of filing: **07.01.2003**

(51) Int Cl.:
H04L 29/06 *(2006.01)* ***G06F 21/00*** *(2006.01)*
H04M 1/725 *(2006.01)*

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PT SE SI SK TR

(30) Priority: **08.01.2002 JP 2002001843**

(62) Document number(s) of the earlier application(s) in accordance with Art. 76 EPC:
03700468.6 / 1 465 039

(71) Applicant: **NTT DoCoMo, Inc.**
Tokyo 100-6150 (JP)

(72) Inventors:
- **Yamada, Kazuhiro**
  Tokyo, 100-6150 (JP)
- **Kondo, Takashi**
  Tokyo, 100-6150 (JP)
- **Watanabe, Nobuyuki**
  Tokyo, 100-6150 (JP)
- **Tsuda, Masayuki**
  Tokyo, 100-6150 (JP)
- **Kamiya, Dai**
  Tokyo, 100-6150 (JP)
- **Asai, Mao**
  Tokyo, 100-6150 (JP)
- **Miura, Fumiaki**
  Tokyo, 100-6150 (JP)
- **Washio, Satoshi**
  Tokyo, 100-6150 (JP)
- **Tomioka, Atsuki**
  Tokyo, 100-6150 (JP)
- **Kawabata, Hiroshi**
  Tokyo, 100-6150 (JP)

(74) Representative: **HOFFMANN EITLE**
Patent- und Rechtsanwälte
Arabellastraße 4
81925 München (DE)

Remarks:
This application was filed on 09-12-2011 as a divisional application to the application mentioned under INID code 62.

(54) Contents transmission method and contents transmission system

(57) A terminal unit comprises:
an executing unit configured to execute at least a first type of program and a second type of program, the first type of program requiring a security descriptive file, the second type of program not requiring the security descriptive file, the security descriptive file including authorization information, the authorization information being information used for controlling access by a program to at least one resource from among a hardware resource, a network resource and a software resource;
a determining unit configured to determine whether a program to be executed by the executing unit is the first type or the second type;
a controlling unit configured to control access to the at least one resource in accordance with a first condition if the program to be executed by the executing unit is determined to be the first type, and configured to control access to the at least one resource in accordance with a second condition if the program to be executed by the executing unit is determined to be the second type, the first condition being a condition based on the authorization information.

EP 2 429 148 A2

Printed by Jouve, 75001 PARIS (FR)

Description

Technical Field

[0001] The present invention relates to transmitting application software to terminal units.

Background Art

[0002] Mobile units equipped with a function of executing Java-AP (Java Application) software by carrying out a program written in accordance with Java (trademark registered) programming language, and downloaded via a network are in widespread use.

[0003] Java-AP software includes a Jar (Java Archive) file and an ADF (Application Descriptor File). The Jar file contains a program, which provides a user with certain Java-AP. The ADF is dependent upon the Jar file, and contains, for instance, URL showing where a Jar file is stored (hereinafter, referred to as package URL), the size of a Jar file, the most recent date when a Jar file is changed and other necessary information.

[0004] A mobile unit downloads the relevant software to the desired Java-AP by following the process described below. First, the mobile unit obtains an ADF pertaining to the desired Java-AP from a server unit, which constitutes WWW (World Wide Web).

[0005] The mobile unit, which obtains an ADF checks the content of the ADF, and determines whether the Jar file pertaining to the desired Java-AP can be installed in the mobile unit. When the mobile unit determines that Java-AP software can be installed, the mobile unit obtains from a server unit, which constitutes WWW a Jar file by using package URL contained in the ADF. The process of downloading Java-AP software is complete when the Jar file is obtained. Hereinafter, in the mobile unit, installation of the downloaded Java-AP software is carried out, and the Java-AP software can be activated when required.

[0006] Incidentally, when the Java-AP software is installed in a mobile unit the activation of Java-AP is subject to greater restriction than the activation of functions that are native to the mobile unit, such as a communication application function. The activation of a Java-AP is restricted in that it is unable to access certain data contained in a mobile unit, such as telephone numbers for example. By imposing strict restrictions in this manner, leakage or falsification of confidential data contained in a mobile unit, occurring due to malfunctioning Java-AP or caused intentionally, can be prevented.

[0007] However, imposing the above-mentioned restriction on all Java-AP uniformly does not adequately meet the needs of a user of a mobile unit or an IP (information provider). For instance, some users seem to feel that Java-AP could be allowed to refer to someof the private information stored in a mobile unit as long as security is guaranteed. Also, some IPs wish to provide Java-AP, which uses some of the private information stored in a mobile unit, or some of the functions a mobile unit is equipped with.

[0008] To fulfill these requirements, a system in which a trustworthy organization such as a communication provider providing a communication service to users of mobile units working as an authority, is entrusted with the responsibility of authorizingJava-AP to operate with greater flexibility. Mobile units using Java-AP are notified by the authorization of the operational rules set for Java-AP, and the mobile units can restrict the operation of Java-AP on the basis of the prescribed rules. In this system, only a trustworthy organization should be entrusted to administer the authorization of a more flexible operation of Java-AP.

[0009] When the above-mentioned system is applied to the downloading process of Java-AP software, information showing the authorization in an ADF or a Jar file must be included. Since a Jar file is updated by an IP as required, and it is appropriate for an IP to own a Jar file, it is appropriate for the corresponding ADF to contain information on the validity of the authorization.

[0010] However, since the content of anADF is dependent upon a Jar file, an ADF owned by a trustworthy organization needs to be updated oncean IP updates a Jar file. Also, updating an ADF becomes necessary at times even without the updating of a Jar file, as in the instance where access to a certain Jar file gets deluged, and the Jar file is moved to another server unit in the IP. In this instance, since the location where the Jar file is stored is changed, a package URL contained in the ADF needs to be changed. However, since the ADF is administered by a trustworthy organization and excludes the involvement of other agents, the updating operation of an ADF could become a very busy one.

Disclosure of Invention

[0011] The present invention was developed to overcome the stated problems of the conventional art, and its object is to provide a transmission method and a transmission system for transmitting without restricting the freedom of an IP to a terminal unit which allows an application to operate in accordance with an authorization, software for achieving an application which is transmitted by transmitting a plurality of files which are dependent upon each other.

[0012] To solve the above-mentioned problem of the conventional art, the present invention provides a transmission method comprising: a process for transmitting an authorization file in a communication system in which an originator originates a requestincluding information showing a storage ofa file and the file is transmitted in response to the request, the process for transmitting including transmitting a security descriptive file as the authorization file from an administering server unit storing the security descriptive file through a secure link to a terminal unit, the security descriptive file containing first identification information and authorization information,

Nintendo Exhibit 1003
Part 2 of 2
Page 001064

the first identification information showing a storagelocation of an application descriptive file, the application descriptive file having information dependent upon an entity file, which includes software for executing an application, along with information showing a storage location of the entity file, the authorization information of the security descriptive file indicating an allowable range of an operation of the application which is executed in accordance with the software, the terminal unit executing operation of the application within a range shown by the authorization information; a dependent information obtaining process for the terminal unit to obtain, by using the first identification information contained in the security descriptive file transmitted from the communication system in the authorization transmission process, the application descriptive file from one or a plurality of server units in which the application descriptive file is stored; and a program obtaining process for the terminal unit to obtain the entity file from the communication system by using the application descriptive file obtained in the dependent information obtaining process.

[0013] By this transmission method, the terminal unit obtains before obtaining the application descriptive file and the entity file corresponding to the application, the security descriptive file, which is transmitted by the communication system after security has been assured. In the security descriptive file, the authorization given to the application is denoted, and in the terminal unit, the application corresponding to the security descriptive file is allowed to operate in accordance with the authorization denoted by the obtained security descriptive file.

[0014] Also, the present invention provides a transmission system comprising: a communication system for returning, when a stored location of a file is notified, the file which comprises one or a plurality of server units in which an entity file containing software for achieving an application and an application descriptive file which has a content dependent upon the entity file showing a stored location of the entity file, and an administering server unit in which a security descriptive file containing a first identification information showing a stored location of the application descriptive file and authorization information showing authorization given to an application which is achieved when a terminal unit executes the software is stored; and a terminal unit which approves operation of an application in accordance with authorization given to the application, wherein the administering server unit transmits the security descriptive file to the terminal unit by assuring security, and wherein the terminal unit obtains the application descriptive file by using the first identification information contained in the security descriptive file transmitted by the communication system, and the entity file from the communication system by using the application descriptive file.

[0015] By this transmission system, the terminal unit, before obtaining the application descriptive file and the entity file corresponding to the application, obtains the security descriptive file transmitted by the transmission system after security has been assured. The authorization given to the application is denoted in the security descriptive file, and in the terminal unit, the application corresponding to the security descriptive file is allowed to operate in accordance with the permission of the authorization denoted by the obtained security descriptive file.

Brief Description of Drawings

[0016]

Fig. 1 is a block diagram showing the configuration of the transmission system of one embodiment for executing the present invention.
Fig. 2 is a conceptual figure showing data configuration of anADF inherent to the transmission system.
Fig. 3 is a block diagram showing the configuration of mobile unit 16 constituting the transmission system.
Fig. 4 is a conceptual figure showing the functional configuration of mobile unit 16.
Fig. 5 is a flowchart showing the process of mobile unit 16 for downloading and installing Java-AP software.
Fig. 6 is a conceptual figure showing data configuration ofan SDF stored in administering server unit 18 in the transmission system.
Fig. 7 is a conceptual figure showing the content of policy information contained in the SDF.
Fig. 8 is a block diagram for explaining the operation of the transmission system.
Fig. 9 is a diagram showing a list page transmitted in the transmission system.
Fig. 10 is a diagram showing the content of an explanatory file stored in IP server unit 12 constituting the transmission system.
Fig. 11 is a diagram showing an explanatory page transmitted in the transmission system.
Fig. 12 is a diagram showing the content of an explanatory file stored in IP server unit 12.
Fig. 13 is a diagram showing an explanatory page transmitted in the transmission system.
Fig. 14 is a diagram showing the content of an explanatory file stored in IP server unit 13 constituting the transmission system.
Fig. 15 is a diagram showing an explanatory page transmitted in the transmission system.
Fig. 16 is a sequence diagram for explaining the operation of the transmission system.
Fig. 17 is a sequence diagram for explaining the operation of the transmission system.
Fig. 18 is a sequence diagram for explaining the operation of the transmission system.
Fig. 19 is a block diagram for explaining another operation of the transmission system.
Fig. 20 is a sequence diagram for explaining another operation of the transmission system.

Nintendo Exhibit 1003
Part 2 of 2
Page 001065

Best Mode of Carrying out the Invention

[0017] Hereinafter, by referring to figures, a transmission system, which is one mode of the present invention is explained. In figures, identical codes are given to common portions.

(1) Configuration

[0018] As shown in Fig. 1, in the transmission system, IP server units 12 to 14 are connected to Internet 11. IP server unit 12 is administered by the first IP (Internet Provider), and IP server units 13 and 14 are administered by the second IP, which differs from the first IP. IP server units 12 to 14 constitute WWW, and each is equipped with similar hardware and functions to those of a general WWW server unit. Mobile packet communication network 15 is a network a communication provider uses to provide a mobile packet communication service. Mobile unit 16 can perform radio packet communication with mobile packet communication network 15. Gateway server unit 17 is administered by an identical communication provider to that of mobile packet communication network 15. Gateway server unit 17 is a unit for connecting mobile packet communication network 15 and Internet 11, and has similar configuration to that of a general gateway server unit. Administering server unit 18 is connected to gateway server unit 17 by an exclusive line. Administering server unit 18 also constitutes WWW, and has similar hardware and a function to those of a general WWW unit. Gateway server unit 17 performs packet communication between mobile packet communication network 15 and Internet 11, packet communication between administering server unit 18 and mobile packet communication network 15, and packet communication between administering server unit 18 and Internet 11. Mobile unit 16, by using the relaying function, is able to perform packet communication with IP server units 12 to 14 via mobile packet communication network 15 and Internet 11, Several mobile units exist in the actual transmission system, but only one mobile unit 16 isshown to avoid complicating the figures. For the same reason, only IP server units 12 to 14 are shown.

[0019] In the transmission system, mobile unit 16 is capable of receiving Java-AP software from the desired site on Internet 11. Software which mobile unit 16 is capable of receiving is distinguished between the one pertaining to a trusted Java-AP and the one pertaining to a non-trusted Java-AP. A trusted Java-AP software is one which the communication provider administering mobile packet communication network 15 guarantees authenticity of on the basis of the contract with IP administering IP server units 12 to 14. A non-trusted Java-AP software is any Java-AP software other than a trusted Java-AP software.

[0020] Administering server unit 18 stores each SDF (Security Descriptive File) pertaining to each trusted Java-AP software, which is transmitted in the transmission system. SDF is a file, which is produced by the communication provider administering mobile packet communication network 15, and is the file necessary for downloading into a mobile unit the Java-AP software, which uses trusted API (APplication Interface) of the mobile unit. An explanation of trusted API will fellow later. As shown in Fig.6, an SDF contains an APID for detecting trusted Java-AP software, policy information, ADF-URL showing the location where an ADF corresponding to the Java-AP software is stored, and public key CA (Certifying Agent) gives to IP which provides the Java-AP software. Policy information is the information showing restriction on the operation of Java-AP. Policy information and the restriction on Java-APs operation carried out on the basis of the policy information will be explained in detail afterwards.

[0021] In the present embodiment, when a trusted Java-AP software is transmitted to mobile unit 16 by one of IP server units 12 to 14, an SDF corresponding to the trusted Java-AP software is transmitted to mobile unit 16 by administering server unit 18 in response to the request from mobile unit 16. In mobile unit 16, when a trusted Java-AP software is executed, restriction on the trusted Java-AP's operation is carried out on. the basis of the SDF corresponding to the trusted Java-AP. This is one characteristic of the present embodiment. As shown in Fig.1, transmission of the SDF is carried out via mobile packet communication network 15, and administering server unit 18 and gateway server unit 17 are connected by an exclusive line. Also, for transmission, the SDF is encrypted.

[0022] Hereinafter, with regard to relevance to the characteristics, the configuration of each element of the transmission system will be explained.

[0023] IP server units 12, 13 and 14 are equipped with fixed memory 12A, 13A and 14A respectively.

[0024] Fixed memory 12A, 13A and 14A are fixed memories such as a hard disc, and store Java-AP software constituting Jar files and ADF, and explanatory files on the content of Java-AP software for users of mobile units.

[0025] Each Java-AP software stored in fixed memory 12A, 13A and 14A might be either trusted Java-AP software or non-trusted Java AP software. Whether Java-AP is a trusted Java-AP or a non-trusted Java-AP, in each ADF of Java-AP software, information such as a package URL showing the location where a Jar file in WWW is stored, information showing the size of the Jar file, and information showing the date of the most recent update are written. Such information is generally known as items to be written in the ADF of a Java-AP software. Also, the ADF of a trusted Java-AP software, as shown in Fig.2, contains an APID, of the trusted Java-AP and the hash value of the Jar file besides the generally known information. Furthermore, the ADF of a trusted Java-AP software is encrypted by a secret key which CA gives to an IP which provides the software.

[0026] Also, the explanatory file is a text file written in

Nintendo Exhibit 1003
Part 2 of 2
Page 001066

accordance with HTML. A mobile unit, when a certain Java-AP software is downloaded, needs to download beforehand the explanatory file corresponding to the Java-AP software. The explanatory file contains information for forming UI (User Interface) for receiving from the user the command for downloading Java-AP software. Mobile unit 16 displays the UI screen in accordance with the information. The user can carry out the operationon mobile unit 16 for specifying the object showing the desired Java-AP on the UI screen. The explanatory file is written for the object specified by the user in this manner so as to correspond to the URL showing where in WWW the SDF (ADF if SDF does not exist) corresponding to Java-AP software which is the object for downloading is located.

[0027] Each of IP server units 12 to 14 is equipped with the function of producing and updating each of the above-mentioned files in accordance with the command of an IP.

[0028] Administering server unit 18 is equipped with fixed memory 18A such as a hard disc. Administering server unit 18 establishes a TCP connection with the party. When administering server unit 18 receives a request message which uses the GET method of HTTP from the party via a TCP connection, administering server unit 18 reads out the file identified by the URL specified by the GET method from fixed memory 18A, and returns a response message of HTTP containing the file, and severs the connection.

[0029] Also, in the above-mentioned fixed memory 18A are stored, list file 200 for introducing to the user of mobile unit 16 downloadable Java-AP software, and respective SDF corresponding to each Java-AP software which is listed in list file 200.

[0030] List file 200, as SDF, is the file produced by the communication provider in accordance with the contract signed by each IP which administers IP server units 12 to 14, and the communication provider which administers administering server unit 18. Listfile 200 is a text file written in accordance with HTML. As already explained, the mobile unit needs to obtain the explanatory file containinga URL of an SDF to download the SDF of certain Java-AP software. Mobile unit 16 can obtain the explanatory file directly by accessing the IP server unit in which the explanatory file is stored. However, in the present embodiment, mobile unit 16 can obtain the explanatory file of the desired Java-AP software also by the following process as opposed to the above-mentioned direct method. First, mobile unit 16, by accessing administering server unit 18, obtains list file 200, and displays the UI screen accordingly. The user can carry out the operation on mobile unit 16 to specify the object showing the desired Java-AP on the UI screen. List file 200 matches the object specified by the user to a URL showing the location of the explanatory file of Java-AP software in WWW which is the object of downloading. Mobile unit 16, by using the URL obtained via list file 200, obtains the explanatory file from the IP server unit.

[0031] Mobile unit 16 consists of, as shown in Fig.3, OS (Operating System) software; ROM 16A in which Java-AP environment software for establishing the environment for executing the Java-AP, and several types of native AP software are stored; CPU 16B which is connected to ROM 16A for reading out a program from ROM 16A and executing the program; displaying unit 16C which is connected to CPU 16B; fixed memory 16D; RAM 16E; communication unit 16F; and operation unit 16G.

[0032] Displaying unit 16C has, for instance, a liquid crystal displaying panel, and displays data provided by CPU 16B as an image. Fixed memory 16D is, for instance, SRAM or EEPROM, and data is read and written by CPU 16B. Fixed memory 16D is used to store Java-AP software (ADF and Jar) downloaded from a server unit (hereinafter, referred to as a Web server unit) constituting WWW, and an SDF.

[0033] Communication unit 16F performs radio packet communication with mobile packet communication network 15, and relays packets between CPU 16B and mobile packet communication network 15. Also, communication unit 16F is equipped with CODEC, a microphone, a speaker and so forth for communication besides an antenna or a radio transmission and reception unit. Hence, mobile unit 16, by communication unit 16F, can perform communication by circuit switching via a mobile communication network (not shown). Operation unit 16G is equipped with an operation controller, and provides CPU 16B a signalin accordance with the operation carried out by the operation controller.

[0034] As the switch (not shown) is turned on, CPU 16B reads out the program contained in OS software from ROM 16A, and executes with RAM 16E as a work area. As a result, functions for providing UI and so forth are executed in CPU 16B. In other words, CPU 16B activates OS software, and executes OS of Fig.4 in mobile unit 16. OS identifies the command of the user on the basis of the signal provided by operation unit 16G and the status of UI, and executes the process in accordance with the command.

[0035] When the command of the user requests activation of communication software, which is native AP software, OS activates the communication software, and executes communication AP in mobile unit 16. By using communication AP, the user can communicate with the party.

[0036] When the command of the user requests the activation of telephone directory AP, which is native AP software, OS activates the telephone directory software, and executes telephone directory AP in mobile unit 16. By using telephone directory AP, the user can refer to, use, and change the content of the telephone directory (hereinafter, referred to as telephone directory data) stored in fixed memory 16D.

[0037] When the command of the user requests the activation of Web browser software, which is native AP software, OS activates Web browser software, and executes the Web browser in mobile unit 16. Also, the Web browser provides UI. Then, as the user gives the com-

Nintendo Exhibit 1003
Part 2 of 2
Page 001067

mand by operating operation unit 16G, the Web browser identifies the command of the user on the basis of the status of UI and the signal provided by operation unit 16G, and executes the process in accordance with the command. For instance, when the command is for obtaining the specified file from WWW,a TCP connection is established by operating communication unit 16F with the Web server unit in which the file is stored, a request message of HTTP using the GET method is transmitted by URL showing the specified location, a response message corresponding to the request message is received, and the connection is severed. Furthermore, the Web browser interprets the file contained in the received response message in accordance with HTML, produces UI containing the Web page, and provides the user. Also, when a user sends a command for downloading Java-AP software, the command is notified to JAM (Java Application Manager). Specifically, in a Web page, either by clicking or pressing, when an anchor tag to which the object tag is specified is designated, the Web browser extracts URL which is specified as data property of the object tag, and notifies JAM that downloading of Java-AP software by URL is requested.

[0038] When the command of the user requests the activation of JAM software, which is native AP software, OS activates JAM software, and executes JAM in mobile unit 16. JAM shows to the user a list of Java-AP software installed in mobile unit 16, and activates the Java-AP software specified by the user. Specifically, when the command of the user to JAM requests the activation of Java-AP software, Java-AP environment software is activated, and Java-AP environment is executed in mobile unit 16. Then, the specified Java-AP software is activated, and Java-AP is executed in Java-AP environment. Java-AP environment contains KVM, which is a light-weight Java Virtual Machine appropriate to a cellular terminal, and API provided for Java-AP. API provided for Java-AP is divided into trusted API which only Java-AP whose trustworthiness is guaranteed by the communication provider on the basis of the contract with the IP (hereinafter, referred to as trusted AP) is allowed to use, and non-trusted API which any Java-AP is allowed to use.

(2) Operation

[0039] Hereinafter, the operation of the present embodiment is explained. JAM, when a command for requesting the downloading of Java-AP is notified by the Web browser, carries out the process for downloading and installing Java-AP software into mobile unit 16. The flow of the process is shown in Fig. 5. In Fig.5, the process of mobile unit 16 for obtaining the explanatory file is omitted. Since some different modes exist for the process of obtaining the explanatory file, the process will belater explained with some specific examples of the operation. As shown in Fig.5, JAM first determines whether Java-A.P software, which is about to be downloaded, is trusted Java-AP software (Step S11). Specifically, when mobile unit 16 obtains the explanatory file, the Web browser provides the user UI corresponding to the explanatory file, and receives the command for downloading Java-AP software from the user. The Web browser notifies JAM of URL of the Java-AP software specified by the user. JAM refers to the file name at the end of the URL notified by the Web browser, and determines that the software is trusted Java-AP software if the extension of the file is "sdf", and non-trusted Java-AP software if the extension of the file is not "sdf." When the Java-AP software, which is about to be downloaded, is determined as trusted Java-AP software, the downloading and installation processes identical to the conventional processes are carried out (Step S12).

[0040] When the Java-AP software, which is about to be downloaded, is determined as trusted Java-AP software, JAM obtains the SDF corresponding to the software from administering server unit 18 (Step S13). In other words, JAM establishes the TCP connection with administering server unit 18, produces and transmits via the TCP connection a request message requesting administering server unit 18 for transmission of the SDF stored in the location shown by URL notified by the Web browser, receives the response message to the request message, and severs the above-mentioned connection.

[0041] Then, JAM extracts APID, ADF-URL, and the public key from SDF contained in the response message, and writes SDF into fixed memory 15D.

[0042] Next, JAM obtains ADF (Step S14). Specifically, JAM establishes the TCP connection with the Web server unit in which ADF identified by ADF-URL extracted from SDF is stored, produces and transmits a request message for requesting transmission of ADF, receives a response message to the request message, and severs the TCP connection.

[0043] As already explained, an ADF corresponding to trusted Java-AP software, contains the hash values of an APID, and a Jar file, and further signed (encrypted) by the secret key CA gives to the IP which provides the trusted Java-AP software. Then, JAM checks (decodes) the signature of the ADF contained in the response message by using the public key extracted from the SDF, and determines the authenticity of the ADF (Step S15).

[0044] When an ADF is determined to be authentic, the JAM compares the APID extracted from an SDF to an APID contained in the ADF, and determines whether these APIDs match (Step S16). When these APIDs are determined to match, JAM determines whether the trusted Java-AP software can be installed in mobile unit 16 on the basis of the content of the ADF (Step S 17). The basis of determination is identical to the conventional basis.

[0045] When installation is determined to be possible, JAM obtains the Jar file. Specifically, JAM writes the ADF into mobile unit 16, and extracts the hash value and package URL from the ADF. Furthermore, JAM establishes the TCP connection with the Web server unit in which the Jar file identified by package URL is stored, produces

Nintendo Exhibit 1003
Part 2 of 2
Page 001068

and transmits a request message for requesting transmission of the Jar file, receives the response message to the request message, and severs the TCP connection (Step S18).

[0046] Furthermore, JAM calculates the hash value to the obtained Jar file (Step S19). Any hash function can be used to calculate the hash value, but the hash value used by the mobile unit and the hash value the IP uses for calculating the hash value contained in the ADF must be identical.

[0047] JAM compares the hash value calculated by JAM to the hash value extracted from the ADF (Step S20), writes the obtained Jar file into administering server unit 18 when these hash values match, carries out several kinds of processes pertaining to installation of trusted Java-AP software (Step S21), and notifies the user that installation is successful (Step S22).

[0048] When an ADF is determined not to be authentic, whm an APID of an SDF and an APID of an ADF do not match, when the Java-AP software which is about to be installed is determined not to be installable, and when the calculated hash value and the hash value of an ADF do not match, the JAM notifies the user that installation has failed, and returns the status of mobile unit 16 to the status that exists before the acquisition of the SDF starts.

[0049] Also, JAM supervises the operation of Java-AP, and restricts the use of the trusted API. The restriction is carried out in accordance with policy information in SDF stored in fixed memory 16D. Policy information in SDF, for instance, is the content conceptually shown in Fig. 7. In the policy information shown in Fig.7, the use of the necessary trusted API "getPhoneList()" for referring to telephone number directory data, and the necessary trusted API "get MsStatus()" for obtaining the status of the mobile unit stored in the mobile unit are allowed, and the use of the necessary trusted API for referring to history data of transmission and reception "getcallHistory()" stored in the mobile unit is forbidden.

(3) Specific Operation

[0050] Next, the operation of the above-mentioned system is explained.

[0051] In the operation explained below, establishment of the TCP connection and severing operation are general operations of HTTP; therefore, the explanation is omitted. Also, the above-mentioned operations carried out by OS, the Web browser, JAM, Java-AP, native AP and so forth are operations of mobile unit 16; therefore, in the following explanation, the main unit which carries out the operation is mobile unit 16.

[0052] In the operation explained here in after, the following case is the presupposition. First, as shown in Fig. 8, in fixed memory 18A of administering server unit 18, are stored list file 200 and SDF 204. Atthis stage, list file 200 is written to provide list page 201 shown in Fig.9 when interpreted and executed by mobile unit 16. Also, list file 200 is written, when option 201A constituting list page 201 is specified either by being clicked or pressed, to produce a request message containing URL of explanatory file 202 ("http://www.mam.bbb.co.jp/ghi.html") as a parameter of the GET method. Moreover, list file 200 is written, when option 201B constituting list page 201 is specified, to produce a request message containing URL of explanatory file 207 ("http://www.ccc.co.jp/jkl.html") as a parameter of the GET method.

[0053] Also, SDF 204 contains "0001" as APID, information shown in Fig.7 as policy information, "http://www.main.bbb.co.jp/viewer.jam" as ADF-URL, and public key CA gives to IP which administers IP server unit 13 and IP server unit 14.

[0054] Also, in fixed memory 12A of IP server unit 12, are stored explanatory file 211 corresponding to Java-AP software of the title "tsume-shogi" (a game similar to "chess ") (hereinafter, referred to as the first Java-JP software), ADF 213, and Jar file 214. Explanatory file 211, ADF 213, and Jar file 214 are produced by IP administering IP server unit 12. The content of explanatory file 211 is shown in Fig. 10. Explanatory file 211 is written to provide explanatory page 212 shown in Fig. 11 when interpreted and executed by mobile unit 16. Also, ADF 213 contains URL of Jar file 214 ("http:llwww.ccc.co.jp/shogi.jar") as package URL.

[0055] Also, in fixed memory 12A of IP server unit 12, are stored explanatory file 207 corresponding to Java-AP software of the title "horoscope" (hereinafter, referred to as the second Java-AP software), ADF 209, and Jar file 210. Explanatory file 207, ADF 209, and Jar file 210 are produced by IP administering IP server unit 12. The content of explanatory file 207 is shown in Fig.12. Explanatory file 207 is written to provide explanatory page 208 shown in Fig. 13 when interpreted and executed by mobile unit 16. Also, ADF 209 contains URL of Jar file 210 ("http://www.ccc.co.jp/horoscope.jar") as package URL.

[0056] Also, in fixed memory 13A of IP server unit 13, are stored explanatory file 202 corresponding to Java-AP software of the title "telephone number directory viewer" (hereinafter, referred to as the third Java-AP software), ADF 205, and Jar file 206. Explanatory file 202, ADF 205, and Jar file 206 are produced by IP administering IP server unit 13 and IP server unit 14. The content of explanatory file 202 is shown in Fig.14. Explanatory file 202 is written to provide explanatory page 203 shown in Fig. 15 when interpreted and executed by mobile unit 16. ADF 205 contains "0001" as APID, the hash value of Jar file 206, URL of Jar file 206 ("http://www.main.bbb.co.jp/viewer.jar") as package URL, and is signed by secret key CA gives to IP administering IP server unit 13, and IP server unit 14.

[0057] Also, mobile unit 16 is in the status in which the first to the third Java-AP software can be downloaded.

(2-1) Installation Operation

[0058] First, the operation of installing Java-AP soft-

Nintendo Exhibit 1003
Part 2 of 2
Page 001069

ware in mobile unit 16 is explained with reference to each Java-AP software.

(2-1-1) First Java-AP Software

[0059] The installation operation of the first Java-AP software begins when the user finds the IP server unit in which Java-software he/she desires is stored, and then tries to obtain explanatory file 211 in the Web browser by operating mobile unit 16. First, in mobile unit 16, request message tm 12 containing URL of explanatory file 211 ("http://www.ccc.co.ip/mno.html") as a parameter of the GET method is produced. Request message tm 12 is, as shown in Fig. 16, transmittedby mobile unit 16, and is received by IP server unit 12. In IP server unit 12, response message tm 13 containing explanatory file 211 is produced in response to the content of request message tm 12. Response message tm 13 is transmitted by IP server unit 12, and is received by mobile unit 16. In mobile unit 16, UI corresponding to the content of explanatory file 211 is provided to the user. As a result, in displaying unit 16C, explanatory page 212 shown, for instance, in Fig. 11 is displayed.

[0060] When the user sees explanatory page 212, and operates mobile unit 16 to hit anchor 212A in explanatory page 212, the value specified as ijam property of the anchor tag written in explanatory file 211 of Fig. 10 (the tag which begins with "<A") identifies the object tag specified as id property (the tag which begins with"<OBJECT") in mobile unit 16. Then, URL specified as data property of the object tag ("http://www.ccc.co.jp/shogi.jam") is extracted, and determination of Step S11 of Fig.5 is carried out. In the present example, since the extension of URL is not sdf, the ordinary process (Step S12) is carried out. In other words, the process is carried out as follows. First, request message tm 16 for requesting transmission of ADF 213 identified by the URL is produced. Request message tm 16 is transmittedby mobile unit 16, and is received by IP server unit 12. In IP server unit 12, response message tm 17 containing ADF 213 is produced in response to the content of request message tm 16. Response message tm 17 is transmittedby IP server unit 12, and is received by mobile unit 16.

[0061] In mobile unit 16, on the basis of the content of ADF 213, whether the first Java-AP software can be installed is determined. As mentioned above, since mobile unit 16 is in the status in which the first Java-AP software can be installed, installation of the first Java-AP software is determined to be possible in mobile unit 16.

[0062] Then, in mobile unit 16, ADF 213 is written into fixed memory 16D1. Also, in mobile unit 16, package URL ("http://www.ccc.co.jp/shogi.jar") is extracted from ADF 213, and request message tm 18 requesting transmission of Jar file 214 identified by the package URL is produced. Request message tm 18 is transmitted by mobile unit 16, and is received by IP server unit 12. in IP server unit 12, response message tm 19 containing Jar file 214 is produced in response to the content of request message tm 18. Response message tm 19 is transmitted by IP server unit 12, and is received by mobile unit 16. In mobile unit 16, Jar file 214 is written into fixed memory 16DI, and installation of the first Java-AP software is completed.

[0063] When the first Java-AP software is determined as not installable in mobile unit 16, the status of mobile unit 16 returns to the statusthat existed before the acquisition of ADF 213 began.

(2-1-2) Second Java-AP software

[0064] The installation operation of the second Java-AP software begins when the user tries to obtain explanatory file 207 by operating mobile unit 16. As already explained, explanatory file 207 can be obtained either by directly accessing the relevant IP server or through list file 200, but only the operation which begins with trying to obtain list file 200 is explained.

[0065] As shown in Fig. 17, in mobile unit 16, request message tm 20 containing URL of list file 200 ("http: //ww.aaa.co.jp/def,html") as a parameter of the GET method is produced. Request message tm 20 is transmitted by mobile unit 16, and is received by administering server unit 18. In administering server unit 18, response message tm 21 containing list file 200 is produced in response to the content of request message tm 20. Response message tm 21 is transmitted by administering server unit 18, and is received by mobile unit 16. In mobile unit 16, when response message tm 21 is received, list file 200 in response message tm 21 is interpreted in accordance with HTML, and UI corresponding to the content of list file 200 is provided to the user of mobile unit 16. As a result, in displaying unit 16C of mobile unit 16, list page 201 shown, for instance, in Fig. 9 is displayed.

[0066] When the user, after seeing list page 201, operates mobile unit 16 to hit option 201B in list page 201, request message tm 22 containing URL ("http: //www.ccc.co.jp.jkl.html") corresponding to option 201B as a parameter of the GET method is produced. Request message tm 22 is transmitted by mobile unit 16, and is received by IP server unit 12. In IP server unit 12, response message tm 23 containing explanatory file 207 is produced in response to the content of request message tm 22. Response message tm 23 is transmitted by IP server unit 12, and is received by mobile unit 16. In mobile unit 16, UI corresponding to the content of explanatory file 207 is provided to the user. As a result, in displaying unit 16C, explanatory page 208 shown, for instance, in Fig. 13 is displayed.

[0067] When the user, after seeing explanatory page 208, operates mobile unit 16 to hit anchor 208A in explanatory page 208, the value specified as ijam property of the anchor tag written in explanatory file 207 of Fig. 12 (the tag which begins with "<A") identifies the object tag specified as id property (the tag which begins with "<OBJECT"). Then, URL specified as data property of the object tag ("http://www.ccc.co.jp/horoscope.jam") is

Nintendo Exhibit 1003
Part 2 of 2
Page 001070

extracted, and determination of Step S11 of Fig.5 is carried out. In this example, since the extension of URL is not sdf,an ordinary process (Step S 12) is carried out. In other words, the process is carried out as follows. First, request message tm 26 requesting transmission of ADF 209 identified by the URL is produced. Request message tm 26 is transmittedby mobile unit 16, and is received by IP server unit 12. In IP server unit 12, response message tm 27 containing ADF 209 corresponding to the content of request message tm 26 is produced. Response message tm 27 is transmittedby IP server unit 12, and is received by mobile unit 16.

[0068] In mobile unit 16, on the basis of the content of ADF 209, whether the second Java-AP software can be installed is determined. As mentioned above, since mobile unit 16 is in the status in which the second Java-AP software can be installed, the second Java-AP software is determined as installable in mobile unit 16.

[0069] Next, in mobile unit 16, ADF 209 is written into fixed memory 16D1. Also, in mobile unit 16, package URL ("http://www.ccc.co.jp/horoscope.jar") is extracted from ADF 209, and request message tm 28 requesting transmission of jar file 210 identified by package URL is produced. Request message tm 28 is transmitted by mobile unit 16, and is received by IP server unit 12. In IP server unit 12, response message tm 29 containing Jar file 210 in response to the content of request message tm 28 is produced. Response message tm 29 is transmitted by IP server unit 12, and is received by mobile unit 16. In mobile unit 16, Jar file 210 is written into fixed memory I6D1, and installation of the second Java-AP software. is completed.

[0070] When the second Java-AP software is determined as not installable in mobile unit 16, the status of mobile unit 16 returns toa previous status, that which existed before the acquisition of ADF 209 began.

(2-1-3) Third Java-AP Software

[0071] The installation operation of the third Java-AP software begins when the user tries to obtain explanatory file 202 by operating mobile unit 16. In the operation, mobile unit 16 obtains relevant list file 200, determines where explanatory file 202 exists, and tries to obtain explanatory file 202.

[0072] As shown in Fig. 18, in the operation which begins by trying to obtain list file 200, an operation identical to the operation shown in Fig. 17 is carried out till list page 201 shown, for instance, in Fig. 9 is displayed after mobile unit 16 receives response message tm 21. When the user, after seeing list page 201, operates mobile unit 16 to hit option 201A in list page 201, request message tm 32 containing URL corresponding to option 201A ("http://www.main.bbb.co.jp/ghi.html") as a parameter of the GET method is produced in mobile unit 16. Request message tm 32 is transmitted by mobile unit 16, and is received by IP server unit 13. In IP server unit 13, response message tm 33 containing explanatory file 202 in response to the content of request message tm 32 is produced. Response message tm 33 is transmitted by IP server unit 13, and is received by mobile unit 16. In mobile unit 16, the user is provided with UI corresponding to the content of explanatory file 202. As a result, in displaying unit 16C, explanatory page 203 shown, for instance, in Fig. 15 is displayed.

[0073] When the user, after seeing explanatory page 203, operates mobile unit 16 to hit anchor 203A in explanatory page 203, the value specified as ijam property of the anchor tag written in explanatory file 202 in Fig. 14 (the tag which begins with "<A") identifies the object tag specified as id property (the tag which begins by "<OBJECT"). Then, URL specified as data property of the object tag ("http://www.aaa.co.jp/abc.sdf") is extracted, and determination of Step S 11 in Fig. 5 is carried out. In the example, the extension of URL is sdf; therefore, the process of step S 13 and thereafter is carried out. In other words, the process is carried out as follows. First, request message tm 34 requesting transmission of SDF 204 identified by the URL is produced. Request message tm 34 is transmitted by mobile unit 16, and is received by administering server unit 18. In administering server unit 18, response message tm 35 containing SDF 204 in response to the content of request message tm 34 is produced. Response message tm 35 is transmitted by administering server unit 18, and received by mobile unit 16 via gateway server unit 17 and mobile packet communication network 15. The communication path between administering server unit 18 and gateway server 17 is an exclusive line, and since gateway server unit 17 is directly connected to mobile packet communication network 15 whose security is assured, SDF 204 can not be falsified until SDF 204 is received by mobile unit 16 (thereinbefore, Step S13).

[0074] In mobile unit 16, SDF 204 is written into fixed memory 16D1 of fixed memory 16D. Also, in mobile unit 16, APID ('0001"), ADF-URL ("http://www.main.bbb.co.jp/viewer.jam"), and the public key are extracted from SDF 204, and request message tm 36 requesting transmission of ADF 205 identified by ADF-URL is produced. Request message tm 36 is transmitted by mobile unit 16, and is received by IP server unit 13. In IP server unit 13, response message tm 37 containing ADF 205 in response to the content of request message tm 36 is produced. Response message tm 37 is transmitted by IP server unit 13, and is received by mobile unit 16 (thereinbefore, Step S14).

[0075] In mobile unit 16, authenticity of ADF 205 is determined by using the public key extracted from SDF 204 (Step S15). As mentioned above, since the public key contained in SDF 204 corresponds to the key used for the signature on ADF 205, ADF 205 is determined to be authentic as far as ADF 205 is not changed in IP server unit 13, or on the communication path between IP server unit 13 and mobile unit 16.

[0076] When ADF 205 is determined to be authentic, in mobile unit 16, APID extracted from SDF 204 and APID

Nintendo Exhibit 1003
Part 2 of 2
Page 001071

contained in ADF 205 are compared (Step S16). As mentioned above, since APID which matches APID in SDF 204 is written in ADF 205 in IP server unit 13, APID extracted from SDF 204 and APID contained in ADF 205 match as long as the description is correct.

[0077] When APID match, in mobile unit 16, whether the third Java-AP software can be installed is determined on the basis of the content of ADF 205 (Step S17). As mentioned above, since mobile unit 16 is in the status in which the third Java-AP software can be installed, the third Java-AP software is determined to be installable in mobile unit 16.

[0078] Then, in mobile unit 16, ADF 205 is written into fixed memory 16D1. Also, in mobile unit 16, the hash value and package URL ("http://www.main.bbb.co.jp/viewer.jar") are extracted, and request message tm 38 requesting transmission of Jar file 206 identified by the package URL is produced. Request message tm 38 is transmitted by mobile unit 16, and is received by IP server unit 13. In IP server unit 13, response message tm 39 containing Jar file 206 corresponding to the content of request message tm 38 is produced. Response message tm 39 is transmitted by IP server unit 13, and is received by mobile unit 16 (thereinbefore, Step S18).

[0079] In mobile unit 16, the hash value is calculated by using Jar file 206 and the specific hash function (Step S19), and the calculated hash value and the hash value extracted from ADF 205 are compared (Step S20). As mentioned above, the hash value of the Jar file corresponding to ADF 205 is written in ADF 205; therefore, these hash values would match as long as the description is correct. When these hash values match, in mobile unit 16, Jar file 206 is written into fixed memory 16D1, and the installation of the third Java-AP software is completed (Step S21 and S22).

[0080] When ADF 205 is determined to be not authentic in mobile unit 16, when APID extracted from SDF 204 and APID contained in ADF 205 do not match, when the third Java-AP software is determined to be not installable, or when the calculated hash value and the hash value extracted from ADF 205 do not match, a failure notice is sent to the user (Step S23), and the status of mobile unit 16 returns to a previous status that which existed before the acquisition of SDF 204 began.

(2-2) Operation of Mobile Unit 16 When Java-AP Software Is Activated

[0081] Next, the operation of mobile unit 16 when Java-AP software is activated will be explained.

(2-2-1) Fist Java-AP Software

[0082] The operation of mobile unit 16, when the first Java-AP software installed by the above-mentioned installation operation is activated in mobile unit 16 in which JAM is achieved, and the functions corresponding to the software (hereinafter, referred to as the first Java-AP) are achieved will be explained.

[0083] When API, which the first Java-AP is about to use is non-trusted API, the use of the API is approved by JAM. Hence, the first Java-AP can use the API.

[0084] On the other hand, when API, which the first Java-AP is about to use is trusted API, JAM checks whether SDF corresponding to the Java-AP is stored in fixed memory 16D. Since such SDF is not stored in fixed memory 16D, JAM forbids the use of the API by the first Java-AP. Hence, the first Java-AP will not be able use the API.

(2-2-2) Second Java-AP Software

[0085] The operation of mobile unit 16 when the installed second Java-AP software is activated in mobile unit 16 in which JAM is achieved, and the functions corresponding to the software are achieved, is identical to the operation of mobile unit 16 when the first Java-AP software is activated.

(2-2-3) Third Java-AP Software

[0086] The operation of mobile unit 16 when the installed third Java-AP software is activated in mobile unit 16 in which JAM is achieved, and the functions corresponding to the software (hereinafter, referred to as the third Java-AP) are achieved, will be explained.

[0087] When API, which the third Java-AP is about to use is non-trusted API, the use of the API is approved by JAM. Hence, the third Java-AP can use the API.

[0088] When API, which the third Java-AP is about to use is trusted API, the operation of mobile unit 16 depends upon the API. Hereinafter, the operation of mobile unit 16 is explained with regard to each API.

(2-2-3-1) getPhoneList()

[0089] Since "getPhoneList()" is trusted API, whether the API can be used is determined by JAM on the basis of policy information in SDF 204 stored in fixed memory 16D. The content of the policy information is the content shown in Fig. 7; therefore, the use of"getPhoneList()" is approved by JAM. Hence, the third Java-AP can use "getPhoneList()". In other words, the third Java-AP can read out telephone number directory data.

(2-2-3-2) getCallHistory()

[0090] Since "getCallHistory()" is trusted API, whether the API can be used is determined by JAM on the basis of policy information in SDF 204. Since the content of the policy information is the content shown in Fig. 7, the use of "getCallHistory()" is forbidden by JAM. Hence, the third Java-AP cannot use "getCallHitory()". In other words, the third Java-AP cannot read out history data of transmission and reception.

Nintendo Exhibit 1003
Part 2 of 2
Page 001072

(2-3) Operation after Third Java-AP Software Is Changed

[0091] Next, the operation of the present system after IP which administers IP server unit 13 and IP server unit 14 changes the transmission mode or the content of the third Java-AP software will be explained. However, the present change includes the change of the content of Jar file 206 for the purpose such as the improvement of the third Java-AP software, and the change of the transmission mode for the purpose such as alleviating the burden on IP server unit 13. To achieve the latter change, IP which administers IP server unit 13 and IP server unit 14, as shown in Fig. 19, stores Jar file 206 after the change (hereinafter, referred to as Jar file 215) in fixed memory 14A of IP server unit 14, and produces ADF 216 by changing the content of ADF 205 in accordance with Jar file 215. The above-mentioned operation is required for transmission of the third Java-AP software after the change, and no operation is required for the communication provider, which administers administering server unit 18.

[0092] The installation operation of the third Java-AP software after such changes is shown in Fig. 20. The operation shown in Fig. 20 begins to differ from the operation shown in Fig. 18 when, in IP server unit 13, response message tm 47 containing ADF 216 is produced as opposed to response message tm 37 containing ADF 205. Response message tm 47 corresponds to response message tm 37, response message tm 48 corresponds to response message tm 38, and response message tm 49 corresponds to response message tm 39.

[0093] The operation after response message tm 47 is produced in IP server unit 13 essentially differs from the operation shown in Fig. 18 in that ADF 216 and Jar file 215 are the objects of the process; request message tm 48 requesting transmission of Jar file 215 identified by package URL contained in ADF 216 ("http: //www.sub.bbb.co.jp/viewer.jar") is produced in mobile unit 16; request message tm 48 is transmittedby mobile unit 16, and received by IP server unit 14; response message tm 49 containing Jar file 215 is produced in IP server unit 14; and response message tm 49 is transmitted by IP server unit 14, and is received by mobile unit 16.

(3) Modification

[0094] In the above-mentioned transmission system, ADF and a Jar file are transmitted by the IP server unit, but one or both of these can be transmitted by the administering server unit.

[0095] Also, in the above-mentioned transmission system, the mobile unit confirms the authenticity of the correspondence of the producer of SDF with the producer of ADF on the basis of signature data using the secret key and the public key, but the length of the process in the mobile unit and the IP server unit, or the amount of communication among the mobile unit, the administering server unit, and the IP server unit can be alleviated by not including the public key in SDF; by not signing ADF by using the secret key in the IP server unit; or by omitting the confirmation process in the mobile unit, according to the security level required for the system.

[0096] Also, in the above-mentioned transmission system, the hash value of the Jar file is included in ADF corresponding to the Jar file; and the hash value is calculated in the mobile unit; then the authenticity of the correspondence of the Jar file and ADF is confirmed by comparing the hash value in ADF to the calculated hash value, but the length of the process in the mobile unit and the IP server unit and the amount of communication between the mobile unit and the IP server unit can be alleviated by omitting the confirmation process without including the hash value in ADF depending upon the security level required for the system.

[0097] Also, in the above-mentioned transmission system, whether the correspondence of SDF with ADF (and the Jar file) is authentic is determined by using the inherent APID to trusted Java-AP, but authenticity of the correspondence of SDF with ADF (and the Jar file) can be determined by using the CID inherent to the information provider which provides trusted Java-AP. Also, depending upon the security level required for the system, the determination made on the basis of APID and CID can be omitted.

[0098] Also, in the above-mentioned transmission system, the server is specified by using the domain name, but the server canalso be specified by using the IP address.

[0099] Also, in the mobile unit, by comparing the domain name of the server unit, which transmits SDF to a preset letter string, SDF can be determined to be authentic only when the domain name is that of a server unit administered by a trustworthy organization. In this mode, the letter string to be compared (for instance, the letter string showing the domain name of the communication provider) is pre-stored in ROM or the fixed memory of the mobile unit. When the letter string is pre-stored in ROM, higher security can be assured since the letter string cannot be rewritten. Also, if the letter string is pre-stored in the fixed memory, trustworthy organizations can be stored after the purchase of the mobile unit; therefore, excellent convenience can be provided to a user and a trustworthy organization.

[0100] Also, in the above-mentioned transmission system, a high level of security is assured with a communication provider, which provides the communication path used for transmission of SDF as a trustworthy organization, but the technological scope of the present invention includes the mode in which the communication path is not provided by a trustworthy organization. For instance, by connecting a trustworthy organization to a mobile unit by using an encrypted communication path, the trustworthy organization can transmit SDF via the encrypted communication path. Also, even if the security of the communication path is not assured, by transmitting after encrypting SDF, and decoding SDF in the mobile unit, SDF can

Nintendo Exhibit 1003
Part 2 of 2
Page 001073

be transmitted with a certain degree of security.

[0101] In the above-mentioned transmission system, a file is transmitted and received in accordance with HTTP, but the system can be modified to assure higher security by using HTTPS.

[0102] Also, in the above-mentioned transmission system, a trustworthy organization can be an IP, in other words, the administering unit could comprise an IP server unit.

[0103] Moreover, in the above-mentioned transmission system, API is the object for restricting the use by Java-AP, but any resource can be the object. The resource can be a hardware resource. Also, the resource can be a network resource, or a software resource. A hardware resource can be a resource such as a memory, a speaker, a microphone, an infrared controller, LED (Light Emitting Diode) which a mobile unit can be equipped with, or an external hardware box such as UIM (User Identity Module) or SIM (Subscriber Identity Module) which functions with the mobile unit.

[0104] Next, a network resource is explained. As mentioned above, the mobile unit performs radio communication with the mobile communication network. During radio communication, the mobile unit uses a radio resource such as a radio channel provided by the mobile communication network. The radio resource is one of network resources. Also, the mobile unit, in a higher communication protocol layer than the communication protocol layer the radio resource belongs to, uses a communication resource such as a transmission path of packets or a communication path of the connecting network. The communication resources such as these are included as a network resource.

[0105] Next, a software resource is explained. A software resource can be API, a class, a package and so forth. Various functions are provided by a software resource, but a typical function can be a computation process such as encryption computation, or a function of transmitting or receiving data with other applications such as a Web browser. Also, the technological scope of the present invention includes the mode of restricting the use of a software resource, which the above-mentioned external hardware box is equipped with.

[0106] Incidentally, the use of a hardware resource or a network resource by Java-AP generally takes place by using a software resource. A mobile unit of the above-mentioned transmission system is also equipped with a software resource to use a hardware resource or a network resource, and by restricting the use of a software resource of this type the use of a hardware resource or a network resource is indirectly restricted. By indirectly restricting in this manner, and by preparing various software resources, restrictions which cannot be achieved unless restriction on a plurality of resources are changed in detail, such as giving the right to change the authorization of Java-AP to only trusted Java-AP, lifting the restriction on allowing to communicate only with a server unit accessed for downloading, or allowing to access a specific memory domain can easily be specified. Also, the mode of indirectly restricting the use of a software resource of the above-mentioned external hardware box by restricting the use of a software resource installed in the mobile unit is included in the technological scope of the present invention.

[0107] With regard to a method of expressing permission, a flag (permit/forbid) corresponding to one resource can be used, or permissions of a plurality of resources can be denoted by one expression.

[0108] Also, in the present invention, permission can be denoted to permit (or forbid) the use of a resource with a plurality of types. In this case, in the mobile unit, a more precise control can be achieved. For instance, since two modes (reading out and writing in) exist in the memory, the memory can be used for both reading out and writing in by trusted Java-AP although memory is used only for reading out by non-trusted Java-AP. Also, for instance, when the Web browser and so forth are activated while Java-AP with the right to use a packet transmission path is activated in a mobile unit in which a plurality of applications can share one packet transmission path, control can be such that Java-AP which is permitted to "exclusively use a packet transmission path" can exclusively use a packet transmission path although the Java-AP which is not permitted to "exclusively use a packet transmission path" cannot exclude the sharing of packet transmission path by a Web browser and so forth. Also, by further modifying the above-mentioned modification, the following control can be possible. In other words, Java-AP with a certain type of permission can exclusively use the packet communication path without the user's consent. Also, Java-AP with another permission can use the packet communication path without the user's consent, but needs to obtain the user's consent to exclusively use the packet communication path. Also, Java-AP with another permission can use the packet communication path without the user's consent, but cannot exclusively use the packet communication path. Also, Java-AP with another permission can use the packet communication path only with the user's permission. Also, Java-AP with another permission cannot even use the packet communication path. As it is obvious from these examples, "types of use" of the present invention also contain types of a process when a resource is used (a process of obtaining the user's consent/ a process of not obtaining the user's consent)

[0109] Also, in the above-mentioned transmission system,an identical list page is provided for all mobile units, but a different list page can be provided for each mobile unit.

[0110] Also, in the above-mentioned transmission system, the operation of Java-AP is restricted when the Java-AP is executed. Instead, by including policy information in the Jar file stored in the IP server unit, and when the Jar file is downloaded in the mobile unit, activation of the Java-AP corresponding to the Jar file, or installation of Java-AP software containing the Jar file can be forbidden

Nintendo Exhibit 1003
Part 2 of 2
Page 001074

if comparison between the policy information and the policy information in SDF results in a mismatch. Only the permission given to the item as a result of the match in policy information can be valid.

[0111] Also, SDF can be transmitted after being signed by the secret key which CA gives to the communication provider, and the signature on SDF can be inspected in the mobile unit by the public key, which CA gives to the communication provider. The public key of the communication, provider must necessarily be pre-stored in the mobile unit. The public key can be pre-stored in the fixed memory after being transmitted by means of communication. Also, the mobile unit can be sold after writing the key into ROM.

[0112] Also, in the above-mentioned transmission system, software is delivered to a mobile unit, but the technological scope of the present invention includes the mode of transmitting software to a terminal unit besides a mobile unit.

[0113] Further embodiments are as follows:

E1. A transmission method comprising:

a process for transmitting an authorization file in a communication system in which an originator originates a request including information showing a storage of a file and the file is transmitted in response to said request, said process for transmitting including transmitting a security descriptive file as said authorization file from an administering server unit storing said security descriptive file through a secure link to a terminal unit, the security descriptive file containing first identification information and authorization information, the first identification information showing a storage location of an application descriptive file, the application descriptive file having information dependent upon an entity file, which includes software for executing an application, along with information showing a storage location of said entity file, the authorization information of the security descriptive file indicating an allowable range of an operation of the application which is executed in accordance with said software, said terminal unit executing operation of the application within a range shown by said authorization information;

a dependent information obtaining process for said terminal unit to obtain, by using said first identification information contained in said security descriptive file transmitted from said communication system in said authorization transmission process, said application descriptive file from one or a plurality of server units in which said application descriptive file is stored; and

a program obtaining process for said terminal unit to obtain said entity file from said communication system by using said application descriptive file obtained in said dependent information obtaining process.

E 2. A transmission method according to Embodiment 1,
wherein said application descriptive file is obtained from said administering server unit by said dependent information obtaining process.

E 3. A transmission method according to Embodiment 1,
wherein said entity file is obtained from said administering server unit by said program obtaining process.

E 4. A transmission system according to Embodiment 1,
wherein said application descriptive file is obtained from said administering server unit in said dependent information obtaining process, and
wherein said entity file is obtained from said administering server unit in said program obtaining process.

E 5. A transmission method according to Embodiment 1 further comprising:

an encryption process for said communication system to encrypt said security descriptive file; and

a decoding process for said terminal unit to decode said security descriptive file transmitted by said communication system in said process for transmitting an authorization file,

wherein, in said process for transmitting an authorization file, said security descriptive file encrypted in said encryption process is transmitted to said terminal unit, and
wherein, in said dependent information obtaining process, said terminal unit obtains said application descriptive file by using said security descriptive file decoded in said decoding process.

E 6. A transmission process according to Embodiment 1,
wherein said authorization information shows restriction on use of a resource.

E 7. A transmission method according to Embodiment 6,
wherein said resource is a hardware resource inside said terminal unit.

E 8. A transmission method according to Embodiment 6,
wherein said resource is a hardware resource outside said terminal unit which said terminal unit can

Nintendo Exhibit 1003
Part 2 of 2
Page 001075

use.

E 9. A transmission method according to Embodiment 6,
wherein said resource is a software resource inside said terminal unit.

E 10. A transmission method according to Embodiment 6,
wherein, said resource is a software resource outside said terminal unit which said terminal unit can use.

E 11. A transmission method according to Embodiment 6,
wherein said resource is a network resource which said terminal unit can use.

E 12. A transmission method according to Embodiment 1,
wherein said authorization information shows a type of use of a resource.

E 13. A transmission method according to Embodiment 1,
wherein an application descriptive file corresponding to said application is signed by a secret key a certifying agent gives to an information provider providing said application,
wherein a security descriptive file corresponding to said application contains a public key a certifying agent gives to said information provider, and
wherein, in said program obtaining process, said terminal unit inspects authenticity of an application descriptive file obtained in said dependent information obtaining process by using said public key, and obtains said entity file from said communication system by using said application descriptive file only when authenticity is verified.

E 14. A transmission method according to Embodiment 1,
wherein said application descriptive file and said security descriptive file contain an application identifier an administrator which administers said administering server unit gives, and
wherein, in said program obtaining process, said terminal unit compares an application identifier contained in a security descriptive file transmitted by said administering server unit in said authorization transmission process to an application identifier contained in an application descriptive file obtained in said dependent information obtaining process, and obtains said entity file from said communication system by using said application descriptive file only when both identifiers match.

E 15. A transmission method according to Embodiment 1,
wherein said comunication system further comprises an information providing server unit in which a file for downloading containing a second identification information showing a stored location of said security descriptive file;
an advance transmission process for said communication system to transmit said file for downloading to a terminal unit; and
a process for requesting transmission of an authorization file for said terminal unit to request said communication system for transmission of said security descriptive file by using said file for downloading transmitted by said communication system in said advance transmission process,
wherein, in said authorization transmission process, said communication system transmits to said terminal unit said security descriptive file requested by said process for requesting transmission of an authorization file.

E 16. A transmission method according to Embodiment 1,
wherein a process after said dependent information obtaining process is carried out only when a security descriptive file transmitted in said dependent information transmission process is stored in said administering server unit.

E 17. A transmission method according to any one of Embodiment 1 to Embodiment 16,
wherein said terminal unit is a mobile unit.

E 18. A transmission system comprising:

a communication system for returning, when a stored location of a file is notified, said file which comprises one or a plurality of server units in which an entity file containing software for achieving an application and an application descriptive file which has a content dependent upon said entity file showing a stored location of said entity file, and an administering server unit in which a security descriptive file containing a first identification information showing a stored location of said application descriptive file and authorization information showing authorization given to an application which is achieved when a terminal unit executes said software is stored; and
a terminal unit which approves operation of an application in accordance with authorization given to said application,

wherein said administering server unit transmits said security descriptive file to said terminal unit by assuring security, and
wherein said terminal unit obtains said application

Nintendo Exhibit 1003
Part 2 of 2
Page 001076

descriptive file by using said first identification information contained in said security descriptive file transmitted by said communication system, and said entity file from said communication system by using said application descriptive file.

E 19. A transmission system according to Embodiment 18,
wherein said administering server unit stores said application descriptive file.

E 20. A transmission system according to Embodiment 18,
wherein said administering server unit stores said entity file.

E 21. A transmission system according to Embodiment 18,
wherein, said administering server unit stores said application descriptive file, and wherein said administering server unit stores said entity file.

Claims

1. A terminal unit, comprising:

   an executing unit configured to execute at least a first type of program and a second type of program, the first type of program requiring a security descriptive file, the second type of program not requiring the security descriptive file,
   the security descriptive file including authorization information, the authorization information being information used for controlling access by a program to at least one resource from among a hardware resource, a network resource and a software resource;
   a determining unit configured to determine whether a program to be executed by the executing unit is the first type or the second type;
   a controlling unit configured to control access to the at least one resource in accordance with a first condition if the program to be executed by the executing unit is determined to be the first type, and configured to control access to the at least one resource in accordance with a second condition if the program to be executed by the executing unit is determined to be the second type, the first condition being a condition based on the authorization information.

2. The terminal unit according to Claim 1, wherein
the first condition is more permissive that the second condition.

3. The terminal unit according to Claim 1, wherein
the first condition is a condition to permit access to a resource in accordance with the authorization information as well as access to a resource in accordance with a predetermined condition, and
the second condition is a condition to permit access to a resource in accordance with the predetermined condition.

4. The terminal unit according to Claim 1, wherein:

   the terminal unit further includes a first obtaining unit configured to obtain the authorization information through a network;
   the security descriptive file shows a stored location of the application descriptive file;
   the application descriptive file shows a stored location of an entity file, the entity file including a program to be executed by the executing unit;
   the terminal unit further includes a second obtaining unit configured to obtain the application descriptive file from the stored location shown by the security descriptive file;
   the terminal unit further includes a third obtaining unit configured to obtain the entity file from the stored location shown by the application descriptive file;
   the terminal unit further includes an install unit configured to install in the terminal unit the program included in the entity file; and
   the executing unit is configured to execute the program installed by the installed unit.

5. The terminal unit according to Claim 1, wherein
the software resource includes at least one of API (Application Program Interface), class and package.

6. The terminal unit according to Claim 1, wherein
the authorization information shows a type of use of a resource.

7. The terminal unit according to Claim 1, further comprising
at least one of a memory, a speaker, a microphone, an infrared controller, and a light emitting diode as the hardware resource.

8. The terminal unit according to Claim 1, wherein
the hardware resource includes an external device which is a device other than the terminal unit,

9. The terminal unit according to Claim 1, further comprising
a wireless communication unit configured to communicate through a network, and
the network resource includes at least one of a wireless communication channel, a packet transmission path, a connection provided by the network.

10. A computer-readable storage medium storing a pro-

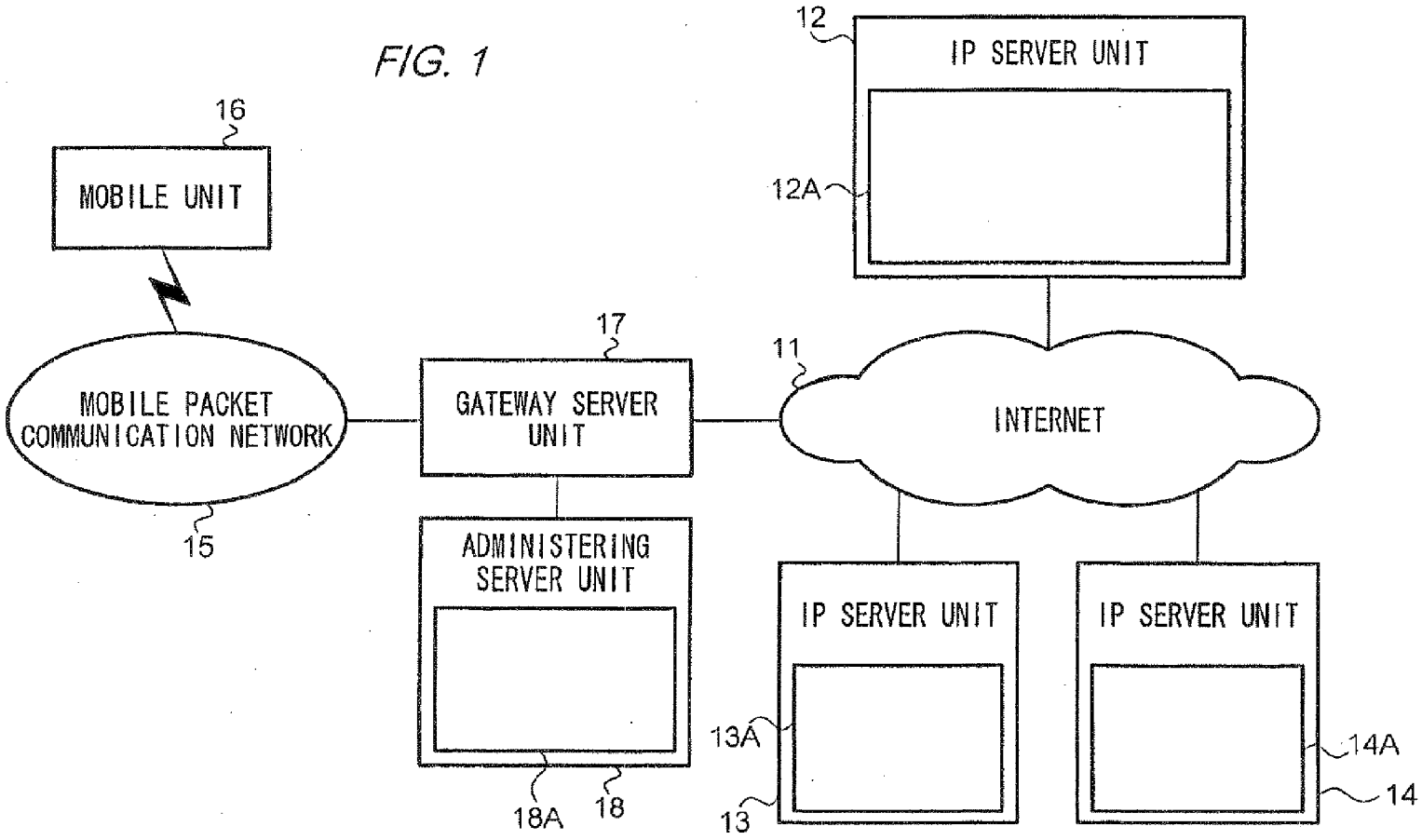
Nintendo Exhibit 1003
Part 2 of 2
Page 001077

gram causing a computer device to execute a process, the computer device including an executing unit configured to execute at least a first type of program and a second type of program, the first type of program requiring a security descriptive file, the second type of program not requiring the security descriptive file, the security descriptive file including authorization information, the authorization information being information used for controlling an access by a program to at least one resource from among a hardware resource, a network resource and a software resource; the process comprising:

determining whether a program to be executed by the executing unit is the first type or the second type;
controlling an access to the at least one resource in accordance with a first condition if the program to be executed by the executing unit is determined to the first type, and configured to control an access to the at least one resource in accordance with a second condition if the program to be executed by the executing unit is determined to the second type, the first condition being a condition based on the authorization information.

Nintendo Exhibit 1003
Part 2 of 2
Page 001078

FIG. 1

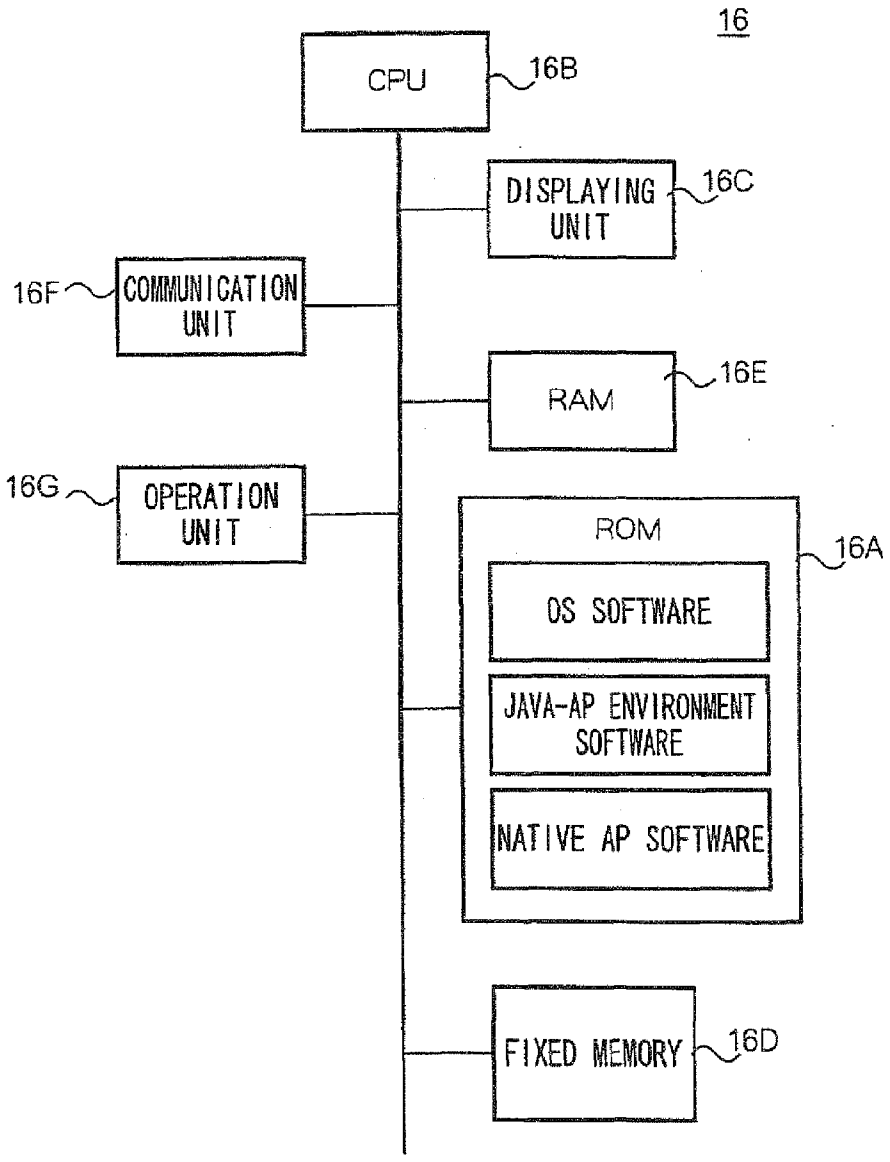


Nintendo Exhibit 1003
Part 2 of 2
Page 001079

FIG. 2

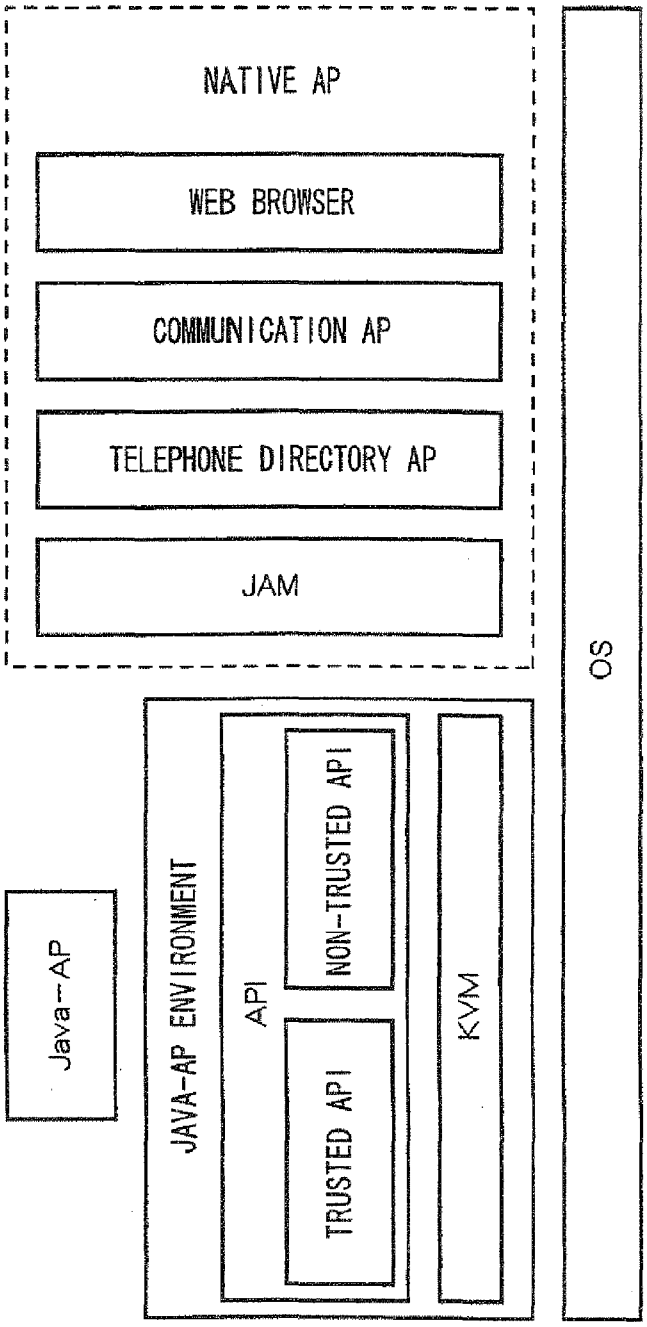


FIG. 3



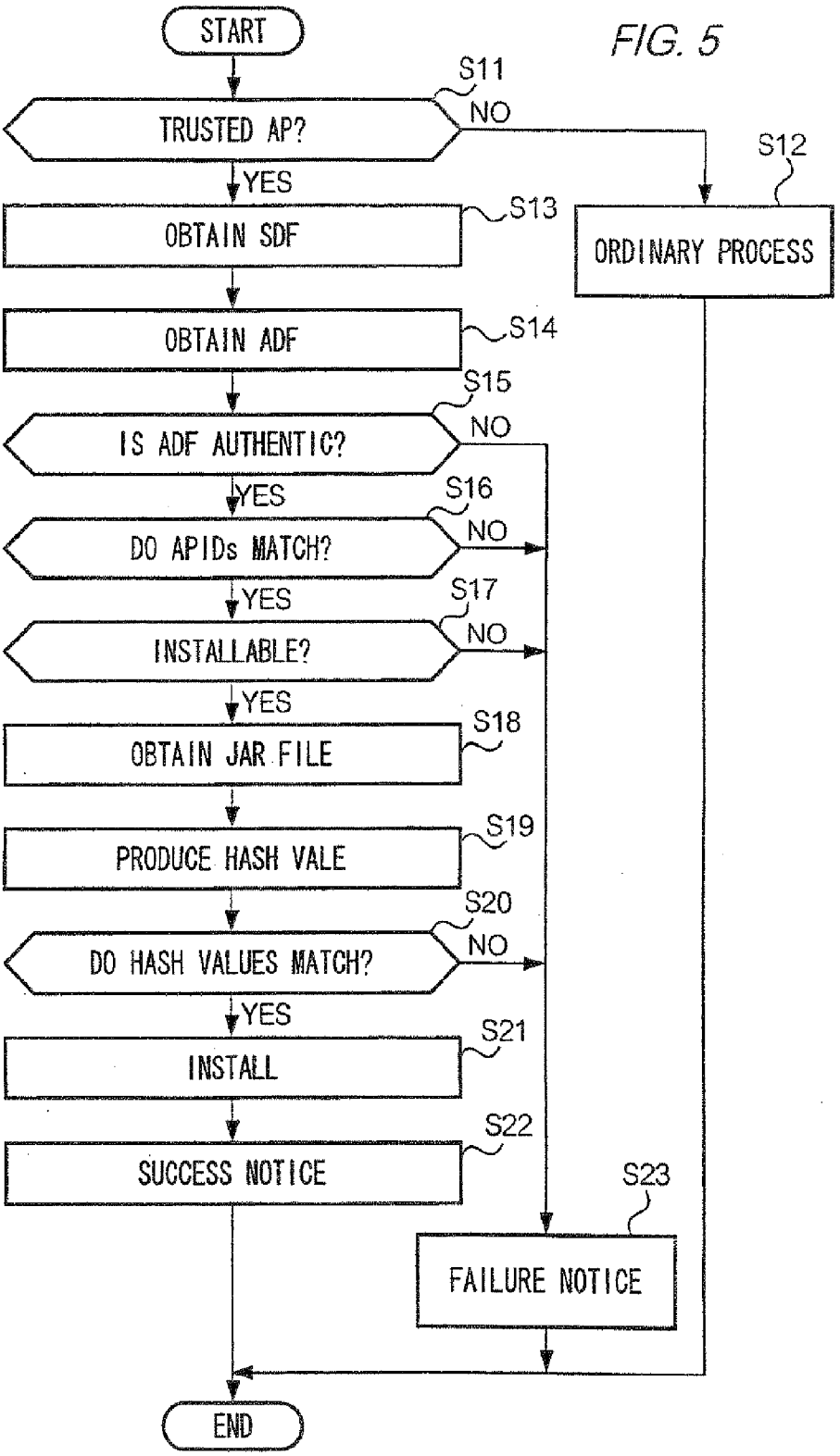
Nintendo Exhibit 1003
Part 2 of 2
Page 001080

FIG. 4



Nintendo Exhibit 1003
Part 2 of 2
Page 001081

FIG. 5



Nintendo Exhibit 1003
Part 2 of 2
Page 001082

FIG. 6

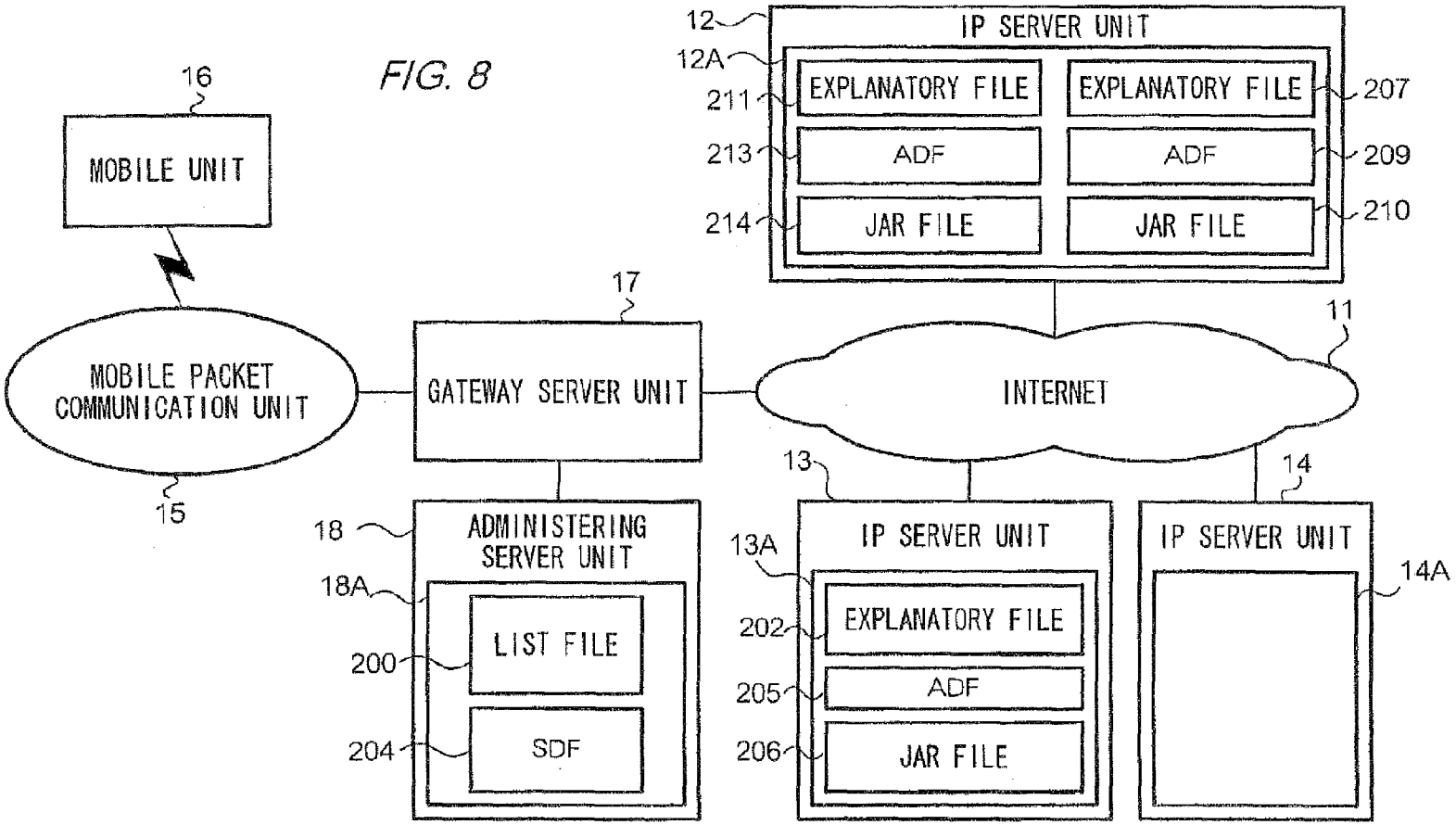
APID	POLICY INFORMATION	ADF-URL	PUBLIC KEY

FIG. 7

TRUSTED API	PERMISSION
getPhoneList()	○
getCallHistory()	×
getMsStatus()	○

Nintendo Exhibit 1003
Part 2 of 2
Page 001083

FIG. 8



Nintendo Exhibit 1003
Part 2 of 2
Page 001084

FIG. 9

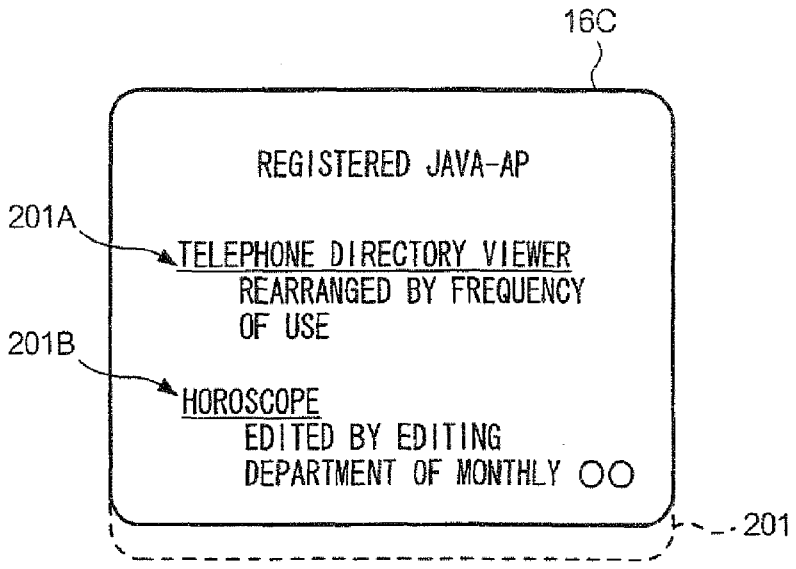


FIG. 10

```
<OBJECT declare id="application.declaration"
data="http://www.ccc.co.jp/horoscope.jam>
TSUME-SHOGI
</OBJECT>
SOFTWARE FOR ~.CLICK
<A ijam="#application.declaration">HERE</A>
TO DOWNLOAD.
```


Nintendo Exhibit 1003
Part 2 of 2
Page 001085

FIG. 11

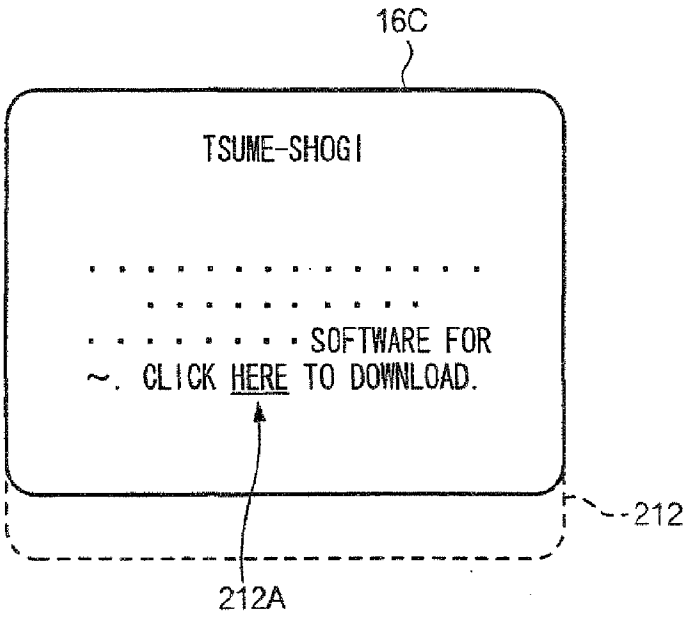


FIG. 12

<OBJECT declare id="application.declaration"
data="http://www.ccc.co.jp/viewer.jam>
HOROSCOPE
</OBJECT>
SOFTWARE FOR ~. CLICK
<A ijam="#application.declaration">HERE</A>
TO DOWNLOAD.

Nintendo Exhibit 1003
Part 2 of 2
Page 001086

FIG. 13

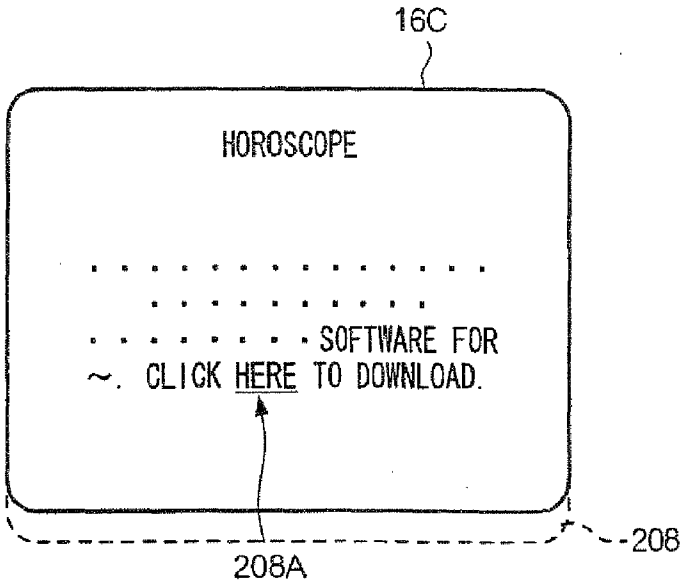
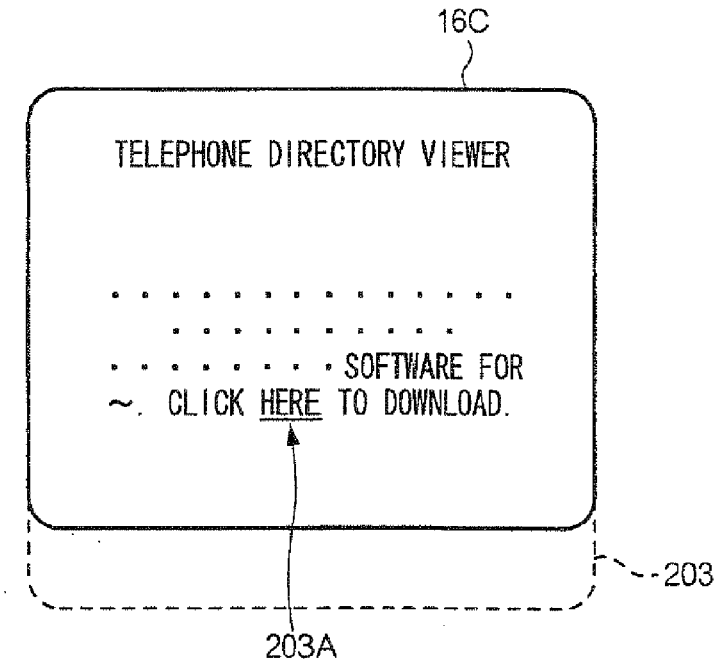


FIG. 14

```
<OBJECT declare id="application.declaration"
data="http://www.aaa.co.jp/abc.sdf"
type="application/x-jam">
TELEPHONE DIRECTORY VIEWER
</OBJECT>
SOFTWARE FOR ~. CLICK
<A ijam="#application.declaration">HERE</A>
TO DOWNLOAD.
```

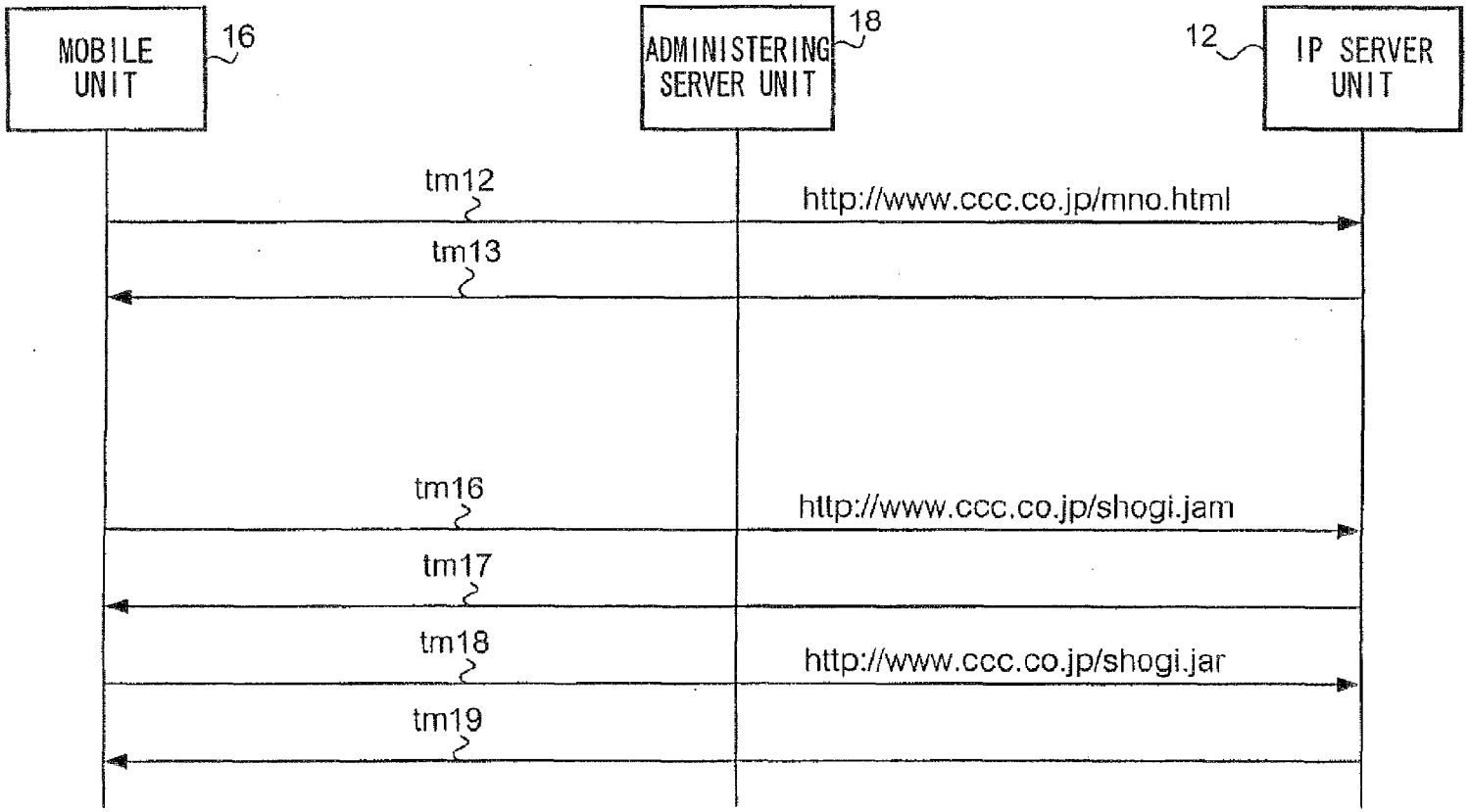

Nintendo Exhibit 1003
Part 2 of 2
Page 001087

FIG. 15



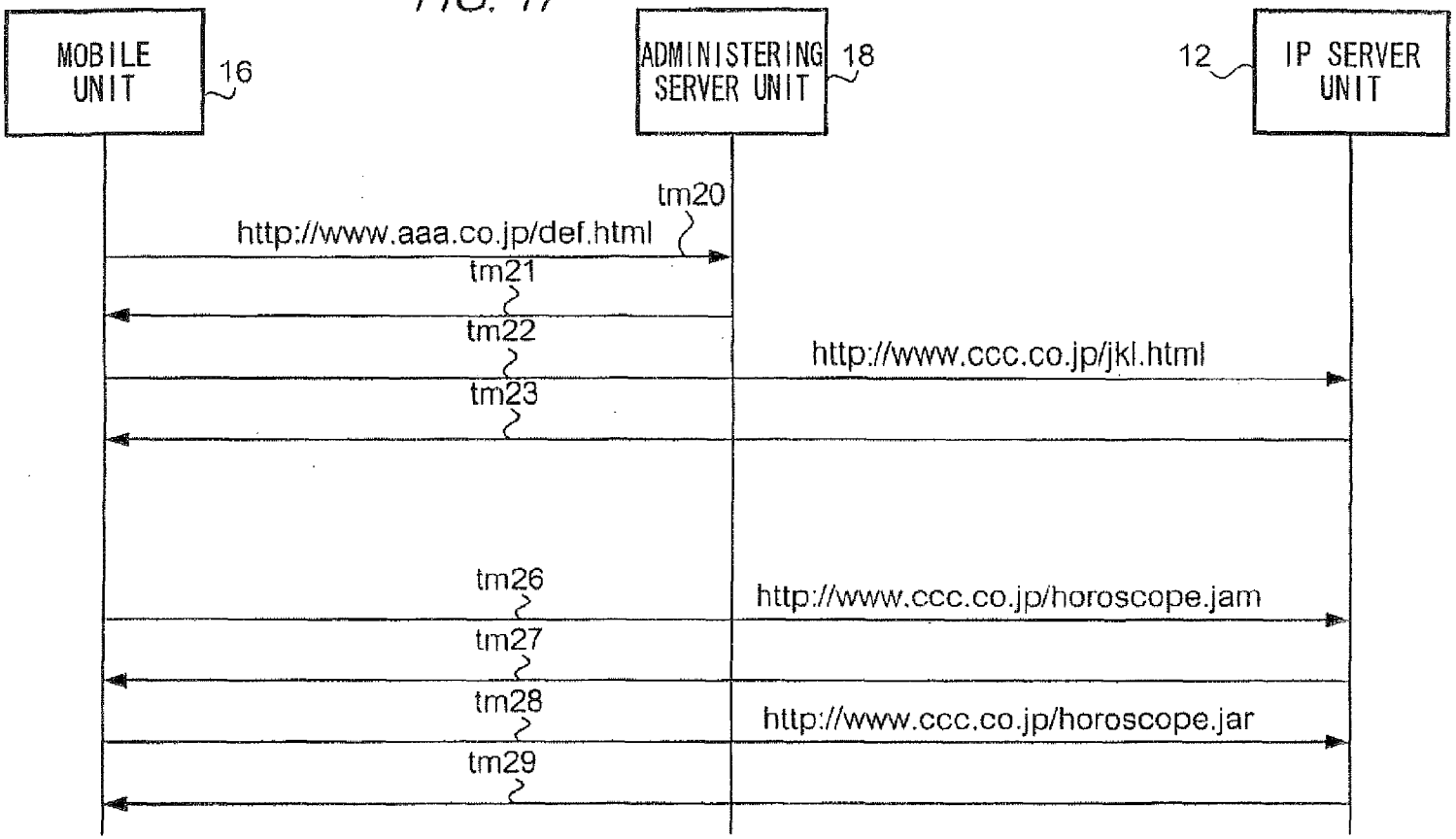
Nintendo Exhibit 1003
Part 2 of 2
Page 001088

FIG. 16



Nintendo Exhibit 1003
Part 2 of 2
Page 001089

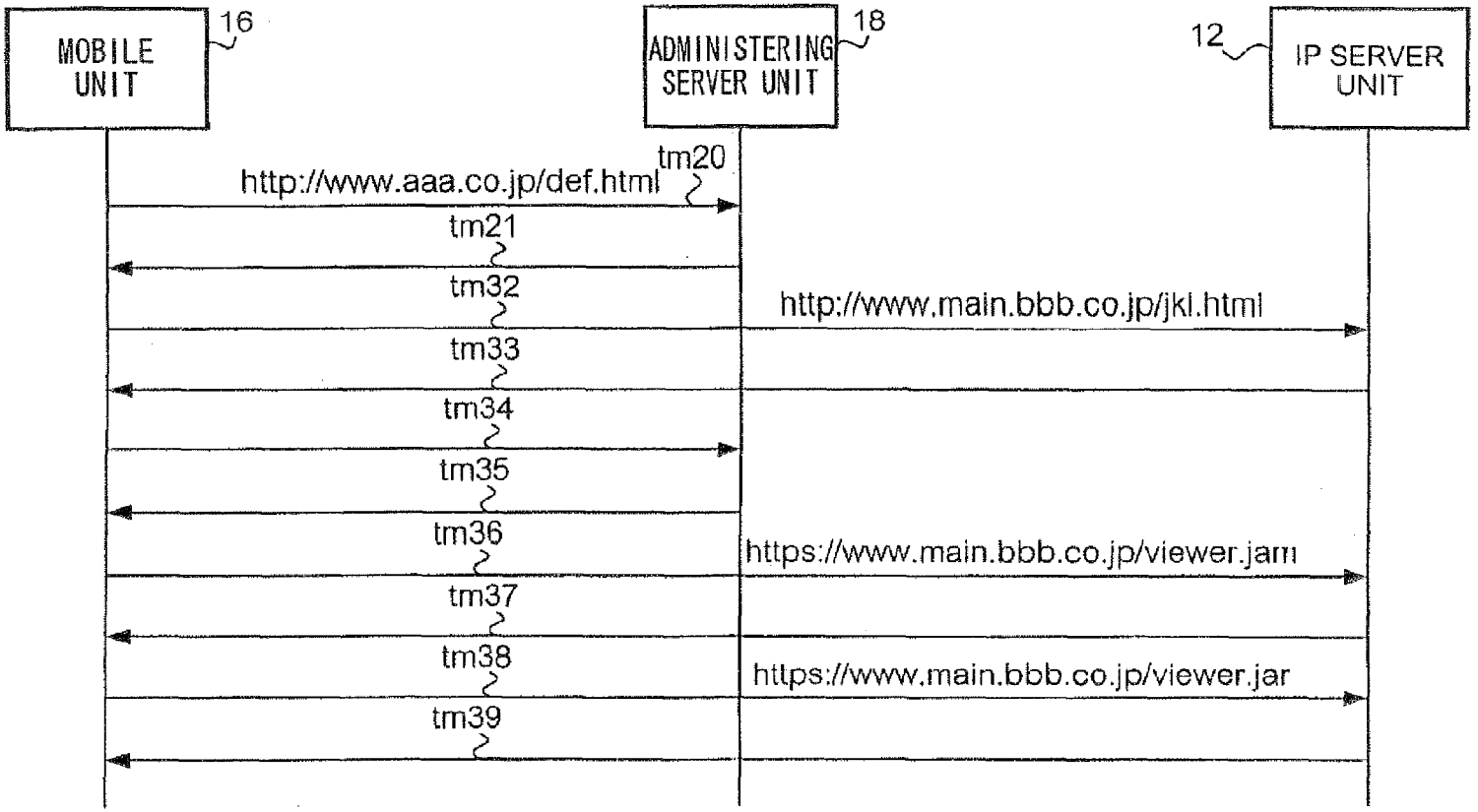
FIG. 17



EP 2 429 148 A2

Nintendo Exhibit 1003
Part 2 of 2
Page 001090

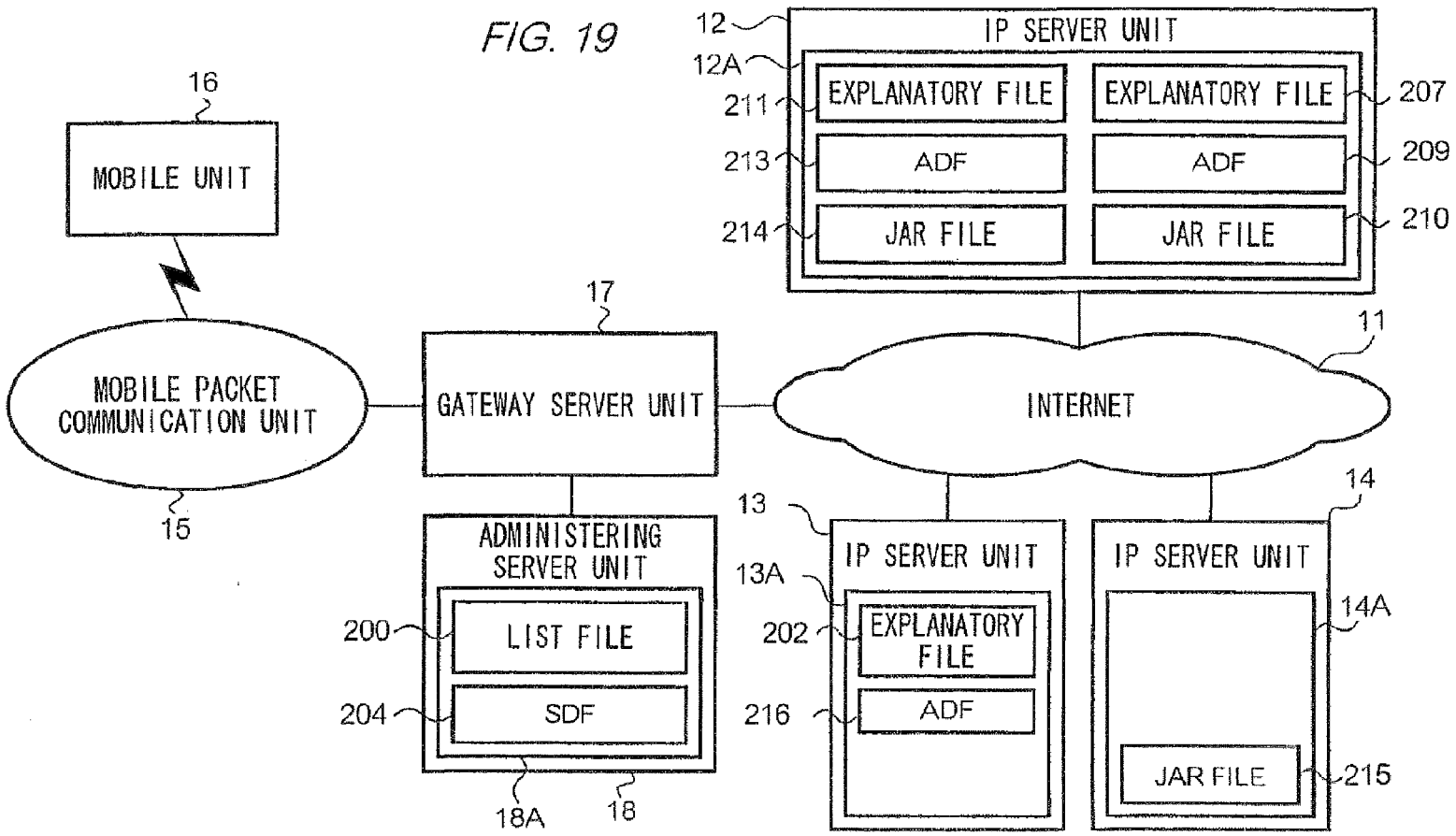
FIG. 18



EP 2 429 148 A2

Nintendo Exhibit 1003
Part 2 of 2
Page 001091

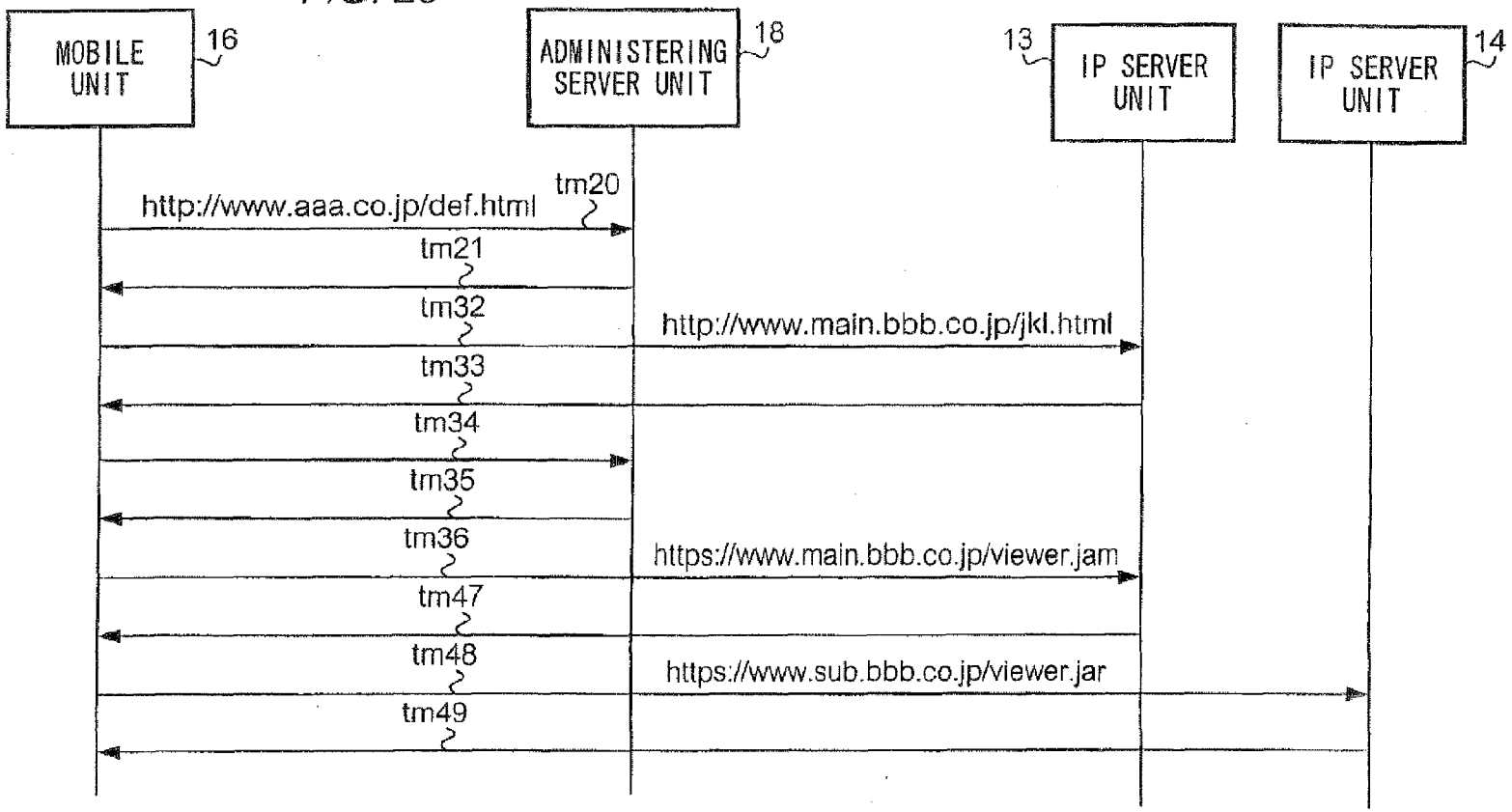
FIG. 19



EP 2 429 148 A2

Nintendo Exhibit 1003
Part 2 of 2
Page 001092

FIG. 20



Nintendo Exhibit 1003
Part 2 of 2
Page 001093

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

(11) EP 1 465 041 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
06.10.2004 Bulletin 2004/41

(51) Int Cl.[7]: **G06F 1/00**

(21) Application number: **04007646.5**

(22) Date of filing: **30.03.2004**

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

(30) Priority: **31.03.2003 JP 2003096015**

(71) Applicant: **NTT DoCoMo, Inc.**
Tokyo 100-6150 (JP)

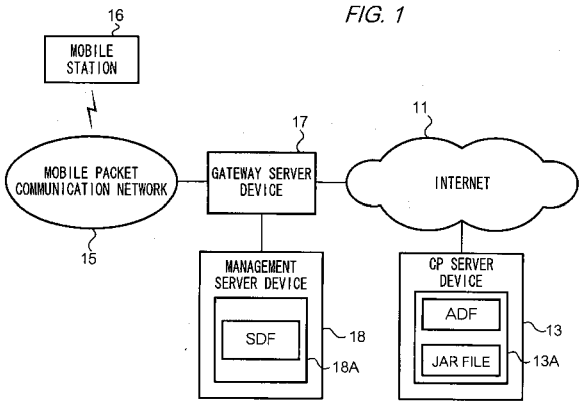
(72) Inventors:
- **Ichikawa, Yuichi**
  2-chome Chiyoda-ku Tokyo 100-6150 (JP)
- **Naruse, Naoki**
  2-chome Chiyoda-ku Tokyo 100-6150 (JP)
- **Oi, Tatsuro**
  2-chome Chiyoda-ku Tokyo 100-6150 (JP)
- **Watanabe, Nobuyuki**
  2-chome Chiyoda-ku Tokyo 100-6150 (JP)
- **Hattori, Yasunori**
  2-chome Chiyoda-ku Tokyo 100-6150 (JP)
- **Takeshita, Masato**
  2-chome Chiyoda-ku Tokyo 100-6150 (JP)
- **Nishida, Masakazu**
  2-chome Chiyoda-ku Tokyo 100-6150 (JP)
- **Asai, Mao**
  2-chome Chiyoda-ku Tokyo 100-6150 (JP)
- **Tsuda, Masayuki**
  2-chome Chiyoda-ku Tokyo 100-6150 (JP)
- **Tomioka, Atsuki**
  2-chome Chiyoda-ku Tokyo 100-6150 (JP)
- **Yamada, Kazuhiro**
  2-chome Chiyoda-ku Tokyo 100-6150 (JP)
- **Kamiya, Dai**
  2-chome Chiyoda-ku Tokyo 100-6150 (JP)
- **Washio, Satoshi**
  2-chome Chiyoda-ku Tokyo 100-6150 (JP)
- **Yamane, Naoki**
  2-chome Chiyoda-ku Tokyo 100-6150 (JP)
- **Murakami, Keiichi**
  2-chome Chiyoda-ku Tokyo 100-6150 (JP)

(74) Representative: **HOFFMANN - EITLE**
Patent- und Rechtsanwälte
Arabellastrasse 4
81925 München (DE)

(54) Communication device, method and program for checking sofware execution permission

(57) The present invention provides a means for reflecting modifications made in a server to data with regard to a scope of rights, which are granted to an application program operable in a communication device such as a mobile station, on data stored in the communication device. To achieve the aim, in a system according to the present invention, Java-AP software is provided to a mobile station by transmitting an ADF, a SDF and a JAR file from servers to the mobile station in that order. The SDF is a file containing data indicating restrictions of behavior of a Java-AP in a mobile station. The SDF also contains data indicating a validity state of the SDF, namely 'valid' or 'invalid', which is managed by management server device (18). Before a mobile station runs a Java-APP which is installed in the mobile station, the mobile station accesses management server device (18) and checks whether a SDF corresponding to the Java-APP is valid. When the SDF is valid, the mobile station runs the Java-APP following the restrictions indicated in the SDF.

FIG. 1



EP 1 465 041 A2

Description

TECHNICAL FIELD

[0001] The present invention relates to a technology for granting application behavior rights.

BACKGROUND ART

[0002] Recently, many new functions for use in mobile stations have been developed. These include, for example, functions for downloading software containing application programs written in Java (registered trademark) programming language (the application program will be referred to as 'Java-APP', and the software will be referred to as 'Java-AP software' hereinafter), through communication networks, and also functions for running Java-APPs.

[0003] Behavior of applications, which are groups of functions realized in mobile stations when Java-APPs are run (the application will be referred to as 'Java-AP' hereinafter) is generally subject to greater restriction than that of native application programs, which application programs are installed in mobile stations prior to distribution to users; and include email communication applications, and the like. (For example, refer to Japanese patent laid-open No. 10-254783.)

[0004] Specifically, Java-APs are generally not permitted to access confidential data, such as telephone directory data stored in a mobile station. The rationale for restricting the behavior of Java-APs in this way is to prevent leakage or falsification of confidential data by Java-APs which are realized by Java-APPs containing either malicious code or code errors.

[0005] However, imposing strict behavior restriction on all Java-APs makes it impossible to meet demands of users who wish to use a variety of new and useful functions of Java-APs. Similarly, such comprehensive restriction also makes it impossible for Contents Providers (referred to as 'CPs' hereinafter) to provide to users Java-APPs which are capable of providing a variety of such functions.

[0006] In a case that reasonable security (reliability) of such Java-APPs can be assured, it is likely that users would not object to Java-APs being able to access personal data stored in a mobile station in which they are run. Moreover, it would be desirable for CPs to be able to provide users with more functional Java-APPs than is currently the case, and which can provide various functions in utilizing personal data stored in mobile stations.

[0007] To meet the demands explained above, different systems have been proposed, such as the one outlined below.

[0008] In this example, a carrier which provides communication services to users of mobile stations grants rights of behavior with fewer restrictions to Java-APs, than those that are normally applied; and the carrier informs the mobile stations of the kind of rights that have been granted to the Java-APs. The carrier would normally be one that is considered to be reliable in terms of security by users; thus, in this explanation the expression 'reliable agency' is used. In the system in question, restriction of behavior of Java-APs is controlled at a mobile station on the basis of rights informed to the mobile station by the reliable agency.

[0009] However, this system suffers from a problem. Namely, during use, the reliable agency informs mobile stations of those rights of behavior for which restrictions are relaxed for Java-APPs operable in the mobile stations. Then, the mobile stations accordingly set restrictions for the behavior of the Java-APs. However, it may become necessary for the reliable agency to modify information about such behavior rights. This would be the case, say, in a situation where it is retrospectively realized that a behavior right relaxation of a Java-AP is having a detrimental effect on users. Such a case could include that in which a Java-AP reads out personal data stored in a mobile station memory and transmits the data to an external device without the consent of the owner. In this case, it would be necessary for the reliable agency to immediately modify allowable behavior of the Java-AP. Simply put, there will be occasions on which to ensure data security it becomes necessary to (rapidly) modify rights of behavior granted by a reliable agency to a Java-AP.

[0010] In consideration of such problems, the present invention has been made to provide a system which enables modifications of rights, which have been granted to communication devices such as mobile stations, to be reflected as restrictions of behaviors in the communication devices.

DISCLOSURE OF THE INVENTION

[0011] To solve the problem of the conventional art described above, the object of the present invention is to provide a communication device comprising: an executing means for executing software; a permission data storing means for storing permission data indicating permissible behavior for an application, which is a group of functions provided by execution of the software; a checking means for checking, by accessing an external device before the software is executed, whether the permission data are valid; and an execution control means for permitting the executing means to execute the software when, on the basis of a result of the check carried out by the checking means, the permission data are determined to be valid, and for not permitting the executing means to execute the software when, on the basis of the result of the check executed by the checking means, the permission data are determined to be invalid.

[0012] According to the communication device of the present invention, since a validity of permission data is checked by accessing an external device before software is executed, secure execution of software can be

Nintendo Exhibit 1003
Part 2 of 2
Page 001095

realized.

[0013] In a preferred embodiment, the present communication device further comprises a determining means for determining whether it is necessary to carry out the check by the checking means, the determining means either carrying out or not carrying out the check on the basis of a determination made by the determination means.

[0014] As will be apparent from the foregoing description, in the communication device of the present invention, it is possible to determine whether a check is actually required. As a result, communication operations can be simplified in comparison with a system in which a check must be conducted each time software is to be executed.

[0015] In another preferred embodiment, in the present communication device, the determining means comprises: a counting means for counting a number of executions of the software; and a frequency data storing means for storing frequency data indicating how frequently it is necessary to carry out the check; and wherein, the determining means determines, on the basis of a number of executions of the software, as counted by the counting means, and on the basis of the frequency data stored in the frequency data storing means, whether it is necessary to carry out a check by the checking means.

[0016] According to the present communication device, the check is carried out on the basis of a number of execution times of the software.

[0017] In another preferred embodiment, in the communication device, the determining means comprises: a timer means for providing time data indicating a present time; and a time interval data storing means for storing time interval data indicating a time interval at which it is necessary to carry out the check; and wherein, the determining means calculates, on the basis of time data provided by the timer, a time period between a present time and a time recorded at a most recent execution of the software, and determines whether it is necessary to carry out the check by the checking means on the basis of the calculated time period and the time interval data stored in the time interval data storing means.

[0018] Thus, according to the present communication device, the check is carried out on the basis of a determined time interval.

[0019] In another preferred embodiment, the communication device further comprises count data storing means for storing count data indicating a count of times that the software is allowed to be executed in a condition that the checking means is unable to make the check; and the execution control means permits the executing means to execute the software in a condition that the checking means is unable to make the check up to a number of times which is indicated by the count data stored in the count data storing means.

[0020] According to the present communication device, especially when the communication device is a mobile station which communicates with an external device wirelessly, a user can start the software up to a certain number of times even when the communication device is unable to make the check in a situation, for example, where a communication failure of some sort occurs, which is not uncommon in wireless systems.

[0021] In another preferred embodiment, the communication device further comprises an updating means for obtaining update data from the external device, and updating the permission data stored in the permission data storing means on the basis of the update data.

[0022] In another preferred embodiment, in the present communication device, the updating means transmits, to the external device, update timing data indicating a timing of a most recent update of the permission data stored in the permission data storing means, when the checking means makes the check; receives update data transmitted from the external device in response to the transmission of the update timing data; and updates the permission data stored in the permission data storing means on the basis of the update data.

[0023] In another preferred embodiment, the communication device further comprises a terminating means for instructing the executing means to terminate execution of the software when the application attempts to conduct behavior which the application is not permitted to conduct.

[0024] In another preferred embodiment, in the communication device, the permission data contain information on usage of at least one of an internal hardware resource of the communication device, an external hardware resource of the communication device, a software resource and a communication network resource.

[0025] The present invention also aims to provide a method for controlling a communication device comprising: a step for transmitting to the communication device permission data, which indicates permissible behavior for an application, which is a group of functions provided by execution of software in the communication device; a step for checking, by communicating data between the communication device and an external device, whether the permission data are valid, before the software is executed in the communication device; and a step for permitting the software to be executed only when the permission data are determined to be valid on the basis of a result of the check.

[0026] The present invention also aims to provide a program for instructing a computer to execute: a process for storing, in a storing means, permission data indicating permissible behavior for an application, which is a group of functions provided by execution of software; a process for checking, by accessing an external device, whether the permission data are valid, before the software is executed; and a process for permitting the software to be executed only when the permission data are determined to be valid on the basis of a result of the check.

Nintendo Exhibit 1003
Part 2 of 2
Page 001096

[0027] The present invention further aims to provide a communication method comprising: a step for transmitting from a communication system comprising (a) a software data providing server device which stores software data containing software for providing a group of functions forming an application, (b) a management server device which stores security descriptor data containing permission data indicating permissible behavior for the application, and (c) an application descriptor data providing server device which stores application descriptor data indicating a storage location of the software data and a storage location of the security descriptor data, to the communication device the application descriptor data; a step for transmitting the application descriptor data from the communication system to the communication device; a step for transmitting data indicating the storage location of the security descriptor data contained in the application descriptor data from the communication device to the communication system; a step for transmitting the security descriptor data from the communication system to the communication device on the basis of the data indicating the storage location of the security descriptor data; a step for storing the security descriptor data in the communication device; a step for transmitting data indicating the storage location of the software data contained in the security descriptor data from the communication device to the communication system; a step for transmitting the software data from the communication system to the communication device on the basis of the data indicating the storage location of the software data; a step for installing, in the communication device, the software contained in the software data transmitted from the communication system to the communication device; a step for checking, by communicating data between the communication device and the communication system before the software is executed in the communication device, whether the security descriptor data stored in the communication device are valid; and a step for permitting the software to be executed in the communication device only when the security descriptor data are determined to be valid on the basis of a result of the check.

[0028] As mentioned above, according to the present invention, software which should not be executed is prevented from executing in a communication device such as a mobile station, since it is checked whether permission data, indicating behavior, which an application realized by the software is permitted to conduct, is valid or not by accessing an external device; and according to a result of the check, undesirable software is not permitted to be started.

BRIEF DESCRIPTION OF THE DRAWINGS

[0029]

Fig. 1 is a block diagram showing configurations of a communication system according to an embodiment of the present invention.

Fig. 2 is a figure showing an example of contents of SDF used in the communication system.

Fig. 3 is a block diagram showing configurations of a mobile station, which is a component of the communication system.

Fig. 4 is a figure showing an example of contents stored in a nonvolatile memory of the mobile station.

Fig. 5 is a conceptual diagram showing functional structures of the mobile station.

Fig. 6 is a flowchart showing processes of the mobile station for downloading and installing Java-AP software.

Fig. 7 is a flowchart showing processes of mobile station for starting the Java-AP software.

Fig. 8 is a figure showing an example of SDF check response used in the communication system.

Fig. 9 is a figure showing an example of SDF check response used in the communication system.

Fig. 10 is a sequence chart showing a series of operations in the communication system.

BEST MODE FOR CARRYING OUT THE INVENTION

[0030] In the following, an embodiment of the present invention will be explained with reference to the drawings. In the drawings, common components are referred to by the same reference numerals.

[0031] First, procedures which are followed by a mobile station for downloading Java-AP software in a known communication system will be explained. The mobile station first receives an Application Descriptor File or an Application Description File (referred to as 'ADF' hereinafter) from a server device contained in World Wide Web (referred to as 'WWW' hereinafter). Then, the mobile station receives a Java Archive file (referred to as 'JAR file' hereinafter).

[0032] An ADF contains data depending on a corresponding JAR file. For example, the ADF contains data showing a URL indicating a location where the JAR file is stored (referred to as 'package URL' hereinafter), data showing a size of the JAR file, data showing last update time of the JAR file, and so on. These data are indispensable to the ADF. After the mobile station receives the ADF, the mobile station determines whether a Java-APP to be downloaded, which is contained in the JAR file, is installable in the mobile station by referring to data contained in the ADF, checking available memory capacity, and so on. When the mobile station determines that the Java-APP is installable, the mobile station receives the JAR file from a server device in WWW using a package URL contained in the ADF. The mobile station completes the download processes when it receives the JAR file which contains the Java-APP. After the completion of the download, the downloaded Java-APP is installed in the mobile station and it becomes possible to be run. Usually, ADFs and JAR files are prepared by CPs who provide Java-AP. 'Java-AP software' is a term for de-

Nintendo Exhibit 1003
Part 2 of 2
Page 001097

scribing a combination of an ADF and a JAR file in the known communication system.

[0033] On the other hand, when a Java-APP is downloaded into a mobile station in the present embodiment, first an ADF corresponding to Java-APP which a user of the mobile station wishes to download is delivered to the mobile station, then a Security Descriptor File or a Security Description File (referred to as 'SDF' hereinafter) which corresponds to the Java-APP is delivered to the mobile station, and finally a JAR file, which contains the Java-APP, is delivered to the mobile station. In short, three different files of ADF, SDF and JAR file are delivered to the mobile station in this order. ADF and JAR file are prepared by a CP who provides the Java-APP in the same way as in the known communication system. However, SDF is prepared by the above-mentioned carrier, following a contract made between the carrier and the CP. In the present embodiment, 'Java-AP software' is also used for describing a combination of an ADF, a SDF and a JAR file.

[0034] A SDF is a file describing a scope of rights which are granted to a corresponding Java-APP stored in a mobile station, and a mobile station runs Java-APP following a scope of rights described in a SDF. A server in the communication network stores data indicating whether each SDF is in a valid state or in an invalid state. Before a mobile station runs a Java-APP which is installed in the mobile station, the mobile station accesses the above-mentioned server to verify whether the SDF corresponding to the Java-APP is valid or invalid. In the following explanation, this verification process will be called 'SDF check'. If a result of the SDF check is 'in a valid state', the mobile station runs Java-APP following the scope of rights described in the SDF. On the other hand, if the result of the SDF check is 'in an invalid state', the mobile station is not allowed to run the Java-APP.

[0035] As explained above, since a state of validity with regard to each SDF can be changed in the server in the communication network, the carrier can easily prevent harmful Java-APP from being run in mobile stations only by setting the SDF corresponding to the Java-APP as in an invalid state.

[0036] 'Harmful Java-APP' means, for example, a Java-APP which may instruct a mobile station to execute an operation which is not expected by a user of the mobile station, such as reading secret data stored in a memory of the mobile station without consent of the user and transmitting the data to an external device. When the carrier finds any harmful Java-APP, the carrier has to only set the state of the SDF of the harmful Java-APP as in an invalid state. When a contract made between a CP and the carrier becomes invalid, for example, because a period of the contract has expired or the CP has not paid the charge indicated in the contract within the deadline, the carrier can also make the Java-APP so as to not be run just by setting a state of the SDF of the Java-APP as in an invalid state. Moreover, in a case that Java-APP is provided for a trial use only during a specified period, the carrier can make the Java-APP to not be run in the same way as explained above when the period for the trial use has passed.

[0037] Conversely, a state of a SDF which is set as in an invalid state may be changed to in a valid state when the carrier wants to allow a Java-APP corresponding to the SDF to be run.

[0038] In the following explanation, to describe a situation where a CPU runs a Java-APP to provide a group of functions, namely a Java-AP, the expression 'a Java-AP is realized' will be used.

(1) Configuration

[0039] In the following, configurations of the communication system according to the present embodiment will be explained.

[0040] As shown in Fig. 1, the communication system comprises CP server device 13 connected to Internet 11, mobile packet communication network 15 which is used by the carrier to provide mobile packet communication services to mobile stations, mobile station 16 which can exchange data packets wirelessly with mobile packet communication network 15 for communicating with an external device through mobile packet communication network 15, gateway server device 17 which interconnects Internet 11 and mobile packet communication network 15, and management server device 18 which is connected to gateway server device 17 via dedicated lines. Practically, the communication system may comprise plural mobile stations and plural CP server devices, but only one of mobile stations 16 and one of CP server devices 13 are shown in Fig. 1 to simplify the figure.

[0041] In the following, details of each component of the communication system will be explained.

(1-1) CP server device

[0042] CP server device 13 has hardware components and features of a general WWW server device. Nonvolatile memory 13A of CP server device 13 stores several JAR files, each of which contains a group of programs written in Java programming language, namely a Java-APP, and several ADFs, each of which contains data with regard to a corresponding JAR file.

[0043] In the communication system according to the present embodiment, some Java-APPs are allowed whose corresponding SDFs do not exist. Behavior of a Java-AP which is realized by a Java-APP whose SDF exists is restricted on the basis of its corresponding SDF. Such a Java-APP is called, in this specification, 'trusted Java-APP' since its reliability is guaranteed by the carrier according to a contract made between the carrier and a CP who provides the Java-APP. Similarly, in this specification a Java-AP which is realized by a trusted Java-APP is called a 'trusted Java-AP'. A Java-APP whose ADF does not exist is a usual Java-APP used in

Nintendo Exhibit 1003
Part 2 of 2
Page 001098

a known communication system, and is called, in this specification, 'untrusted Java-APP' since its reliability is not guaranteed. In the same way, a Java-AP which is realized by an untrusted Java-APP is called, in this specification, an 'untrusted Java-AP'. CP server device 13 can store both ADFs for trusted Java-APPs and ADFs for untrusted Java-APPs.

[0044] An ADF for a trusted Java-APP contains the same data which are contained in an ADF for an untrusted Java-APP such as a package URL indicating a storage location of a JAR file in WWW, data indicating size of the JAR file, data indicating last update time of the JAR file, and so on. An ADF for a trusted Java-APP further contains data called 'APID' for uniquely identifying the JAR file of the trusted Java-APP, and URL called 'SDF-URL' for indicating a storage location of a corresponding SDF in WWW. Mobile station 16 obtains an ADF corresponding to a trusted Java-APP from CP server device 13, and obtains a corresponding SDF using SDF-URL contained in the ADF.

[0045] In the following explanation of the present embodiment, as mentioned above, 'Java-AP software' may mean a combination of an ADF, a SDF and a JAR file when the Java-APP contained in the JAR file is a trusted Java-APP, or a combination of an ADF and a JAR file when the Java-APP is an untrusted Java-APP.

(1-2) Gateway server device

[0046] Gateway server device 17 is managed by the above-mentioned carrier, and has hardware components and features of a general gateway server device which interconnects a mobile packet communication network and Internet. Gateway server device 17 relays data mutually communicated between mobile packet communication network 15 and Internet 11.

(1-3) Management server device

[0047] Management server device 18 is managed by the above-mentioned carrier, and has hardware components and features of a general WWW server device. Nonvolatile memory 18A of management server device 18 stores several SDFs each corresponding to a trusted Java-APP. Each SDF contains, as explained above, data indicating a scope of rights which are granted to the trusted Java-APP. Each SDF further contains data which indicates whether the SDF is valid or invalid. Before mobile station 16 runs a trusted Java-APP, mobile station 16 needs to access management server device 18 and check if the SDF is valid or invalid. On the basis of a result of the check, mobile station 16 determines to execute or not to run the trusted Java-APP.

[0048] Fig. 2 illustrates an example of HTTP message which is transmitted when contents of a SDF are provided by management server device 18 to mobile station 16.

[0049] The line starts with 'Content-Type' shown in Fig. 2, which is a field indicating a type of entity body of the HTTP message. In this case, the field contains a parameter of 'application/x-sdf' which indicates that the entity body relates to a SDF.

[0050] Parameter 'Sts' is for indicating whether the SDF is valid or invalid, and called 'SDF status' in the following explanation. When SDF status is '00', it means the SDF is valid, and when SDF status is '10', it means the SDF is invalid. For example, in the example shown in Fig. 2, since the parameter 'Sts' is '00', the SDF is valid. The parameter 'Package URL' is the same package URL as contained in an ADF as explained above.

[0051] Parameter 'CheckCnt' is for indicating a number of times up to which a mobile station is allowed to run the trusted Java-APP in succession without carrying out a SDF check, and the number of times is called 'count for SDF check' in the following explanation. From a different point of view, a count for SDF check is frequency data showing how frequently a mobile station is required to carry out SDF checks.

[0052] In the present embodiment, for example, a count for SDF check is one of integral numbers between 1 and 999. In the example shown in Fig. 2, the parameter 'CheckCnt' is '005', and it means that mobile station 16 is allowed to run the trusted Java-APP in succession up to 5 times without carrying our a SDF check.

[0053] Parameter 'CheckInt' is for indicating a time interval between timings when SDF checks should be carried out, and called 'interval for SDF check' in the following explanation. More concretely, an interval for SDF check means number of days during which mobile station 16 is allowed to run the trusted Java-APP without carrying out a SDF check after the last SDF check. In the present embodiment, for example, an interval for SDF check is one of integral numbers between 1 and 999. In the example shown in Fig. 2, the parameter 'CheckInt' is '020', and it means that mobile station 16 is allowed to run the trusted Java-APP without executing a SDF check for 20 days once mobile station 16 carries out a SDF check, but when 20 days pass after the last SDF check, mobile station 16 is required to carry out a SDF check prior to running the trusted Java-APP.

[0054] Parameter 'SuspendedCnt' is for indicating a number of times up to which a mobile station is allowed to run the trusted Java-APP in succession in a case that the mobile station is not able to carry out a SDF check, and the number of times is called 'expanded count for SDF check' in the following explanation.

[0055] Mobile station 16 accesses management server device 18 wirelessly and carries out SDF checks. Accordingly, it may become impossible for mobile station 16 to carry out a SDF check when, for example, mobile station 16 is outside the service area of mobile packet communication network 15, namely when it is not reachable, or when a communication failure occurs which may happen more frequently in wireless communications than in cable communications. Therefore, if there is no parameter like 'SuspendedCnt', it may not be pos-

Nintendo Exhibit 1003
Part 2 of 2
Page 001099

sible for a user to use a trusted Java-APP when the user wishes to use it because of a temporary communication failure, and so on, which is highly inconvenient to the user. To avoid such inconvenience, in the present embodiment, it is allowed for a mobile station to run a trusted Java-APP for a limited number of times in succession without carrying out a SDF check even when the mobile station is unable to carry out a SDF check since it is unreachable or in a state of communication failure. In the present embodiment, for example, an expanded count for SDF check is one of integral numbers between 1 and 999. In the example shown in Fig. 2, the parameter 'SuspendedCnt' is '005', and it means that mobile station 16 is allowed to run the trusted Java-APP without carrying out a SDF check up to 5 times in succession even if it is unable to carry out a SDF check.

[0056] Parameter 'Lmd' is for indicating a time and date when the SDF was updated last in management server device 18, and the time and date is called 'last update time' in the following explanation.

[0057] In the example shown in Fig. 2, the parameter 'Lmd' is '20020614120552', and it means that the SDF was last updated 5 minutes and 52 seconds past 12 o'clock on June 14th in year 2002. Last update times are used by management server device 18 for determining whether it is necessary to reflect updated contents of SDFs to data in mobile station 16.

[0058] Parameter 'GetPrivateInfo', 'UserMail', 'MessageApp', 'SetPhoneTheme', 'SetLaunchTime' and 'AllowedHost' are for indicating a scope of rights which are granted to the trusted Java-APP run in mobile station 16, and they are called 'permission data' in the following explanation.

[0059] For example, parameter 'GetPrivateInfo' is for indicating whether a trusted Java-AP realized by the trusted Java-APP is allowed to use a trusted Application Programming Interface (referred to as 'trusted API' hereinafter), which is called indispensably by the Java-AP when it needs to refer to confidential data stored in mobile station 16 such as telephone directory data and unread emails. When the trusted Java-AP is allowed to read confidential data, the parameter 'GetPrivateInfo' is set as 'Yes'. Similarly, parameters 'UserMail', 'MessageApp', 'SetPhoneTheme' and 'SetLaunchTime' are for indicating whether the trusted Java-AP is allowed to call trusted APIs, each of which corresponds to the respective parameter, and when it is allowed, the parameter is set as 'Yes'. In the example shown in Fig. 2, all the parameters are set as 'Yes', and it means that the trusted Java-AP is allowed to call the corresponding trusted APIs.

[0060] Parameter 'AllowedHost' is for indicating URLs of communication devices which a mobile station is allowed to access while the trusted Java-APP is run in the mobile station, and the URLs are called 'access permitted URLs' in the following explanation.

[0061] A Java-APP which is downloaded to mobile station 16 is usually run following a security model which is generally called a sandbox model. According to the sandbox model, while mobile station 16 is running a Java-APP, mobile station 16 is allowed to communicate only with a server from which the Java-APP was downloaded. Under such a strict restriction, it is difficult to provide various kinds of application programs to a user of mobile station 16. Therefore, in the present embodiment, when a trusted Java-APP is run, mobile station 16 is permitted to communicate with certain external communication devices which are specified in advance, in addition to the server from which the trusted Java-APP was downloaded. The parameter 'AllowedHost' shows URLs of external communication devices with which a mobile station is permitted to communicate. In the example shown in Fig. 2, the parameter shows that a mobile station is permitted to communicate with external communication devices whose URLs contain 'http://aaa.co.jp' or 'http://bbb.co.jp' using the port of '8080'.

[0062] When the parameter of access permitted URLs is set as 'any', it means that a mobile station is permitted to communicate with any external communication device while the trusted Java-APP is run. Since it may become difficult to keep security of data stored in the mobile station if the mobile station is permitted to communicate with any external communication device following the parameter 'any', it is also possible to specify URLs of certain external communication devices with which a mobile station is not permitted to communicate. Parameter 'DisallowedHost' is prepared for the purpose, and URLs of external communication devices with which a mobile station is not permitted to communicate, which are called 'access prohibited URLs' in the following explanation, are set to the parameter.

(1-4) Mobile station

[0063] Mobile station 16 comprises, as illustrated in Fig. 3, Operating System (referred to as 'OS' hereinafter) software, ROM 16A which stores a Java environment program for constructing an environment for making Java-APPs operable and several other native application programs, CPU 16B which runs programs stored in ROM 16A, display unit 16C, nonvolatile memory 16D, RAM 16E, communicating unit 16F, operating unit 16G and timer 16H. These components are connected to one another through a data bus.

[0064] Display unit 16C comprises, for example, a liquid crystal display panel and a panel driver circuit, and displays images following data which are provided by CPU 16B. Communicating unit 16F comprises an antenna and a wireless communication unit, wirelessly communicates data packets with mobile packet communication network 15, and relays data packets between CPU 16B and mobile packet communication network 15. Communicating unit 16F comprises a CODEC, a microphone and a speaker for voice communications, and enables mobile station 16 to conduct voice communica-

Nintendo Exhibit 1003
Part 2 of 2
Page 001100

tions through a mobile telephone network (not shown) which has a line switching system. Operating unit 16G comprises hardware such as keypad for operations of a user, and provides CPU 16B with certain signals in accordance with operations done by the user. Timer 16H keeps the present year, month, date and time (referred to just as 'present time' hereinafter). To make timer 16H keep the correct present time, mobile station 16 may regularly receive data indicating the accurate present time using a control channel from a base station (not shown) in mobile packet communication network 15, and adjust the time kept by timer 16H.

[0065] Nonvolatile memory 16D is, for example, a Static Random Access Memory (SRAM) or an Electrically Erasable and Programmable Read Only Memory (EEPROM). Nonvolatile memory 16D stores Java-AP software which are downloaded from servers comprised in WWW.

[0066] Details of an ADF and a SDF of trusted Java-AP software, which are stored in nonvolatile memory 16D, will be explained below.

[0067] Fig. 4 shows an example of data which are generated in accordance with an ADF and the SDF shown in Fig. 2 of trusted Java-AP software, and stored in nonvolatile memory 16D. As shown in Fig. 4, data stored in nonvolatile memory 16D contains a data field of 'number of execution times', a data field of 'number of passed days' and a data field of 'used expanded count for SDF check' which correspond to 'count for SDF check', 'interval for SDF check' and 'expanded count for SDF check' respectively.

[0068] A number of execution times indicates a number of times that mobile station 16 ran the trusted Java-APP after the most recent execution of a SDF check. After the number of execution times reaches the count for SDF check, namely 5 in the example case shown in Fig. 4, mobile station 16 determines that it is required to carry out a SDF check before it runs the trusted Java-APP the next time.

[0069] A number of passed days indicates a time period, namely number of days, which passed after the most recent execution of a SDF check by mobile station 16. The passed days are calculated on the basis of the time kept by timer 16H continuously. After the number of passed days reaches the interval for SDF check, namely 20 in the example case shown in Fig. 4, mobile station 16 determines that it is required a SDF check to be carried out before it runs the trusted Java-APP the next time. A used expanded count for SDF check means a number of times that mobile station 16 ran the trusted Java-APP without carrying out SDF check when a SDF check is required since mobile station 16 was not able to carry out a SDF check. After the used expanded count for SDF check reaches the expanded count for SDF check, namely 5 in the example case shown in Fig. 4, mobile station 16 does not run the trusted Java-APP unless it carries out a SDF check successfully.

[0070] When a power supply (not shown) of mobile station 16 is turned on, CPU 16B reads the OS program stored in ROM 16A and runs the OS program using RAM 16E as a work area. Following instructions of the OS program, CPU 16B establishes an OS environment in mobile station 16 as illustrated in Fig. 5. After the OS environment is established, CPU 16B is enabled to identify instructions given by the user on the basis of signals provided by operating unit 16G, and execute required data processing in accordance with the instructions.

[0071] In the following, when a description like 'the OS executes an operation' is used, it means that CPU 16B executes the operation following instructions provided by the OS program. In the same way, when a description like 'the application executes an operation' is used, it means that CPU 16B runs the application program and executes the operation following instructions provided by the application program. Several applications shown in Fig. 5, such as Java-AP, Java Application Manager (referred to as 'JAM' hereinafter) and telephone directory AP are examples of such application.

[0072] When the user instructs mobile station 16 to run a communication program, which is one of native application programs, the OS starts the communication program and realizes a communication AP in mobile station 16. Once the communication AP is realized, the user is able to make a voice call to an external communication device. When the user instructs mobile station 16 to run a telephone directory program, which is also one of native application programs, the OS starts the telephone directory program and realizes a telephone directory program in mobile station 16. The user is able to make mobile station 16 display, use and edit the data of telephone directory (referred to as 'telephone directory data' hereinafter) stored in nonvolatile memory 16D by using telephone directory AP. When the user instructs mobile station 16 to run a Web browser program, which is also one of native application programs, the OS starts the Web browser program and realizes a Web browser in mobile station 16.

[0073] When the user instructs mobile station 16 to run a JAM program, which is also one of the native application programs, the OS starts the JAM program and realizes a JAM in mobile station 16. JAM shows the user a list of Java-APPs which are installed in mobile station 16, and executes one of the Java-APPs which is selected by the user. More concretely, if the instruction provided by the user is a request for starting one of the Java-APPs, first a Java-AP environment program is executed to establish a Java-AP environment in mobile station 16, then the Java-APP selected by the user is run to realize the Java-AP in the Java-AP environment. A Java-AP environment contains, for example, K Virtual Machine (KVM) which is a lightweight Java virtual machine tuned for a mobile communication device such as mobile station 16, and APIs which provide several functions to Java-APs. APIs which are prepared for Java-APs are categorized into trusted APIs which only trusted Java-APs, namely Java-APs which are realized

Nintendo Exhibit 1003
Part 2 of 2
Page 001101

by trusted Java-APPs, are permitted to use, and untrusted Java-APIs which all Java-APs are permitted to use.

[0074] The JAM further provides functions for managing behavior of Java-APs.

[0075] For example, when the JAM receives a request for running a Java-APP, the JAM determines whether a SDF check is required with regard to the SDF of the Java-APP, and if the JAM determines that the SDF check is required, it carries out the SDF check. For the purpose, the JAM provides functions for generating and updating data indicating the number of execution times, the number of passed days and the used expanded count for SDF check which are to be stored in nonvolatile memory 16D. Moreover, when a result of the SDF check indicates that the SDF is valid, the JAM permits the Java-APP to be run, and after the Java-AP is realized, the JAM restricts behavior of the Java-AP according to permission data contained in the SDF. On the other hand, when the result of the SDF check indicates that the SDF is invalid, the JAM does not permit the Java-APP to be run. When the JAM determines on the basis of the result of the SDF check that it is necessary to update the SDF, the JAM again accesses management server device 18, obtains the updated SDF, and updates data stored in mobile station 16.

(2) Operation

[0076] In the following, operations of the communication system, whose configurations are described above, will be explained.

(2-1) Installation of Java-APP

[0077] When the JAM receives a request for downloading a Java-APP from the Web browser, the JAM starts a series of processes for installing the Java-APP into mobile station 16. Fig. 6 illustrates a flow of the processes.

[0078] As illustrated in Fig. 6, the JAM first downloads from CP server device 13 an ADF corresponding to the Java-APP to be downloaded (step S11). More concretely, the JAM generates a HTTP request containing an ADF-URL, transmits the HTTP request to CP server device 13, and receives a HTTP response containing the ADF from CP server device 13 in response to the HTTP request. In this process, an ADF-URL contained in a HTTP request may be input manually by the user, or selected by the user from candidate URLs embedded in a web page described in HyperText Markup Language (referred to as 'HTML' hereinafter). The JAM instructs nonvolatile memory 16D to store data contained in the downloaded ADF such as an APID, a package URL and a SDF-URL an ADF-URL as shown in Fig. 4.

[0079] Next, the JAM determines whether it is possible for the Java-APP to be downloaded and installed into mobile station 16 on the basis of the data of the ADF stored in nonvolatile memory 16D (step S12). The JAM may carry out the determination in the same way as in the known system such as by comparing the size of a JAR file indicated in the ADF and available free space of nonvolatile memory 16D where the JAR file is to be stored.

[0080] If the JAM determines that the installation is possible (step S12; Yes), the JAM determines whether the Java-APP to be downloaded is a trusted Java-APP or an untrusted Java-APP (step S13). More concretely, the JAM checks whether an APID and a SDF-URL are contained in the data stored in step S11, and if they are contained, the JAM determines that there is a SDF corresponding to the Java-APP, which means that the Java-APP is a trusted Java-APP. Contrarily, if an APID and a SDF-URL are not contained, the JAM determines that the Java-APP is an untrusted Java-APP. Now it is supposed that an APID and a SDF-URL are contained in the downloaded ADF, and the JAM determines that the Java-APP to be downloaded is a trusted Java-APP (step S13; Yes).

[0081] The JAM receives a SDF corresponding to the trusted Java-APP from management server device 18 (step S14). More concretely, the JAM establishes a TCP connection between management server device 18, generates a HTTP request containing the SDF-URL obtained from the ADF, transmits the HTTP request to management server device 18, receives a HTTP response (please refer to Fig. 2) in response to the HTTP request, and disconnects the TCP connection.

[0082] The JAM verifies whether the received SDF contains appropriate data (step S15). More concretely, the JAM checks whether the received SDF is described following a certain format, checks whether an APID contained in the SDF and the APID stored in nonvolatile memory 16D matches, and so on, and all results of the checks are affirmative, the JAM determines that the received SDF contains correct appropriate data (step S15; Yes). Then, the JAM instructs nonvolatile memory 16D to store the data contained in the SDF as shown in Fig. 4.

[0083] The JAM downloads the JAR file from CP server device 13 (step S16). More concretely, the JAM generates a HTTP request containing the package URL stored in nonvolatile memory 16D, transmits the HTTP request to CP server device 13 and receives a HTTP response containing the JAR file from CP server device 13 in response to the HTTP request.

[0084] The JAM instructs the received JAR file to store nonvolatile memory 16D and installs the trusted Java-APP contained in the JAR file following normal procedures for installing a Java-APP (step S17). Then the JAM notifies to the user that the installation is completed successfully (step S 18).

[0085] After the installation is completed, when the trusted Java-APP is run, the JAM monitors behavior of the trusted Java-AP which is realized by the Java-APP, and restricts the trusted Java-AP to using only certain trusted APIs according to permission data contained in the SDF and stored in nonvolatile memory 16D.

Nintendo Exhibit 1003
Part 2 of 2
Page 001102

[0086] During the above-mentioned processes, when it is determined that the Java-APP is unable to be installed (step S12; No), or when the SDF does not contain appropriate data (step S15; No), the JAM notifies the user that the installation failed (step S20), and restores software configurations in mobile station 16 to a state previous to step S11.

[0087] During the above-mentioned processes, when it is determined that the Java-APP to be installed is an untrusted Java-APP (step S13; No), the JAM downloads the JAR file from CP server device 13 according to a package URL contained in the ADF following an ordinary procedure (step S19), and installs an untrusted Java-APP contained in the JAR file following normal procedures for installing a Java-APP (step S 17). Then the JAM notifies to the user that the installation is completed successfully (step S 18).

(2-2) Starting of Java-APP

[0088] In the following, operations for starting a Java-APP will be explained with reference to Fig. 7.

[0089] When the JAM receives from the user a request for starting a Java-APP, the JAM determines whether the Java-APP to be started is a trusted Java-APP or an untrusted Java-APP (step S31). When it is determined that a trusted Java-APP is to be started (step S31; Yes), the JAM determines whether a SDF check is required (step S32). More concretely, the JAM checks data stored in nonvolatile memory 16D, and determines that a SDF check is required when the number of execution times is more than the count for SDF check, or when the number of passed days is more than the interval for SDF check. On the other hand, the JAM determines that a SDF check is not required when both of the conditions are not met.

[0090] For example, according to the data shown in Fig. 4, both of the above-mentioned conditions are not met, the JAM determines that a SDF check is not required (step S32; No). Then, the JAM adds 1 to the number of execution times and updates the number stored in nonvolatile memory 16D (step S33), and starts the trusted Java-APP (step S34).

[0091] After the same trusted Java-APP is repeatedly started by the JAM as explained above, the number of execution times may become the same as the count for SDF check. In the case, the JAM determines that a SDF check is required (step S32; Yes), generates a HTTP request for requesting a SDF check operations, and transmits the HTTP request to management server device 18 (step S35). The HTTP request contains an APID and data indicating last update time of the SDF corresponding to the trusted Java-APP to be started, namely the value of parameter 'Lmd'.

[0092] After receiving the HTTP request, management server device 18 extracts the APID from the HTTP request, reads a SDF containing the APID stored in nonvolatile memory 18A and checks a validity status of the SDF. Management server device 18 also extracts the last update time from the HTTP request, compares the extracted last update time and a last update time contained in the SDF read from nonvolatile memory 18A, and determines whether it is necessary to update the SDF stored in mobile station 16. More concretely, management server device 18 determines that the update is necessary if the last update time extracted from the HTTP request is older than the last update time in the SDF stored in nonvolatile memory 18A, and determines that the update is not necessary if both of the last update times are the same. According to a result of the above-mentioned determination, management server device 18 generates a HTTP response containing data indicating a validity status of the SDF, and transmits the HTTP response to mobile station 16 as a SDF check response.

[0093] Fig. 8 and Fig. 9 show examples of SDF check response.

[0094] Fig. 8 shows a SDF check response which is generated when the SDF is valid. In the case, as shown in Fig. 8, the value of parameter 'Sts', which indicates the validity status of the SDF, is set as '00' meaning 'valid'. On the other hand, Fig. 9 shows a SDF check response which is generated when the SDF is invalid. In the case, as shown in Fig. 9, the value of parameter 'Sts' is set as '10' meaning 'invalid'. When the SDF is valid but it is updated, the value of parameter 'Sts' is set as '01' indicating that it is necessary for the SDF to be updated (not shown in the drawings). When a SDF containing the APID received from mobile station 16 is not stored in nonvolatile memory 18A, the value of parameter 'Sts' is set as '99' indicating that the SDF could not be found (not shown in the drawings).

[0095] When mobile station 16 receives the SDF check response (step S36; Yes), the JAM checks the contents of the SDF check response (step S37).

[0096] For example, when the JAM receives a SDF check response as shown in Fig. 8, the JAM determines that the SDF is valid (step S37; valid), clears the number of execution times, the number of passed days and the used expanded count for SDF check stored in nonvolatile memory 16D (step S38), and starts the trusted Java-APP (step S34).

[0097] On the other hand, when the JAM receives a SDF check response as shown in Fig. 9, the JAM determines that the SDF is invalid (step S37; invalid), displays a message that the Java-APP is unable to be started, and restores software configurations in mobile station 16 to a state previous to step S31 (step S39).

[0098] When the received SDF check response indicates that the SDF is updated, the JAM determines that the SDF stored in nonvolatile memory 16D is required to be updated (step S37; update), and carries out procedures for updating the SDF (step S40). More concretely, similarly to operations in step S14 of Fig. 6, the JAM receives data of the updated SDF from management server device 18, verifies that the received data of the updated SDF are appropriate, and stores the data

Nintendo Exhibit 1003
Part 2 of 2
Page 001103

of the updated SDF in nonvolatile memory 16D.

[0099] During the above-mentioned operations, if the JAM is unable to start or fails to complete a SDF check because, for example, mobile station 16 is in an unreachable state or a communication failure happens (step S36; No), the JAM checks the data stored in nonvolatile memory 16D and determines whether the used expanded count for SDF check is less than the expanded count for SDF check (step S41).

[0100] When the used expanded count for SDF check is less than the expanded count for SDF check (step S41; Yes), the JAM adds 1 to the used expanded count for SDF check and updates the count stored in nonvolatile memory 16D (step S42). Then, the JAM starts the trusted Java-APP (step S34).

[0101] On the other hand, when the used expanded count for SDF check is the same as or more than the expanded count for SDF check (step S41; No), the JAM displays a message that it is unable to carry out a SDF check and the Java-APP is unable to be started, and restores software configurations in mobile station 16 to a state previous to step S31 (step S39).

(2-3) Example of operations of the overall communication system for making a trusted Java-APP operable

[0102] In the following, an example of a series of operations for making a trusted Java-APP operable, which follows the above-mentioned steps, will be explained with reference to Fig. 10.

[0103] In the following explanation, operations which are carried out by the JAM or other applications, such as Java-APs, are described as operations which are carried out by 'mobile station 16'.

[0104] In Fig. 10, when mobile station 16 receives from the user a request for downloading a Java-APP, mobile station 16 generates HTTP request m1 which contains an ADF-URL of an ADF corresponding to the Java-APP to be downloaded, and transmits HTTP request m1 to CP server device 13.

[0105] In response to HTTP request m1, CP server device 13 generates HTTP response m2 which contains the corresponding ADF, and transmits HTTP response m2 to mobile station 16.

[0106] After receiving HTTP response m2, mobile station 16 stores data of the ADF in nonvolatile memory 16D. If mobile station 16 determines that it is possible to install the Java-APP into mobile station 16, mobile station 16 generates HTTP request m3 which contains a SDF-URL of a SDF, and transmits HTTP request m3 to management server device 18.

[0107] After receiving HTTP request m3, management server device 18 generates HTTP response m4 which contains the SDF, and transmits HTTP response m4 to mobile station 16 in response to HTTP request m3.

[0108] After receiving HTTP response m4, mobile station 16 verifies that contents of the SDF are appropriate, and stores the SDF in nonvolatile memory 16D. Then, mobile station 16 generates HTTP request m5 which contains a package URL, and transmits HTTP request m5 to CP server device 13.

[0109] , In response to HTTP request m5, CP server device 13 generates HTTP response m6 which contains a JAR file, and transmits HTTP response m6 to mobile station 16. The JAR file contains the trusted Java-APP to be downloaded.

[0110] After receiving HTTP response m6, mobile station 16 stores the received JAR file in nonvolatile memory 16D, and carries out procedures for installing the trusted Java-APP.

[0111] After the installation of the trusted Java-APP is completed, when mobile station 16 receives an instruction for starting the Java-APP, mobile station 16 determines whether a SDF check is required or not, and if it is determined that a SDF check is required, mobile station 16 transmits HTTP request m7 which contains an APID and a last update time of the SDF to management server device 18 as a SDF check request.

[0112] After receiving HTTP request m7, management server device 18 generates HTTP response m8 which contains data indicating a validity status of the SDF corresponding to the APID contained in HTTP request m7, and transmits HTTP response m8 to mobile station 16 as a SDF check response.

[0113] After receiving HTTP response m8, mobile station 16 checks the contents of the SDF check response. For example, when the SDF check response indicates that it is necessary for the SDF to be updated, mobile station 16 updates the SDF. Namely, mobile station 16 generates HTTP request m9 which contains the SDF-URL, and transmits HTTP request m9 to management server device 18.

[0114] After receiving HTTP request m9, management server device 18 generates HTTP response m10 which contains the SDF identified by the SDF-URL contained in HTTP request m9, and transmits HTTP response m10 to mobile station 16.

[0115] After receiving HTTP response m10, mobile station 16 starts the Java-APP, and runs the Java-APP according to the contents of the updated SDF.

(2-4) Behavior management of mobile station 16 while Java-APP is run

[0116] In the following, management of behavior of mobile station 16 when the above-mentioned Java-APP is run will be explained.

(2-4-1) Untrusted Java-AP

[0117] After an untrusted Java-APP is installed in mobile station 16 according to the above-mentioned procedures for installation, the untrusted Java-APP is started, which results in an untrusted Java-AP being realized

Nintendo Exhibit 1003
Part 2 of 2
Page 001104

in mobile station 16. Then, the following behavior management is conducted by mobile station 16.

[0118] When an API which is requested to be used by the untrusted Java-AP is an untrusted API, mobile station 16 permits the untrusted Java-AP to use the untrusted API, since, as explained above, untrusted API is permitted to be used by any sort of Java-AP. In short, the untrusted Java-AP is allowed to call the untrusted API.

[0119] When an API which is requested to be used by the untrusted Java-AP is a trusted API, mobile station 16 checks whether a SDF corresponding to the Java-AP is stored in nonvolatile memory 16D. Since no SDF corresponding to the untrusted Java-AP is stored in nonvolatile memory 16D, mobile station 16 prohibits the untrusted Java-AP to use the trusted API. In short, the untrusted Java-AP is not allowed to call the trusted API.

(2-4-2) Trusted Java-AP

[0120] After a trusted Java-APP is installed in mobile station 16, the trusted Java-APP is started, and a trusted Java-AP is realized in mobile station 16, the following behavior management is conducted by mobile station 16.

[0121] When an API which is requested to be used by the trusted Java-AP is an untrusted API, as a matter of course, mobile station 16 permits the trusted Java-AP to use the untrusted API as explained above. In short, the trusted Java-AP is allowed to call the untrusted API.

[0122] When an API which is requested to be used by the trusted Java-AP is a trusted API, since a SDF corresponding to the trusted Java-AP is stored in nonvolatile memory 16D, mobile station 16 may permit the trusted Java-AP to use the trusted API. However, in the case, behavior of the trusted Java-AP is restricted according to permission data contained in the SDF.

[0123] For example, if parameter 'GetPrivateInfo' in the permission data is set as 'Yes', the trusted Java-AP is allowed to read the telephone directory data and the data containing unread emails stored in nonvolatile memory 16D. At the same time, the trusted Java-AP is allowed to communicate with only external devices whose URLs are contained in parameter 'AllowedHost' in the permission data.

[0124] As explained above, mobile station 16 permits behavior of the trusted Java-AP which meet conditions indicated in the permission data contained in the SDF corresponding to the trusted Java-AP, and prohibits behavior which does not meet the conditions. Accordingly, the user can use not only Java-APs in the known system, namely untrusted Java-APs, but also trusted Java-APs, which are allowed to behave more flexibly than untrusted Java-APs, thus enjoying a far greater degree of convenience.

[0125] In the communication system according to this embodiment, mobile station 16 receives an ADF, a SDF and a JAR file separately in this order, which also bring some effects as follows.

[0126] As explained above, Java-AP software, each of which comprises an ADF and a JAR file, are usually designed and made by CPs, and each CP offers, in their own site (CP server device 13 in Fig. 1) on the Internet, to provide their Java-AP software to any user. Accordingly, a user usually decides to download or not to download Java-AP software after accessing sites of CPs and reading introductory explanations of their Java-AP software shown on pages of the sites. When the user decides to download certain Java-AP software, the user needs to operate his/her mobile station for instructing it to carry out download procedures. For the purpose, the above-mentioned pages of the sites usually contain URLs of the files to be downloaded along with anchor tags. From a viewpoint of CPs, it is the easiest way to embed URLs of ADFs in their pages since ADFs are managed by CPs and URLs of ADFs are well known by CPs. On the other hand, for embedding URLs of SDFs in pages of CPs, CPs have to obtain the URLs from the carrier and keep updating the URLs. However, if an ADF, a SDF and a JAR file are delivered to a mobile station separately in this order, there is no need for a CP to obtain, embed and update a URL of a SDF, which is quite convenient.

[0127] Moreover, the above-mentioned method of delivering three sorts of files in a certain order is suitable when procedures for upgrading Java-AP software in the known communication system are taken into account. In the known communication system currently used, when a user requests upgrading Java-AP software, his/her mobile station first checks data contained in an ADF corresponding to the Java-AP software, and tries to obtain an upgraded version of the JAR file using a package URL contained in the ADF. Namely, a mobile station first checks an ADF, then downloads a target JAR file.

[0128] Compared to the procedures for upgrading Java-AP software currently adopted as explained above, the procedures for upgrading Java-AP software adopted in the communication system according to the present embodiment have a similar flow. Namely, in the communication system according to the present embodiment, a mobile station first checks an ADF in the same way as in the known system, and after obtaining a SDF-URL contained in the ADF, the mobile station obtains a JAR file using the SDF-URL in a similar way to that of the known system. Accordingly, it is possible to introduce a communication system according to the present embodiment without making significant changes to the system which is currently used. Compared to the method proposed above, if a SDF, an ADF and a JAR file are delivered to a mobile station in this order when Java-AP software is upgraded, once the mobile station obtains the ADF, the mobile station may obtain an upgraded version of the JAR file the next time by referring to the ADF obtained before, namely without referring to the most recent SDF. Since the SDF may be modified when the upgraded version of the JAR file is downloaded into the mobile station, the upgraded ver-

Nintendo Exhibit 1003
Part 2 of 2
Page 001105

sion of the JAR file without referring to the most recent SDF may cause security problems. Therefore, an ADF, a SDF and a JAR file should be delivered to a mobile station in this order. ,

(3) Modification

[0129] The present invention is not limited to the embodiment as explained above, and it may be modified within the technical scope of the present invention. Following are examples of such modifications.

[0130] In the above-mentioned communication system, behavior of Java-APs is managed with regard to APIs which are used by Java-APs and external devices with which Java-APs communicate, but any other sort of conditions on resources may be set for managing the behavior. 'Resources' may be software resources such as APIs, communication network resources such as communication device with which the Java-AP communicates with, and hardware resources.

[0131] 'Hardware' may be hardware which a mobile station comprises such as a memory, a speaker, a microphone, an infrared radiation controller and Light Emitting Diode (LED), and 'hardware' may also be external devices which operates cooperatively with the mobile station such as User Identity Module (UIM) and Subscriber Identity Module (SIM).

[0132] As a matter of course, 'communication network resources' should not be limited to resources in the communication network such as communication devices with which the mobile station communicates. As mentioned above, the mobile station communicates with mobile communication network wirelessly. During the wireless communications, the mobile station uses wireless communication resources such as wireless channels assigned by the mobile communication network. The wireless communication resources may also be used as the above-mentioned 'communication network resources'. At the same time, the mobile station may use communication resources such as transmission paths for data packet communications and communication paths of a line switching system for voice communications on communication protocol layers which are located higher than the communication protocol layer on which wireless communication resources are used. The communication resources on the higher communication protocol layers may also be used as the above-mentioned 'communication network resources'.

[0133] 'Software resources' may be APIs, classes, packages, and so on, as explained above. Software resources may provide various kinds of functions, and data processing functions such as encrypting data and data communicating functions such as transmitting and receiving data with other applications such as a Web browser are typical functions. In the present invention, software resources comprised by the above-mentioned external devices may also be used as parameters of conditions for restricting behavior of Java-APs.

[0134] In the above-mentioned embodiment, the JAM prohibits certain types of behavior of trusted Java-APs before they behave according to permission data. However, the JAM may manage the behavior of the Java-APs by compulsorily terminating processes of the trusted Java-APs when the JAM detects that the trusted Java-APs attempt to behave in a manner which does not meet conditions indicated in the permission data.

[0135] In the above-mentioned communication system, software is delivered to mobile stations, but software may be delivered to other types of communication devices.

[0136] In the above-mentioned communication system, the carrier may function as a CP. Namely, a management server device and a CP server device may be located in the same communication node.

[0137] The above-mentioned programs run by a CPU of a mobile station, such as JAM program and an OS program, and their related data, may be provided being stored in recording media from which the CPU is able to read the programs and the data, such as magnetic recording media, optical recording media, and ROM. The programs and the data may also be provided to mobile stations through communication networks such as the Internet.

Claims

1. A communication device comprising:

   an executing means for executing software;
   a permission data storing means for storing permission data indicating permissible behavior for an application, which is a group of functions provided by execution of the software;
   a checking means for checking, by accessing an external device before the software is executed, whether the permission data are valid; and
   an execution control means for permitting said executing means to execute the software when, on the basis of a result of the check carried out by said checking means, the permission data are determined to be valid, and for not permitting said executing means to execute the software when, on the basis of the result of the check executed by said checking means, the permission data are determined to be invalid.

2. A communication device according to claim 1, wherein,

   said communication device further comprises a determining means for determining whether it is necessary to carry out the check by said checking means, said determining means either carrying out or not carrying out the check on the basis of a de-

Nintendo Exhibit 1003
Part 2 of 2
Page 001106

termination made by said determination means.

3. A communication device according to claim 2, wherein,
said determining means comprises:

a counting means for counting a number of executions of the software; and
a frequency data storing means for storing frequency data indicating how frequently it is necessary to carry out the check; and wherein,
said determining means determines, on the basis of a number of executions of the software, as counted by said counting means, and on the basis of the frequency data stored in said frequency data storing means, whether it is necessary to carry out a check by said checking means.

4. A communication device according to claim 2, wherein,
said determining means comprises:

a timer means for providing time data indicating a present time; and
a time interval data storing means for storing time interval data indicating a time interval at which it is necessary to carry out the check; and wherein,
said determining means calculates, on the basis of time data provided by said timer, a time period between a present time and a time recorded at a most recent execution of the software, and determines whether it is necessary to carry out the check by said checking means on the basis of the calculated time period and the time interval data stored in said time interval data storing means.

5. A communication device according to claim 1, wherein:

said communication device further comprises count data storing means for storing count data indicating a count of times that the software is allowed to be executed in a condition that said checking means is unable to make the check; and
said execution control means permits said executing means to execute the software in a condition that said checking means is unable to make the check up to a number of times which is indicated by the count data stored in said count data storing means.

6. A communication device according to claim 1 further comprising:

an updating means for obtaining update data from said external device, and updating the permission data stored in said permission data storing means on the basis of the update data.

7. A communication device according to claim 6, wherein:

said updating means
transmits, to said external device, update timing data indicating a timing of a most recent update of the permission data stored in said permission data storing means, when said checking means makes the check;
receives update data transmitted from said external device in response to the transmission of the update timing data; and
updates the permission data stored in said permission data storing means on the basis of the update data.

8. A communication device according to claim 1 further comprising:

a terminating means for instructing said executing means to terminate execution of the software when the application attempts to conduct behavior which the application is not permitted to conduct.

9. A communication device according to claim 1, wherein:

the permission data contain information on usage of at least one of an internal hardware resource of said communication device, an external hardware resource of said communication device, a software resource and a communication network resource.

10. A method for controlling a communication device comprising:

a step for transmitting to said communication device permission data, which indicates permissible behavior for an application, which is a group of functions provided by execution of software in said communication device;
a step for checking, by communicating data between said communication device and an external device, whether the permission data are valid, before the software is executed in said communication device; and
a step for permitting the software to be executed only when the permission data are determined to be valid on the basis of a result of the check.

Nintendo Exhibit 1003
Part 2 of 2
Page 001107

11. A program for instructing a computer to execute:

a process for storing, in a storing means, permission data indicating permissible behavior for an application, which is a group of functions provided by execution of software;
a process for checking, by accessing an external device, whether the permission data are valid, before the software is executed; and
a process for permitting the software to be executed only when the permission data are determined to be valid on the basis of a result of the check.

12. A communication method comprising:

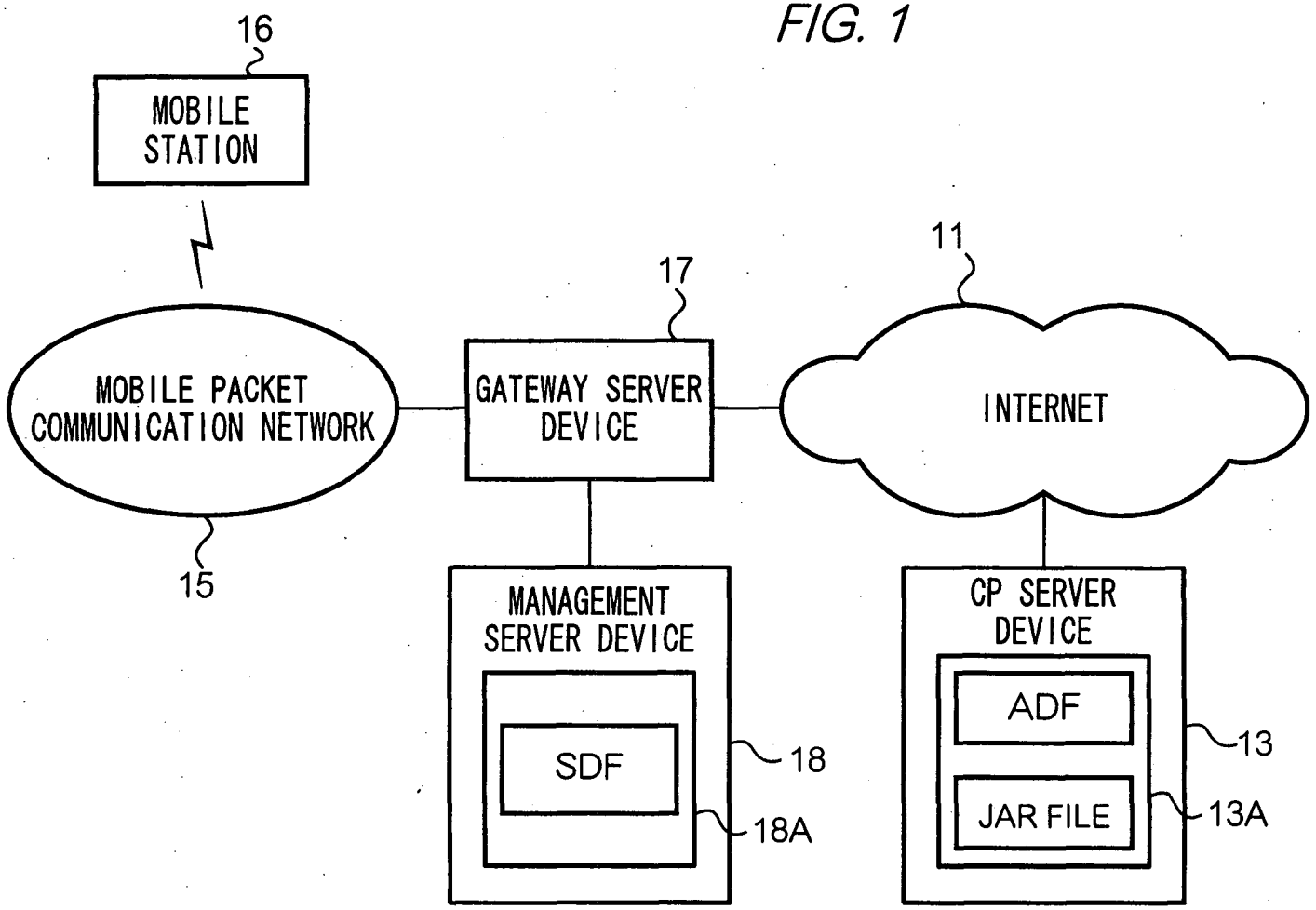
a step for transmitting from a communication system comprising

(a) a software data providing server device which stores software data containing software for providing a group of functions forming an application,
(b) a management server device which stores security descriptor data containing permission data indicating permissible behavior for the application, and
(c) an application descriptor data providing server device which stores application descriptor data indicating a storage location of the software data and a storage location of the security descriptor data,

to said communication device the application descriptor data;
a step for transmitting the application descriptor data from said communication system to said communication device;
a step for transmitting data indicating the storage location of the security descriptor data contained in the application descriptor data from said communication device to said communication system;
a step for transmitting the security descriptor data from said communication system to said communication device on the basis of the data indicating the storage location of the security descriptor data;
a step for storing the security descriptor data in said communication device;
a step for transmitting data indicating the storage location of the software data contained in the security descriptor data from said communication device to said communication system;
a step for transmitting the software data from said communication system to said communication device on the basis of the data indicating the storage location of the software data;
a step for installing, in said communication device, the software contained in the software data transmitted from said communication system to said communication device;
a step for checking, by communicating data between said communication device and said communication system before the software is executed in said communication device, whether the security descriptor data stored in said communication device are valid; and
a step for permitting said software to be executed in said communication device only when the security descriptor data are determined to be valid on the basis of a result of the check.

Nintendo Exhibit 1003
Part 2 of 2
Page 001108

FIG. 1



EP 1 465 041 A2

FIG. 2

HTTP/1.0 200 OK
Content-Type:application/x-sdf
•
•
•
<CR><LF>
Sts = 00 <CR><LF>
PackageURL = http://cpserver.com:8080/Demo.JAR <CR><LF>
CheckCnt = 005 <CR><LF>
CheckInt = 020 <CR><LF>
SuspendedCnt = 005 <CR><LF>
Lmd=20020614120552 <CR><LF>
GetPrivateInfo = yes <CR><LF>
UseMailer = yes <CR><LF>
MessageApp = yes <CR><LF>
SetPhoneTheme = yes<CR><LF>
SetLaunchTime = yes <CR><LF>
AllowedHost = http://aaa.co.jp http://bbb.co.jp:8080 <CR><LF>

FIG. 5

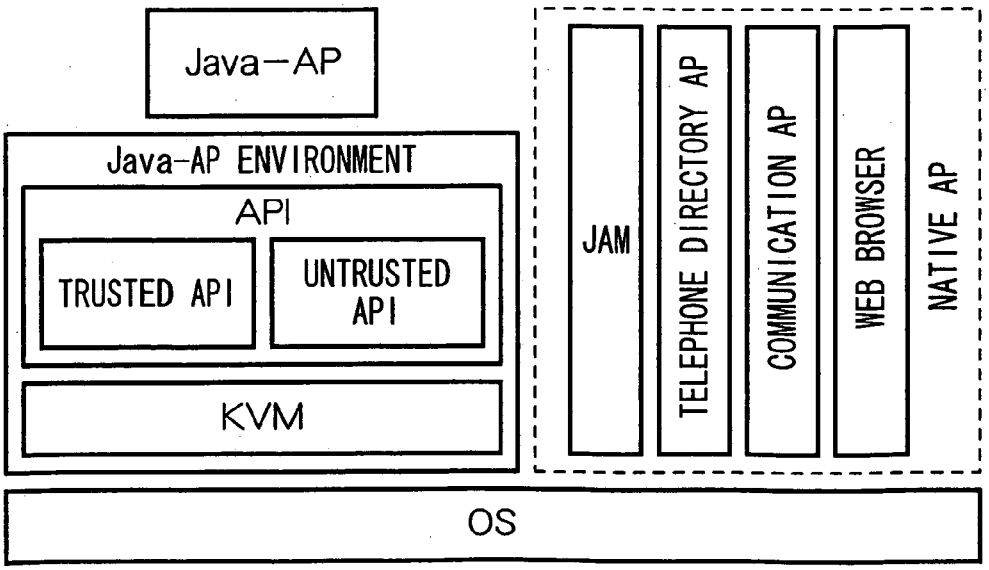
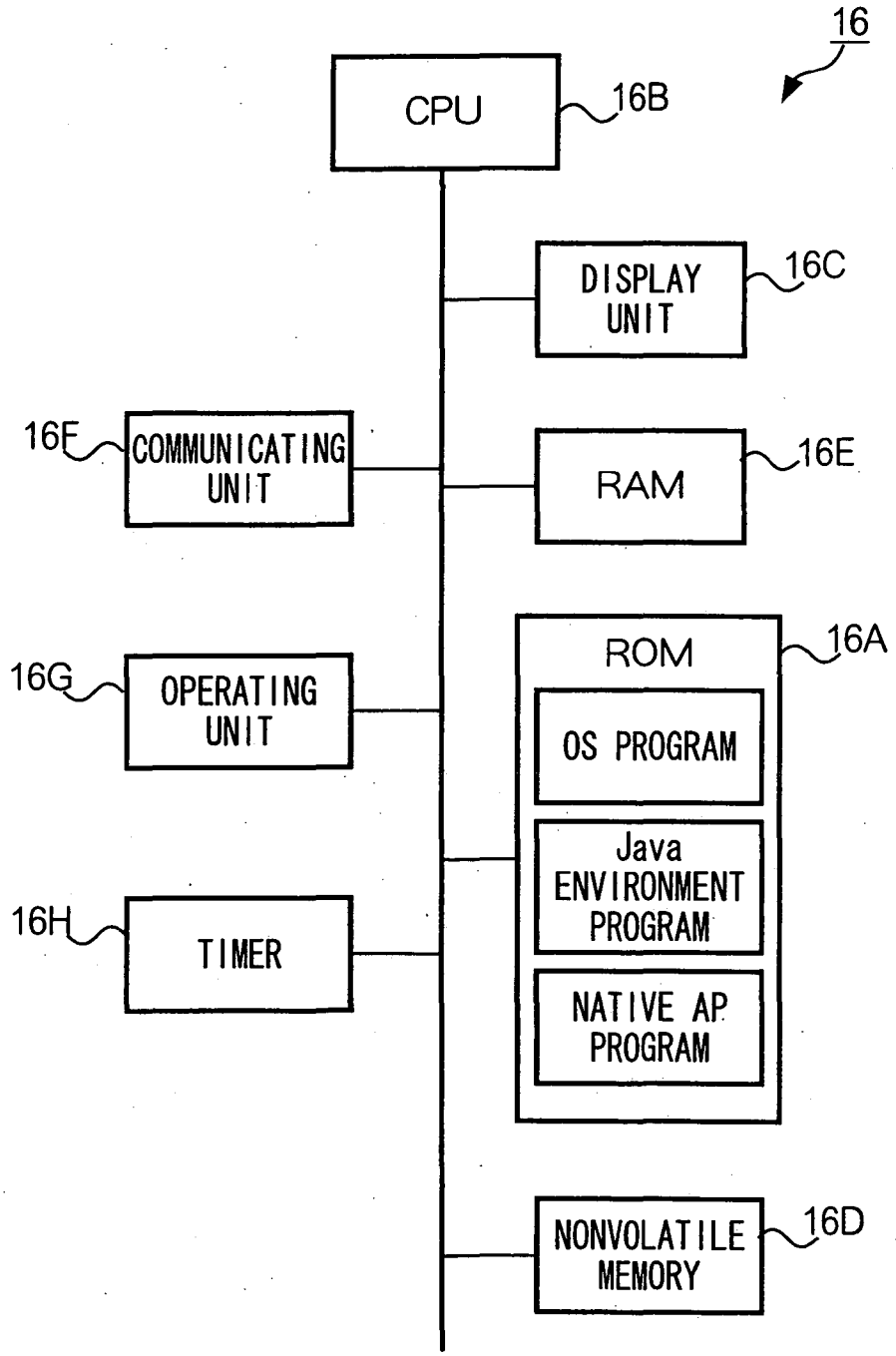


FIG. 3



Nintendo Exhibit 1003
Part 2 of 2
Page 001111

FIG. 4

APID		1000001
ADF-URL		http://CPserver.com:8080/Demo.adf
SDF-URL		http://MNserver.com:8080/Demo.sdf
PACKAGE URL		http://CPserver.com:8080/Demo.JAR
VALIDITY STATUS OF SDF		00(VALID)
COUNT FOR SDF CHECK		5 (TIMES)
INTERVAL FOR SDF CHECK		20 (DAYS)
EXPANDED COUNT FOR SDF CHECK		5 (TIMES)
LAST UPDATE TIME		5 MINUTES AND 52 SECONDS PAST 12 O'CLOCK ON JUNE 14TH IN YEAR 2002
PERMISSION DATA	GetPrivateInfo	YES
	Usermailer	YES
	MessageApp	YES
	SetPhoneTheme	YES
	SetLaunchTime	YES
	AllowedHost	aaa.co.jp:8080
	AllowedHost	bbb.co.jp:8080

NUMBER OF EXECUTION TIMES	2 (TIMES)
NUMBER OF PASSED DAYS	15 (DAYS)
USED EXPANDED COUNT FOR SDF CHECK	1 (TIME)

EP 1 465 041 A2

FIG. 6

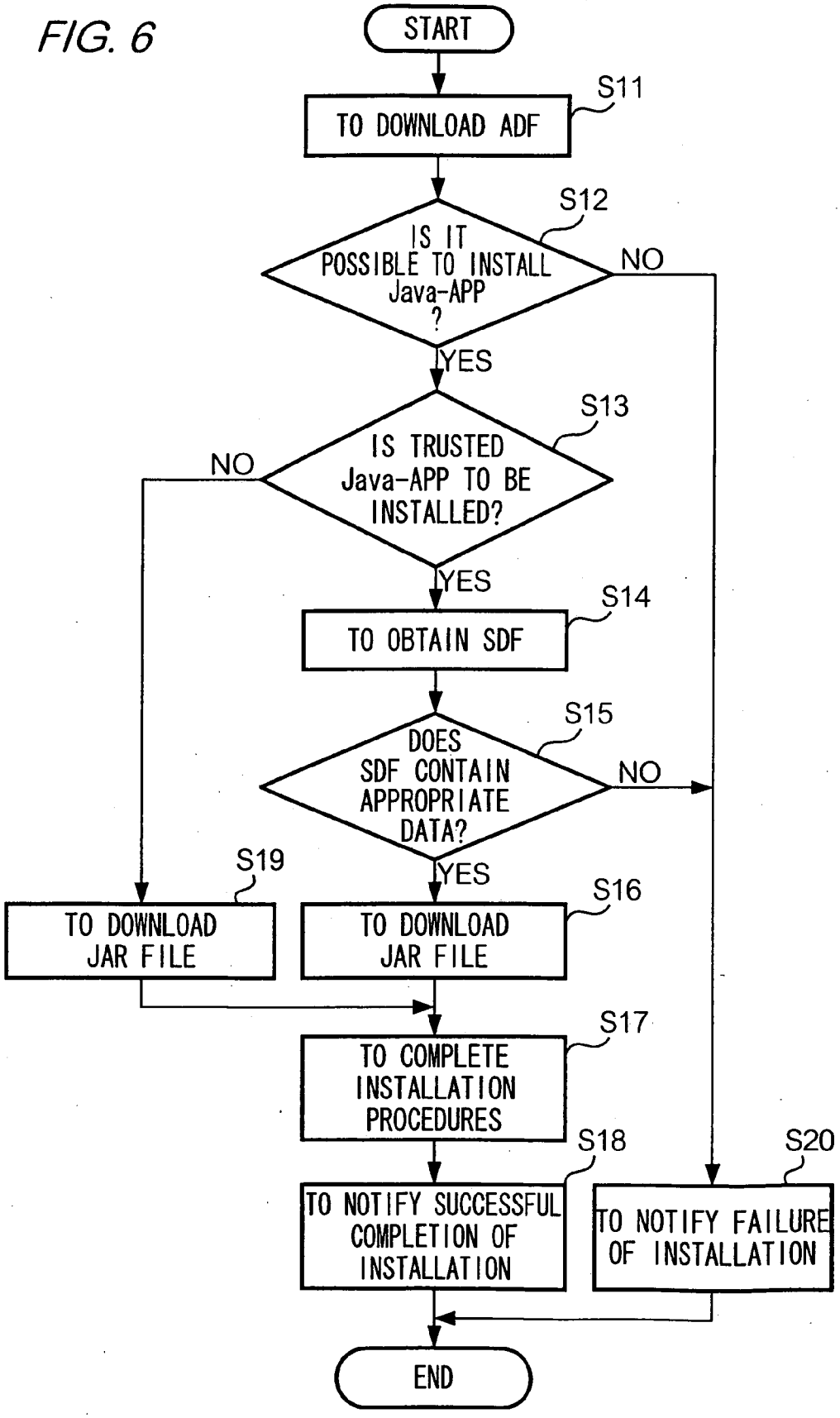

START
S11
TO DOWNLOAD ADF
S12
IS IT POSSIBLE TO INSTALL Java-APP ?
NO
YES
S13
IS TRUSTED Java-APP TO BE INSTALLED?
NO
YES
S14
TO OBTAIN SDF
S15
DOES SDF CONTAIN APPROPRIATE DATA?
NO
YES
S19
TO DOWNLOAD JAR FILE
S16
TO DOWNLOAD JAR FILE
S17
TO COMPLETE INSTALLATION PROCEDURES
S18
TO NOTIFY SUCCESSFUL COMPLETION OF INSTALLATION
S20
TO NOTIFY FAILURE OF INSTALLATION
END

FIG. 7

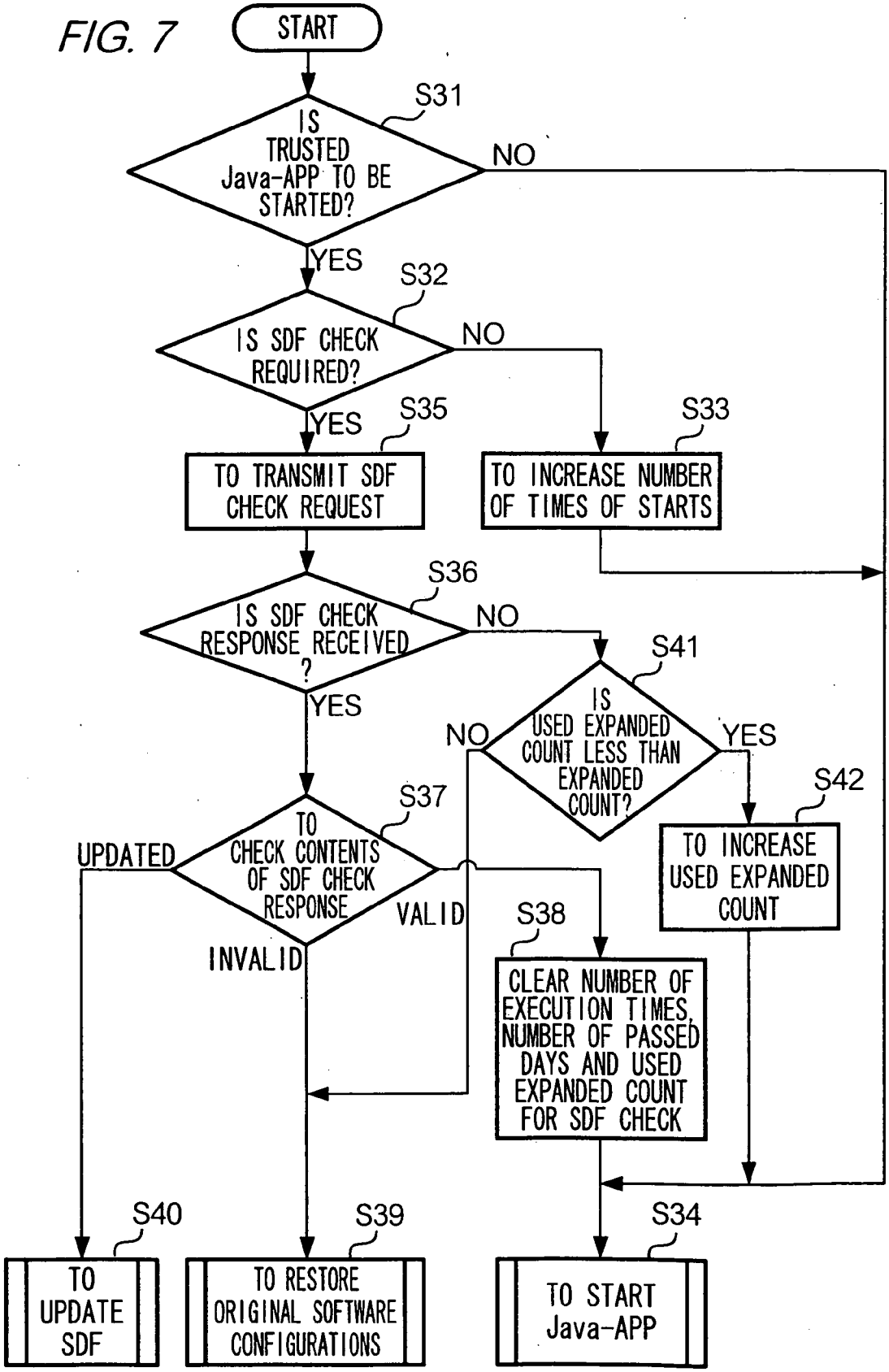

START
S31
IS TRUSTED Java-APP TO BE STARTED?
NO
YES
S32
IS SDF CHECK REQUIRED?
NO
YES
S35
TO TRANSMIT SDF CHECK REQUEST
S33
TO INCREASE NUMBER OF TIMES OF STARTS
S36
IS SDF CHECK RESPONSE RECEIVED ?
NO
YES
S41
IS USED EXPANDED COUNT LESS THAN EXPANDED COUNT?
NO
YES
S42
TO INCREASE USED EXPANDED COUNT
S37
TO CHECK CONTENTS OF SDF CHECK RESPONSE
UPDATED
VALID
INVALID
S38
CLEAR NUMBER OF EXECUTION TIMES, NUMBER OF PASSED DAYS AND USED EXPANDED COUNT FOR SDF CHECK
S40
TO UPDATE SDF
S39
TO RESTORE ORIGINAL SOFTWARE CONFIGURATIONS
S34
TO START Java-APP

FIG. 8

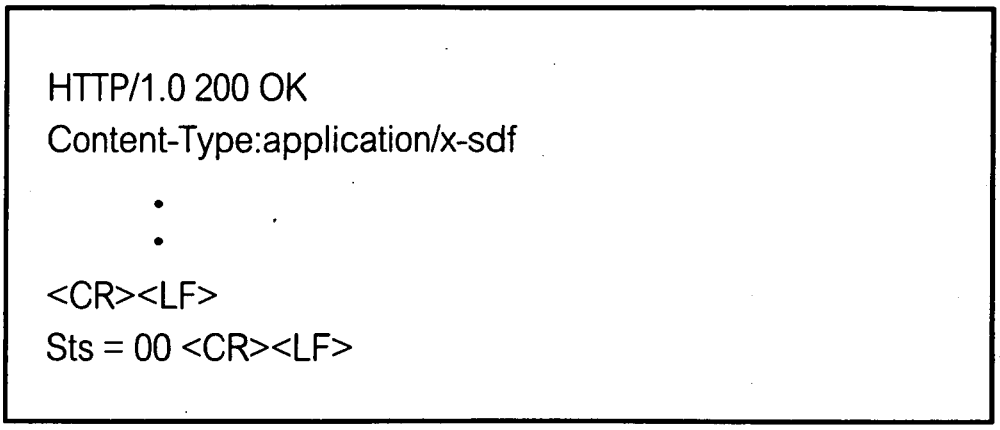

HTTP/1.0 200 OK
Content-Type:application/x-sdf
<CR><LF>
Sts = 00 <CR><LF>

FIG. 9

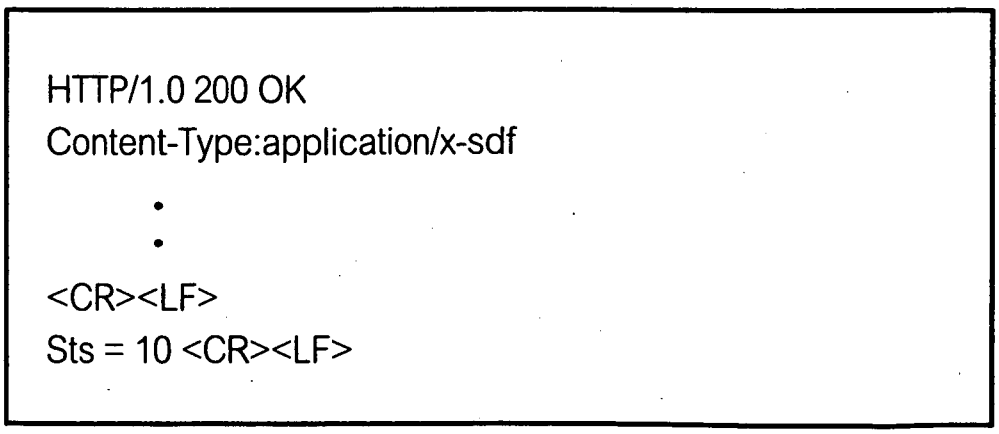
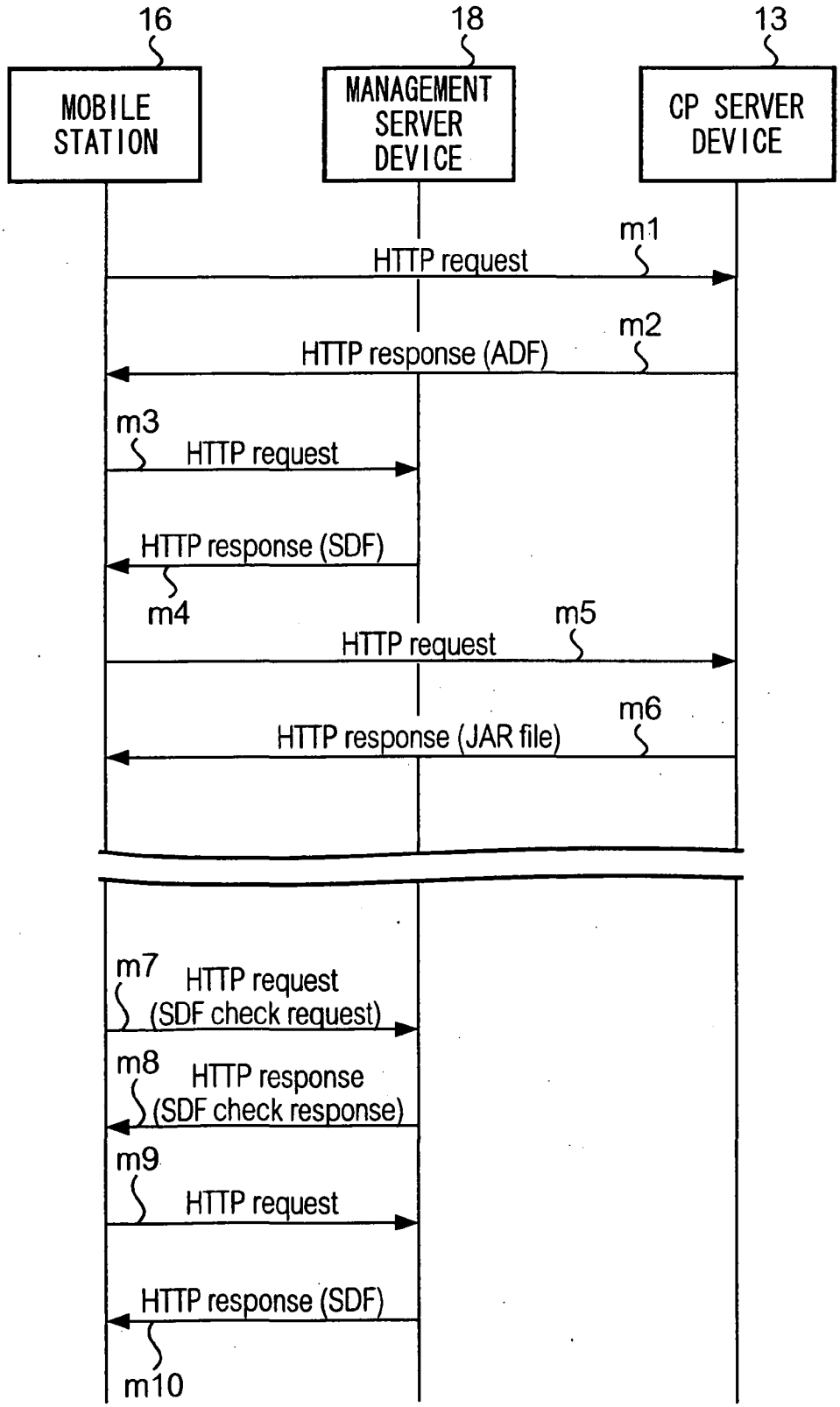

HTTP/1.0 200 OK
Content-Type:application/x-sdf
<CR><LF>
Sts = 10 <CR><LF>

FIG. 10


16
18
13
MOBILE STATION
MANAGEMENT SERVER DEVICE
CP SERVER DEVICE
m1
HTTP request
m2
HTTP response (ADF)
m3
HTTP request
HTTP response (SDF)
m4
m5
HTTP request
m6
HTTP response (JAR file)
m7
HTTP request
(SDF check request)
m8
HTTP response
(SDF check response)
m9
HTTP request
HTTP response (SDF)
m10

(12) UK Patent Application (19) GB (11) 2 312 767 (13) A

(43) Date of A Publication 05.11.1997

(21) Application No 9708608.6

(22) Date of Filing 28.04.1997

(30) Priority Data
(31) 08638807 (32) 29.04.1996 (33) US

(51) INT CL[6]
G06F 1/00

(52) UK CL (Edition O)
G4A AAP

(56) Documents Cited
EP 0570123 A1
Dialog record 01887837 of Computer Shopper, v16, n2, Feb 199 6, p 581(2) Dialog record 01939746 of PC Magazine, v15, n11, 11 June 199 6, p 221(5)

(58) Field of Search
UK CL (Edition O) G4A AAP
INT CL[6] G06F 1/00
Online: COMPUTER, INSPEC, WPI

(71) Applicant(s)
Mitel Corporation

(Incorporated in Canada - Ontario)

P O Box 13089, Kanata, Ontario K2K 1X3, Canada

(72) Inventor(s)
Richard Deadman

(74) Agent and/or Address for Service
John Orchard & Co
Staple Inn Buildings North, High Holborn, LONDON, WC1V 7PZ, United Kingdom

(54) **Protected persistent storage access by applets**

(57) A method of processing an applet comprises storing a file in a persistent storage medium (PSM), the file including an access control list 20; transmitting an applet from a server, the applet including at least one of applet identification data and a private key encrypted domain identifier and public key pair, a maximum file size indicator, and a specification of required operations; receiving the applet by an application engine 16; in the event the domain identifier is included in the applet, decrypting the domain identifier using the public key; checking at least one of the applet identification data and decrypted domain identifier against the access control list for a match, and in the event a match is found, allowing the operations specified in the applet on a file stored in a persistent storage medium and as specified in the file access control list.

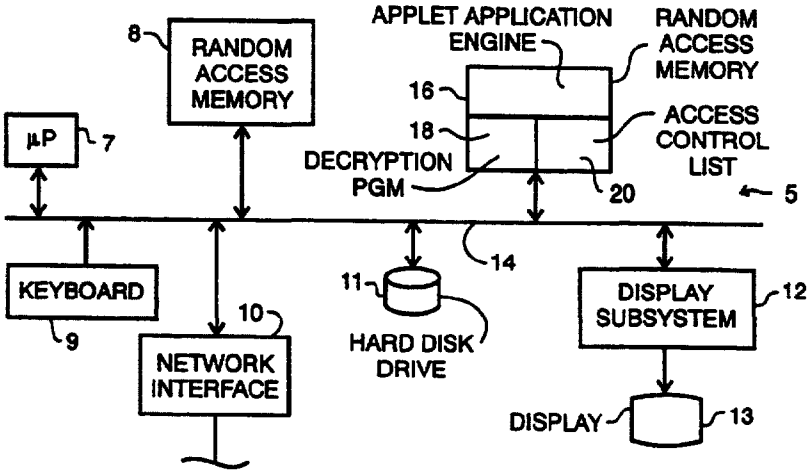


FIG. 2

GB 2 312 767 A

Nintendo Exhibit 1003
Part 2 of 2
Page 001117

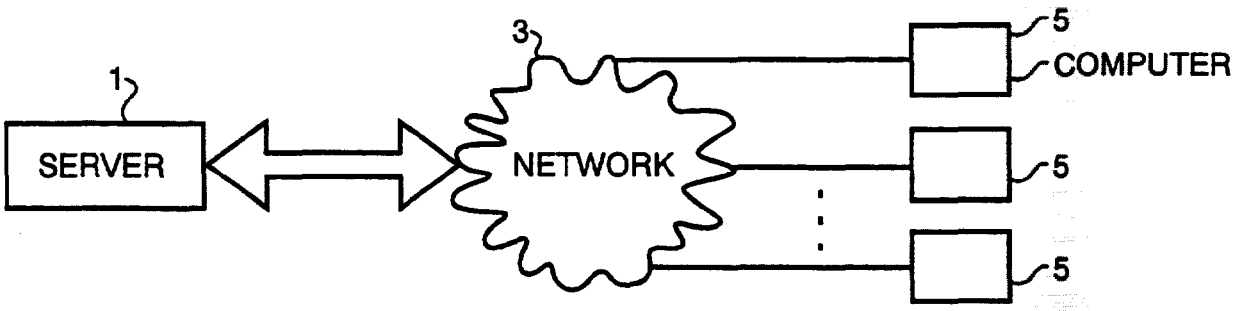


FIG. 1

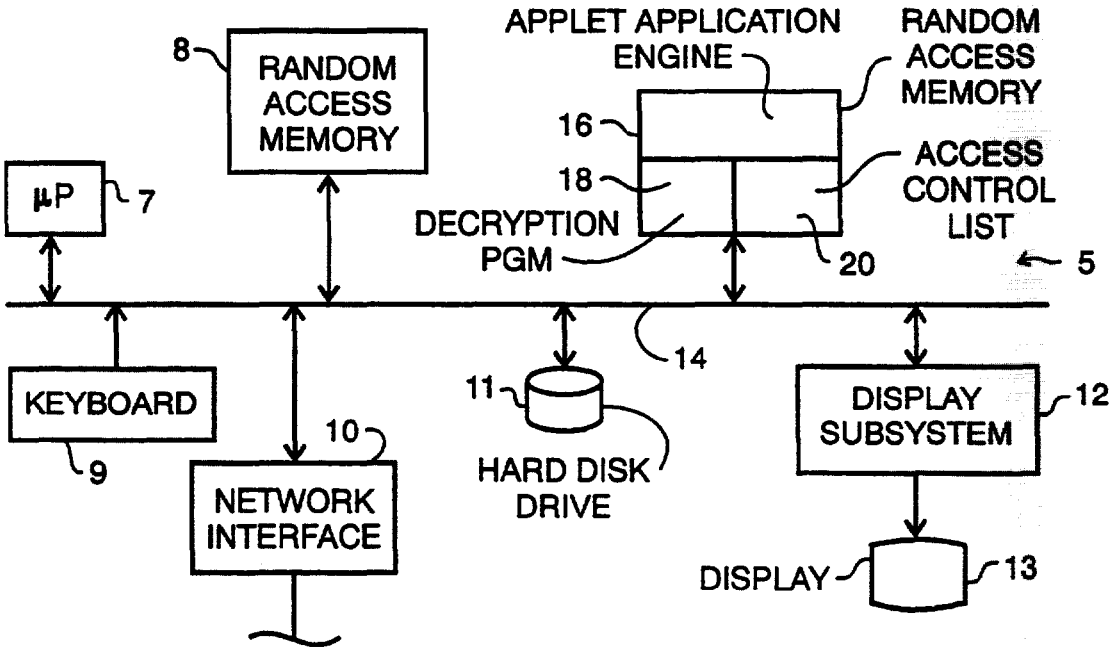


FIG. 2

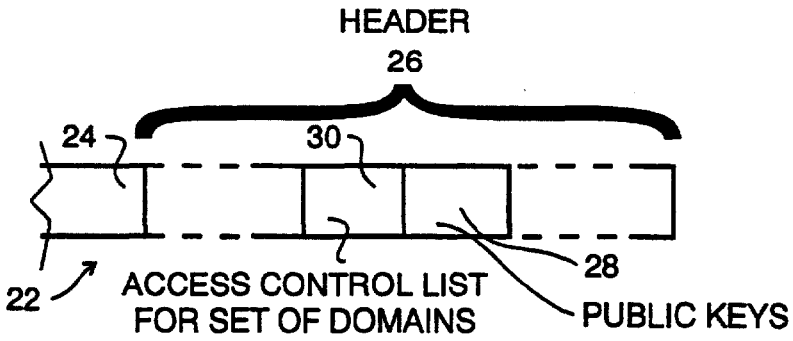


FIG. 3

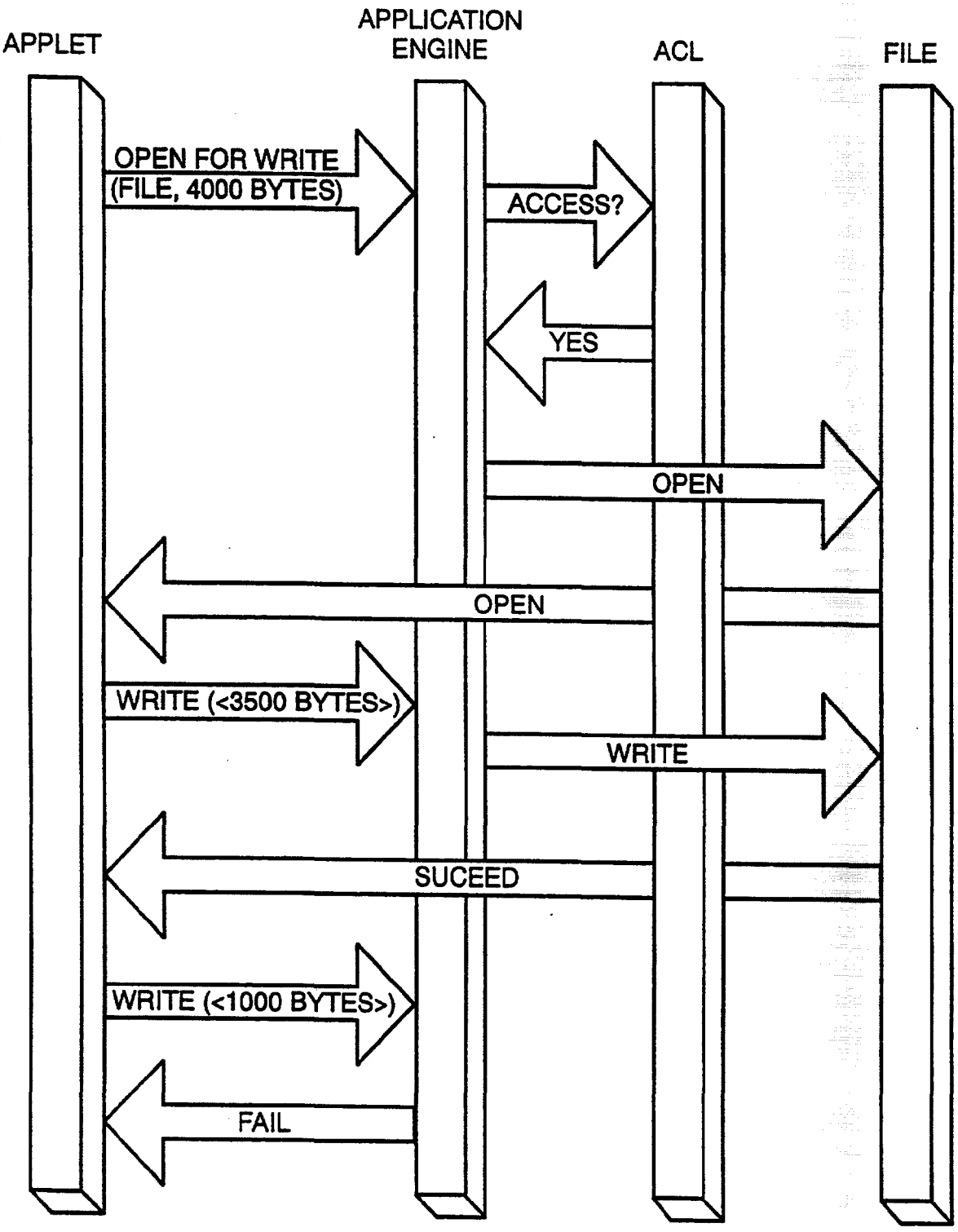

APPLET
APPLICATION ENGINE
ACL
FILE
OPEN FOR WRITE (FILE, 4000 BYTES)
ACCESS?
YES
OPEN
OPEN
WRITE (<3500 BYTES>)
WRITE
SUCEED
WRITE (<1000 BYTES>)
FAIL

FIG. 4

PROTECTED PERSISTENT STORAGE ACCESS FOR MOBILE APPLICATIONS

This invention relates to the field of computer networks and in particular to a method of processing computer applets in a secure manner.

With the advent of object based programming, it has become possible to transmit applets via a network such as the internet to various computers connected to the network. An applet is a small self-contained application program which can cause a computer to perform a particular function, and contains both instructions for the computer to operate and data which is to be processed in its operation.

Various programming languages have been designed to create applets, such as Telescript, SafeTCL, Java, CyberAgents and ClearLake Agents. By the use of interpreters, the various applets can be hardware independent, that is, the same applet can be received and processed by computer hardware having different operating systems. For example, the same applet would be able to be received and processed by a MacIntosh operating system and by one of the Microsoft Windows operating systems on an Intel processor based (e.g. IBM) personal computer type operating system.

It has been speculated that the programs and data that computers that receive applets use would provide all that a computer would need, to be able to be fully functional and which would result in a significantly less expensive computer than at present. This is because the computer would be expected to use the network as a mass storage repository of programs and data. Programs would be paid for on a use basis, rather than on an up-front-payment and unlimited use basis. However this presupposes that a user would be prepared to allow mass storage to be out of his control to store

critical data, specially written programs, etc. For the reason of security, it had been believed to be desirable and/or necessary to provide a local mass storage associated with each or with most computers.

With the storage of data on a local mass storage device, the problem of security arises in relation to the reception from the network of applets, since those applets could access and transmit or corrupt files.

The security problem has been handled in one of two ways.

The first is to restrict the use of applets to a closed or managed system, wherein software components can be given domain names and all possible execution sites can be populated with lists of allowed domains. For example, in the CyberAgents program, domains are used to control execution and security on remote sites. A domain is a jurisdiction group of applets that share access rights.

The second way of handling security is to isolate the applets from important system resources, as noted earlier above. Applets received from an uncontrolled system network such as the internet have been restricted from being able to access mass storage devices, and have been relegated to providing fancy graphics and to providing a graphical user interface for client-server computing. Programs to create applets in open uncontrolled systems, are e.g. Telescript, SafeTCL and Java. For example, Java can cause a computer to display a window and provide the content of the window and play sounds, but cannot write to the computer disk drive.

While use of either of these schemes produces the required security to ensure that rogue applets do not cause damage to a computer's resources, such as the

files, they also place restriction on the usability of open system mobile applets. If a downloaded applet cannot access the disk drive to provide persistent local storage for the user, the usability of applets written using such an isolation scheme is very limited.

Operating systems approach security by providing usage rights over directories and files. By assigning the applet application engine to a user group, the applets executed by the application engine are restricted to accessing those files available to the application engine itself. While this provides security to other file system files outside the reach of the application engine, it does not provide security between applets. Similarly, some applet application engines, such as Sun's HotJava, provide access control lists. These lists specify which subset of the application engines accessible files can be accessed by applets. This provides extended security, but does not solve the problem of providing security to data between applets.

HotJava also specifies a method by which access to files can be granted to applets through the use of a dialog box to the user of the applet, wherein the user is required to enter a security authorization. This provides greater security between applets but is intrusive and requires that the user understand the sensitivity and origin (virtual ownership) of data in each file.

If the application engine can determine the source of the applet, access rights can be set according to certain rules. Java can detect if an applet originates within a firewall (a software barrier to outside access to a local (internal) network); the HotJava Browser allows different security rights to applets loaded on each side of the firewall. Read and

write access can be granted to internal applications only.

General Magic's Telescript technology provides a generic set of authorities and permits. Agents within a region all contain the same authority - this is similar to the concept of user domains. Each Telescript agent has only one authority. Permits control the execution of instructions or the access and usage of a resource. Telescript does not specify how a Telescript place negotiates usage rights for a resource with an agent of a certain authority, or stores the access rights for each resource instance.

The term "persistent storage" will be used in this specification as a generic term for all media which store or carry files, data or programs which are to be protected, and can include, but is not restricted to electronic random access memory, floppy or hard disk drives, buses, etc.

Features of an arrangement to be described, by way of example in illustration of the present invention is that it provides comparatively secure access to persistent storage for distributed applet code within an open uncontrolled computing environment. It protects files from unauthorized use by unknown mobile applets while still providing access to the file system for these applets. This involves negotiating access rights for the unknown applet.

In a particular embodiment illustrative of the invention, a method of processing an applet includes storing a file in a persistent storage medium (PSM), the file including an access control list, and transmitting an applet from a server, the applet including at least one of applet identification data and a private key encrypted domain identifier and public key pair, a maximum file size indicator, and a specification of required storage operations, receiving the applet by

an application engine, and in the event the domain identifier is included in the applet, decrypting the domain identifier using the public key, checking at least one of the applet identification data and decrypted domain identifier against the access control list for a match, and in the event a match is found, allowing the operations specified in the applet on a file stored in a persistent storage medium which allow access to the applet as specified in the file access control list.

In another illustrative embodiment, a method of providing an applet includes receiving applets in an application engine, the applets comprising specifications of operations to be performed, and some of the applets being comprised of at least one of applet identification data and private key encrypted domains, processing the applets to an extent to which access to files contained in a persistent storage medium is not required, in the event applets are comprised of private key encrypted domains, decrypting the domains, checking at least one of the applet identification data and decrypted domains against a control list, and processing the applets which the at least one of applet identification data and decrypted domains match entries in the control list to access the files contained in persistent storage medium required by the applets.

In yet another illustrative embodiment, a method of processing an applet includes storing an access control list which contains an identity of applets or domains that can have access to files stored in a persistent medium, and giving access to the access control medium by applets which contain an identity or a domain corresponding to an entry on the list.

In a further illustrative embodiment, an applet application engine includes an access

control list, the engine and list stored in a memory of a computer, and apparatus for providing an applet comprising an identification of at least one of an applet identifier and a domain for comparison with the list and determination of a match, and apparatus for allowing access to an otherwise protected computer resource in the event of the match.

Arrangements which are given by way of example, and which are helpful in understanding the invention, will now be described with reference to the accompanying drawings, in which:

Figure 1 is a block schematic diagram of a network on which the present invention can be implemented,

Figure 2 is a block schematic diagram of a computer used in the network of Figure 1, on which the present invention can be implemented,

Figure 3 is an illustration of a representative applet which may be used in carrying out the present invention, and

Figure 4 illustrates a representative method of operation illustrative of the invention.

Figure 1 illustrates an open and uncontrolled network, and is comprised of a server 1, which interfaces a network 3, to which various computers 5 are or can be connected. The network 3 can be a wide area network, the internet, a telephone network, etc.

In operation, the server 1 applies applets to the network 3 for reception by one or more computers 5. A computer which receives the applet, which is a complete object based program, processes it using the data contained therein. As noted above, in prior art open systems the applet was restricted from being able to access persistent storage.

Figure 2 illustrates the basic architecture of a representative computer system 5. A microprocessor 7, random access memory 8 (RAM), a keyboard 9, a network interface device 10 such as a modem, a hard disk drive 11 persistent storage device, and a display subsystem 12 to which a display 13 is connected, are connected to, or are in communication with, a bus 14.

In addition, random access memory 16 (RAM), which may be part of or merged with RAM 8, is connected or in communication with bus 14. RAM 16 contains a applet application engine, which is comprised of an interpreter for the particular applet language used, and a software command structure to cause microprocessor 7 to process the applets. For example, for the Java language, an applet application engine can be HotJava, available from Sun Microsystems Inc..

In operation of a prior art system, applets are received from the server 1 via the network 3 and network interface 10, are interpreted by microprocessor 7 using the interpreter in application engine 16 and the interpreted instructions are processed by microprocessor 7 to cause the display subsystem 12 to display graphics, etc. on display 13. The application engine will not access persistent storage media such as the hard disk drive 11, to avoid corruption etc. of files as described therein, and therefore cannot use data or other programs stored therein. Such a system is thus considerably limited.

In an arrangement illustrative of the present invention, the application engine 16 contains a public key decryption program 18 and an access control list 20 of authorized applets and/or domains which may be allowed to access persistent storage media.

In the illustrative arrangement, when applets are received via the network interface, access

rights to the persistent storage media are negotiated using the application engine 16. Access rights need not be restricted and access negotiated only to persistent storage media; any resource of the computer or controlled or accessed by the computer can be restricted and access thereto negotiated.

Access rights are negotiated by

(a) authentication; the applet application engine verifies the identity or domain of all applets requesting access to persistent storage;

(b) usage rights negotiation; this involves verifying the applet's authority to read, write or append to a file. Methods to ensure that an applet has access rights to a particular file are described below. It should be noted that in an open uncontrolled system, central authentication or domain management is not possible. For this reason, the negotiation of access rights is conducted between the application engine, the file and the applet;

(c) quality of service; maximum disk usage for an application is set. Limits are preferably placed on access rights which give the bounds within which the granted resource may be used.

By acting as an intermediary between the unknown applet and the disk resource, the application engine negotiates and enforces access rights. For disk files, this involves acting as an intermediary for all disk actions. If a target file will grow beyond a predetermined size, a write can be denied or the user can be asked for override permission, which effectively allows growth of the access rights by an incremented amount. The application engine can specify an override zone, above which the access right will not be allowed to grow.

As a safety mechanism, file versioning may be provided by the application engine to ensure that any modifications or deletions to files can be recovered in the event of damage to the file system. Access control list 20 is used by the application engine to specify what default and file specific access rights are for individual persistent storage objects (files).

Each applet should be associated with a server site that uniquely identifies the applet. If loaded from the World Wide Web, for example, this identity can be the universal resource locator (URL) of the applet. When an applet creates a new file, it can specify which applets have which rights on the file.

Domains are groups of applets which share access rights. Using private key encryption and an encrypted message supplied by the applet, the application engine can ensure that an applet belongs to a specified domain. Applet files contain a header which contains the access control list for a set of domains and public keys for each domain.

Figure 3 illustrates an applet 22, which contains a payload 24 comprised of an object based application program including data, and a header 26. The header 26 contains a public key or keys 28, and an application control list 30. It also contains an identification of the applet (e.g. the URL if used on the World Wide Web), which can be encrypted at the server using a private key, for decryption under control of the application engine 16 using a public key 28. The header also contains an encrypted set of domains e.g. domain messages, which can be decrypted using a public key 28.

An applet can belong to more than one domain at a time. It is the responsibility of the domain

owners to ensure that the private key used to generate encrypted domain authentication messages is kept secure.

As noted above, the encrypted message may encrypt the applet identification using the private key. This ensures that the encrypted message has not been copied from another applet in an attempt to borrow that applet's domain capabilities. Any attempt to clone the server applet to another machine and then be able to read domain owned files would then cause the encrypted message to not match the applet identification, invalidating the applet's domain within the application engine.

A combination of access methods is preferably used to provide secure persistent storage access to applets. Both applet identification and preferably domains are implemented as keys within the access control list for a file.

The language classes for file access in the application engine 16 for the particular language used, e.g. Java should be amended to allow access to the file system based on the applet's identification and domain. Applets should be modified as indicated above with respect to Figure 3 to optionally include their domain, and by the use of public key encryption thereof and/or their identification. Each file with access rights associated with it is contained in the access control list 20 (Figure 2) in a file stored in RAM 16 accessible by the application engine, e.g. a browser program. The application engine retained in RAM 16 negotiates access rights for each loaded applet. Access rights contain quality of service information that defines how each resource can be accessed, including maximum size changes. Each change to the file system is checked to see if it would violate access rights. This includes

the right to access, change or create a file and limits on the size of changes.

The structure of the access control list is preferred to be:

<Filename:...>
<<AccessMode,...>:<IDorDomain,...>;...>

where:

Filename	Contains one or more file names, each of which may contain wild-cards
AccessMode	Is one of Read, Write, Append, Delete, NewFile, List, Query, (MMaxSize:x)
IDorDomain	Is either a URL specifying an applet, or a (DomainName:publicKey) pair

For example, an entry where all files in the "/public" directory are readable, "/private" can have new files created by applet, URL "http://www.foo.bar/applet.class" and applets of domain "MyDomain" have write access to the file "/private/foo.bar" with add mode set to 5000 bytes, could appear as:

/public/*
Read : *
/private/*
NewFile : *
/private/foo.bar
Write, (MaxSize:5000) :
http://www.foo.bar/applet.class; (MyDomain: 987654321)

A message send within the application engine would appear as shown in the flow chart of Figure 4, in which time runs from top to bottom of the figure.

In this example the applet requests to open the file to write 4000 bytes. The access control list (ACL) is queried by the application engine. The request

is approved, and the file is opened. When 3500 bytes are added, the write to the file succeeds. An attempt to write a further 1000 bytes does not succeed and they are not written, since they exceed the application engine's stored access rights (4000 bytes) for the applet over the file.

Thus it may be seen that there has been described a method by which applets can access files stored on persistent storage media, or can access other computer system resources in an open system that would otherwise be protected, in a secure manner, without concern that a rogue applet has adopted the domain of another applet, or concern that a file can be stolen or corrupted. The method also provides protection against overrunning the storage media due to excessive file size production. Yet at the same time it allows more sophisticated functioning of a computer connected to an open system, since files and other resources at, controlled by or connected to the local computer can be used in processing of applets.

A person understanding this description may now conceive of alternative structures and embodiments or variations of the above. All those which fall within the scope of the claims appended hereto are considered to be part of the present invention.

CLAIMS

1. A method of processing an applet including

(a) storing a file in a persistent storage medium (PSM), the file including an access control list,

(b) transmitting an applet from a server, the applet including at least one of applet identification data and a private key encrypted domain identifier and public key pair, a maximum file size indicator, and a specification of required operations,

(c) receiving said applet by an application engine,

(d) in the event the domain identifier is included in the applet, decrypting the domain identifier using the public key,

(e) checking said at least one of said applet identification data and decrypted domain identifier against the access control list for a match, and

(f) in the event a match is found, allowing said operations specified in the applet on a file stored in a persistent storage medium which allow access to the applet as specified in the file access control list.

2. A method as defined in claim 1 including terminating performing of said operation specified in the applet in the event said operation would result in a file size in the persistent storage medium that exceeds said maximum file size indicated in the applet.

3. A method as defined in claim 1 including not performing said operation in the event a match is not found.

4. A method of processing applets including

(a) receiving applets in an application engine, said applets comprising specifications of operations to be performed, and some of said applets being comprised of at least one of applet identification data and private key encrypted domains,

(b) processing said applets to an extent to which access to files contained in a persistent storage medium is not required,

(c) in the event applets are comprised of private key encrypted domains, decrypting said domains,

(d) checking said at least one of said applet identification data and decrypted domains against a control list, and

(e) processing those of said applets of which said at least one of applet identification data and decrypted domains match entries in said control list to access said files contained in persistent storage medium required by said applets.

5. A method of processing an applet including storing an access control list which contains an identity of applets or domains that can have access to files stored in a persistent medium, and giving access to said persistent medium by applets which contain an identity or a domain corresponding to an entry on the list.

6. A method as defined in claim 5 including using an application control engine stored in a computer to inspect said applets, check said access control list and process said applets identified on said access control list to operate on a file stored in said access control medium.

7. A method as defined in claim 6 including checking an applet for a file size limitation specification carried by said applet, and failing to operate on said file in the event operation specified by the applet would result in a file size in excess of said file size.

8. A method as defined in claim 5 in which at least one of said identity and domain carried by the applet is private key encrypted and includes a corresponding public key, and including the steps of decrypting to authenticate said at least one of said identity and domain using the public key, said giving access step being carried out only in the event the decrypted said at least one of said identity and domain is authenticated.

9. An applet application engine including an access control list, said engine and list being stored in a memory of a computer, and means for providing an applet having an identification of at least one of an applet identifier and a domain for comparison with said list and determination of a match, and means for allowing access to an otherwise protected computer resource in the event of said match.

10. An engine as claimed in claim 9, in which the computer resource is a persistent storage medium.

11. An engine as claimed in claim 9 further including a public key decryption means, said applet having a private key encrypted identifier or domain and public key, for decryption by said decryption means.

12. A computer software applet including a private key encrypted authentication code defining a domain of said applet.

13. A method as claimed in any one of claims 1, 4, or 5 substantially as described herein with reference to the accompanying drawings.

14. An applet application engine as claimed in claim 9 substantially as described herein with reference to the accompanying drawings.

15. A computer software applet as claimed in claim 12 substantially as described herein with reference to the accompanying drawings.

The Patent Office

17

Application No: GB 9708608.6 **Examiner:** B G Western
Claims searched: 1-11, 13, 14 **Date of search:** 15 July 1997

Patents Act 1977
Search Report under Section 17

Databases searched:

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

UK Cl (Ed.O): G4A AAP

Int Cl (Ed.6): G06F 1/00

Other: On-line databases: COMPUTER, INSPEC, WPI

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
X	EP-0570123-A1 (FISCHER) N.b. pages 5-13	1,3-6,9,10
X	Dialog record 01887837 of Computer Shopper, v16, n2, Feb 1996, p581(2), Nicolaisen N, "Waking up the Web: Sun Microsystem's Java gives programming on the Web a boost"	1,3-6,9,10
X	Dialog record 01939746 of PC Magazine, v15, n11, June 11 1996, p221(5), Lam J, "Java and Java tools: waking up the Web"	1,3-6,9,10

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

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 98/44404**

(43) International Publication Date: 8 October 1998 (08.10.98)

(21) International Application Number: PCT/US98/06116

(22) International Filing Date: 27 March 1998 (27.03.98)

(30) Priority Data:
08/829,990 1 April 1997 (01.04.97) US

(71) Applicant: SUN MICROSYSTEMS, INC. [US/US]; 901 San Antonio Road, Palo Alto, CA 94303 (US).

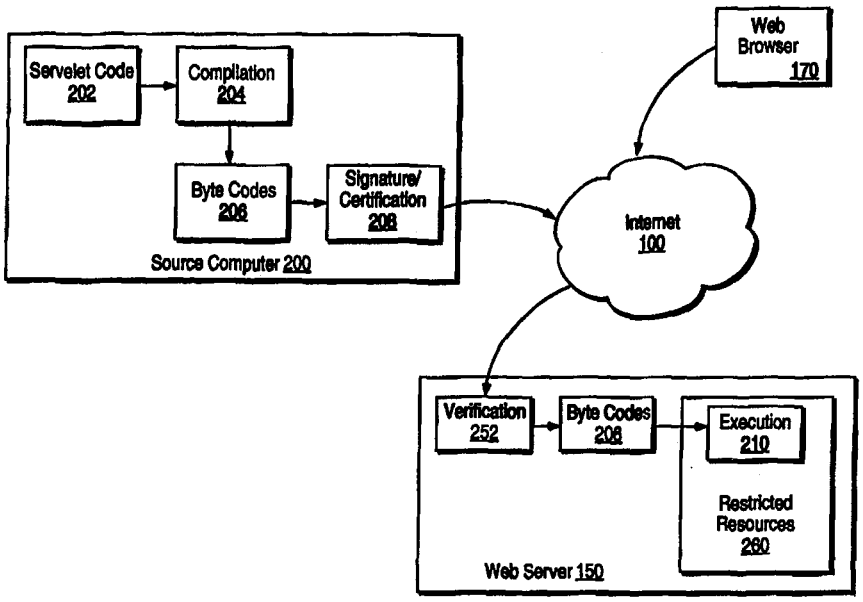
(72) Inventors: LEE, John, Y.; 200 E. Dana Street #D92, Mountain View, CA 94041 (US). DHARMARAJ, Satish, K.; 3375 Honestead Road #8, Santa Clara, CA 95051 (US).

(74) Agents: HYMAN, Eric, S. et al.; Blakely, Sokoloff, Taylor & Zafman, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025–1026 (US).

(81) Designated States: JP, European patent (AT, BE, CH, 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 the receipt of amendments.

(54) Title: METHOD AND APPARATUS FOR PROVIDING SECURITY FOR SERVERS EXECUTING APPLICATION PROGRAMS RECEIVED VIA A NETWORK



(57) Abstract

A method and apparatus for providing security to a server running an application program received over a network is provided. The application program, along with a source identifier is received from a source computer. Access privileges to server resources are granted based on the source identifier. The application program is loaded into a predetermined, bounded region of memory.

Nintendo Exhibit 1003
Part 2 of 2
Page 001137

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 AND APPARATUS FOR PROVIDING SECURITY FOR SERVERS EXECUTING APPLICATION PROGRAMS RECEIVED VIA A NETWORK

FIELD OF THE INVENTION

The present invention relates to network computing, and more specifically, to providing security for servers that receive applications programs via a network.

BACKGROUND OF THE INVENTION

Currently, the largest computer network in existence is the Internet, which is a worldwide interconnection of computer networks that communicate using a common protocol. The Internet grew out of work funded in the 1960s by the United States Department of Defense's Advanced Research Projects Agency (ARPA). Millions of computers, from low end personal computers to high-end super computers are connected to the Internet. For many years, few outside of the academic/research community accessed the Internet.

In 1989, a new type of information system known as the World-Wide Web (the Web) was introduced to the Internet. Early development of the Web took place at CERN, the European Particle Physics Laboratory. The Web is a wide-area hypermedia information retrieval system.

In 1993, the National Center for Supercomputing Applications (NSCA) released a Web browser called "Mosiac" that implemented a graphical user interface (GUI). Mosiac's GUI was simple to learn and allowed users to retrieve documents from the Web using simple point-and-click commands. The introduction of Mosaic simplified document and information retrieval over the Internet and more specifically on the Web.

Nintendo Exhibit 1003
Part 2 of 2
Page 001139

As a result, use of the Internet became more popular outside of the academic/research community.

At that time, architecture of the Web typically followed a conventional client-server model. The terms "client" and "server" are used to refer to a computer's general role as a requester of data (the client) or provider of data (the server). In the Web environment, Web browsers, such as Mosaic, reside in clients and Web documents reside in servers. Web clients and Web servers communicate using a protocol called "HyperText Transfer Protocol" (HTTP). A browser opens a connection to a server and initiates a request for a document. The server delivers the requested document, typically in the form of a text document coded in a standard HyperText Markup Language (HTML) format.

In 1994, a subsidiary of Sun Microsystems released a Web browser based on a programming language developed at Sun called Java™. Programs written in Java are architecture neutral such that the programs run on any platform that supports Java. The programs are architecture neutral because they are compiled into a series of byte-codes that are not hardware specific and thus can be downloaded and executed on many different computers. Byte-codes are transmitted over the Internet and then translated and executed by the receiving computer.

One unique feature of Java not provided by typical Web browsers is an ability to provide applets as part of a Web page. Applets are part of a Web page, but they are downloaded and executed by the computer running a Web browser rather than the Web server. Thus, a user, with a browser that supports Java, accesses a Web page and downloads a small program that is executed locally. This, however, presents security issues not present in the traditional client-server model.

To provide security, Java methods and variables are accessed by name and are executed in a restricted environment that includes a predetermined portion of memory. Furthermore, when byte-codes are received they are subjected to a verification process that determines whether the byte-codes have been modified.

Security for applets may be provided in a straightforward all-or-nothing manner. For example, the applet is loaded and executed, or it is not loaded and executed. Because the applet is self-contained, it does not access local resources other than a designated, bounded region of memory.

Another unique feature of Java is the servlet, which is a program, like an applet, however, a servlet runs on the Web server rather than the Web browser. Servlets are typically larger than applets and/or require more resources. For example, a servlet may be a search program offered by a particular server. A user accesses the servlet by supplying search parameters. The servlet then executes a search of resources available to the server. Thus, a user may have access to certain server resources and the server may have its search software updated by a third party without any effort by the server administrator.

Because servlets may require access to server resources, the all-or-nothing approach is inefficient. In order to provide a secure all-or-nothing scheme in a server that loads servlets, a standard subset of resources would be offered to all servlets. However, different servlets require different resources and different servlet sources may be worthy of different levels of trust. Thus, prior security schemes do not provide a flexible, yet secure, environment for providing servlets on a server. Therefore, it would be desirable to provide security to Web servers from potentially hostile programs, wherein permissions to access the resources of the server are granted based on the source of the program.

Nintendo Exhibit 1003
Part 2 of 2
Page 001141

As the Web grows, it would be desirable to provide greater access to the resources and features of the Web. Thus, many controllers of Web servers may wish to provide access to servlets developed by third parties. Therefore, it would be desirable to provide security to Web servers from potentially hostile servlets, wherein access to the resources of the server are allocated based on the source of a particular servlet.

SUMMARY OF THE INVENTION

A method and apparatus for providing security for a server executing programs received by the server via a network is disclosed. An application program that is to be provided by a Web server along with a source identifier is received by the Web server via a network, such as the Internet. Before loading the application program, the server performs a verification procedure including granting access privileges based on the source identifier. Access privileges are granted or withheld for a plurality of resources available to the server. If an application program is received from a known hostile source or no access privileges are granted, the applications program may be rejected. Thus, the resources defining the application program's universe, or sandbox, is determined individually based on source identifiers.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

Figure 1 is one embodiment of a network that may be used to practice the present invention.

Figure 2 is a block diagram for practicing the present invention.

Figure 3 is a flow diagram of one embodiment of verifying a servlet and allocating resources to the servlet based on the source of the servlet according to the present invention.

DETAILED DESCRIPTION

A method and apparatus for providing security to servers running applications programs based on the source of the application program 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.

An application program that is to be provided by a Web server along with a source identifier is received by the Web server via a network, such as, the Internet. Before loading the application program, the server performs a verification procedure including granting access privileges based on the source identifier. Access privileges are granted or withheld for resources available to the server. If an application program is received from a known hostile source, or if no access privileges are granted, the applications program may be rejected. Thus, the resources defining the application program's universe, or sandbox, are determined individually based on source identifiers.

Figure 1 is one embodiment of a network that may be used to practice the present invention. Internet 100 connects a plurality of hosts, such as

hosts 120, 122, 124 and 126 to Web server 150. The hosts may also act as Web servers to provide files or resources to other hosts. Alternatively, Internet 100 may be a local area network (LAN) or other network that connects a plurality of computers together.

According to the present invention, a host may send a program, such as a servlet, to Web server 150. When Web server 150 receives a servlet and a source identifier from a source computer, the source of the servlet is checked to determine the access privileges available to the servlet. Thus, Web server 150 may reject a servlet from an unknown or known hostile source, or Web server 150 may accept and load a servlet and grant access to server resources based on the source of the servlet. Once the servlet is loaded by web server 150, it may be accessed by a web browser running on a host connected to web server 150.

Figure 2 is one embodiment of a source computer coupled to a Web server via the Internet according to the present invention. The steps that for providing security in the server according to one embodiment are show as blocks in the computer that performs that step.

An application program, such as a servlet, may be developed on a computer other than the one on which it is to be offered over a network. A servlet is written in servlet code, shown by 202. According to one embodiment, the servlet is developed in a Java environment, however, a servlet or other program may be developed in a non-Java environment. The servlet code is compiled, at 204, into a plurality of byte codes, shown by 206. Byte-codes are similar to machine instructions, however, they are not machine specific. Byte-length codes are used in one embodiment of the present invention, however, other code sizes may also be used. It is important to note that all of the codes are the same size.

A digital certificate, which includes a digital signature, represented by 208, is then appended to byte-codes 206. According to one embodiment, the digital certificate is an X.509 certificate. However, other source identifiers may also be used. The source identifier must provide a secure and unique identification of the signer such that the administrator of the receiving web server may trust the source identifier to authenticate the source of the servlet.

The byte-codes are sent to a designated server via the Internet 100, or other network. The receiving computer, such as Web server 150 that receives the byte-codes executes a verification process 252. Verification comprises examining the source identifier, such as digital certificate 208, to a local list of known trusted sources and granting resource privileges based on that source identifier. Additionally, the verification process includes other verification techniques to determine whether the byte-codes have been modified or if the source identifier has been modified. According to one embodiment, an access control list (ACL) is maintained on Web server 150 that stores trusted sources and the privileges extended to servlets received from those sources. Resource privileges include load privileges, permission to read or to write files on the server's disk(s), network read and write privileges, execution privileges, system property access, and socket listen or connect privileges. Other privileges may also be granted. Any combination of one or more of these access privileges may be granted to a servlet.

Once the byte-codes have been verified, they are translated into servlets configured to run on the architecture of Web server 150. The servlet is then loaded into the servlet's sandbox (i.e., a known, bounded area in memory and granted access only to those specific resources listed in the ACL). Once the servlet is loaded in Web server 150, Web browser 170 may access the servlet and any resources available to the servlet. Thus, resource

privileges are granted on a servlet-by-servlet basis, which increases the flexibility of a Web server's security. This improved flexibility allows administrators to grant more privileges to known and trusted sources, while granting fewer privileges to new or unknown sources. By eliminating an all-or-nothing security approach, the web may offer more resources in a more convenient manner.

Figure 3 is a flow diagram of one embodiment of verifying a servlet and allocating resources to the servlet based on the source of the servlet according to the present invention. In step 300, a Web server receives a servlet or other application program via a network, such as the Internet.

In step 310 the local ACL is checked to determine what, if any, privileges are to be granted to the servlet based on the source identifier received with the servlet. The ACL may contain an entry for unsigned servlets to grant a specified level of access to all unsigned servlets. The Web server may also reject all unsigned servlets along with servlets received from a source known to be hostile.

If no access privileges are granted in step 320, the servlet is rejected or not loaded in step 330. If the servlet is granted access privileges in step 320, a sandbox is defined for the servlet in step 340. A sandbox defines the set of access privileges granted to the servlet.

In step 350, the servlet is loaded by Web server 150. In addition to limitations imposed on a servlet by the sandbox to which it is assigned, other checks may be performed. For example, accesses to certain resources may be monitored by the server to guard against hostile behavior.

Thus, by providing identity-based access controls, a Web server can control which servlets have access to which data. This arrangement provides protection against theft or alteration of data. In addition, by

restricting a servlet to a sandbox, the Web server can control the damage that malicious code can cause.

In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

Nintendo Exhibit 1003
Part 2 of 2
Page 001147

CLAIMS

What is claimed is:

1. A method for providing security for a server running an application program comprising the steps of:

receiving the application program from a source computer, wherein the application program has associated with it a source identifier that uniquely identifies the source computer in a secure manner;

determining resources to be provided for use by the application program based on the source identifier;

granting access privileges to the resources determined to be accessible by the application program; and

loading the application program into a predetermined, bounded location in memory of the server.

2. The method of claim 1, wherein the source identifier is a digital signature.

3. The method of claim 1, wherein the step of determining comprises the step of checking an access control list (ACL) to determine the resources available to the application program based on the source identifier.

4. The method of claim 1, wherein the access privileges comprise:

server disk read privileges;

server disk write privileges;

server network read privileges; and

server network write privileges.

5. The method of claim 1, wherein the access privileges comprise:

execution privileges;

system property access privileges;

socket connect privileges; and

socket listen privileges.

6. The method of claim 1, wherein the application program is received from the source computer the Internet.

7. The method of claim 6, wherein the application program is received via the world wide web.

8. The method of claim 1, wherein the application program is received a local area network.

9. A method for providing an application program for execution on a remote server, wherein the server executes the application program in

a bounded range of memory locations and allocates local resource access based on a source of the application program comprising the steps of:

generating the application program;

compiling the application program into a plurality of predetermined length codes;

generating a source identifier;

associating the source identifier with the codes; and

sending the codes to a server, wherein the server verifies the codes and allocates access to local resources based on the source identifier, further wherein the server generates the application program based on the codes received.

10. The method of claim 9, wherein the source identifier comprises a digital signature.

11. The method of claim 9, wherein the plurality of predetermined length codes comprises a plurality of byte-length codes.

12. The method of claim 9, wherein the resources allocated comprise:

server disk read privileges;

server disk write privileges;

server network read privileges; and

server network write privileges.

13. The method of claim 9, wherein the resources allocated comprise:

execution privileges;

system property access privileges;

socket connect privileges; and

socket listen privileges.

14. The method of claim 9, wherein the codes are sent via the Internet.

15. The method of claim 9, wherein the codes are sent via a local area network.

16. An apparatus for loading an application program received from a source computer for use on a server comprising:

means for receiving the application program from a source computer, wherein the application program has associated with it a source identifier that uniquely identifies the source computer in a secure manner;

means for determining resources accessible by the server to be provided for use by the application program based on the source identifier;

means for granting access privileges to the resources determined to be accessible by the application program; and

means for loading the application program into a predetermined, bounded location in memory of the server.

17. The apparatus of claim 16, wherein the source identifier comprises a digital signature.

18. The apparatus of claim 17, wherein the means for determining resources accessible by the server comprises an access control list (ACL) maintained on the server.

19. A machine readable medium having stored thereon data representing sequences of instructions, which when executed by a computer system, cause said computer system to perform the steps of:

receiving the application program from a source computer, wherein the application program has associated with it a source identifier that uniquely identifies the source computer in a secure manner;

determining resources accessible by the server to be provided for use by the application program based on the source identifier;

granting access privileges to the resources determined to be accessible by the application program; and

loading the application program into a predetermined, bounded location in memory of the server.

20. The machine readable medium of claim 19, wherein the source identifier comprises a digital signature.

Nintendo Exhibit 1003
Part 2 of 2
Page 001152

21. The machine readable medium of claim 19, wherein the step of determining comprises the step of checking an access control list (all) to determine the resources available to the application program based on the source identifier.

22. The machine readable medium of claim 19, wherein the access privileges comprise:

server disk read privileges;

server disk write privileges;

server network read privileges; and

server network write privileges.

23. The machine readable medium of claim 19, wherein the access privileges comprise:

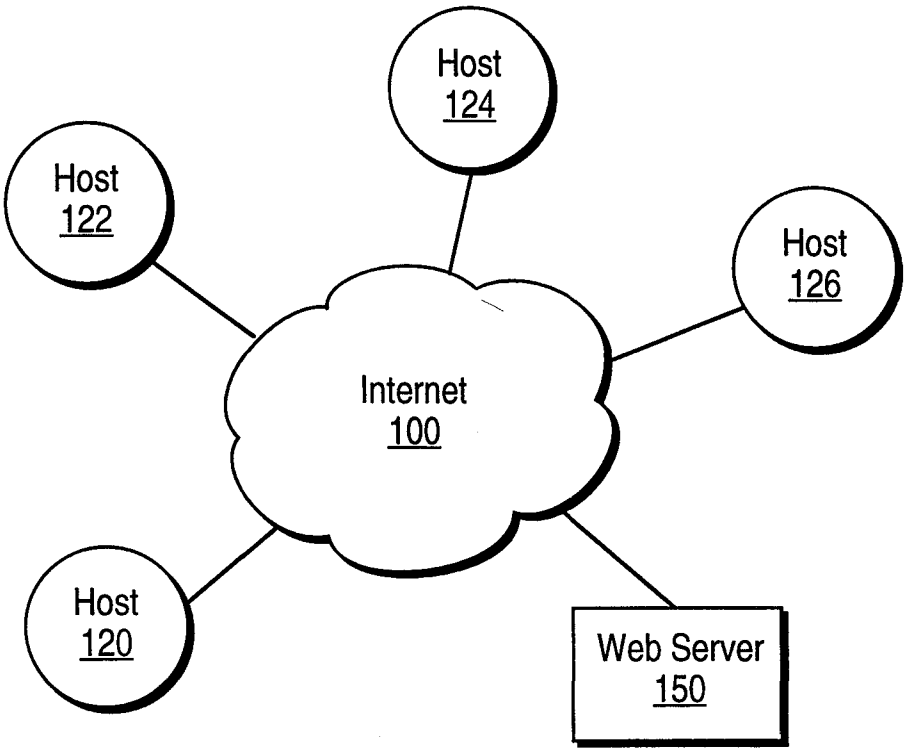
execution privileges;

system property access privileges;

socket connect privileges; and

socket listen privileges.

FIG. 1


Host
124
Host
122
Host
126
Internet
100
Host
120
Web Server
150

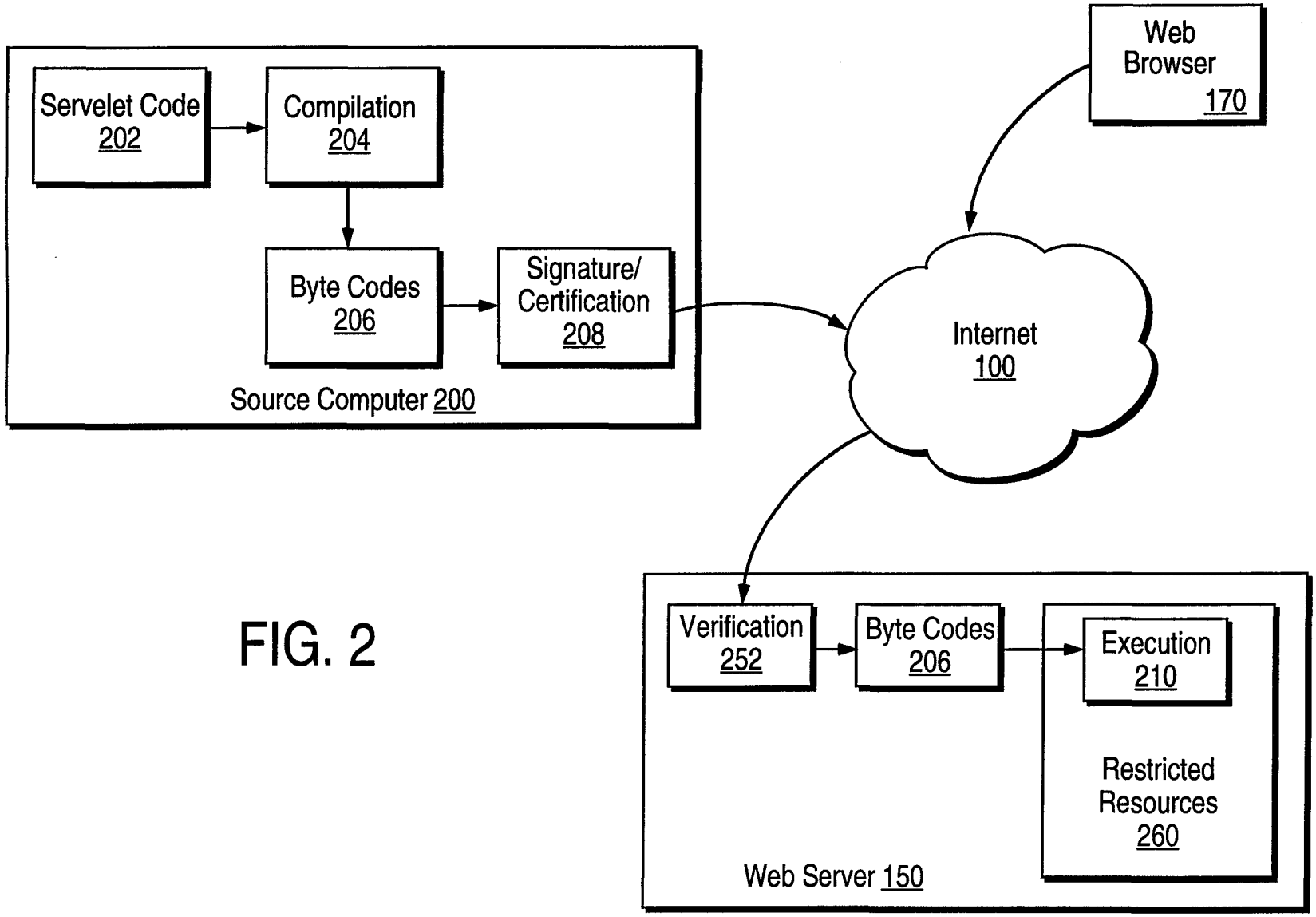

Servelet Code
202
Compilation
204
Byte Codes
206
Signature/
Certification
208
Source Computer 200
Web
Browser
170
Internet
100
Verification
252
Byte Codes
206
Execution
210
Restricted
Resources
260
Web Server 150

FIG. 2

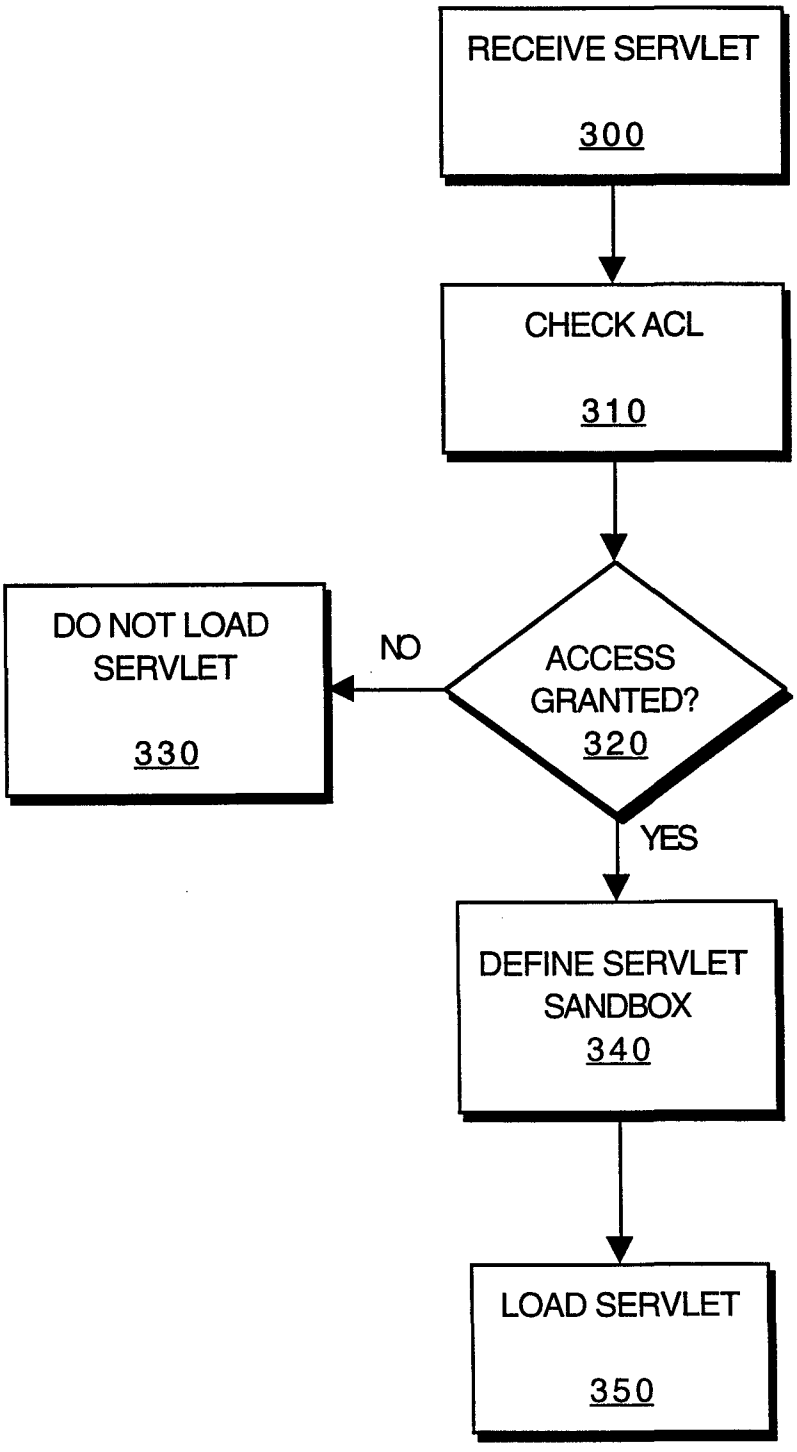

RECEIVE SERVLET
300
CHECK ACL
310
ACCESS GRANTED?
320
NO
DO NOT LOAD SERVLET
330
YES
DEFINE SERVLET SANDBOX
340
LOAD SERVLET
350

FIG. 3

INTERNATIONAL SEARCH REPORT

national Application No
PCT/US 98/06116

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 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 6 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.
X	EP 0 409 397 A (INT COMPUTERS LTD) 23 January 1991 see figures 1-3 see column 2, line 31 - column 3, line 33 see column 4, line 48 - column 6, line 50	1,4,8,9, 12,13, 15,16, 19,21-23
A	EP 0 580 350 A (DIGITAL EQUIPMENT CORP) 26 January 1994	1-5, 8-10,12, 13,15-23
A	EP 0 570 123 A (FISCHER ADDISON M) 18 November 1993 see figures 1-3 see column 5, line 13 - column 10, line 50	1,2,6, 8-11, 14-17, 19-21

☐ 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
21 July 1998	27/07/1998

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	Weiss, P

Nintendo Exhibit 1003
Part 2 of 2
Page 001157

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 98/06116

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0409397 A	23-01-1991	AU 629663 B	08-10-1992
		AU 5914390 A	24-01-1991
		DE 69028362 D	10-10-1996
		DE 69028362 T	03-04-1997
		US 5012515 A	30-04-1991
EP 0580350 A	26-01-1994	US 5235642 A	10-08-1993
		JP 6202998 A	22-07-1994
EP 0570123 A	18-11-1993	US 5412717 A	02-05-1995
		AU 3820993 A	18-11-1993
		CA 2095087 A	16-11-1993
		JP 6103058 A	15-04-1994
		US 5311591 A	10-05-1994

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 14/715,227

APPLICATION AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY FEE($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	280
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	600
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	720
TOTAL CLAIMS (37 CFR 1.16(i))	20 minus 20 =	*			OR	x 80 =	0.00
INDEPENDENT CLAIMS (37 CFR 1.16(h))	3 minus 3 =	*				x 420 =	0.00
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						0.00
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							0.00
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	1600

APPLICATION AS AMENDED - PART II

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT A	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT B	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L 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 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 NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
14/715,227	05/18/2015	2143	1740	11913-US-CNT3 4214-25714	20	3

CONFIRMATION NO. 5874

FILING RECEIPT

82313
Conley Rose - BlackBerry Files
Attn: J. Robert Brown
5601 Granite Parkway, Suite 500
Plano, TX 75024

OC000000075424734

Date Mailed: 06/01/2015

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

Neil P. Adams, Kitchener, CANADA;
Michael G. Kirkup, Waterloo, CANADA;
Herbert A. Little, Waterloo, CANADA;
Russell N. Owen, Waterloo, CANADA;

Applicant(s)

BlackBerry Limited, Waterloo, CANADA;

Power of Attorney: The patent practitioners associated with Customer Number 82313

Domestic Priority data as claimed by applicant

This application is a CON of 13/618,311 09/14/2012 PAT 9033216
which is a CON of 12/885,281 09/17/2010 PAT 8887988
which is a CON of 11/118,844 04/29/2005 PAT 7815100
which claims benefit of 60/567,163 04/30/2004
This application 14/715,227
is a CIP of 13/606,814 09/07/2012 ABN
which is a CON of 12/869,589 08/26/2010 PAT 8302185
which is a CON of 10/732,132 12/10/2003 PAT 7793355
which claims benefit of 60/432,610 12/12/2002

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.) - None.
Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

Permission to Access - A proper **Authorization to Permit Access to Application by Participating Offices** (PTO/SB/39 or its equivalent) has been received by the USPTO.

If Required, Foreign Filing License Granted: 05/27/2015

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 14/715,227**

Projected Publication Date: 09/10/2015

Non-Publication Request: No

Early Publication Request: No

Title

System and Method of Owner Application Control of Electronic Devices

Preliminary Class

715

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific

countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop

technology, manufacture products, deliver services, and grow your business, visit http://www.SelectUSA.gov or call +1-202-482-6800.

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 NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
14/715,227	05/18/2015	Neil P. Adams	11913-US-CNT3 4214-25714

CONFIRMATION NO. 5874

82313
Conley Rose - BlackBerry Files
Attn: J. Robert Brown
5601 Granite Parkway, Suite 500
Plano, TX 75024

INFORMAL NOTICE

OC00000075424735

Date Mailed: 06/01/2015

INFORMATIONAL NOTICE TO APPLICANT

Applicant is notified that the above-identified application contains the deficiencies noted below. No period for reply is set forth in this notice for correction of these deficiencies. However, if a deficiency relates to the inventor's oath or declaration, the applicant must file an oath or declaration in compliance with 37 CFR 1.63, or a substitute statement in compliance with 37 CFR 1.64, executed by or with respect to each actual inventor no later than the expiration of the time period set in the "Notice of Allowability" to avoid abandonment. See 37 CFR 1.53(f).

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

- A properly executed inventor's oath or declaration has not been received for the following inventor(s):
  Neil P. Adams
  Michael G. Kirkup
  Herbert A. Little
  Russell N. Owen

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/fhadera/

Electronic Acknowledgement Receipt

EFS ID:	22385554
Application Number:	14715227
International Application Number:	
Confirmation Number:	5874
Title of Invention:	System and Method of Owner Application Control of Electronic Devices
First Named Inventor/Applicant Name:	Neil P. Adams
Customer Number:	82313
Filer:	J. Robert Brown/Rami Law
Filer Authorized By:	J. Robert Brown
Attorney Docket Number:	11913-US-CNT3 4214-25714
Receipt Date:	19-MAY-2015
Filing Date:	
Time Stamp:	12:34:45
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Power of Attorney	11913-US-CNT3_2015-05-19_SUB_POA_.pdf	453468 146cef290bf3da42dedbf3e656c45a8448b7fe39	no	2

Warnings:	
Information:	

Total Files Size (in bytes):	453468

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82A (07-13)
Approved for use through 11/30/2014. OMB 0651-0051
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 FOR POWER OF ATTORNEY TO ONE OR MORE REGISTERED PRACTITIONERS

NOTE: This form is to be submitted with the Power of Attorney by Applicant form (PTO/AIA/82B) to identify the application to which the Power of Attorney is directed, in accordance with 37 CFR 1.5, unless the application number and filing date are identified in the Power of Attorney by Applicant form. If neither form PTO/AIA/82A nor form PTO/AIA82B identifies the application to which the Power of Attorney is directed, the Power of Attorney will not be recognized in the application.

Application Number	14/715,227
Filing Date	May 18, 2015
First Named Inventor	Neil P. Adams
Title	System and Method of Owner Application Control of Electronic Devices
Art Unit	TBD
Examiner Name	Unknown
Attorney Docket Number	11913-US-CNT3 4214-25714

SIGNATURE of Applicant or Patent Practitioner

Signature	/J. Robert Brown, Jr./	Date (Optional)	2015-05-19
Name	J. Robert Brown, Jr.	Registration Number	45,438
Title (if Applicant is a juristic entity)			
Applicant Name (if Applicant is a juristic entity)			

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4(d) for signature requirements and certifications. If more than one applicant, use multiple forms.

☐ *Total of _____ forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82B (07-13)
Approved for use through 11/30/2014. 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

POWER OF ATTORNEY BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number	Filing Date

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)

☑ I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above: 82313

OR

☐ I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:

☑ The address associated with the above-mentioned Customer Number

OR

☐ The address associated with Customer Number:

OR

Firm or Individual Name	Conley Rose, P.C.				
Address					
City		State		Zip	
Country					
Telephone		Email			

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box).

BlackBerry Limited

☐ Inventor or Joint Inventor (title not required below)

☐ Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)

☑ Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)

☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature	/Kevin Mulvenon/	Date (Optional)	Dec 10, 2013
Name	Kevin Mulvenon		
Title	Senior Vice President, Taxation		

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.

☐ Total of forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Legal OK
JLH as per BB OK
WN

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	System and Method of Owner Application Control of Electronic Devices
First Named Inventor/Applicant Name:	Neil P. Adams
Filer:	J. Robert Brown/Rami Law
Attorney Docket Number:	11913-US-CNT3 4214-25714

Filed as Large Entity

Filing Fees for Utility under 35 USC 111(a)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Utility application filing	1011	1	280	280
Utility Search Fee	1111	1	600	600
Utility Examination Fee	1311	1	720	720
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
			Total in USD ($)	**1600**

CLAIMS:

What is claimed:

1. A method of application control on an electronic device, the method comprising:

receiving, by the electronic device, from an application operating on the electronic device, a request to access local memory on the electronic device;

determining whether the application is allowed to access the local memory based upon an authorization record and an application identifier associated with the application; and

controlling application access to the local memory based upon the determining.

2. The method of claim 1, wherein the request to access the local memory comprises a request to erase data from the local memory.

3. The method of claim 1, wherein the request to access the local memory comprises a request to read data from the local memory.

4. The method of claim 1, wherein the authorization record and the application identifier are included in owner control information stored on the electronic device for each application allowed to be executed on the electronic device, and wherein insertion of additional owner control information onto the electronic device is restricted.

Nintendo Exhibit 1003
Part 2 of 2
Page 001171

5. The method of claim 4, further comprising restricting modification of the owner control information after the owner control information has been stored on the electronic device.

6. The method of claim 1, wherein the authorization record and the application identifier are provided by an external source through a wireless communication channel.

7. The method of claim 1, wherein the authorization record related to determining whether access to the local memory is allowed is provided by an external computer that is used to enforce policies for operating electronic devices within an organization.

8. The method of claim 1, wherein an update to the authorization record is provided by an external source for storage on the electronic device.

Nintendo Exhibit 1003
Part 2 of 2
Page 001172

9. The method of claim 1, wherein a plurality of authorization records associated with a plurality of application identifiers are stored on the electronic device, the method further comprising:

receiving requests to access the local memory from a plurality of applications operating on the electronic device;

determining whether access to the local memory is allowed to be performed by their respective applications based upon the stored authorization records and the application identifiers that are respectively associated with the plurality of applications; and

controlling access by the plurality of applications to local memory based upon whether access to the local memory is allowed.

10. The method of claim 9, wherein a group containing two or more of the applications is associated with one of the authorization records.

11. The method of claim 1, wherein the authorization record is stored in a protected data store, thereby preventing alteration or deletion of the authorization record unless permitted.

12. A device, comprising:

a memory; and

a processor coupled with the memory and configured by instructions stored in the memory to:

receive, from an application operating on the electronic device, a request to access local memory on the electronic device;

determine whether the application is allowed to access the local memory based upon an authorization record and an application identifier associated with the application; and

control application access to the local memory based upon the determining.

13. The device of claim 12, wherein the request to access the local memory comprises a request to erase data from the local memory.

14. The device of claim 12, wherein the request to access the local memory comprises a request to read data from the local memory.

15. The device of claim 12, wherein the authorization record and the application identifier are included in owner control information stored on the electronic device for each application allowed to be executed on the electronic device, and wherein insertion of additional owner control information onto the electronic device is restricted.

Nintendo Exhibit 1003
Part 2 of 2
Page 001174

16. The device of claim 15, further comprising restricting modification of the owner control information after the owner control information has been stored on the electronic device.

17. The device of claim 12, wherein the authorization record and the application identifier are provided by an external source through a wireless communication channel.

18. The device of claim 12, wherein the authorization record related to determining whether access to the local memory is allowed is provided by an external computer that is used to enforce policies for operating electronic devices within an organization.

19. The device of claim 12, wherein an update to the authorization record is provided by an external source for storage on the electronic device.

20. A non-transitory computer-readable medium with instructions stored thereon, wherein the instructions are executable by a processor to cause the processor to perform the steps of:

receiving, from an application operating on the electronic device, a request to access local memory on the electronic device;

determining whether the application is allowed to access the local memory based upon an authorization record and an application identifier associated with the application; and

controlling application access to the local memory based upon the determining.

ABSTRACT

Systems and methods of owner application control of an electronic device are provided. Owner application control information is stored on the electronic device and/or one or more remote servers. Owner application control information is consulted to determine if one or more required applications are available for execution on the electronic device. If not, one or more required applications not available are downloaded and installed. This could be in a manner transparent to the user of the electronic device. If one or more required applications are not available on the electronic device, the device can be functionally disabled in whole, or in part, until one or more required applications are available.

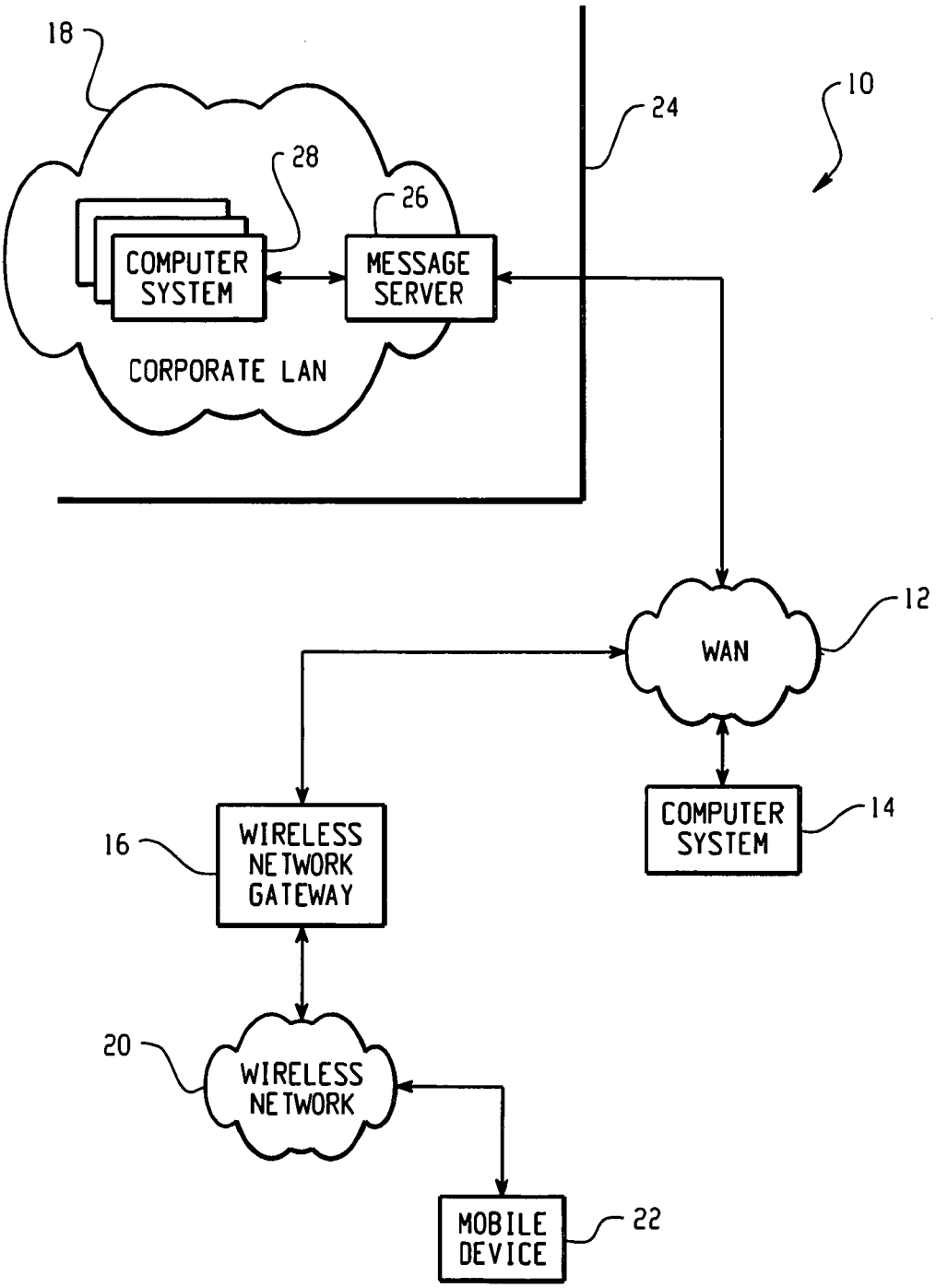

18
10
24
28
26
COMPUTER SYSTEM
MESSAGE SERVER
CORPORATE LAN
12
WAN
16
WIRELESS NETWORK GATEWAY
COMPUTER SYSTEM
14
20
WIRELESS NETWORK
MOBILE DEVICE
22

Fig. 1

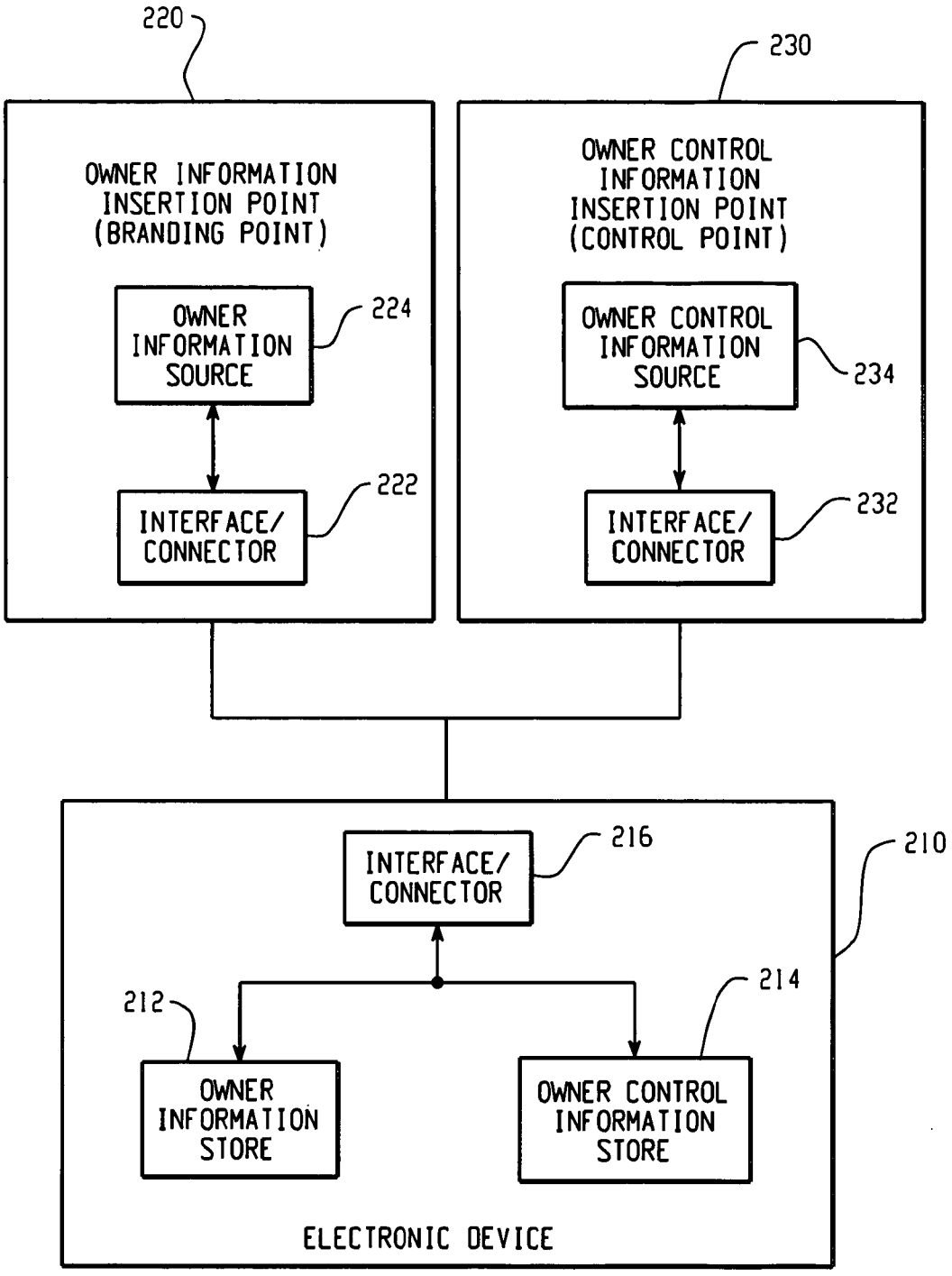

220
230
OWNER INFORMATION INSERTION POINT (BRANDING POINT)
OWNER CONTROL INFORMATION INSERTION POINT (CONTROL POINT)
OWNER INFORMATION SOURCE
224
OWNER CONTROL INFORMATION SOURCE
234
INTERFACE/ CONNECTOR
222
INTERFACE/ CONNECTOR
232
INTERFACE/ CONNECTOR
216
210
212
214
OWNER INFORMATION STORE
OWNER CONTROL INFORMATION STORE
ELECTRONIC DEVICE

Fig. 2

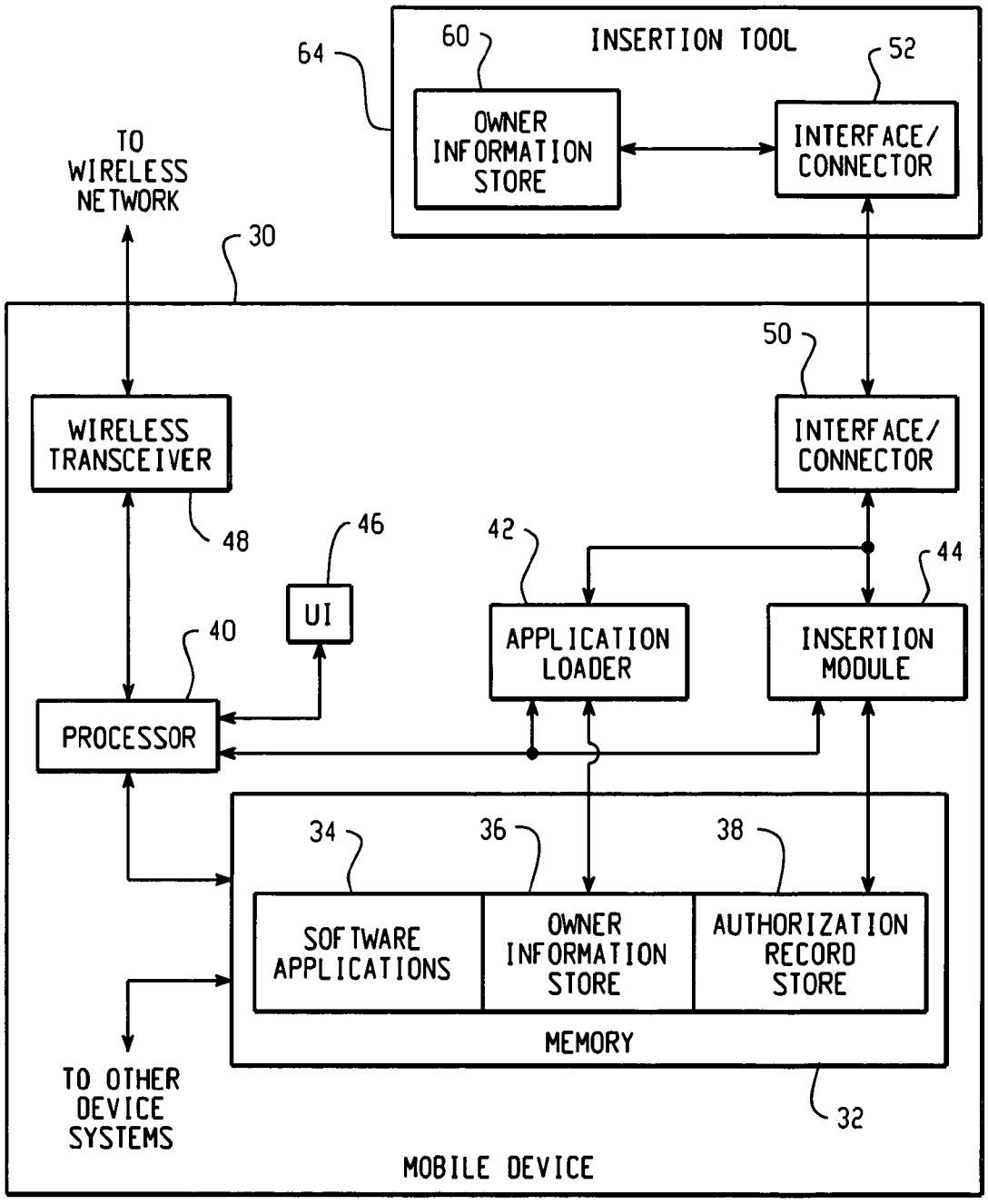

INSERTION TOOL
60
64
52
OWNER INFORMATION STORE
INTERFACE/ CONNECTOR
TO WIRELESS NETWORK
30
50
WIRELESS TRANSCEIVER
INTERFACE/ CONNECTOR
48
46
42
44
UI
APPLICATION LOADER
INSERTION MODULE
40
PROCESSOR
34
36
38
SOFTWARE APPLICATIONS
OWNER INFORMATION STORE
AUTHORIZATION RECORD STORE
MEMORY
TO OTHER DEVICE SYSTEMS
32
MOBILE DEVICE

Fig. 3

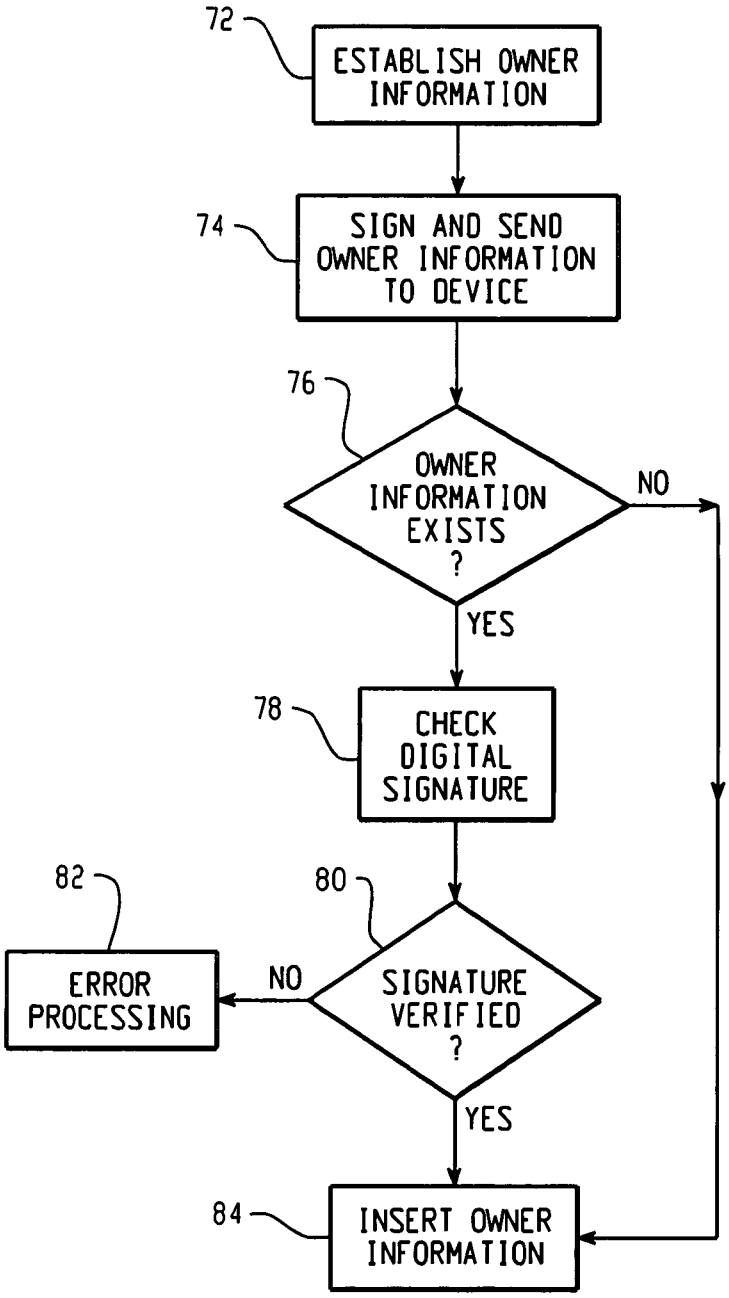

72
ESTABLISH OWNER INFORMATION
74
SIGN AND SEND OWNER INFORMATION TO DEVICE
76
OWNER INFORMATION EXISTS ?
NO
YES
78
CHECK DIGITAL SIGNATURE
82
80
ERROR PROCESSING
NO
SIGNATURE VERIFIED ?
YES
84
INSERT OWNER INFORMATION

Fig. 4

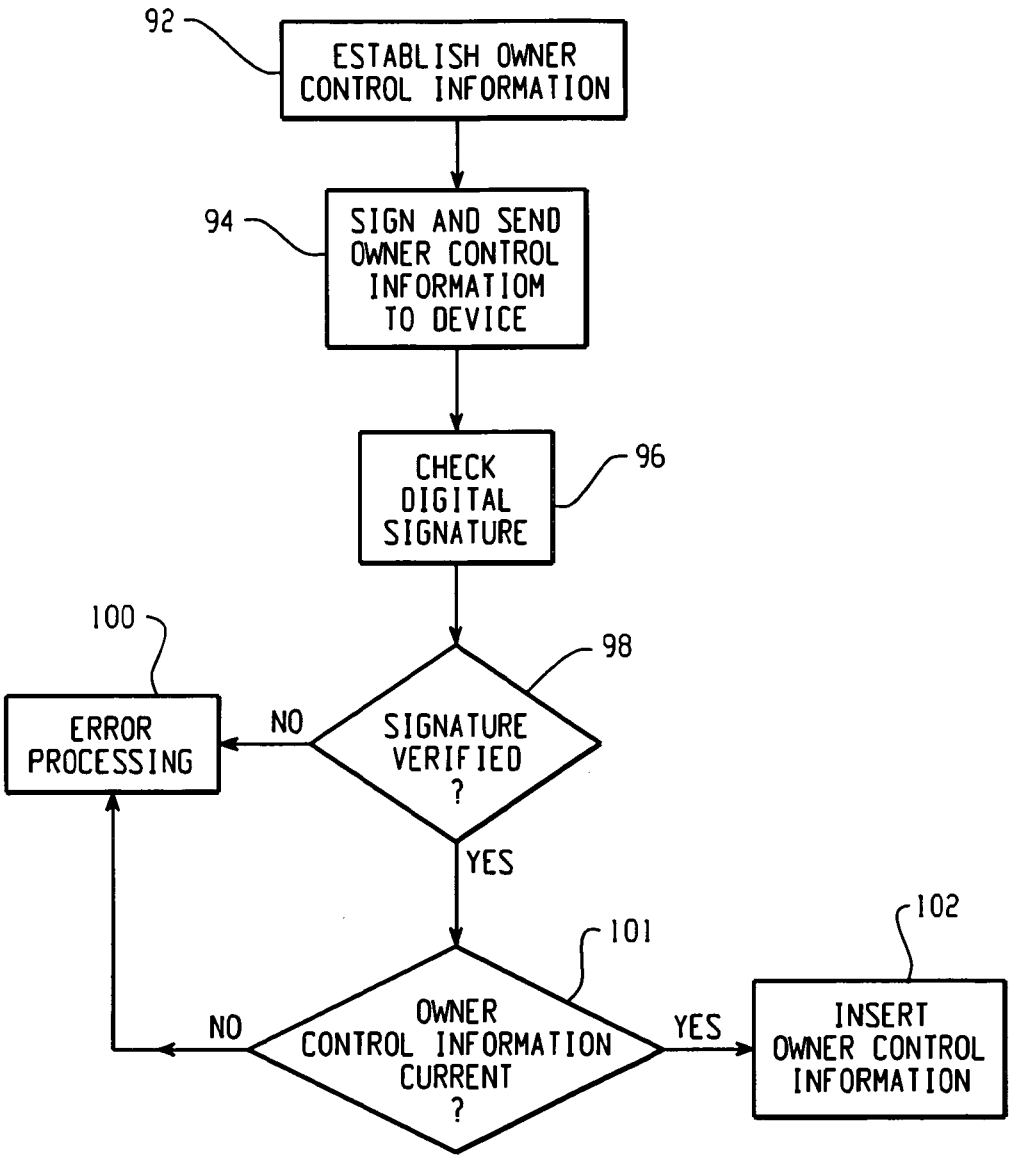

92
ESTABLISH OWNER CONTROL INFORMATION
94
SIGN AND SEND OWNER CONTROL INFORMATIOM TO DEVICE
CHECK DIGITAL SIGNATURE
96
100
98
ERROR PROCESSING
NO
SIGNATURE VERIFIED ?
YES
101
102
NO
OWNER CONTROL INFORMATION CURRENT ?
YES
INSERT OWNER CONTROL INFORMATION

Fig. 5

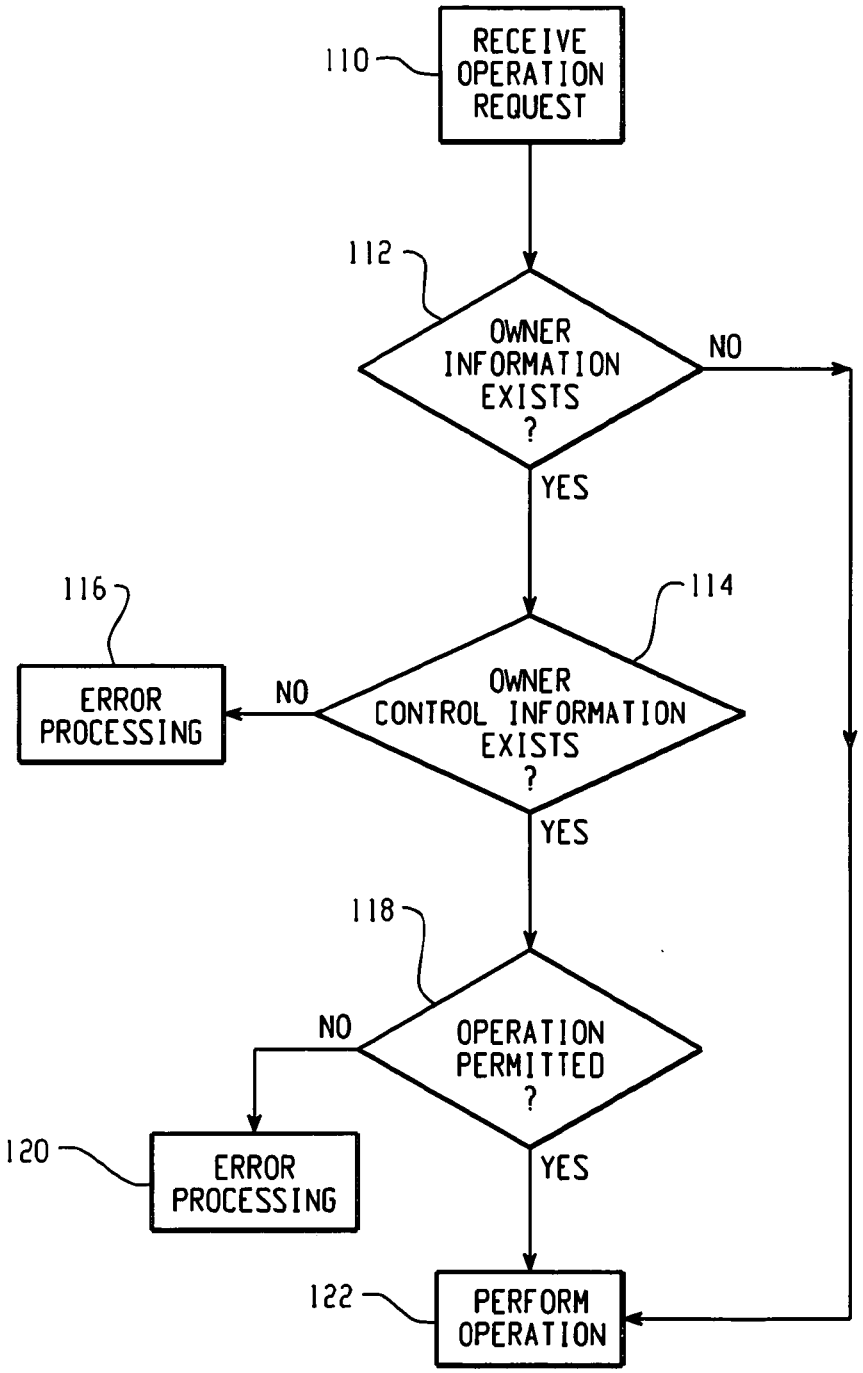

110
RECEIVE OPERATION REQUEST
112
OWNER INFORMATION EXISTS ?
NO
YES
116
ERROR PROCESSING
NO
114
OWNER CONTROL INFORMATION EXISTS ?
YES
118
NO
OPERATION PERMITTED ?
120
ERROR PROCESSING
YES
122
PERFORM OPERATION

Fig. 6

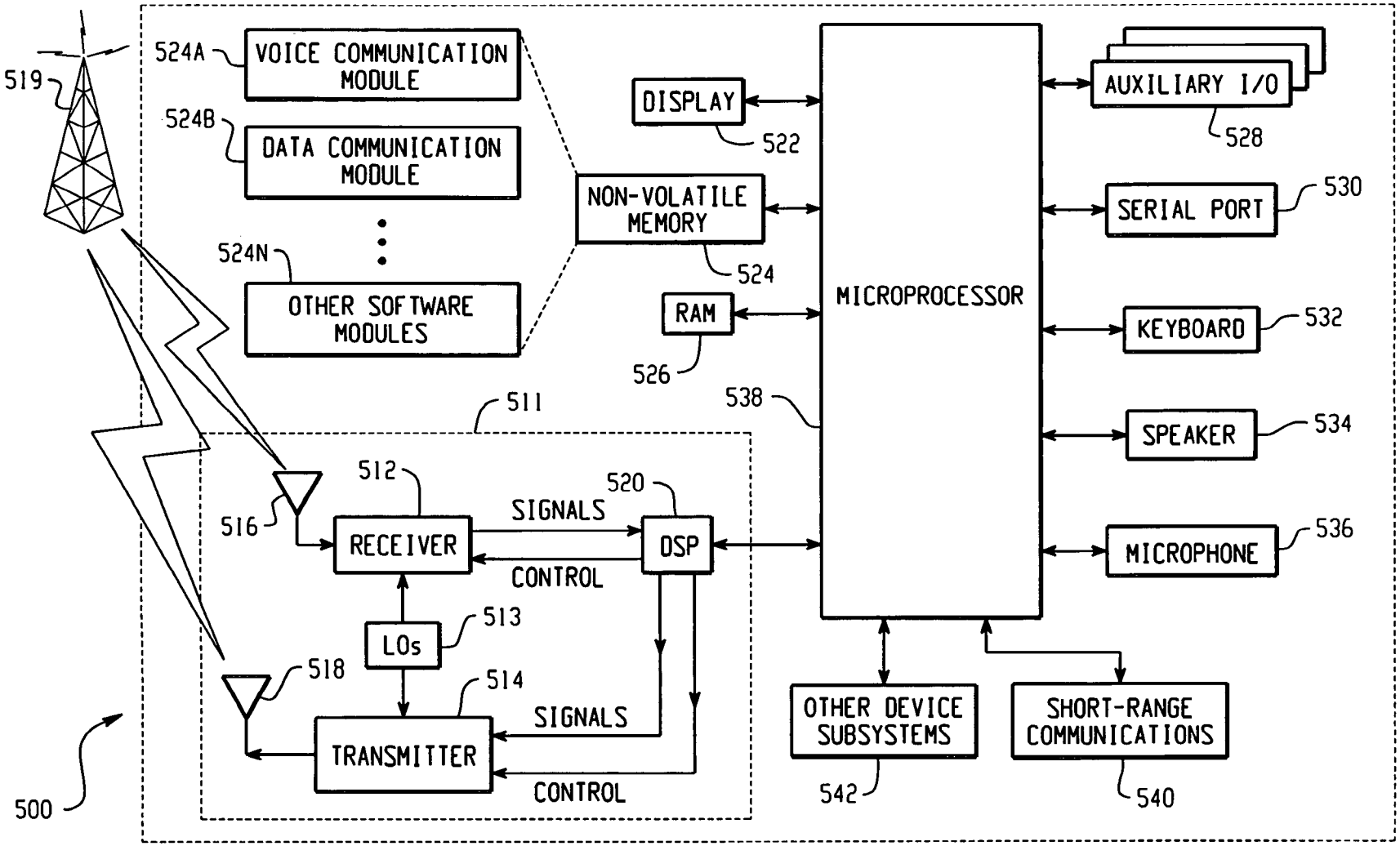


Fig. 7

System and Method of
Owner Application Control of Electronic Devices
Inventors: Neil P. Adams, et al.
Atty. Docket No. 11913-US-CNT[3] (4214-25714)

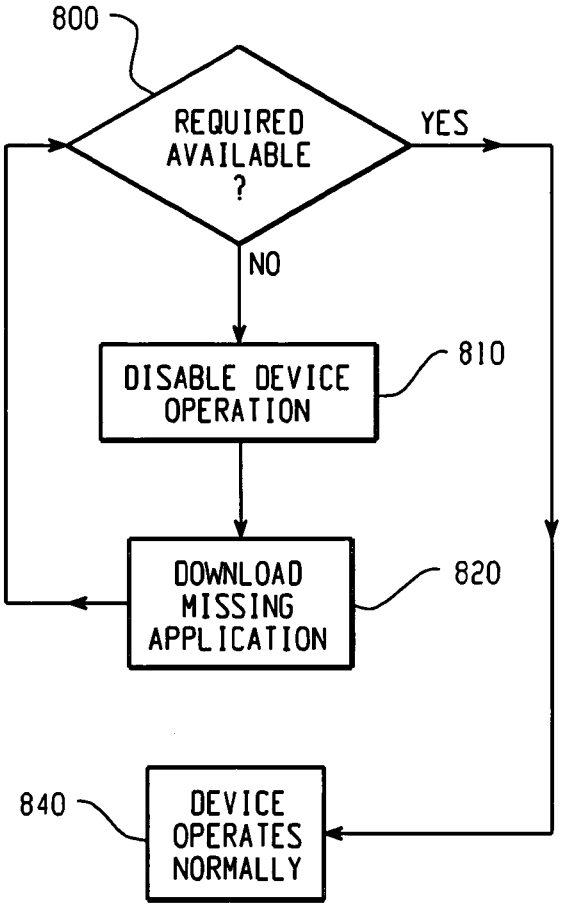

800
REQUIRED AVAILABLE ?
YES
NO
DISABLE DEVICE OPERATION
810
DOWNLOAD MISSING APPLICATION
820
840
DEVICE OPERATES NORMALLY

Fig. 8

Application	Version	Disposition	Delivery Mechanism	Policy Set
⊞ ☑ System Software	<Use Latest>	Required	Wireline Only	N/A
⊞ ☑ Browser	<System Version>	Required	Wireline Only	N/A
⊞ ☑ SSL/TLS Security Package	<System Version>	Required	Wireline Only	N/A
⊞ ☑ WTLS Security Package	<System Version>	Required	Wireline Only	N/A
⊞ ☑ Phone	<System Version>	Required	Wireline Only	N/A
⊟ ☑ MemoPad	<System Version>	Optional	Wireline Only	N/A
⊞ ☑ 950/957	<System Version>	Optional	Wireline Only	N/A
⊞ ☑ 6200 Series	<System Version>	Optional	Wireline Only	N/A
⊞ ☑ 6700 Series	<System Version>	Optional	Wireline Only	N/A
⊞ ☑ 7200 Series	<System Version>	Optional	Wireline Only	N/A
⊞ ☑ 7700 Series	<System Version>	Optional	Wireline Only	N/A
⊟ ☑ Tasks	<System Version>	Optional	Wireline Only	N/A
⊞ ☑ 6200 Series	<System Version>	Optional	Wireline Only	N/A
⊞ ☑ 6700 Series	<System Version>	Optional	Wireline Only	N/A
⊞ ☑ 7200 Series	<System Version>	Optional	Wireline Only	N/A
⊞ ☑ 7700 Series	<System Version>	Optional	Wireline Only	N/A
⊟ ☑ Calender and Tasks	<System Version>	Optional	Wireline Only	N/A
⊞ ☑ 950/957	<System Version>	Optional ▽	Wireless Only	N/A
⊞ ☑ Internal MDS App	1.0	Required	Wireless	Internal App Policies
⊞ ☑ RIMCell	<Any>	Optional	Wireless	External App Policies
⊞ ☑ RIMZee	<Any>	Optional	Wireless	External App Policies

900

Fig. 9

SYSTEM AND METHOD OF OWNER APPLICATION CONTROL OF ELECTRONIC DEVICES

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of U.S. Patent Application No. 13/618,311 filed September 14, 2012 by Neil P. Adams, et al. entitled "System and Method of Owner Application Control of Electronic Devices" (11913-US-CNT[2] – 4214-25713), which is a continuation of U.S. Patent No. 8,887,988 issued on November 18, 2014 entitled "System and Method of Owner Application Control of Electronic Devices" (11913-US-CNT – 4214-25712), which is a continuation of U.S. Patent No. 7,815,100 issued on October 19, 2010 entitled "System and Method of Owner Application Control of Electronic Devices" (11913-US-PAT – 4214-25701), which claims priority to and the benefit of U.S. Provisional Application No. 60/567,163 filed April 30, 2004 entitled "System and Method of Owner Application Control of Electronic Devices" (11913-US-PRV – 4214-25700); this application is also a continuation in part of U.S. Patent Application No. 13/606,814 filed September 7, 2012 by Herbert A. Little, et al. entitled "System and Method of Owner Control of Electronic Devices" (10735-US-CNT[2] – 4214-27116), which is a continuation of U.S. Patent No. 8,302,185 issued on October 30, 2012 entitled "System and Method of Owner Control of Electronic Devices" (10735-US-CNT – 4214-27115), which is a continuation of U.S. Patent No. 7,793,355 issued on September 7, 2010 entitled "System and Method of Owner Control of Electronic Devices" (10735-US-PAT – 4214-27101), which claims priority to and the benefit of U.S. Provisional Application No. 60/432,610 filed December 12, 2002 entitled "System and Method of Owner Control of Electronic

Nintendo Exhibit 1003
Part 2 of 2
Page 001187

Devices" (10735-US-PRV – 4214-27100), all of which are incorporated herein by reference as if reproduced in their entirety.

BACKGROUND

[0002] This system relates generally to electronic devices, and in particular to controlling application installation of such devices by a device owner.

[0003] In a corporate environment, employees are often provided with access to office supplies and equipment to be used in performing job functions. Standard equipment typically includes at least a personal computer (PC), and may also include wireless mobile communication devices and other types of electronic devices. Although such equipment is intended primarily for business or work-related purposes, users sometimes make personal use of office equipment. Employers may be comfortable with some degree of personal use of such equipment, provided that the personal use does not interfere with normal job functions, does not incur additional costs, and conforms with company policies.

[0004] In these types of situations, a user of an electronic device is not the owner of the device, and the user and owner may have different perceptions of acceptable device uses. Acceptable uses may be specified in company policies, for example, which employees are expected to follow, but beyond company policy statements, a corporate device owner often has little if any control over how electronic devices are ultimately used. According to one known scheme for controlling operation of electronic devices, an owner loads a policy file onto a device to restrict the type of operations or software applications that may be executed by the device. However, this type of scheme is sometimes circumvented by a user by either deleting the owner policy file or replacing the

Nintendo Exhibit 1003
Part 2 of 2
Page 001188

owner policy file with a user policy file which may include fewer restrictions than the owner policy file. Therefore, there remains a need for a system and method of owner application control of electronic devices.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] Fig. 1 is a block diagram showing a communication system in which electronic devices are used.

[0006] Fig. 2 is a block diagram illustrating a system of inserting owner information and owner control information onto an electronic device.

[0007] Fig. 3 is a block diagram of an electronic device in which a system and method of owner control are implemented.

[0008] Fig. 4 is a flow diagram illustrating a method of inserting owner information onto an electronic device.

[0009] Fig. 5 is a flow diagram illustrating a method of inserting owner control information onto an electronic device.

[0010] Fig. 6 is a flow diagram showing a method of owner control of an electronic device.

[0011] Fig. 7 is a block diagram of a wireless mobile communication device as an example of an electronic device.

[0012] Fig. 8 is a flow diagram illustrating a method of device initialization of required applications.

[0013] Fig. 9 depicts an exemplary user interface on a remote server for an owner to designate application control information for dissemination to particular devices, or groups of devices.

Nintendo Exhibit 1003
Part 2 of 2
Page 001189

DETAILED DESCRIPTION

[0014] Fig. 1 is a block diagram showing a communication system in which electronic devices are used. The communication system 10 includes a Wide Area Network (WAN) 12, coupled to a computer system 14, a wireless network gateway 16 and a Local Area Network (LAN) 18. The wireless network gateway 16 is also connected to a wireless communication network 20 in which a wireless mobile communication device 22 ("mobile device"), is configured to operate.

[0015] The computer system 14 is a desktop or laptop PC, which is configured to communicate to the WAN 12, the Internet for example. PCs, such as the computer system 14, normally access the Internet through an Internet Service Provider (ISP), Application Service Provider (ASP) or the like.

[0016] The LAN 18 is an example of a typical working environment, in which multiple computers 28 are connected in a network. It is normally located behind a security firewall 24. Within the LAN 18, a message server 26, operating on a computer behind the firewall 24, acts as the primary interface for the corporation to exchange messages both within the LAN 18, and with other external messaging clients via the WAN 12. Known message servers include, for example, Microsoft™ Exchange Server and Lotus Domino™. The LAN 18 includes multiple computer systems 28, each of which implements a messaging client, such as Microsoft Outlook14, Lotus Notes™, Yahoo! ™ Messenger, AOL Instant Messenger, or other client-server or peer-to-peer, or similar messaging clients with various architectures. Messages received by the message server 26 are distributed to mailboxes for user accounts addressed in the received messages, and are then accessed by a user through a messaging client operating on a computer system 28. The fact that

Nintendo Exhibit 1003
Part 2 of 2
Page 001190

the example given illustrates a client-server architecture in no way implies that such architecture is necessary, as other architectures may be used.

[0017] Although only a message server 26 is shown in the LAN 18, those skilled in the art will appreciate that a LAN may include other types of servers supporting resources that are shared between the networked computer systems 28, and that the message server 26 may also provide additional functionality, such as dynamic database storage for data such as, but not limited to, calendars, to-do lists, task lists, e-mail and documentation. The message server 26 and electronic messaging are described for illustrative purposes only. Owner control systems and methods are applicable to a wide range of electronic devices, and are in no way limited to electronic devices with messaging capabilities.

[0018] The wireless gateway 16 provides an interface to a wireless network 20, through which messages may be exchanged with a mobile device 22. Such functions as addressing of the mobile device 22, encoding or otherwise transforming messages for wireless transmission, and any other interface functions are performed by the wireless gateway 16. The wireless gateway 16 may be configured to operate with more than one wireless network 20, in which case the wireless gateway 16 also determines a most likely network for locating a given mobile device 22 and possibly track mobile devices as users roam between countries or networks.

[0019] The mobile device 22 is, for example, a data communication device, a voice communication device, a dual-mode communication device such as many modern cellular telephones having both data and voice communications functionality, a multiple-mode device capable of voice, data and other types of communications, a personal digital

Nintendo Exhibit 1003
Part 2 of 2
Page 001191

assistant (PDA) enabled for wireless communications, or a laptop or desktop computer system with a wireless modem.

[0020] Any computer system with access to the WAN 12 may exchange messages with the mobile device 22 through the wireless network gateway 16. Alternatively, private wireless network gateways such as wireless Virtual Private Network (VPN) routers could be implemented to provide a private interface to a wireless network. A wireless VPN router implemented in the LAN 18 provides a private interface from the LAN 18 to one or more mobile devices such as 22 through the wireless network 20. A private interface to a mobile device 22 may also effectively be extended to entities outside the LAN 18 by providing a message forwarding, or redirection system that operates with the message server 26. Such a message redirection system is disclosed in United States Patent No. 6,219,694, which is hereby incorporated into this application by reference. In this type of system, incoming messages received by the message server 26 and addressed to a user of a mobile device 22 are sent through the wireless network interface, either a wireless VPN router, the wireless gateway 16, or another interface, for example, to the wireless network 20 and to the user's mobile device 22. Another alternate interface to a user's mailbox on a message server 26 may be a Wireless Application Protocol (WAP) gateway. Through a WAP gateway, a list of messages in a user's mailbox on the message server 26, and possibly each message or a portion of each message, may be sent to the mobile device 22.

[0021] A wireless network 20 normally delivers messages to and from communication devices such as the mobile device 22 via RF transmissions between base stations and devices. The wireless network 20 may, for example, be a data-centric wireless network, a

voice-centric wireless network, or a dual-mode network that can support both voice and data communications over the same infrastructure. Recently developed networks include Code Division Multiple Access (CDMA) networks and General Packet Radio Service (GPRS) networks. So-called third-generation (3G) networks like Enhanced Data rates for Global Evolution (EDGE) and Universal Mobile Telecommunications Systems (UMTS) are currently under development. Older data- centric networks include, but are not limited to, the Mobitex™ Radio Network ("Mobitex"), and the DataTAC™ Radio Network ("DataTAC"). Voice-centric data networks such as Personal Communication System (PCS) networks, including Global System for Mobile Communications (GSM) and Time Division Multiple Access (TDMA) systems, have been available in North America and world-wide for several years.

[0022] In the system 10, a company that owns the LAN 18 may provide a computer system 28 and/or a mobile device 22 to an employee. When a computer system 28 issued to an employee is a laptop computer, for example, the computer system 28 may be used either within or outside the corporate LAN 18. When the computer system is operating within the LAN 18, non-local operations may be restricted by configuring permissions and restrictions for the computer system 28, a network account of the user, or both, in such a way that the permissions and restrictions are not configurable by the user. However, if a user is using a computer outside the LAN 18, by connecting the computer to the WAN 12 as shown at 14, for example, network-based controls in place at the LAN 18 can sometimes be bypassed.

[0023] In order to maintain control over an electronic device such as the computer system 28 or mobile device 22, an owner may establish local settings directly on the

Nintendo Exhibit 1003
Part 2 of 2
Page 001193

device. Such local settings control device operations, but only as long as the settings remain intact on the device. A common problem with this type of control mechanism is that local settings may be deleted, replaced, or otherwise altered by a user.

[0024] In some instances, the owner control information can include one or more applications lists. The application lists provide owner control of application installation and deletion on the electronic device. As an example, application lists can address different types of applications, such as but not limited to:

[0025] Required applications: These applications must be present on the device before the user may use the device; alternatively, one or more operations of the device can be disabled until such applications are available for execution on the device. This will allow the owner to install audit and remote administrative applications. The user can in some implementations be prevented from deleting these applications. This list may be small.

[0026] Allowable (or authorized) applications: These applications may or may not be present on a device. Thus the user is free to download these applications if they desire the functionality that the application provides. This list may be small.

[0027] Excluded applications: These applications may not be present on a device. Presumably an excluded application is a malicious application, or otherwise undesirable application. This list is potentially large.

[0028] With respect to allowed and/or required applications, even if the applications are required or allowed on the device, the owner may want to impose restriction on operations that such applications can perform. For instance, the owner may want to control functionality such as:

[0029] 1) Is the application allowed to open network connections inside the firewall (e.g., via MDS)?

[0030] 2) Is the application allowed to open network connections outside the firewall (e.g., via WAP, device TCP, SMS)?

[0031] 3) Is the application allowed to open local connections (e.g., serial, IR, or USB connections)?

[0032] 4) Is the application allowed to interact with other processes? Is the application allowed to access a Runtime Store or a Persistent Store?

[0033] 5) Is the application allowed access to a telephone API (e.g., to make phone calls)?

[0034] Fig. 2 is a block diagram illustrating a system of inserting owner information and owner control information onto an electronic device; such an insertion system may be used in one or more implementations of the described owner application control systems and methods. The foregoing explanation is therefore exemplary in nature. The system in Fig. 2 includes an electronic device 210, an owner information insertion point 220, and an owner control information insertion point 230. The owner information insertion point 220 is alternatively referred to as a branding point, while the owner control insertion point 230 is alternatively referred to as a control point. An owner information store 212, an owner control information store 214, and an interface/connector 216 are provided in the electronic device 210. The owner information insertion point 220 includes an owner information source 224 and an interface/connector 222. The owner control information insertion point 230 similarly includes an owner control information source 234 and an interface/connector 232.

Nintendo Exhibit 1003
Part 2 of 2
Page 001195

[0035] The owner information store 212 stores information, such as an owner name or other identification information, for example, which identifies an owner of the electronic device 210. The owner control information store 214 stores information that is used to control the operation of the electronic device 210. Owner control information may, for example, be specified in an authorization record that lists software applications authorized to be installed and executed on the electronic device 210; authorization records can further constrain operations performed by installed applications. The use of owner control information to control operations of an electronic device is described in further detail below. The owner information source 224 and the owner control information source 234 could be local memory devices, communication modules through which remote memory devices storing owner information and owner control information are accessible, or possibly user interfaces through which owner information and owner control information are entered.

[0036] The interface/connector 222 is compatible with the interface/connector 216 to establish a communication link between the owner information insertion point 220 and the electronic device 210, to thereby enable owner information to be transferred to the electronic device 210 from the owner information source 224. The interface/connector 232 similarly enables transfer of owner control information from the owner control information source 234 onto the electronic device 210 via a communication link established between the interface/connectors 232 and 216. The interface/connectors 216, 222, and 232 may establish wired communication links, where the interface/connectors are serial ports, for example, or wireless communication links such as infrared links where the interface/connectors are infrared modules. Owner information and owner control

Nintendo Exhibit 1003
Part 2 of 2
Page 001196

information transferred to a device are respectively inserted or stored in the owner information store 212 and the owner control information store 214.

[0037] The owner control insertion point 220 is associated with an owner of the electronic device 210. Where the electronic device 210 is provided to a user by an employer, for example, the owner control insertion point 220 may be a computer system or device controlled by a corporate computer system administrator or IT department. The electronic device 210 is "branded" with owner information by establishing a communication link between the owner information insertion point 220 and the electronic device 210 through the interface/connectors 222 and 216 and then inserting owner information into the owner information store 212. Unless otherwise desired, once owner information has been inserted onto the mobile device 210, then there can be a configuration such that only the owner or a party authorized by the owner is able to change the owner information or insert or change owner control information on the electronic device 210.

[0038] Because insertion of owner control information onto the electronic device 210 is restricted once owner information has been inserted, the owner control information insertion point 230 need not necessarily be controlled by the owner of the electronic device 210. When the owner maintains control over the owner control information insertion point 230, the insertion points 220 and 230 may be implemented in the same computer system or device and share the same interface/connector. However, separate insertion points 220 and 230 as shown in Fig. 2 allow an owner of the electronic device to delegate owner control information insertion to a trusted entity. If owner control information insertion is controlled using digital signatures, for example, as described in

Nintendo Exhibit 1003
Part 2 of 2
Page 001197

further detail below, an owner first brands the electronic device 210 and provides the electronic device 210 and digitally signed owner control information to a user. In this case, the owner control information insertion point 230 may be the user's computer system, which is then used to insert the digitally signed owner control information onto the electronic device 210.

[0039] In most implementations, the owner information insertion point 220 and the owner control information control point 230 include the same type of interface/connectors 222 and 232, compatible with the interface/connector 216 in the electronic device 210. However, the electronic device 210 may alternatively include multiple interface/connectors, such that different types of interface/connectors may be implemented at an owner information insertion point 220 and an owner control information insertion point 230. Although only a single owner control information insertion point 220 and owner control information insertion point 230 are shown in Fig. 2, a complete insertion system may include more than one of each type of insertion point. In a large company, for example, corporate computer system administrators may be authorized to perform owner information insertion operations from administrator computer systems, or from any corporate computer system from which administrative functions can be accessed, thereby providing multiple owner information insertion points 220. Similarly, when an owner allows users to insert digitally signed owner control information onto electronic devices, as described above, each user's computer system may be used as an owner control information insertion point 230.

[0040] The systems and methods of owner application control can use the insertion structures and methods described above; however, so long as owner control information

Nintendo Exhibit 1003
Part 2 of 2
Page 001198

store is capable of storing a required application list, and in some instances an allowed application list and/or an excluded application list, the particular control information insertion system and method can vary significantly, and use any conventional insertion/interfacing technology, without impacting the. owner application control systems and methods discussed herein.

[0041] Fig. 3 is a block diagram of an electronic device in which a system and method of owner application control can be implemented. In Fig. 3, the electronic device is a mobile device 30 adapted to operate within a wireless network. Also shown in Fig. 3 is an insertion tool 64 used to insert owner information onto the mobile device 30.

[0042] It should be apparent to those skilled in the art that only the components involved in an owner control system are shown in Fig. 3. A mobile device typically includes further components in addition to those shown in Fig. 3. Also, the mobile device 30 is an illustrative example of an electronic device for which an owner may wish to enforce some sort of usage policy. An owner may also wish to control the usage of other types of electronic devices, such as mobile telephones, laptop computers and PDAs, for example.

[0043] As shown in Fig. 3, a mobile device 30 comprises a memory 32, a processor 40, an application loader 42, an insertion module 44, a user interface (UI) 46, a wireless transceiver 48, and an interface/connector 50. The memory 32 can include a software applications store 34, an owner information store 36, an authorization record store 38, as well as possibly other data stores associated with other device systems in addition to those shown in Fig. 3.

Nintendo Exhibit 1003
Part 2 of 2
Page 001199

[0044] The memory 32 is a writable store such as a RAM or Flash memory into which other device components may write data. However, write and erase access to the software application store 34, the owner information store 36, and the authorization record store 38 may be restricted, but need not be in all implementations. For example, a user of the mobile device 30 may be able to retrieve data from the stores 34, 36, and 38, but write and erase operations for these stores are controlled, as described below. The software application store 34 includes software applications that have been installed on the mobile device 30, and may include, for example, an electronic messaging application, a personal information management (PIM) application, games, as well as other applications. The owner information store 36 stores information such as an owner name or other identification, data integrity and source authentication information, such as a digital signature public key associated with a digital signature private key of the owner. Owner control information, in which an owner of the mobile device 30 specifies usage permissions and restrictions for the mobile device 30, is stored in an authorization record in the authorization record store 38. Such authorization records can include one or more of the aforementioned required, allowed and/or excluded application lists, or more specific operation constraints for specific allowed and/or required applications. It should be appreciated that the supported lists need not be stored as unit; rather, the lists can logically be formed from authentication records associated with individual applications, wherein the each application authentication records includes a field that designates the application appropriately (e.g., allowed, required, excluded, etc.) and can include a field containing an application identifier.

Nintendo Exhibit 1003
Part 2 of 2
Page 001200

[0045] The processor 40 is connected to the wireless transceiver 48 and thus enables the mobile device 30 for communications via a wireless network. The application loader 42 and insertion module 44, described in further detail below, are connected to the interface/connector 50 to allow communication with the insertion tool 64, through the co-operating interface/connector 52.

[0046] The UI 46 includes one or more UI components, such as a keyboard or keypad, a display, or other components which accept inputs from or provide outputs to a user of the mobile device 30. Although shown as a single block in Fig. 3, it should be apparent that a mobile device 30 typically includes more than one UI, and the UI 46 is therefore intended to represent one or more user interfaces.

[0047] The insertion tool 64 includes an owner information store 60 and an interface/connector 52 through which information is exchanged with the mobile device 30, and thus represents an owner information insertion point 220 (Fig. 2). As described above, an owner information insertion point such as the insertion tool 64 is normally controlled by an owner of an electronic device. Therefore, the insertion tool 64 is, for example, implemented on an administrator computer system used by an authorized administrator to enable services for or otherwise configure the mobile device 30. Because networked computer systems can typically be used by any user, the insertion tool 64 may instead be accessible to any computer system in a corporate network, dependent upon the particular user that is currently "logged on" the computer system.

[0048] The owner information store 60 stores owner information to be inserted onto the mobile device 30, and may be implemented, for example, on a local memory component such as a RAM chip, a flash memory device, or a hard disk drive. When the

Nintendo Exhibit 1003
Part 2 of 2
Page 001201

insertion tool 64 is implemented in a networked computer system or other network-connected device, the owner information store 60 may be a remote memory system such as a file server that is accessible to the insertion tool 64 through a network connection. The owner information store 60 may instead incorporate a memory reader such as a smart card reader, a memory card reader, a floppy disk drive, or a CD or DVD drive, for example.

[0049] Information is transferred between the insertion tool 64 and the mobile device 30 via a communication link established between the interface/connectors 50 and 52. The interface/connectors 50 and 52 could be any of a plurality of compatible data transfer components, including, for example, optical data transfer interfaces such as Infrared Data Association (IrDA) ports, other short-range wireless communications interfaces, or wired interfaces such as serial or Universal Serial Bus (USB) ports and connections. Known short-range wireless communications interfaces include, for example, "Bluetooth" modules and 802.11 modules according to the Bluetooth or 802.11 specifications, respectively. It will be apparent to those skilled in the art that Bluetooth and 802.11 denote sets of specifications, available from the Institute of Electrical and Electronics Engineers (IEEE), relating to wireless LANs and wireless personal area networks, respectively. Therefore, a communication link between the insertion tool 64 and the mobile device 30 may be a wireless connection or a physical wired connection.

[0050] Because communications between the insertion tool 64 and the mobile device 30 need not necessarily be accomplished using a physical connection, references to connecting a mobile device to an insertion tool include establishing communications through either physical connections or wireless transfer schemes. Thus, the mobile

Nintendo Exhibit 1003
Part 2 of 2
Page 001202

device 30 could be connected to the insertion tool 64 by connecting serial ports on the mobile device 30 and the insertion tool 64, by positioning the mobile device 30 such that an optical port thereof is in a line of sight of a similar port of the insertion tool 64, or by connecting or arranging the mobile device 30 and the insertion tool 64 in some other manner so that data may be exchanged. The particular operations involved in establishing communications between a mobile device and an insertion tool are dependent upon the types of interfaces and/or connectors available in both the mobile device and the insertion tool.

[0051] Owner branding of the mobile device 30 may be facilitated by inserting owner information onto the mobile device 30 using the insertion tool 64 before the mobile device 30 is operable by a user. This may be accomplished, for example, by pre-loading owner information before the mobile device 30 is provided to the user by the owner, or before the mobile device 30 is configured for use. In the former example, the owner maintains physical control of the mobile device 30 until owner information has been loaded, whereas in the latter example, the user has possession of the mobile device 30 but is in this example unable to make use of the device until it is configured by, or at least under the control of, the owner.

[0052] Pre-loading of owner information onto the mobile device 30 can be performed using the insertion tool 64. The insertion tool 64 may be a computer system associated with an a owner system administrator, or a computer system which may be used by a mobile device user or administrator. Depending upon the owner information pre- loading scheme, the insertion tool 64 is operated by a mobile device user or an administrator.

Nintendo Exhibit 1003
Part 2 of 2
Page 001203

[0053] When the mobile device 30 has been connected to the insertion tool 64, owner information is retrieved from the owner information store 60 and transferred to the mobile device 30 through the interface/connectors 52 and 50, and passed to the insertion module 44 on the mobile device 30, which stores the owner information to the owner information store 36 in the memory 32.

[0054] Although the insertion module 44 is shown in Fig. 3 as being connected to the interface/connector 50, this module can be implemented as a software module or application that is executed by the processor 40. As such, data transfers to and from the interface/connector 50 may actually be accomplished by routing data through the processor 40 to the interface/connector 50. In this case, the processor 40 may be instructed by the insertion tool 64 to start the insertion module 44 before the owner information is transferred to the mobile device 30. Alternatively, the processor 40 may be configured to start the insertion module 44 whenever owner information is received. The insertion tool 64 may similarly be a software module or application that is executed by a processor (not shown) in a computer system or device on which the insertion tool 64 operates.

[0055] The owner information that is pre-loaded onto the mobile device 30 may include data integrity and/or source authentication information, such as a cryptographic system like a digital signature public key which corresponds to a digital signature private key used by the owner to digitally sign information before it is transferred to the mobile device 30. Pre-loading of the data integrity and/or source authentication information enables greater security of owner control operations, as described in further detail below

Nintendo Exhibit 1003
Part 2 of 2
Page 001204

in the context of digital signatures. Owner information may also include, for example, a name or other identifier associated with the owner of the mobile device 30.

[0056] In an owner control scheme in which digital signatures are used to verify data integrity and authenticate a source of data, when the owner's digital signature public key has been inserted into the owner information store 36 on the mobile device 30, owner control information, which specifies permissions and/or restrictions for the mobile device 30, is inserted onto the mobile device 30. Although an owner information insertion point, insertion tool 64, is shown in Fig. 3, it will be apparent from Fig. 2 and the above description that owner control information is usually inserted onto an electronic device after the device has been branded by inserting owner information onto the device. An owner control information insertion tool (not shown) configured for use with the mobile device 30 is similar to the insertion tool 64, including an owner control information store and an interface/connector compatible with the interface/connector 50. Owner control information is inserted onto the mobile device 30 and stored in the form of an authorization record in the authorization record store 38. In an authorization record, an owner of the mobile device 30 specifies a list of software applications that a user is authorized to install on the mobile device 30, as well as possibly a list of required software applications that must be installed on the mobile device 30.

[0057] In order to prevent a user from inserting false owner control information to thereby circumvent owner control, owner control information can be digitally signed using the owner's digital signature private key before being transferred to the mobile device 30. The insertion module 44 may be configured to verify the digital signature before the owner

Nintendo Exhibit 1003
Part 2 of 2
Page 001205

control information is stored on the mobile device 30. If digital signature verification fails, then the owner control information is not stored on the mobile device 30.

[0058] Digital signature schemes can involve some sort of transformation of digitally signed information to provide for checking the integrity of the information and authentication of a source of the signed information. For example, according to one known digital signature technique, a digest of information to be digitally signed is first generated using a non-reversible digest algorithm or transformation. Known digest algorithms include Secure Hashing Algorithm 1 (SHA-1) and Message-Digest algorithm 5 (MD5). Other digest techniques that produce a unique digest for each unique input may also be used. The digest is then further transformed using a digital signature private key and a signature algorithm to generate a digital signature. In digital signature verification, a digital signature public key corresponding to the private key is used.

[0059] In the context of owner control and owner control information, insertion of the owner's digital signature public key on a mobile device 30 as part of the owner information provides for digital signature-based security of owner control information. If some or all of the owner control information is digitally signed before transfer to the mobile device 30, then the insertion module 44 can verify that owner control information has actually been signed using the owner's digital signature private key, known only to the owner, and that the owner control information has not been changed since it was signed. In this example, only owner control information that originates with the owner of a mobile device 30 is stored to and used on the mobile device 30.

[0060] Owner control information is obtained by an owner control information insertion tool from an owner control information store, which may be a remote data store

Nintendo Exhibit 1003
Part 2 of 2
Page 001206

accessible to the insertion tool, a local store, or some form of memory reader, as described above. Owner control information is established based on a set of software applications or functions that the owner wishes to authorize on an electronic device, and may tend to change relatively infrequently once established. Such owner control information could then be digitally signed by a secure computer system or software component to which only administrators have access, using the owner's digital signature private key. In this case, signed owner control information is then stored at a location that is accessible to administrator computer systems and possibly other computer systems, and retrieved by an owner control information insertion tool as required. The owner control information insertion tool then transfers the signed owner control information to the mobile device 30. Depending upon how often owner control information changes or is expected to change, the signed owner control information may be further distributed to each computer system in a network in order to provide local access to signed owner control information. When new owner control information is generated and signed, the signed new owner control information can replace all existing copies of the owner control information, as described in further detail below. Wide distribution of owner control information provides easier access to the owner control information, whereas shared remote storage of owner control information requires fewer updates when new owner control information is established.

[0061] It is also possible to support digital signature generation for owner control information on an owner control information insertion tool. However, in the present example, this would require that the owner control information insertion tool has access to the owner's digital signature private key. Unless otherwise desired, digital signing of

Nintendo Exhibit 1003
Part 2 of 2
Page 001207

owner control information only by secure computer systems or components is generally preferred in that it limits the number of computer systems that can access the owner's digital signature private key.

[0062] When signed owner control information is transferred to the insertion module 44, digital signature verification operations are performed. If the digital signature is verified, then the owner control information is stored on the mobile device 30 in the authorization record store 38. Otherwise, the owner control information is not stored. In the event of a digital signature verification failure, an error or like indication may be output to a user on a UI 46 such as a display, an error message may be returned to the owner control information insertion tool, and an indication of the failure may also be output to a user of the owner control information insertion tool. When owner control information insertion fails, retry or other error processing operations may be performed on the owner control information insertion tool, the mobile device 30, or both.

[0063] Given the importance of the owner digital signature public key in the present example, at least a first owner information insertion operation for any mobile device 30 is preferably either performed or at least authorized by an administrator, in order to ensure that accurate owner control information is inserted onto the mobile device 30. This prevents a user from circumventing owner control by inserting a digital signature public key other than the owner's digital signature public key onto the mobile device 30.

[0064] When owner control information changes, where an owner wishes to expand or further restrict the use of an electronic device, for example, any existing owner control information may be replaced. As described above, new owner control information may be

Nintendo Exhibit 1003
Part 2 of 2
Page 001208

digitally signed, and the signed new owner control information is distributed to one or more locations from which it is retrieved for insertion onto electronic devices.

[0065] Any of several mechanisms for subsequent distribution of signed new owner control information to electronic devices are possible. When new owner control information is distributed to each owner control information insertion tool, the insertion tool may be configured to detect receipt of new owner control information, and to transfer the new owner control information to the mobile device 30 the next time the mobile device 30 is connected to the owner control information insertion tool. As described above, an owner control information insertion point 230 (Fig. 2), such as an owner control information insertion tool, may be controlled by a user of an electronic device. Many modern electronic devices are configured to be synchronized with computer systems. In such systems, this type of owner control information distribution may be supported by implementing an owner information control insertion tool in a user's computer system. New owner control information is then transferred to the electronic device the next time the electronic device is synchronized with the computer system.

[0066] Alternatively, signed new owner control information may be sent by an owner to all owned mobile devices through a wireless network, via the LAN 18, the WAN 12, and the wireless network gateway 16, as shown in Fig. 1, for example. Such signed owner control information could be sent to the owned mobile devices either directly or through one or more owner control information insertion tools. Although the owner's digital signature public key may be initially transferred to a mobile device 30 through the interface/connectors 52 and 50, other communication links which cannot be physically secured or protected, such as wireless or public communication network links, may be

Nintendo Exhibit 1003
Part 2 of 2
Page 001209

used to subsequently transfer signed owner control information to an electronic device that is enabled for communications over such other links. When the owner's digital signature public key has been inserted on a mobile device 30, the insertion module 44 is able to verify both the integrity and the source identity of any signed owner control information received, whether it is received via the interface/connector 50 or the wireless transceiver 48. In this type of implementation, for example, an owner control information insertion tool may include a different type of interface to the mobile device 30 than the owner information insertion tool 64.

[0067] Initial storage of owner control information, as well as replacement of existing owner control information, is in this example thereby dependent upon verification of a digital signature by the insertion module 44. Other checks may also be performed before existing information is replaced. In order to prevent replay attacks, in which old owner control information is received by the electronic device, owner control information can include version information. A configuration can include an existing owner control information being replaced only where received owner control information is newer than the existing owner control information. Generally, newer owner control information has a higher version number.

[0068] Although owner information is inserted onto the mobile device 30 using the insertion tool 64 as described above, changes to existing owner information, such as when the owner's digital signature private/public key pair is changed, may alternatively be updated on the mobile device 30 using digital signature techniques. To this end, the insertion tool 64 may include other types of communication modules (not shown), such as a wireless transceiver or network connector, for example, that are less secure than the

Nintendo Exhibit 1003
Part 2 of 2
Page 001210

interface/connector 52. In that case, any such updates are dependent upon verification of a digital signature using a digital signature public key in existing owner information.

[0069] The foregoing description relates primarily to writing owner information and owner control information to memory on an electronic device such as the mobile device 30. However, an owner may also wish to erase owner information and owner control information, without replacing existing information with new information. In this case, because information is not being written to memory on a device, no signed owner information or owner control information would be sent to the device. Instead, an erase command or request may be sent to the device. Erasure may be a further function supported by the insertion module 44.

[0070] Referring again to Fig. 3, if owner information is to be erased from the owner information store 36, then an erase command or request is digitally signed and sent to the insertion module 44. As with new owner information or owner control information, a signed command or request could be sent to the mobile device 30 through either the interface/connector 50 or the wireless transceiver 48. The insertion module 44, using the owner's digital signature public key, executes the command or completes the request if a digital signature is verified. Otherwise, the command or request may be ignored, and an error or failure indication may be displayed to a user on a UI 46 on the mobile device 30, returned to a sending system or device that sent the command or request, or both. Further error or failure processing routines may then be performed at the sending system or device.

[0071] Since owner information includes the owner's digital signature public key in a signature-based owner control scheme, erasure of owner information can be tightly

Nintendo Exhibit 1003
Part 2 of 2
Page 001211

controlled. For example, only owner system administrators may be authorized to send erase commands or requests. Sending of signed commands or requests to the mobile device 30 can be restricted to administrator computer systems or accounts, an owner information insertion tool, or an owner-controlled erasure tool. For example, an insertion tool such as the insertion tool 64 could be adapted to erase existing owner information from the mobile device 30 by providing an erase command generator or store which is also coupled to the interface/connector 52. Alternatively, owner information erasure could be accomplished using a specialized, owner-controlled erasure tool incorporating such an erase command generator or store and an interface to the mobile device 30. Erasure of owner control information can be controlled in a similar manner.

[0072] Where an owner control system is configured to support erasure and possibly other owner information and owner control information management functions, access to the owner's digital signature private key may be restricted in order to control the information, requests, and commands that can be digitally signed and sent to an electronic device. The digital signature private key or digital signature generation functions may be accessible only to specific computer systems or administrator login accounts, for example.

[0073] As shown in Fig. 3, other systems on the mobile device 30 can have access to the memory 32. Configurations may be used wherein no device system is able to insert, change, or erase owner information or owner control information without submitting properly signed information or commands. Any data stores, such as the owner information store 36 and the authorization record store 38, that store owner information or owner control information can therefore be located in protected memory areas.

Configuration may be used where only the insertion module 44 has write and erase access to these stores, such that digital signature-based control of insertion and erasure of owner information and owner control information are maintained. Other device systems have read only access to owner information and owner control information. In one possible implementation, any systems or components through which the memory 32 is accessible are configured to allow memory read operations from any locations in the memory 32, but deny any write or erase operations to memory locations storing owner information or owner control information unless the operations originate with or are authorized by the insertion module 44. In an alternative implementation, a memory manager (not shown) is provided to manage all memory access operations. Such a memory manager is configured to direct any write or erase operations involving owner information or owner control information stores to the insertion module 44 for digital signature checking and authorization before completing the operations. Owner information and owner control information may thereby be read by other device systems, but preferably may only be inserted, changed, or erased when a digital signature is verified.

[0074] It should be appreciated that the above public key digital signature operations are intended only as an illustrative example. Other digital signature schemes, or other data integrity checking and source authentication schemes, may instead be used to verify the integrity and source of owner control information or commands. Further, the authentication and security described herein above can be used to transfer the owner application control information; however, various systems and methods of owner

Nintendo Exhibit 1003
Part 2 of 2
Page 001213

application control need not use authentication and/or secure transmission in order to achieve the desired owner application control as described herein.

[0075] In the mobile device 30, owner control information is included in an authorization record that is stored in the authorization record store 38. An authorization record specifies particular software applications that are authorized for installation on the mobile device 30, and may also specify required software applications that must be installed on the mobile device 30. Such an authorization record provides an electronic device owner with relatively tight control of how a user makes use of the mobile device 30, since only authorized software applications can be loaded onto the device.

[0076] For authorized and/or required applications, some systems can provide a more fine grained control within the authorization record(s). In such systems, the owner can provide more specific controls on the operations that installed application can perform. Such controls can be specified on an individual application basis, or in some cases by groups of applications. Such operation controls can determine whether an application can connect to external resources, and if so, the channels (that may be used for such connections) can communicate with other applications executing on the device and/or can access part or all of local memory on the device.

[0077] Software application loading operations are enabled on the mobile device 30 by the application loader 42. As described above in regard to the insertion module 44, although the application loader 42 is shown as being connected to the interface/connector 50, information may actually be exchanged between the application loader 42 and the interface/connector 50 or the wireless transceiver 48 through the processor 40.

Nintendo Exhibit 1003
Part 2 of 2
Page 001214

[0078] Like owner information and owner control information, software applications may be received by the mobile device 30 via the interface/connector 50 or the wireless transceiver 48. One possible source of software applications configured for operation on the mobile device 30 is a user's computer system equipped with an interface/connector compatible with the interface/connector 50. When the computer system is connected to a corporate LAN, for example, software applications provided by a corporate owner of the mobile device 30 may be retrieved from a file server on the LAN or other store on the LAN, and transferred to the mobile device. A computer system may also or instead obtain software applications for the mobile device 30 from a local store, or other sources, such as Internet-based sources, with which the computer system may communicate.

[0079] The application loader 42 may be configured to determine whether owner control information is stored on the mobile device 30 whenever a software application is received. If no owner control information is present on the mobile device 30, then no owner controls have been established for the mobile device 30, and the software application is installed. Alternatively, the application loader 42 could consult a remote server for an owner control information update prior to attempting the installation. Software application installation typically involves such operations as storing a received application file to the software application store 34 in the memory 32, extracting files for storage to the software application store 34, or possibly executing an installation program or utility. If owner control information is subsequently inserted onto the mobile device 30, existing software applications may be checked by either the application loader 42 or the insertion module 44 to ensure that all software applications resident on the mobile device

Nintendo Exhibit 1003
Part 2 of 2
Page 001215

30 are authorized software applications. Any software applications that have not been authorized are erased from the mobile device 30 or otherwise rendered inoperable.

[0080] In some circumstances, owner information may have been inserted onto an electronic device, but owner control information has yet to be inserted. In order to prevent loading of a software application onto the mobile device 30 that subsequently inserted owner control information does not authorize, the mobile device 30 may be substantially disabled, permitting only a limited subset of device functions to be executed, until owner control information is inserted. Alternatively, the application loader 42 may be configured to determine whether owner information is present on the mobile device 30 when a software application is received. Where owner information is found, indicating that owner control information will be established and used for the mobile device 30, the application loader 42 then determines whether owner control information has been inserted. In the event that owner information but not owner control information is found, the application loader 42 does not load the received software application. Error processing operations may then be performed, such as purging the received software application from any temporary memory location in which it was stored when received, and, if memory resources on the mobile device 30 permit, storing the received software application on the mobile device 30 in such a way that it is not executable. Any software applications stored in this manner are then processed by the application loader 42 when owner control information is inserted onto the mobile device 30. Although software applications are stored on the mobile device 30 in this example, they would not be usable until owner control information is inserted onto the mobile device 30, and it is confirmed that the software applications are authorized for installation. The amount of memory space made

Nintendo Exhibit 1003
Part 2 of 2
Page 001216

available for such software applications may occupy can be limited, so that available memory space will not be depleted by storing unchecked and possibly unauthorized software applications.

[0081] When the application loader 42 determines that owner control information has been inserted onto the mobile device 30, the application loader 42 then determines whether the received software application is authorized for installation on the mobile device 30. If the owner control information includes an authorized software application list, the application loader 42 searches the list to determine whether the received software application is one of the authorized software applications. Alternatively, an authorized (allowed) software application list residing on a remote or external device (e.g., remote computer system, external card or memory device, etc.) can be consulted to determine whether a particular application is authorized for installation. In some such cases, the approval response from the remote or external device can include the application for installation, or information from which a source for the to-be installed application can be obtained; upon receipt, the device can download and/or install the application based upon the received approval response.

[0082] An authorized software application list can include information that uniquely identifies the authorized software applications, such as a hash of the software application source code or executable code, for example. Because a software application developer is free to choose a file name for any software application, file names may not provide a reliable authorization check. However, if an owner generates a hash of each authorized software application and includes the hash in the owner control information that is inserted onto the mobile device 30, then only particular versions of authorized software

Nintendo Exhibit 1003
Part 2 of 2
Page 001217

applications can be installed on the mobile device 30. The application loader 42 generates a hash of any received software application, and installs the software application only if the generated hash matches a hash in the owner control information. In order to support different hashing algorithms on different electronic devices, a device owner generates more than one hash of each software application and includes each hash in the owner control information inserted onto each owned electronic device. An electronic device may then use any of a number of different hashing algorithms to generate a hash of a received software application. Other unique transformations than hashes could also be used to generate owner control information and to determine whether received software applications are authorized for installation.

[0083] In some instances, prior to checking the authorized application list, at periodic intervals or upon a remote authorization change, the device can receive an authorized application list, or an authorized application list update, from a remote server or external device controlled by the device owner. The list or list update can be received in response to a request by the device (e.g., request a list or update upon installation attempt) or without such a request (e.g., responsive to an authorization modification by the owner on a remote owner administration server). Upon receipt of an authorized list, the device can install the list overwriting any prior list; upon receiving an update, the update is processed and integrated into an existing list, or used to create a list if none was present prior. In some instances the secure insertion tools, and/or other encryption/authentication, approaches as described herein can be used to provide the authorized application list to the device.

[0084] Owner control information may also include a required software application list that uniquely identifies software applications that the owner of an electronic device establishes as mandatory; alternatively, such a required software application list could reside on a remote or external device (e.g., remote computer system, external card or memory device, etc.) that can be consulted at need. A required software application list allows an owner to ensure that every owned electronic device supports certain core functions, such as electronic messaging and secure communications, for example.

[0085] Software applications in a required software application list may be uniquely identified by one or more hashes, as described above in the context of authorized applications. The processor 40, application loader 42, insertion module 44, or a further device component or system is configured to periodically check to ensure that each required software application is present on the mobile device 30, and that a hash of each required software application matches a hash in the required software application list. In addition, or instead, at power up or other initialization of the device, presence of required applications can be checked. Where a required software application is not present on the device or its hash does not match a hash in the required software application list, which would occur when a software application has been changed, the mobile device 30, or at least some of its functions, can be rendered unusable. Alternatively, the mobile device 30 can download and install missing or corrupted applications transparently to the user of the device; after successful installation of all required programs, the device is restored to operability.

[0086] In some instances, device initialization may include use of the required software application list. Such a process is shown in Fig. 8. A determination is made as to

Nintendo Exhibit 1003
Part 2 of 2
Page 001219

whether required applications are available on the device in step 800. The device examines installed applications to determine if applications on the required software application list are available on the device. This can occur through examination of the required software application list residing in the owner control information store. Alternatively, identification information associated with installed applications can be transmitted to a remote server managed by the owner that performs the comparison and returns the results of such a comparison to the device. If required applications are missing the device is disabled in part, or in whole, in step 810. The device may transparently initiate download of required applications that were determined to be unavailable 820. In implementations using a remote server to perform the comparison, some such implementations may allow the remote server to directly return any missing applications to the device. When all required applications are present the device operates normally 840.

[0087] Prior to checking for the presence of required applications, at periodic intervals or upon a remote authorization change, the device can receive a required list, or a required list update, from a remote server or external device controlled by the device owner. The list or list update can be received in response to a request by the device (e.g., request a list or update at initialization) or without such a request (e.g., responsive to an authorization modification by the owner on a remote owner administration server). Upon receipt of a required list, the device can install the list overwriting any prior list; upon receiving an update, the update is processed and integrated into an existing list, or used to create a list if none was present prior. In some instances the secure insertion tools,

Nintendo Exhibit 1003
Part 2 of 2
Page 001220

and/or other encryption/authentication, approaches as described herein can be used to provide the required application list to the device.

[0088] In order to provide further control over required software applications, erasure or other operations involving such applications are controlled. Digital signature- based control of such functions is implemented by requiring a digital signature on any erase or write command that affects a required software application. When an erase or write command is received from a system on the mobile device 30 or from a remote system via the interface/connector 50 or wireless transceiver 48, the processor 40 or another device system such as a memory manager (not shown) determines whether the command involves the software application store 34. Such a write or erase command is not executed unless a digital signature is verified using the owner's digital signature public key stored on the mobile device 30. Although software applications may be executed by device systems without requiring digital signatures, required software applications, if so desired, may only be changed or erased when a digital signature is verified. As above, digital signatures represent one possible data integrity and source authentication mechanism.

[0089] Owner control information may also include an excluded software application list that uniquely identifies software applications that the owner of an electronic device establishes cannot be installed on the device. An excluded software application list allows an owner to ensure that every owned electronic device does not contain particular malicious and/or counter productive software applications. Software applications in an excluded software application list may be uniquely identified by one or more hashes, as described above in the context of authorized applications. The processor 40, application

Nintendo Exhibit 1003
Part 2 of 2
Page 001221

loader 42, insertion module 44, or a further device component or system is configured to periodically check to ensure that no excluded software application is present on the mobile device 30, and that a hash of each present software application does not match a hash in the excluded software application list. Where an excluded software application is present on the device or its hash does match a hash in the excluded software application list, which would occur when a software application has been changed, the mobile device 30, or at least some of its functions, can be rendered unusable. Alternatively, the mobile device 30 can delete an excluded application found present on the device transparently to the user of the device; after successful deletion, the device is restored to operability.

[0090] In many cases, the excluded application list can be maintained remotely on a remote server or on an external memory device rather than in a memory area local to the device. In such instances, application loader can transmit a request to the remote server or search the external memory device (e.g., memory card, network attached disk, etc.). In such cases, the remote server consults the excluded list, or the device searches the external memory device, to determine whether the to-be installed application has been designated as excluded. In the case of a remote server, the remote server would transmit to the device either an approval or a denial as appropriate. The device could determine approval or denial directly from its consultation of an external memory device.

[0091] When an application installation request is received by a device supporting an excluded application list, the excluded application list is consulted based upon the application to-be installed. If the application to-be installed is found on the excluded application list, the installation request is denied and the application is not installed on the device. In machines supporting an allowed list and/or a required list in addition to an

Nintendo Exhibit 1003
Part 2 of 2
Page 001222

excluded list, a list priority could be established to determine how the device handles installation requests for applications that are on multiple lists. For instance, if a particular application appears on both the required list and the excluded list, a conflict exists. A priority scheme can be used to resolve such conflicts. In one such scheme, if an application is on the required list, its presence on other lists is not considered; if an application is on the authorized list and the excluded lists, then the application is considered excluded as the more conservative approach.

[0092] In instances where allowed, required and/or excluded lists are supported, some or all these lists can be maintained locally within the devices owner control information store; alternatively, one or more supported lists could be maintained remotely or on an external memory device.

[0093] Fig. 4 is a flow diagram illustrating a method of inserting owner information onto an electronic device; this method may be used in connection with inserting the owner application control information onto the electronic device. The method in Fig. 4 begins at step 72, when an electronic device owner establishes owner information. This involves such operations as selecting an owner name or identifier and generating or obtaining an owner digital signature private/public key pair, for example. The owner information is then digitally signed and sent to the electronic device at step 74.

[0094] At step 76, a determination is made as to whether owner information already exists on the electronic device, by checking an owner information store, for example. When owner information does not exist on the electronic device, such as for an initial insertion of owner information, the owner information is inserted onto the electronic device at step 84, by storing the owner information to a memory on the electronic device. When

Nintendo Exhibit 1003
Part 2 of 2
Page 001223

the owner information is initially being inserted onto the electronic device, it need not necessarily be digitally signed. As described above, initial owner information insertion may be performed directly by or at least under the authorization of the owner or an owner system administrator.

[0095] A digital signature associated with the owner information is checked at step 78 where owner information already exists on the electronic device. If the digital signature is not verified, as determined at step 80, the owner information cannot be inserted onto the electronic device, and error processing is invoked at step 82. As described above, error processing may include such operations as indicating an error or failure on a UI of the electronic device and sending an error or failure message to an insertion tool or system from which the owner information was sent. The owner information is inserted onto the electronic device at step 84 where the digital signature was verified.

[0096] Once owner information has been inserted onto an electronic device, owner control information is inserted onto the electronic device to set up owner controls. Fig. 5 is a flow diagram illustrating a method of inserting owner control information onto an electronic device.

[0097] At step 92, owner control information is established, based on how an owner wishes to control an electronic device. Owner control information, as described above, may include an authorized software application list and a required software application list, for example. The owner control information is then signed and sent to the electronic device at step 94. The digital signature on the owner control information is then checked at step 96. At step 98, it is determined whether the digital signature is verified. Error processing, which may involve operations similar to those described above in conjunction

Nintendo Exhibit 1003
Part 2 of 2
Page 001224

with step 82 in Fig. 4, is performed at step 100. If owner information including the owner's digital signature public key has not been previously inserted onto the electronic device, or the owner control information was not signed using the digital signature private key corresponding to the owner digital signature public key inserted onto the electronic device, then the digital signature is not verified at step 98.

[0098] When the digital signature is verified at step 98, it is then determined at step 101 whether the received owner control information is current, such as by determining whether a version number of the received owner control information is greater than the version number of existing owner control information. The owner control information is inserted onto the electronic device at step 102 when the digital signature was verified and the received owner control information is current, by storing the information to an appropriate data store on the electronic device, for example. Otherwise, error processing is performed at step 100.

[0099] Other operations may also be dependent upon verification of digital signatures. For example, commands or requests to write data to or erase data from an owner information store, an owner control information store, or a software application store may be similarly processed to verify associated digital signatures before the commands or requests are completed.

[00100] The owner control information, such as the software application lists and the application operation restrictions described above, can be maintained on a remote server managed by the device owner. The remote server can maintain a device data store, that may be in the form of a database, that stores the control information, including application control information, associated with owned devices.

Nintendo Exhibit 1003
Part 2 of 2
Page 001225

[00101] For each device, or for groups of devices, particular application lists (e.g., required, authorized and/or excluded) and allowable operations for particular applications can be created, modified, stored and distributed. In some implementations, a graphical user interface can be provided through which the owner can specify the particular control information associated with a device or device group.

[00102] Owner control information regarding particular applications can be provided to the electronic device from a remote source (e.g., remote computer system server, local memory device such as memory card or disk, etc.) via a wired (or other direct connection based upon physical contact of the device with the source) or a wireless (e.g., IR, 802.11, Bluetooth, etc.) communication channel. In such cases, the policy information including installation constraints and/or application operation constraints can be provided in a predefined format. The received policy information can then be used to create, update and/or delete authentication records.

[00103] In one implementation, the following format can be used to encode the owner control information regarding an application:

<encoding> = <version><connection set>*<cod file data>*

<version> is a byte. The current version is 0.

<connection set>=a UTF8 string of comma separated domains

<cod file data> = <hash><flags><internal connections set index (a byte)><external connections set index (a byte)>

[00104] The download default is specified by a cod file data with a hash of all zeros.

<hash> is a 20 byte SHA1 hash of the cod file

<flags> is a 32 bit int.

Nintendo Exhibit 1003
Part 2 of 2
Page 001226

Required App = 1

Excluded App = 2

Inter-Process Communication Allowed = 4

Internal Network Connections Allowed = 8

External Connections Allowed = 16

Local Connections Allowed = 32

[00105] The policy information associated with specific applications can be stored either on a remote server or memory device for delivery to the electronic device. Such storage can in some instances conform to the above described format. The policy information can be for an application can be associated with a particular electronic device or a particular group of electronic devices. Alternatively, some or all of such policy information can be stored remotely and queried upon request from the device.

[00106] Fig. 9 depicts one possible user interface provided via a remote server for an owner to configure application policy information for particular applications. In this particular example interface, particular groups of electronic devices by device type have particular owner control information associated therewith for particular applications. An information technology manager for the device owner can control the policy for a given set of devices by changing the provided configuration information. Such changes could then be transmitted to individual devices. In the depicted example, the application list is being used in the context of constructing a "target configuration" in which the handheld will be required to have the browser and security applications, as well as the phone application, but the memo pad and tasks applications are optional (allowed but not

required). A user can select a row as shown at 900 in order to modify one or more of the values associated with an entry.

[00107] A remote server can be used to store owner control information associated with one or more electronic devices, or groups of devices. The remote server can communicate with the one or more devices via any suitable communication channel (e.g. wired or wireless connection). The remote server can use owner control information insertion tools as described herein. The owner control information on the remote server can be managed in a variety of ways including through provision of a management user interface such as the one depicted in Fig. 9. The management user interface can be provided directly by the remote server, or alternatively be provided by a computer system that communicates with the remote server.

[00108] As has been discussed previously, the required, allowed and/or excluded application lists can be either locally or remotely (including an external memory device) queried, depending upon particular implementation. In either case, the supported lists can be maintained by a remote server that the owner could control. In the case of an external memory device used for list consultation, the memory device could be a shared access device (e.g., network disk) accessible by the device and an owner management system, or the memory device could be a removable media or memory card that is temporarily connected to either the remote server, or a separate management system in communication with the remote server. In the latter case, list information would be stored on the memory device for later insertion or access by the owner controlled device.

[00109] If lists are locally maintained, the device may periodically or upon occurrence of specific events (e.g., initialization, installation request, etc.) query the remote server for a

Nintendo Exhibit 1003
Part 2 of 2
Page 001228

list or list update. Upon receipt of a request by the remote server, the server would determine the list appropriate for the requested device based upon device type and/or an individual device identifier. The remote server would then transmit the determined list or list update to the requesting device. A server side change to the list could trigger an unsolicited push of the list to the device. In which case, the server would determine impacted devices based upon the server side change and transmit the list or list update to the impacted devices.

[00110] Owner control information is then used to control the electronic device. Fig. 6 is a flow diagram showing a method of owner control of an electronic device. At step 110, an operation request is received at the electronic device. Operation requests include, for example, receipt of a software application for installation, a function call from a software application executing on the electronic device, an attempt by a user, software application, or a system on the electronic device to perform an operation, and the like. Such requests may originate with a user, a software application, a device system, or possibly a remote system or device. If owner information does not exist on the electronic device, as determined at step 112, then owner controls have not been established and the operation is performed at step 122. In the example of a received software application, step 122 involves installation of the software application on the electronic device.

[00111] When owner information exists, it is determined at step 114 whether owner control information exists. Error processing operations are performed at step 116 if owner information, but not owner control information, exists. As described above, determining whether owner information exists at step 112, and then reverting to error processing at step 116 where it is determined at step 114 that owner control information does not exist

Nintendo Exhibit 1003
Part 2 of 2
Page 001229

prevents certain operations, such as software application loading and installation, when an owner information has been inserted onto an electronic device, but owner control information has not yet been inserted. Step 116 may include such operations as presenting an error message to a user of the electronic device and returning an error indication to a source from which the operation request was received. Alternatively, a default action in response to a negative determination at step 114 could be to revert to step 122, when an owner does not wish to restrict device operations before owner control information is inserted.

[00112] When both owner information and owner control information have been inserted onto an electronic device, it is determined at step 118 whether the operation is permitted. In the case of a received software application, step 118 involves determining whether software application installation is permitted, and possibly whether the software application is an authorized software application. In the case of authorized applications, the requested operation could include, for example, an application requesting opening a connection (e.g., network connection - MDS, WAP, SMS, TCP, etc. or local - USB, serial, etc.), accessing the telephone API, accessing local memory or communicating with other applications executing on the device. The operation is performed at step 122 where the operation is permitted. Otherwise, error processing is performed at step 120. As described above, owner control information may include not only permissions and restrictions for electronic device operations and software applications, but also a list of required software applications or modules which may be checked from time to time to ensure that all required software applications are present on an electronic device. For example, an electronic device may be configured to check for required software

Nintendo Exhibit 1003
Part 2 of 2
Page 001230

applications at step 118 when certain types of operation requests are received, and to perform the operation at step 122 only when all required software applications are found.

[00113] It will be appreciated that the above description relates to the invention by way of example only. Many variations on the systems and methods described above will occur to those knowledgeable in the field, and such variations are within the scope of this application, whether or not expressly described. As an example, a system of owner application control of an electronic device can comprise an owner control information store configured to store owner control information for controlling operation of the electronic device, and more specifically installation and/or deletion of applications on the electronic device and/or restrictions on operations by applications once installed. The owner control information can comprise a list of one or more required applications. In some instances, the owner control information can further comprise a list of one or more allowed applications and/or a list of one or more excluded applications. The owner control information store can reside on the controlled device or on one or more remote servers; alternatively, portions of the owner control information store can be distributed between the controlled device and one or more remote servers. In addition to, or instead of, the required list, the allowed list and/or the excluded list, the owner control information can comprise one or more constraints that control operations that can be performed by an application or a group of applications.

[00114] A method of owner application control may include variety of steps that in some implementations can be in the form of computer executable instructions stored on one or more computer readable media and/or be distributed via one or more data signals. Such a method can includes verifying that an application in a required list in an owner control

Nintendo Exhibit 1003
Part 2 of 2
Page 001231

information store is available for execution on the device, and if the application in the required list is not verified as available, initiating download and installation of such application from an external application source such as a computer system or memory device. The electronic device can in some instances be disabled from operating normally until applications on the required list are available for execution on the device. More specifically, at least one operation of the device can be disabled until the required application is available for execution.

[00115] As another example, a method may in some instances include steps of storing owner information having a required list having data integrity and/or source authentication information on an electronic device, receiving owner control information at the electronic device, and checking integrity of the received owner control information and/or determining whether the authorized source is authentic.

[00116] In addition to, or instead of, the steps provided above, a device can perform the steps of receiving an operation request from an application and determine whether the application is allowed to perform the requested operation based upon owner control information associated with the application. The requested operation is only executed if the requested operation is determined to be allowed for the application.

[00117] As yet another example, a system can be configured for owner application installation control of an electronic device. An owner control information store can be configured to store owner control information having a required list identifying one or more applications required for full operation of the electronic device. Initialization processor instructions (e.g., as implemented in a software module, etc.) can be used to consult (e.g., directly or indirectly have access to) owner control information in the owner control

information store in order to verify if an application in the required list is available for execution on the electronic device. An application loader module that is invoked by the initialization instructions when the application in the required list is not available for execution and that downloads the application from an external application source and installs it on the electronic device. Operation control instructions can be used to disable at least one operation of the electronic device until the application loader module completes installation of the application.

[00118] A method may also be configured for designating owner contral of application operations for an electronic device by receiving an operation indication of one or more operations associated with a particular application; generating an authorization record based upon the received operation indication and the particular application; storing the generated authorization record; receiving a device indication of one or more electronic devices subject to owner control; receiving a correspondence indication that associates the received device indication with one or more stored authorization records; and communicating one or more stored authorization records to one or more electronic devices based upon the received device indication and the received correspondence indication.

[00119] Still further as another example, owner information and owner control information operations may be secured by other means than digital signatures. Instead of checking digital signatures on owner information, owner control information, and restricted commands or requests, an electronic device might issue a cryptographic challenge using a previously inserted encryption key associated with the owner. The encryption key could be a public key of the owner or a secret key shared between the owner and the electronic

Nintendo Exhibit 1003
Part 2 of 2
Page 001233

device. Operations such as owner information or owner control information insertion or erasure would then be performed only when a valid challenge response is returned. A valid challenge response may only be generated using a corresponding encryption key. Data integrity and source authentication could instead be assumed, for example, where owner information and owner control information are sent to an electronic device over a secure channel. If the device properly decrypts information received via the secure channel, then it is assumed that the information is valid and was sent by an authorized source. In this latter scheme, the source and device share a public/private key pair, or a common symmetric key.

[00120] In some instances, owner control information such as owner application control information can reside on a remote server rather than on the electronic device. For instance, one or more of a required, authorized and/or excluded application list can reside on a remote server. In such instances, an operation request such as application installation or device initialization can generate a query to the remote server where such lists reside. The proper list can be consulted and an appropriate response returned to the inquiring device.

[00121] In addition, an electronic device in which systems and methods described above are implemented may include fewer, further, or additional components than shown in Figs. 2 and 3. Fig. 7 is a block diagram of a wireless mobile communication device as an example of such an electronic device. However, it should be understood that the systems and methods disclosed herein may be used with many different types of devices, such as personal digital assistants (PDAs) and desktop computers.

Nintendo Exhibit 1003
Part 2 of 2
Page 001234

[00122] As shown in Fig. 7, mobile device 500 is preferably a two-way communication device having at least voice and data communication capabilities. The mobile device 500 preferably has the capability to communicate with other computer systems on the Internet. Depending on the functionality provided by the mobile device, the mobile device may be referred to as a data messaging device, a two-way pager, a cellular telephone with data messaging capabilities, a wireless Internet appliance, or a data communication device (with or without telephony capabilities).

[00123] The mobile device 500 includes a transceiver 511, a microprocessor 538, a display 522, non-volatile memory 524, random access memory (RAM) 526, auxiliary input/output (I/O) devices 528, a serial port 530, a keyboard 532, a speaker 534, a microphone 536, a short-range wireless communications sub-system 540, and may also include other device sub-systems 542. The transceiver 511 preferably includes transmit and receive antennas 516, 518, a receiver (Rx) 512, a transmitter (Tx) 514, one or more local oscillators (LOs) 513, and a digital signal processor (DSP) 520. Within the non-volatile memory 524, the mobile device 500 includes a plurality of software modules 524A524N that can be executed by the microprocessor 538 (and/or the DSP 520), including a voice communication module 524A, a data communication module 524B, and a plurality of other operational modules 524N for carrying out a plurality of other functions.

[00124] The mobile device 500 is preferably a two-way communication device having voice and data communication capabilities. Thus, for example, the mobile device 500 may communicate over a voice network, such as any of the analog or digital cellular networks, and may also communicate over a data network. The voice and data networks are depicted in Fig. 7 by the communication tower 519. These voice and data networks

Nintendo Exhibit 1003
Part 2 of 2
Page 001235

may be separate communication networks using separate infrastructure, such as base stations, network controllers, etc., or they may be integrated into a single wireless network. References to the network 519 should therefore be interpreted as encompassing both a single voice and data network and separate networks.

[00125] The communication subsystem 511 is used to communicate with the network 519. The DSP 520 is used to send and receive communication signals to and from the transmitter 514 and receiver 512, and also exchange control information with the transmitter 514 and receiver 512. If the voice and data communications occur at a single frequency, or closely-spaced set of frequencies, then a single LO 513 may be used in conjunction with the transmitter 514 and receiver 512. Alternatively, if different frequencies are utilized for voice communications versus data communications or the mobile device 500 is enabled for communications on more than one network 519, then a plurality of Los 513 can be used to generate frequencies corresponding to those used in the network 519. Although two antennas 516, 518 are depicted in Fig. 7, the mobile device 500 could be used with a single antenna structure. Information, which includes both voice and data information, is communicated to and from the communication module 511 via a link between the DSP 520 and the microprocessor 538.

[00126] The detailed design of the communication subsystem 511, such as frequency band, component selection, power level, etc., is dependent upon the communication network 519 in which the mobile device 500 is intended to operate. For example, a mobile device 500 intended to operate in a North American market may include a communication subsystem 511 designed to operate with the Mobitex or DataTAC mobile data communication networks and also designed to operate with any of a variety of voice

Nintendo Exhibit 1003
Part 2 of 2
Page 001236

communication networks, such as AMPS, TDMA, CDMA, PCS, etc., whereas a mobile device 500 intended for use in Europe may be configured to operate with the GPRS data communication network and the GSM voice communication network. Other types of data and voice networks, both separate and integrated, may also be utilized with the mobile device 500.

[00127] Communication network access requirements for the mobile device 500 also vary depending upon the type of network 519. For example, in the Mobitex and DataTAC data networks, mobile devices are registered on the network using a unique identification number associated with each device. In GPRS data networks, however, network access is associated with a subscriber or user of the mobile device 500. A GPRS device typically requires a subscriber identity module ("SIM"), which is required in order to operate the mobile device 500 on a GPRS network. Local or non-network communication functions (if any) may be operable, without the SIM, but the mobile device 500 is unable to carry out functions involving communications over the network 519, other than any legally required operations, such as '911' emergency calling.

[00128] After any required network registration or activation procedures have been completed, the mobile device 500 is able to send and receive communication signals, preferably including both voice and data signals, over the network 519. Signals received by the antenna 516 from the communication network 519 are routed to the receiver 512, which provides for signal amplification, frequency down conversion, filtering, channel selection, etc., and may also provide analog to digital conversion. Analog to digital conversion of the received signal allows more complex communication functions, such as digital demodulation and decoding, to be performed using the DSP 520. In a similar

Nintendo Exhibit 1003
Part 2 of 2
Page 001237

manner, signals to be transmitted to the network 519 are processed, including modulation and encoding, for example, by the DSP 520 and are then provided to the transmitter 514 for digital to analog conversion, frequency up conversion, filtering, amplification and transmission to the communication network 519 via the antenna 518. Although a single transceiver 511 is shown for both voice and data communications, in alternative embodiments, the mobile device 500 may include multiple distinct transceivers, such as a first transceiver for transmitting and receiving voice signals, and a second transceiver for transmitting and receiving data signals, or a first transceiver configured to operate within a first frequency band, and a second transceiver configured to operate within a second frequency band.

[00129] In addition to processing the communication signals, the DSP 520 also provides for receiver and transmitter control. For example, the gain levels applied to communication signals in the receiver 512 and transmitter 514 may be adaptively controlled through automatic gain control algorithms implemented in the DSP 520. Other transceiver control algorithms could also be implemented in the DSP 520 in order to provide more sophisticated control of the transceiver 511.

[00130] The microprocessor 538 preferably manages and controls the overall operation of the mobile device 500. Many types of microprocessors or microcontrollers could be used here, or, alternatively, a single DSP 520 could be used to carry out the functions of the microprocessor 538. Low-level communication functions, including at least data and voice communications, are performed through the DSP 520 in the transceiver 511. High-level communication applications, including the voice communication application 524A, and the data communication application 524B are stored in the non-volatile memory 524

Nintendo Exhibit 1003
Part 2 of 2
Page 001238

for execution by the microprocessor 538. For example, the voice communication module 524A may provide a high-level user interface operable to transmit and receive voice calls between the mobile device 500 and a plurality of other voice devices via the network 519. Similarly, the data communication module 524B may provide a high-level user interface operable for sending and receiving data, such as e-mail messages, files, organizer information, short text messages, etc., between the mobile device 500 and a plurality of other data devices via the network 519.

[00131] The microprocessor 538 also interacts with other device subsystems, such as the display 522, RAM 526, auxiliary I/O devices 528, serial port 530, keyboard 532, speaker 534, microphone 536, a short-range communications subsystem 540 and any other device subsystems generally designated as 542. For example, the modules 524A-N are executed by the microprocessor 538 and may provide a high-level interface between a user of the mobile device and the mobile device. This interface typically includes a graphical component provided through the display 522, and an input/output component provided through the auxiliary I/O devices 528, keyboard 532, speaker 534, or microphone 536. Such interfaces are designated generally as UI 46 in Fig. 3.

[00132] Some of the subsystems shown in Fig. 7 perform communication-related functions, whereas other subsystems may provide "resident" or on-device functions. Notably, some subsystems, such as keyboard 532 and display 522 may be used for both communication-related functions, such as entering a text message for transmission over a data communication network, and device-resident functions such as a calculator or task list or other PDA type functions.

Nintendo Exhibit 1003
Part 2 of 2
Page 001239

[00133] Operating system software used by the microprocessor 538 is preferably stored in a persistent store such as the non-volatile memory 524. In addition to the operating system and communication modules 524A-N, the non-volatile memory 524 may include a file system for storing data. The non-volatile memory 524 may also include data stores for owner information and owner control information. The operating system, specific device applications or modules, or parts thereof, may be temporarily loaded into a volatile store, such as RAM 526 for faster operation. Moreover, received communication signals may also be temporarily stored to RAM 526, before permanently writing them to a file system located in the non-volatile memory 524. The non-volatile memory 524 may be implemented, for example, with Flash memory, non-volatile RAM, or battery backed-up RAM.

[00134] An exemplary application module 524N that may be loaded onto the mobile device 500 is a PIM application providing PDA functionality, such as calendar events, appointments, and task items. This module 524N may also interact with the voice communication module 524A for managing phone calls, voice mails, etc., and may also interact with the data communication module 524B for managing e-mail communications and other data transmissions. Alternatively, all of the functionality of the voice communication module 524A and the data communication module 524B may be integrated into the PIM module.

[00135] The non-volatile memory 524 preferably provides a file system to facilitate storage of PIM data items on the device. The PIM application preferably includes the ability to send and receive data items, either by itself, or in conjunction with the voice and data communication modules 524A, 524B, via the wireless network 519. The PIM data

Nintendo Exhibit 1003
Part 2 of 2
Page 001240

items are preferably seamlessly integrated, synchronized and updated, via the wireless network 519, with a corresponding set of data items stored or associated with a host computer system, thereby creating a mirrored system for data items associated with a particular user.

[00136] The mobile device 500 is manually synchronized with a host system by placing the mobile device 500 in an interface cradle, which couples the serial port 530 of the mobile device 500 to a serial port of the host system. The serial port 530 may also be used to insert owner information and owner control information onto the mobile device 500 and to download other application modules 524N for installation on the mobile device 500. This wired download path may further be used to load an encryption key onto the mobile device 500 for use in secure communications, which is a more secure method than exchanging encryption information via the wireless network 519.

[00137] Owner information, owner control information and additional application modules 524N may be loaded onto the mobile device 500 through the network 519, through an auxiliary I/O subsystem 528, through the short-range communications subsystem 540, or through any other suitable subsystem 542, and installed by a user in the non-volatile memory 524 or RAM 526. Such flexibility in application installation increases the functionality of the mobile device 500 and may provide enhanced on-device functions, communication-related functions, or both. For example, secure communication applications may enable electronic commerce functions and other such financial transactions to be performed using the mobile device 500.

[00138] When the mobile-device 500 is operating in a data communication mode, a received signal, such as a text message or a web page download, will be processed by

Nintendo Exhibit 1003
Part 2 of 2
Page 001241

the transceiver 511 and provided to the microprocessor 538, which preferably further processes the received signal for output to the display 522, or, alternatively, to an auxiliary I/O device 528. Owner information, owner control information, commands or requests related to owner information or owner control information, and software applications received by the transceiver 511 are processed as described above. A user of mobile device 500 may also compose data items, such as email messages, using the keyboard 532, which is preferably a complete alphanumeric keyboard laid out in the QWERTY style, although other styles of complete alphanumeric keyboards such as the known DVORAK style may also be used. User input to the mobile device 500 is further enhanced with the plurality of auxiliary I/O devices 528, which may include a thumbwheel input device, a touchpad, a variety of switches, a rocker input switch, etc. The composed data items input by the user are then transmitted over the communication network 519 via the transceiver 511.

[00139] When the mobile device 500 is operating in a voice communication mode, the overall operation of the mobile device 500 is substantially similar to the data mode, except that received signals are output to the speaker 534 and voice signals for transmission are generated by a microphone 536. In addition, the secure messaging techniques described above might not necessarily be applied to voice communications. Alternative voice or audio I/O devices, such as a voice message recording subsystem, may also be implemented on the mobile device 500. Although voice or audio signal output is accomplished through the speaker 534, the display 522 may also be used to provide an indication of the identity of a calling party, the duration of a voice call, or other voice call related information. For example, the microprocessor 538, in conjunction with the voice

Nintendo Exhibit 1003
Part 2 of 2
Page 001242

communication module 524A and the operating system software, may detect the caller identification information of an incoming voice call and display it on the display 522.

[00140] A short-range communications subsystem 540 is also to be included in the mobile device 500. For example, the subsystem 540 may include an infrared device and associated circuits and components, or a Bluetooth or 802.11 short-range wireless communication module to provide for communication with similarly-enabled systems and devices. Thus, owner information insertion, owner control information insertion, and application loading operations as described above may be enabled on the mobile device 500 via the serial port 530 or other short-range communications subsystem 540.

[00141] It is to be understood that Fig. 7 represents an example of an electronic device in which owner control systems and methods described above may be implemented. Implementation of such systems and methods in other electronic devices having further, fewer, or different components than those shown in Fig. 7 would occur to one skilled in the art to which this application pertains and are therefore considered to be within the scope of the present application. For example, although a SIM card has not been explicitly shown in Fig. 7, it should be appreciated that implementation of owner control systems and methods in electronic devices with SIM cards is contemplated. Since SIM cards currently incorporate a memory component, owner information, owner control information, or both, may be inserted onto a SIM card and used to maintain owner control of an electronic device when the SIM card is installed in the electronic device. In this case, a SIM card could be branded by inserting owner information onto the SIM card, and owner control information could then be inserted onto the SIM card or an electronic device in which the SIM card is installed.

Nintendo Exhibit 1003
Part 2 of 2
Page 001243

PTO/AIA/14 (12-13)
Approved for use through 01/31/2014. OMB 0651-0032
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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	11913-US-CNT3 4214-25714
	Application Number	
Title of Invention	System and Method of Owner Application Control of Electronic Devices	

The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.
This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

Secrecy Order 37 CFR 5.2

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Inventor Information:

Inventor 1 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Neil	P.	Adams	

Residence Information (Select One) ○ US Residency ◉ Non US Residency ○ Active US Military Service

City	Kitchener	Country of Residence i	CA

Mailing Address of Inventor:

Address 1	295 Phillip Street		
Address 2			
City	Waterloo	**State/Province**	ON
Postal Code	N2L 3W8	**Country** i	CA

Inventor 2 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Michael	G.	Kirkup	

Residence Information (Select One) ○ US Residency ◉ Non US Residency ○ Active US Military Service

City	Waterloo	Country of Residence i	CA

Mailing Address of Inventor:

Address 1	295 Phillip Street		
Address 2			
City	Waterloo	**State/Province**	ON
Postal Code	N2L 3W8	**Country** i	CA

Inventor 3 Remove

Legal Name

PTO/AIA/14 (12-13)
Approved for use through 01/31/2014. OMB 0651-0032
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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	11913-US-CNT3 4214-25714
	Application Number	
Title of Invention	System and Method of Owner Application Control of Electronic Devices	

Prefix	Given Name	Middle Name	Family Name	Suffix
	Herbert	A.	Little	

Residence Information (Select One) ○ US Residency ◉ Non US Residency ○ Active US Military Service

City	Waterloo	Country of Residence i	CA

Mailing Address of Inventor:

Address 1	295 Phillip Street		
Address 2			
City	Waterloo	**State/Province**	ON
Postal Code	N2L 3W8	**Country** i	CA

Inventor 4 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Russell	N.	Owen	

Residence Information (Select One) ○ US Residency ◉ Non US Residency ○ Active US Military Service

City	Waterloo	Country of Residence i	CA

Mailing Address of Inventor:

Address 1	450 Chesapeake Drive		
Address 2			
City	Waterloo	**State/Province**	ON
Postal Code	N2K 4B8	**Country** i	CA

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

☐ **An Address is being provided for the correspondence Information of this application.**

Customer Number	82313		
Email Address	dallaspatents@dfw.conleyrose.com	Add Email	Remove Email

PTO/AIA/14 (12-13)
Approved for use through 01/31/2014. OMB 0651-0032
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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	11913-US-CNT3 4214-25714
	Application Number	
Title of Invention	System and Method of Owner Application Control of Electronic Devices	

Application Information:

Title of the Invention	System and Method of Owner Application Control of Electronic Devices		
Attorney Docket Number	11913-US-CNT3 4214-25714	**Small Entity Status Claimed** ☐	
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	9	**Suggested Figure for Publication (if any)**	1

Filing By Reference :

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)

☐ **Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32).
Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.

Please Select One:	◉ Customer Number	○ US Patent Practitioner	○ Limited Recognition (37 CFR 11.9)
Customer Number	82313		

PTO/AIA/14 (12-13)
Approved for use through 01/31/2014. OMB 0651-0032
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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	11913-US-CNT3 4214-25714
	Application Number	
Title of Invention	System and Method of Owner Application Control of Electronic Devices	

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, or 365(c) or indicate National Stage entry from a PCT application. Providing this information in the application data sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.
When referring to the current application, please leave the application number blank.

Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Continuation of	13618311	2012-09-14

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
13618311	Continuation of	12885281	2010-09-17	8887988	2014-11-18

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
12885281	Continuation of	11118844	2005-04-29	7815100	2010-10-19

Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
11118844	Claims benefit of provisional	60567163	2004-04-30

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Continuation in part of	13606814	2012-09-07

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
13606814	Continuation of	12869589	2010-08-26	8302185	2012-10-30

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
12869589	Continuation of	10732132	2003-12-10	7793355	2010-09-07

Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
10732132	Claims benefit of provisional	60432610	2002-12-12

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button. Add

Foreign Priority Information:

PTO/AIA/14 (12-13)
Approved for use through 01/31/2014. OMB 0651-0032
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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	11913-US-CNT3 4214-25714
	Application Number	
Title of Invention	System and Method of Owner Application Control of Electronic Devices	

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55(d). When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX) [i] the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(h)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

Remove

Application Number	Country [i]	Filing Date (YYYY-MM-DD)	Access Code [i] (if applicable)

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button. Add

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.
NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

Authorization to Permit Access:

☒ Authorization to Permit Access to the Instant Application by the Participating Offices

PTO/AIA/14 (12-13)
Approved for use through 01/31/2014. OMB 0651-0032
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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	11913-US-CNT3 4214-25714
	Application Number	
Title of Invention	System and Method of Owner Application Control of Electronic Devices	

If checked, the undersigned hereby grants the USPTO authority to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the World Intellectual Property Office (WIPO), and any other intellectual property offices in which a foreign application claiming priority to the instant patent application is filed access to the instant patent application. See 37 CFR 1.14(c) and (h). This box should not be checked if the applicant does not wish the EPO, JPO, KIPO, WIPO, or other intellectual property office in which a foreign application claiming priority to the instant patent application is filed to have access to the instant patent application.

In accordance with 37 CFR 1.14(h)(3), access will be provided to a copy of the instant patent application with respect to: 1) the instant patent application-as-filed; 2) any foreign application to which the instant patent application claims priority under 35 U.S.C. 119(a)-(d) if a copy of the foreign application that satisfies the certified copy requirement of 37 CFR 1.55 has been filed in the instant patent application; and 3) any U.S. application-as-filed from which benefit is sought in the instant patent application.

In accordance with 37 CFR 1.14(c), access may be provided to information concerning the date of filing this Authorization.

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Applicant 1 Remove

If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.

Clear

◉ Assignee	○ Legal Representative under 35 U.S.C. 117	○ Joint Inventor
○ Person to whom the inventor is obligated to assign.	○ Person who shows sufficient proprietary interest	

If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:

Name of the Deceased or Legally Incapacitated Inventor :

If the Applicant is an Organization check here. ☒

Organization Name	BlackBerry Limited		
Mailing Address Information:			
Address 1	2200 University Avenue East		
Address 2			
City	Waterloo	**State/Province**	ON
Country [i]	CA	Postal Code	N2K 0A7
Phone Number		Fax Number	

PTO/AIA/14 (12-13)
Approved for use through 01/31/2014. OMB 0651-0032
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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	11913-US-CNT3 4214-25714
	Application Number	
Title of Invention	System and Method of Owner Application Control of Electronic Devices	

Email Address	

Additional Applicant Data may be generated within this form by selecting the Add button. Add

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not subsitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Assignee 1

Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication . An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.

Remove

If the Assignee or Non-Applicant Assignee is an Organization check here. ☐

Prefix	**Given Name**	Middle Name	**Family Name**	Suffix

Mailing Address Information For Assignee including Non-Applicant Assignee:

Address 1			
Address 2			
City		**State/Province**	
Country i		Postal Code	
Phone Number		Fax Number	
Email Address			

Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button. Add

Signature:

Remove

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications

Signature	/J. Robert Brown, Jr./			Date (YYYY-MM-DD)	2015-05-18
First Name	J. Robert	Last Name	Brown, Jr.	Registration Number	45438

Additional Signature may be generated within this form by selecting the Add button. Add

PTO/AIA/14 (12-13)
Approved for use through 01/31/2014. OMB 0651-0032
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 contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	11913-US-CNT3 4214-25714
	Application Number	
Title of Invention	System and Method of Owner Application Control of Electronic Devices	

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Nintendo Exhibit 1003
Part 2 of 2
Page 001252

Electronic Acknowledgement Receipt

EFS ID:	22376792
Application Number:	14715227
International Application Number:	
Confirmation Number:	5874
Title of Invention:	System and Method of Owner Application Control of Electronic Devices
First Named Inventor/Applicant Name:	Neil P. Adams
Customer Number:	82313
Filer:	J. Robert Brown/Rami Law
Filer Authorized By:	J. Robert Brown
Attorney Docket Number:	11913-US-CNT3 4214-25714
Receipt Date:	18-MAY-2015
Filing Date:	
Time Stamp:	17:57:09
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$1600
RAM confirmation Number	5464
Deposit Account	501515
Authorized User	BROWN, J. ROBERT

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		11913-US-CNT3_2015-05-18_APP.pdf	257412 f564b97f4b8ee14eec7360b602b3b6acf9c9be03	yes	64

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Specification	1	57
	Claims	58	63
	Abstract	64	64

Warnings:					
Information:					
2	Drawings-only black and white line drawings	11913-US-CNT3_2015-05-18_DRW.pdf	158868 3a7b6420098a1ab244efff516dc8d98577451760	no	9
Warnings:					
Information:					
3	Application Data Sheet	11913-US-CNT3_2015-05-18_ADS.pdf	1562268 288ea0981b5df938e33011a7849df84c3591392b	no	9
Warnings:					
Information:					
4	Fee Worksheet (SB06)	fee-info.pdf	35329 506d84338e6864e3f81f877bf2b64f84b2128421	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	2013877		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Document code: WFEE

United States Patent and Trademark Office
Sales Receipt for Accounting Date: 05/27/2015

ASAHLE SALE #00000025 Mailroom Dt: 05/18/2015 501515 14715227
01 FC : 1051 140.00 DA

PTO/SB/06 (09-11)
Approved for use through 1/31/2014. OMB 0651-0032
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.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 14/715,227	Filing Date 05/18/2015	☐ To be Mailed

ENTITY: ☒ LARGE ☐ SMALL ☐ MICRO

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT **01/11/2015**

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 20	Minus	** 20	= 0	X $80 =	0
Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X $420 =	0
☐ Application Size Fee (37 CFR 1.16(s))						
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
					TOTAL ADD'L FEE	0

AMENDMENT

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))						
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
					TOTAL ADD'L 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.

LIE
/PAULA BRITTON/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.