More specifically, this public key stored on the card will allow the individual card to verify data signed with the CA's private key. The public key of the CA, which is stored on the card, is used only for determining if the data sent to the card was signed with the proper CA private key. This allows the card to verify the source of any message coming from the CA.

Step 205 inserts a card enablement key in a secure portion of EEPROM in the card to facilitate card specific confidentiality during enablement, and step 207 inserts a card identifier in EEPROM of the card. The identifier, which can be accessed by any terminal, will allow the system to determine the identity of the card in later processes. The identifier is freely available and will not be used to authenticate messages.

Step 209 stores the operating system code in ROM on the card including any primitives which are called or supported by the operating system. The primitives are written in native language code (e.g., assembly language) and are stored in ROM. The primitives are subroutines which may be called by the operating system or by applications residing on the card such as mathematic functions (multiply or divide), data retrieval, data manipulation or cryptographic algorithms. The primitives can be executed very quickly because they are written in the native language of the processor.

After the IC cards are manufactured, they are sent to a personalization bureau ("PB") to enable and personalize the card by storing card personalization data in the memory of the card. The terms enablement and personalization are used interchangeably herein to indicate the preparatory steps taken to allow the card to be loaded securely with

an application. The individual cards are preferably manufactured in batches and are sent to a personalization bureau in a group for processing.

Card Enablement/Personalization

Figure 3 shows the steps of the card enablement process when the card arrives at a personalization bureau. The personalization bureau may be the card issuer (e.g., a bank or other financial institution) or may be a third party that performs the service for the card issuer. The personalization bureau configures the card to a specific user or user class.

Figure 3 specifically shows the steps taken to enable and personalize each IC card which will work within the system. The cards can be placed in a terminal which communicates with IC cards and which reads the card identifier data (previously placed on the card during the manufacturing process -- see step 207). This card identification data is read from the card in step 301. The terminal will effectively send a "get identification data" command to the card and the card will return the identification data to the terminal.

The PB typically processes a group of cards at the same time, and will first compile a list of IC card identification data for the group of cards it is personalizing. The PB then sends electronically (or otherwise) this list of identification data to the Certification Authority ("CA") which creates a personalization (or enablement) data block for each card identifier. The data block includes the card personalization data organized in a number of identity fields and an individual key set for the card, discussed below. These data blocks are then encrypted and sent to the PB in step 302. By using the

card identification data, the PB then matches the cards with the encrypted data blocks and separately loads each data block onto the matched card. To insure that the CA controls the identity of the card and the integrity of the system, the PB never obtains knowledge of the content of the data blocks transferred. Some aspects of the personalization are requested by the card issuer to the CA in order to affect their preferred management of the cards they issue. The following additional steps are performed.

Step 303 first checks to see if an enablement bit stored in EEPROM of the card has been already set. If it already has been set, the card has already been configured and personalized and the enablement process will end as shown in step 304. A card cannot be enabled and personalized twice. If the bit has not been set, then the process continues with step 305.

In step 305, the individualized card key set for the card being enabled (which key set is generated at the CA) is stored on the card. The keys can be used later in off-card verification (i.e., to verify that the card is an authentic card). This verification is necessary to further authenticate the card as the one for which the application was intended.

Step 307 generates four different MULTOS Security Manager (MSM) characteristic data elements (otherwise referred to herein as personalization data) for the card at the CA which are used for securely and correctly loading and deleting applications from a particular card. The MSM characteristics also allow for the loading of applications on specific classes of identified cards. (These MSM characteristics are further described in connection with Figure 5.)

Other data can also be stored on the card at this time as needed by the system design such as an address table or further subroutines.

Step 311 sets the enablement bit in EEPROM of the card which indicates that the enablement process has been completed for the particular card. When this bit is set, another enablement process cannot occur on the card. This ensures that only one personalization and enablement process will occur to the card thus preventing illegal tampering of the card or altering the card by mistake. In the preferred embodiment, the enablement bit is initially not set when the card is manufactured and is set at the end of the enablement process.

Figure 4 shows an example of a block diagram of an IC card chip which has been manufactured and personalized. The IC card chip is located on an IC card for use. The IC card preferably includes a central processing unit 401, a RAM 403, a EEPROM 405, a ROM 407, a timer 409, control logic 411, an I/O ports 413 and security circuitry 415, which are connected together by a conventional data bus.

Control logic 411 in memory cards provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports. CPU 401 with its control logic can perform calculations, access memory locations, modify memory contents, and manage input/output ports. Some cards have a coprocessor for handling complex computations like cryptographic algorithms. Input/output ports 413 are used under the control of a CPU and control logic alone, for communications between the card and a card acceptance device. Timer 409 (which generates or provides a clock pulse) drives the control logic 411 and CPU 401 through the sequence of steps that

accomplish memory access, memory reading or writing, processing, and data communication. A timer may be used to provide application features such as call duration. Security circuitry 415 includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed ("blown") upon completion of testing to prevent later access. The personalization data to qualify the card is stored in a secured location of EEPROM 405. The comparing of the personalization data to applications permissions data is performed by the CPU 401.

Figure 5 shows the steps of generating and loading the four elements of the card personalization data into the memory of the IC cards, and Fig. 5A shows a schematic of bit maps for each identity field residing in the memory of an IC card containing personalization data in accordance with the present invention. Each data structure for each identity field has its own descriptor code. Step 501 loads the data structure for the identity field "card ID" called "msm_mcd_permissions_mcd_no." This nomenclature stands for MULTOS system manager _ MULTOS card device _ permissions_ MULTOS card device number. Although this number is typically 8 bytes long as shown in Fig. 5A, the data could be any length that indicates a unique number for the card. In the preferred embodiment, 2 bytes are dedicated as a signal indicator, 2 bytes comprise a MULTOS Injection Security Module ID (MISM ID) indicating which security module injected the card with its injected keys when it was manufactured, and 4 bytes comprise an Integrated Circuit Card (ICC) serial number which identifies the individual card produced at the particular MISM.

Step 503 loads the data structure for the identity field "issuer ID" called "msm_mcd_permissions_ mcd_issuer_id." This nomenclature stands for a MULTOS card device issuer identification number. Each card issuer (such as a particular bank, financial institution or other company involved with an application) will be assigned a unique number in the card system. Each IC card in the MULTOS system will contain information regarding the card issuer which personalized the card or is responsible for the card. A card issuer will order a certain number of cards from a manufacturer and perform or have performed the personalization process as described herein. For example, a regional bank may order 5,000 cards to be distributed to its customers. The "mcd_issuer_id" data structure on these cards will indicate which issuer issued the cards. In the preferred embodiment, the data structure is 4 bytes long (as shown in Fig. 5A at 503A) to allow for many different issuers in the system although the length of the data structure can vary with the needs of the card system.

Step 505 loads the data structure for the identity field "product ID" called "msm_mcd_permissions_mcd_ issuer_product_id." This nomenclature stands for MULTOS card device issuer product identification number. Each card issuer may have different classes of products or cards which it may want to differentiate. For example, a bank could issue a regular credit card with one product ID, a gold credit card with another product ID and a platinum card with still another product ID. The card issuer may wish to load certain applications onto only one class of credit cards. A gold credit card user who pays an annual fee may be entitled to a greater variety of applications than a regular credit card user who pays no annual fee. The product ID field identifies the card as a

particular class and will later allow the card issuer to check the product ID and only load applications onto cards which match the desired class.

Another way to differentiate products is by application type, such as by categorizing the application as financial, legal, medical and/or recreational, or by assigning particular applications to a group of cards. For example, one card issuer may have different loyalty programs available with different companies to different sets of card users. For example, a bank may have an American Airlines® loyalty program and a British Airways® loyalty program for different regions of the country dependent on where the airlines fly. The product type allows the issuer to fix the product classification of the card during the personalization process. When loading applications onto the card, the product type identification number on each card will be checked to make sure it matches the type of card onto which the issuer desires to load. The product type data structure is preferably an indexing mechanism (unlike the other personalization data structure) of 8 bits (as shown at 505A in Fig. 5A) but could be any length depending upon the needs of the card system. In the illustrated embodiment, the resulting instruction would be to locate the second bit (since the byte's indicated value is 2) in the array to be searched (see discussion of step 809 below).

Step 507 loads the data structure for the identity field data called "msm_mcd_permissions_mcd_ controls_data_ date." This nomenclature stands for the MULTOS card device controls data date or, in other words, the date on which the card was personalized so that, for example, the application loader can load cards dated only after a certain date, load cards before a certain date (e.g., for application updates) or load

cards with a particular data date. The information can include the year, month and day of personalization or may include less information, if desired. The data_date data structure is preferably 1 byte in length (see 507A in Fig. 5A) although it could be any length depending upon the needs of the particular card system used.

Once all of the personalization data structures are loaded and stored in the card, the card has been identified by issuer, product class, date and identification number (and other data fields, if desired), and the card cannot change its identity; these fields cannot be changed in the memory of the card. If a card user wants to change the product_id stored in the card to gain access to different applications available to another product type, a new card will have to be issued to the user containing the correct personalization data. This system is consistent with a gold card member receiving a new card when the classification is changed to platinum.

After the card has been enabled and personalized by storing its individual card key set, MSM personalization characteristics and enablement bit as described in Fig. 3, the card is ready to have applications loaded into its memory.

Loading Applications

The application loading process contains a number of security and card configuration checks to ensure the secure and proper loading of an application onto the intended IC card. The application loading process is preferably performed at the personalization bureau so that the card will contain one or more applications when the card is issued. The card may contain certain common applications which will be present on every card the issuer sends out, such as an electronic purse application or a credit/debit

application. Alternatively, the personalization bureau could send the enabled cards to a third party for the process of loading applications. The multiple application operating system stored in the ROM of each card and the card MSM personalization data is designed to allow future loading and deleting of applications after the card has been issued depending upon the desires of the particular card user and the responsible card issuer. Thus, an older version of an application stored on the IC card could be replaced with a new version of the application. An additional loyalty application could also be added to the card after it has been initially sent to the card user because the application is newly available or the user desires to use the new application. These loading and deleting functions for applications can be performed directly by a terminal or may be performed over telephone lines, data lines, a network such as the Internet or any other way of transmitting data between two entities. In the present IC card system, the process of transmitting the application program and data ensures that only IC cards containing the proper personalization data and which fit on application permissions profile will be qualified and receive the corresponding application program and data.

Figure 6 shows the preferred steps performed in loading an application onto an IC card in the MULTOS IC card system. For this example, the personalization bureau is loading an application from a terminal which enabled the same card. Step 601 performs an "open command" initiated by the terminal which previews the card to make sure the card is qualified to accept the loading of a specific application. The open command provides the card with the application's permissions data, the application's size, and instructs the card to determine (1) if the enablement bit is set indicating the card

has been personalized; (2) whether the application code and associated data will fit in the existing memory space on the card; and (3) whether the personalization data assigned to the application to be loaded allows for the loading of the application onto the particular card at issue. The open command could also make additional checks as required by the card system. These checking steps during the open command execution will be described in detail in conjunction with Figure 7.

After the open command has been executed, the application loader via the terminal will be advised if the card contains the proper identification personalization data and if enough room exists in the memory of the card for the application code and related data. If there is insufficient memory, then a negative response is returned by the card and the process is abended (abnormally ended). If the identification personalization data does not match the applications permissions data, a warning response is given in step 603, but the process continues to the load and create steps. Alternatively, if there is no match, the process may automatically be abended. If a positive response is returned by the card to the terminal in step 605, the application loader preferably proceeds to next steps. The open command allows the application to preview the card before starting any transfer of the code and data.

Step 607 then loads the application code and data onto the IC card into EEPROM. The actual loading occurs in conjunction with create step 609 which completes the loading process and enables the application to execute on the IC card after it is loaded. The combination of the open, load and create commands are sent by the terminal, or another application provider source, to the IC card to perform the application

loading process. The operating system in the IC cards is programmed to perform a specific set of instructions with respect to each of these commands so that the IC card will communicate with and properly carry out the instructions from the terminal.

Step 609 performs the create command which at least: (1) checks if an application load certificate is signed (encrypted) by the CA and therefore authenticates the application as a proper application for the system; and (2) checks the card personalization data stored on the card against the permissions profile for the application to be loaded to qualify the card for loading. It may do other checks as required. If one of the checks fails, then a failure response 610 is given and the process aborts. The application after it has passed these checks will be loaded into the memory of the card.

Figure 7 shows the various steps of the open step 601 of Fig. 6 in more detail. Step 701 determines if the enablement (i.e., control) bit is set. This bit is set when the card has completed its personalization process and has been assigned its personalization data. An application can be loaded on an IC card in the card system only if the card contains the personalization data. If the enablement bit is not set, the card has not been personalized and therefore the card returns a negative response 703 to the terminal. If the enablement bit is set, then the card has been enabled and the test conditions continue with step 711.

Step 711 checks if there is sufficient space in the memory on the card to store the application code and its associated data. Applications will typically have associated data related to their functions. This data will be used and manipulated when the application is run. Storage space in the memory of an IC card is a continuing concern

due to the relatively large physical space required for EEPROM and how it fits in the integrated circuit which is desired to be small enough to fit on a credit card sized card. An example of the size of a preset EEPROM on an IC card is 16K bytes although the actual size varies. Applications can range from 1K byte or less for a very simple application up to the size of available memory for a more sophisticated application. The data associated with an application can range from no data being stored in the card memory to a size constrained by the amount of available memory. These varied sizes of application code and data continually increase as applications become more advanced and diverse.

MULTOS as an operating system is not limited by the number of applications and associated data it can store on the card. Thus, if five applications can fit in the available memory of the card, the card user will have greatly increased functionality than if one or two applications were stored on the card. Once a card's memory is filled to its capacity, however, a new application cannot be loaded onto the card unless another application including its code and data of sufficient size can be deleted. Therefore, checking the amount of available space on the card is an important step. If there is not sufficient space, then an insufficient space response 713 will be returned to the terminal. The application loader can then decide if another existing application on the card should be deleted to make room for the new application. Deletion depends upon the card issuer having an application delete certificate from the CA. If there is sufficient space on the card, then the process continues with step 715.

An example of the testing of memory spaces in step 711 is now described. The numbers used in this example in no way limit the scope of the invention but are used only to illustrate memory space requirements. An IC card may have 16K available EEPROM when it is first manufactured. The operating system data necessary for the operating system may take up 2K of memory space. Thus, 14K would remain. An electronic purse application's code is stored in EEPROM and may take up 8K of memory space. The purse application's required data may take up an additional 4K of memory space in EEPROM. The memory space which is free for other applications would thus be 2K (16K-2K-8K-4K=2K). If a card issuer wants to load a credit/debit application whose code is 6K bytes in size onto the card in this example, the application will not fit in the memory of the IC card. Therefore, the application cannot load the new application without first removing the purse application from the card. If a new credit/debit application was loaded into EEPROM of the IC card, then it would have to overwrite other application's code or data. The application loader is prevented from doing this.

Figure 8 shows the steps performed in determining whether the card's personalization data falls within the permissible set of cards onto which the application at issue may be loaded. These steps are preferably performed during the execution of the "create" command. However, these steps may be performed at any time during the loading or deleting of an application. As described previously, the card is personalized by storing data specific to the card (MSM personalization data) including: a card ID designation specific to an individual card, the card issuer number indicating the issuer of the card, the product type of the card, such as a gold or platinum card, and the date the

card was personalized. This data uniquely identifies the card apart from all other IC cards in the system.

Accordingly, applications can be selectively stored on individual cards in the IC card system on virtually any basis, including the following. An application can be loaded selectively to cards containing one or more specific card numbers. An application can be selectively loaded on one or more cards containing a specified card issuer ID. Moreover, an application can be loaded only upon one type of product specified by the particular card issuer, and/or the application can be loaded only on cards which have a specified date or series of dates of personalization. Each of the personalization data allows an application to be selectively loaded onto certain cards or groups of cards and also ensures that cards without the proper permissions will not receive the application. Personalization data types in addition to the four described can also be used as needed.

The selection of IC cards upon which a particular application may be loaded is made possible by the use of "applications permissions data" which is assigned to the application and represents at least one set of cards upon which the application may be loaded. The set may be based on virtually any factor, including one or more of the following: card numbers, card issuers, product types or personalization dates. Although the individual card's personalization data typically identify one specific number, one card issuer, one product type and one date, the application's permissions data may indicate a card numbers or a blanket permission, a card issuer or a blanket permission, and a number of product types and dates.

For example, a frequent loyalty program may be configured to allow its loading and use on cards in different product classes belonging to one card issuer. In addition, the application permissions data may indicate that the loyalty program can be used on gold and platinum product types if the card was issued after May, 1998. Thus, the MSM permissions check will determine if the card's individual personalization data is included in the allowed or permissible set of cards upon which the application may be loaded. If it is, the application will be loaded.

To expedite the comparison process, an alternative embodiment may include setting one or more permissions data at zero representing a blanket permission for that particular data. For instance, by placing a zero for the "card number" entry in the application permissions data or some other value indicating that all cards may be loaded regardless of their number, the system knows not to deny any cards based on their card number. Moreover, if a zero is placed in the application's permissions data "issuer ID," then all cards similarly will pass the "issuer" test comparison. This feature allows greater flexibility in selecting groups of cards. The zero indicator could also be used for other permissions data, as required.

Referring to Figure 8, each of the permissions data is checked in the order shown, but other orders could be followed because if any one of the permissions fails, the application will be prevented from being loaded on the IC card being checked. The permissions are preferably checked in the order shown. Step 801 checks if the application permissions product type set encompasses the card's product type number stored in the memory of the card. Each card product type is assigned a number by the

system operator. The product types are specified for each card issuer because different card issuers will have different product types. The cards are selectively checked to ensure that applications are loaded only on cards of authorized product type. The application permissions product type set can be 32 bytes long which includes multiple acceptable product types or can be a different length depending upon the needs of the system. Using data structure 505A as an example, the operating system would check bit number 2 in the 256 bit array (32 bytes x 8 bits per byte) resulting from the 32 byte long application permissions data structure. If the permissions check fails, then the card returns a failure message to the terminal in step 803. If the product type check passes (for example, the value of bit no. 2 being 1), then the process continues with step 805.

Step 805 checks if the application permissions allowable card issuer number set encompasses the card's issuer number stored in the memory of the card or if the application permissions issuer data is zero (indicating all cards pass this individual permissions check). Each card issuer is assigned a number by the system operator and the cards are selectively checked to ensure that applications are loaded only on cards distributed by authorized card issuers. The application permissions card issuer number set can be 4 bytes long if one issuer is designated or can be longer depending upon the needs of the system. If the issuer check fails, then the card returns a failure message to the terminal in step 807. If the check passes, then the process continues with step 809.

Step 809 checks if the application permissions date set encompasses the card's data date stored in the memory of the card. The date that the IC card was personalized will be stored and will preferably include at least the month and year. The

cards are selectively checked to ensure that applications are loaded only on cards with the authorized personalization date. The application permissions date set can be 32 bytes long which includes multiple dates or can be a different length depending upon the needs of the system. If the date permissions check fails, then the card returns a failure message to the terminal in step 811. If the date check passes, then the process continues with step 813.

Step 813 checks if the application permissions allowable card number set encompasses the card's ID number stored in the card memory or if the application permissions allowable card number data is zero (indicating all cards pass this individual permissions check). The testing of the permissions is performed on the card during the execution of the open, load and create commands. The application permissions card number data set can be 8 bytes long if one number is designated or can be longer depending upon the needs of the system. If the card number check fails, then the card returns a failure message to the terminal in step 815. If the check passes, then the process continues with step 817.

Summary of IC Card System's Process

Figure 9 shows the components of the system architecture for the card initialization process of an IC card in a secure multiple application IC card system. The system includes a card manufacturer 102, a personalization bureau 104, an application loader 106, the IC card 107 being initialized, the card user 109 and the certification authority 111 for the entire multiple application secure system. The card user 131 is the

person or entity who will use the stored applications on the IC card. For example, a card user may prefer an IC card that contains both an electronic purse containing electronic cash (such as MONDEX™) and a credit/debit application (such as the MasterCard® EMV application) on the same IC card. The following is a description of one way in which the card user would obtain an IC card containing the desired applications in a secure manner.

The card user would contact a card issuer 113, such as a bank which distributes IC cards, and request an IC card with the two applications both residing in memory of a single IC card. The integrated circuit chip for the IC card would be manufactured by manufacturer 102 and sent to the card issuer 113 (or an entity acting on its behalf) in the form of an IC chip on a card. As discussed above (see steps 201-209), during the manufacturing process, data is transmitted 115 via a data conduit from the manufacturer 102 to card 107 and stored in IC card 107's memory. (Any of the data conduits described in this figure could be a telephone line, Internet connection or any other transmission medium.) The certification authority 111, which maintains encryption/decryption keys for the entire system, transmits 117 security data (i.e., global public key) to the manufacturer over a data conduit which is placed on the card by the manufacturer along with other data, such as the card enablement key and card identifier. The card's multiple application operating system is also stored in ROM and placed on the card by the manufacturer. After the cards have been initially processed, they are sent to the card issuer for personalization and application loading.

The card issuer 113 performs, or has performed by another entity, two separate functions. First, the personalization bureau 104 personalizes the IC card 107 in the ways described above, and second, the application loader 106 loads the application provided the card is qualified, as described.

Regarding personalization, an individualized card key set is generated by the CA and stored on the card (see Fig. 3). The card is further given a specific identity using MSM personalization (see Fig. 3, step 307 and Fig. 5) including a card ID number, an issuer ID number identifying the card issuer which processed the card, a card product type number which is specified by the card issuer and the date upon which the personalization took place. After the card has been personalized, applications need to be loaded onto the card so that the card can perform desired functions.

The application loader 106, which could use the same terminal or data conduit as personalization bureau 104, first needs to have determined if the card is qualified to accept the application. This comparison process takes place on the card itself (as instructed by its operating system) using the permissions information. The card, if it is qualified, thus selectively loads the application onto itself based upon the card's identity and the card issuer's instructions. The application loader communicates 119 with the IC card via a terminal or by some other data conduit. After the applications have been loaded on the card, the card is delivered to the card user 109 for use.

The secure multiple application IC card system described herein allows for selective loading and deleting of applications at any point in the life cycle of the IC card after the card has been personalized. Thus, a card user could also receive a personalized

card with no applications and then select a desired application over a common transmission line such as a telephone line or Internet connection.

Figure 10 is a system diagram of entities involved with the use of an IC card once it has been personalized. The system includes an IC card 151, a terminal 153, an application load/delete entity 155, the certification authority 157, a card issuer 171 and other IC cards 159 in the system. The arrows indicate communication between the respective entities. The CA 157 facilitates loading and deleting of applications. After providing the MSM permissions data and card specific keyset to the card during card enablements, the CA allows applications to be later loaded and deleted preferably by issuing an application certificate. Application specific keys are required to authenticate communication between a card and terminal. The IC card 151 also can communicate with other IC cards 159. Card issuer 171 is involved with all decisions of loading and deleting applications for a card which it issued. All communications are authenticated and transmitted securely in the system.

For instance, IC card 151 will use the following procedure to load a new application onto the card. IC card 101 is connected to terminal 153 and the terminal requests that an application be loaded. Terminal 153 contacts application load/delete entity 155 which, as a result and in conjunction with card issuer 171, sends the application code, data and application permissions data (along with any other necessary data) to terminal 153. Terminal 153 then queries card 151 to ensure it is the correct card onto which the application may be loaded. If IC card passes the checks discussed above, the application is loaded onto card 151. The CA 157 provides the application load or

delete certificate that enables the application to be loaded or deleted from the card. This example shows one way to load the application, but other variations using the same principles could be performed, such as directly loading the application at the application load/delete entity 155.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

For example, it will be appreciated that the MSM personalization and permissions data may not only be used for loading applications onto IC cards but also for deleting applications from said cards. The same checks involving MSM permissions and loading applications are made for deleting applications. A delete certificate from the CA authorizing the deletion of an application will control from which cards the application may be deleted. This is accomplished through the personalization data stored on each IC card and the permissions check as described herein.

Moreover, the data may also be applicable to personal computers or other units onto which applications may be loaded which are not physically loaded on cards. In addition, the application's permissions data may actually include data representative of a set or sets of cards to be excluded, instead of included -- cards that cannot be loaded with the application.

The scope of the present disclosure includes any novel feature or combination of features disclosed therein either explicitly or implicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or mitigates any or all of the problems addressed by the present invention. The applicant hereby gives notice that new claims may be formulated to such features during the prosecution of this application or of any such further application derived therefrom. In particular, with reference to the appended claims, features from dependent claims may be combined with those of the independent claims in any appropriate manner and not merely in the specific combinations enumerated in the claims.

CLAIMS:

1. An IC card system comprising at least one IC card, an application to be loaded onto said card and means for determining whether said card is qualified to accept the loading of said application onto said card.

2. The IC card system of claim 1, wherein said IC card contains card personalization data, and said application is assigned application permissions data representing at least one set of IC cards upon which said application may be loaded.

3. The IC card system of claim 2, wherein said determining means compares said card personalization data with said application permissions data.

4. The IC card system of claim 3, wherein whether said application is loaded onto said IC card depends on the result of said comparison, such that in the event the card personalization data matches said permissions data set the card is qualified and the application is loaded.

5. The IC card system of any of claims 2 to claim 4, wherein said personalization data comprises data representative of a unique card identification designation.

6. The IC card system of any of claims 2 to claim 5, wherein said personalization data comprises data representative of a card issuer.

7. The IC card system of any of claims 2 to claim 6, wherein said personalization data comprises data representative of a product class.

8. The IC card system of any of claims 2 to claim 7, wherein said personalization data comprises data representative of a date.

9. An IC card system comprising at least one IC card and an application, wherein said IC card contains personalization data representative of that card and said application is assigned a permissions data set representing at least one IC card upon which said application may be loaded, said system further comprising means for determining whether said personalization data falls within said permissions data set.

10. The IC card system of claim 9 wherein said application is loaded onto said IC card in the event said determining means determines that said personalization data falls within said set.

11. The IC card system of claim 9 or claim 10 wherein said personalization data comprises data representing a card identification designation, and an issuer of said card.

12. The IC card system of any of claims 9 to claim 11 wherein said personalization data comprises data representing a product class and a date.

13. The IC card system of any of claims 9 to 12 wherein said permissions data set includes a plurality of card identification designations.

14. The IC card system of any of claims 9 to 13 wherein said permissions data set includes one or more issuers of IC cards.

15. The IC card system of any of claims 9 to 14 wherein said permissions data set includes one or more product classes.

16. The IC card system of any of claims 9 to 15 wherein said permissions data set includes a plurality range of dates.

17. The IC card system of any of claims 9 to 16 wherein said permissions data set includes all IC cards which attempt to load the application.

18. An IC card system comprising at least one IC card, an application to be loaded onto said card and means for enabling said card to be loaded with said application.

19. The IC card system of claim 18 wherein said enabling means comprises means for storing personalization data onto said card.

20. The IC card system of claim 18 wherein said enabling means comprises means for setting an enablement bit.

21. The IC card system of claim 19 wherein said enabling means comprises means for setting an enablement bit.

22. The IC card system of claim 20 further comprising means for checking the enablement bit prior to enabling said IC card to determine whether or not said card has already been enabled.

23. The IC card system of claim 21 further comprising means for checking the enablement bit prior to enabling said IC card to determine whether or not said card has already been enabled.

24. A process for loading an application onto an IC card comprising the step of determining whether said IC card is qualified to accept the loading of said application onto said card.

25. The process of claim 24 wherein said determining step includes the steps of: providing said card with personalization data;

assigning to said application permissions data representing at least one set of IC cards upon which said application may be loaded;

comparing said personalization data with said permissions data; and

loading said application onto said IC card provided said personalization data falls within said set of cards upon which said application may be loaded.

26. The process of claim 25, wherein said personalization data comprises data representative of a card identification designation.

27. The process of claim 25 or claim 26, wherein said personalization data comprises data representative of a card issuer.

28. The process of any of claims 25 to claim 27, wherein said personalization data comprises data representative of a product class.

29. The process of any of claims 25 to claim 28, wherein said personalization data comprises data representative of a date.

30. The process of any of claims 25 to claim 29 further comprising the first step of enabling said card to be loaded with said application.

31. The process of claim 30 wherein said enabling step includes the step of storing personalization data onto said card.

32. The process of claim 30 wherein said enabling step includes the step of setting an enablement bit indicating that the card has been enabled.

33. The process of claim 31 wherein said enabling step further includes the step of setting an enablement bit indicating that the card has been enabled.

34. The process of claim 32 wherein prior to said enabling step a checking step is performed to determine whether said card has been enabled.

35. The process of claim 33 wherein prior to said enabling step a checking step is performed to determine whether said card has been enabled.

36. A process for deleting an application from an IC card comprising the step of determining whether said IC card is qualified to delete said application based upon permissions data associated with said application.

37. The process of claim 36 wherein said determining step includes the steps of:

providing said card with personalization data;

assigning to said application permissions data representing at least one set of IC cards from which said application may be deleted;

comparing said personalization data with said permissions data; and

deleting said application from said IC card provided said personalization data falls within said set of cards from which said application may be deleted.

38. The process of claim 37, wherein said personalization data comprises data representative of a card identification designation.

39. The process of claim 37 or claim 38, wherein said personalization data comprises data representative of a card issuer.

40. The process of any of claims 37 to claim 39, wherein said personalization data comprises data representative of a product class.

41. The process of any of claims 37 to claim 40, wherein said personalization data further comprises data representative of a date.

42. An IC card system comprising at least one IC card, an application to be deleted from said card and means for determining whether said card is qualified to delete said application from said card.

43. The IC card system of claim 42, wherein said IC card contains card personalization data, and said application is assigned application permissions data set representing at least one set of IC cards from which said application may be deleted.

44. The IC card system of claim 43, wherein said determining means compares said card personalization data with said application permissions data.

45. The IC card system of claim 44, wherein whether said application is deleted from said IC card depends on the result of said comparison, such that in the event the card personalization data matches said permissions data set the card is qualified and the application is deleted.

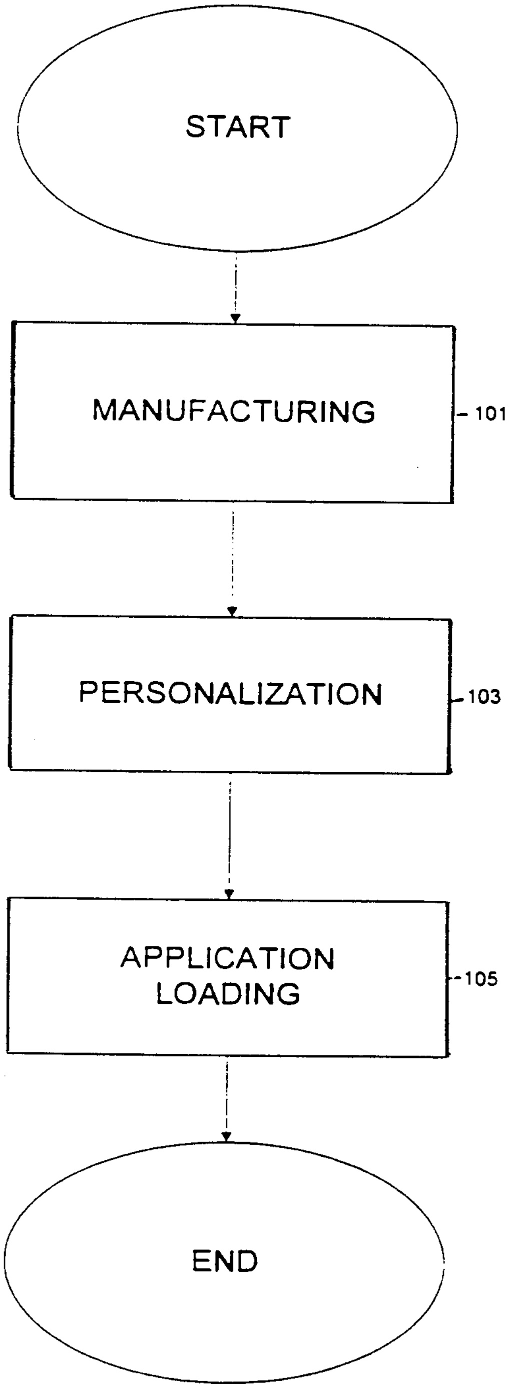

START
MANUFACTURING
101
PERSONALIZATION
103
APPLICATION LOADING
105
END

FIG. 1

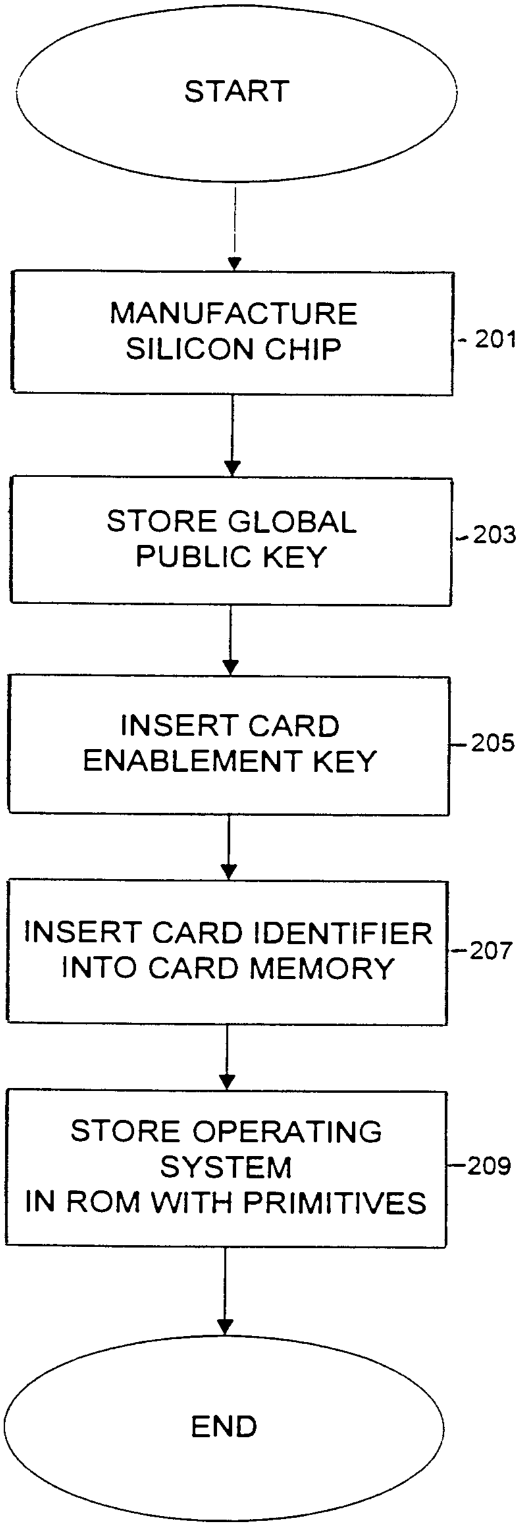

START
MANUFACTURE SILICON CHIP
201
STORE GLOBAL PUBLIC KEY
203
INSERT CARD ENABLEMENT KEY
205
INSERT CARD IDENTIFIER INTO CARD MEMORY
207
STORE OPERATING SYSTEM IN ROM WITH PRIMITIVES
209
END

FIG. 2

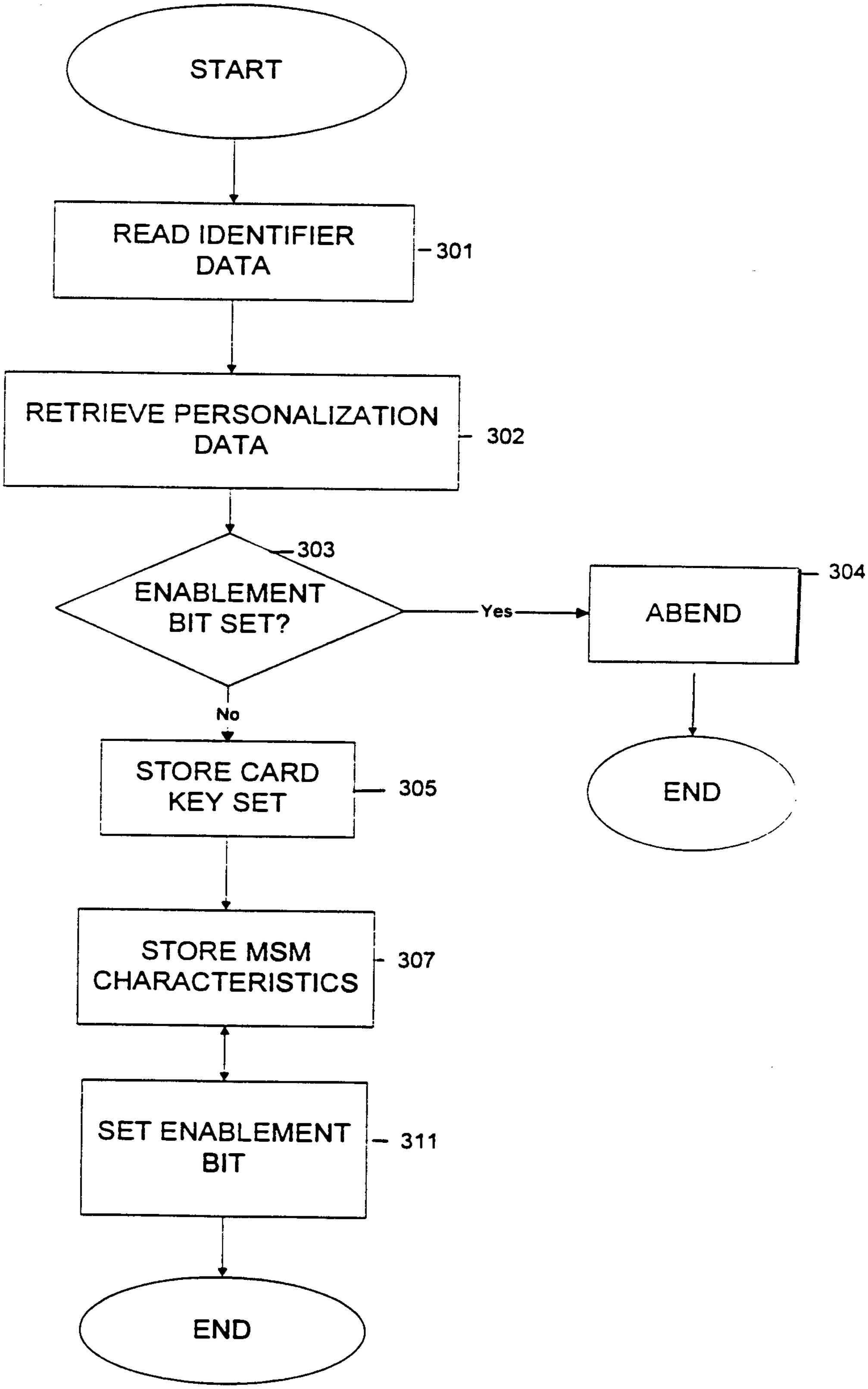

START
READ IDENTIFIER DATA
301
RETRIEVE PERSONALIZATION DATA
302
303
ENABLEMENT BIT SET?
Yes
ABEND
304
END
No
STORE CARD KEY SET
305
STORE MSM CHARACTERISTICS
307
SET ENABLEMENT BIT
311
END

FIG. 3

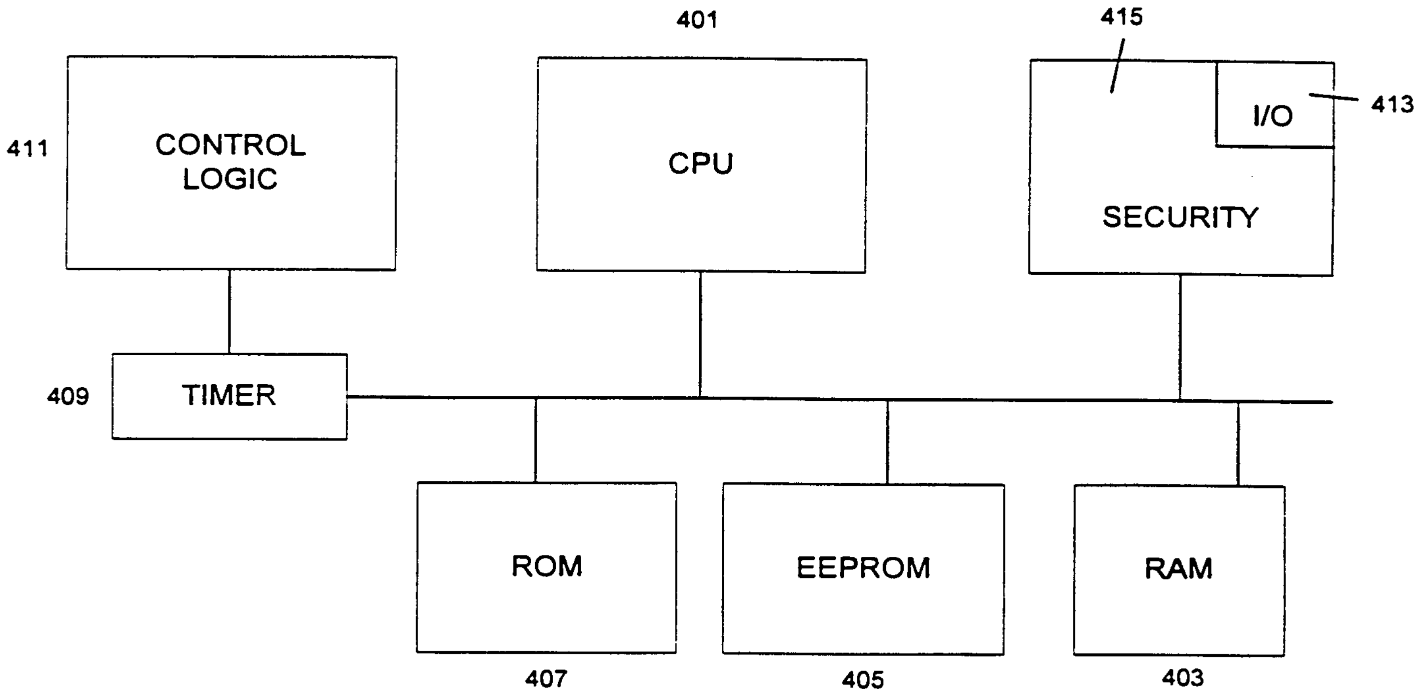

401
CPU
415
I/O
413
SECURITY
411
CONTROL LOGIC
409
TIMER
ROM
407
EEPROM
405
RAM
403

FIG. 4

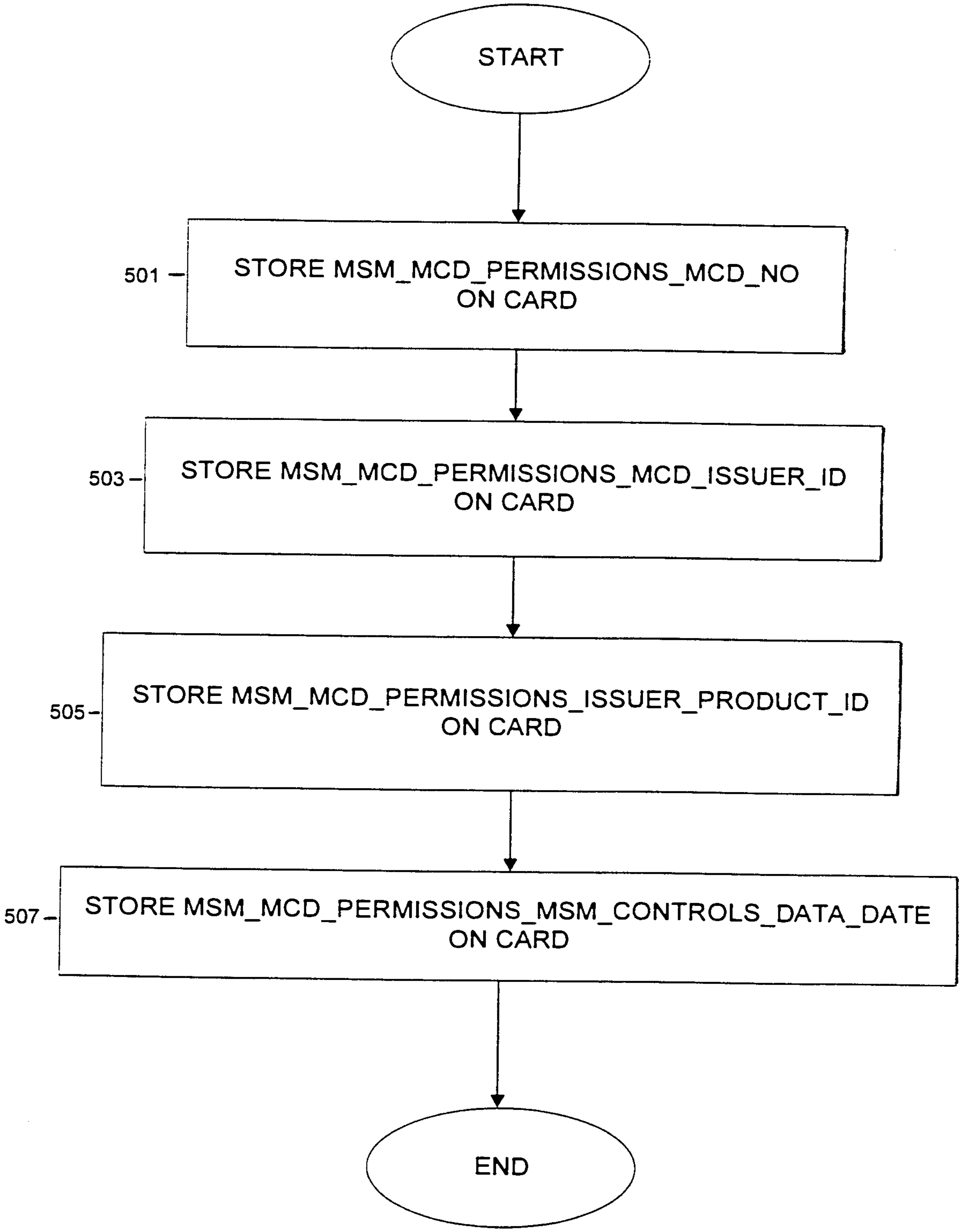

START
501
STORE MSM_MCD_PERMISSIONS_MCD_NO ON CARD
503
STORE MSM_MCD_PERMISSIONS_MCD_ISSUER_ID ON CARD
505
STORE MSM_MCD_PERMISSIONS_ISSUER_PRODUCT_ID ON CARD
507
STORE MSM_MCD_PERMISSIONS_MSM_CONTROLS_DATA_DATE ON CARD
END

FIG.. 5

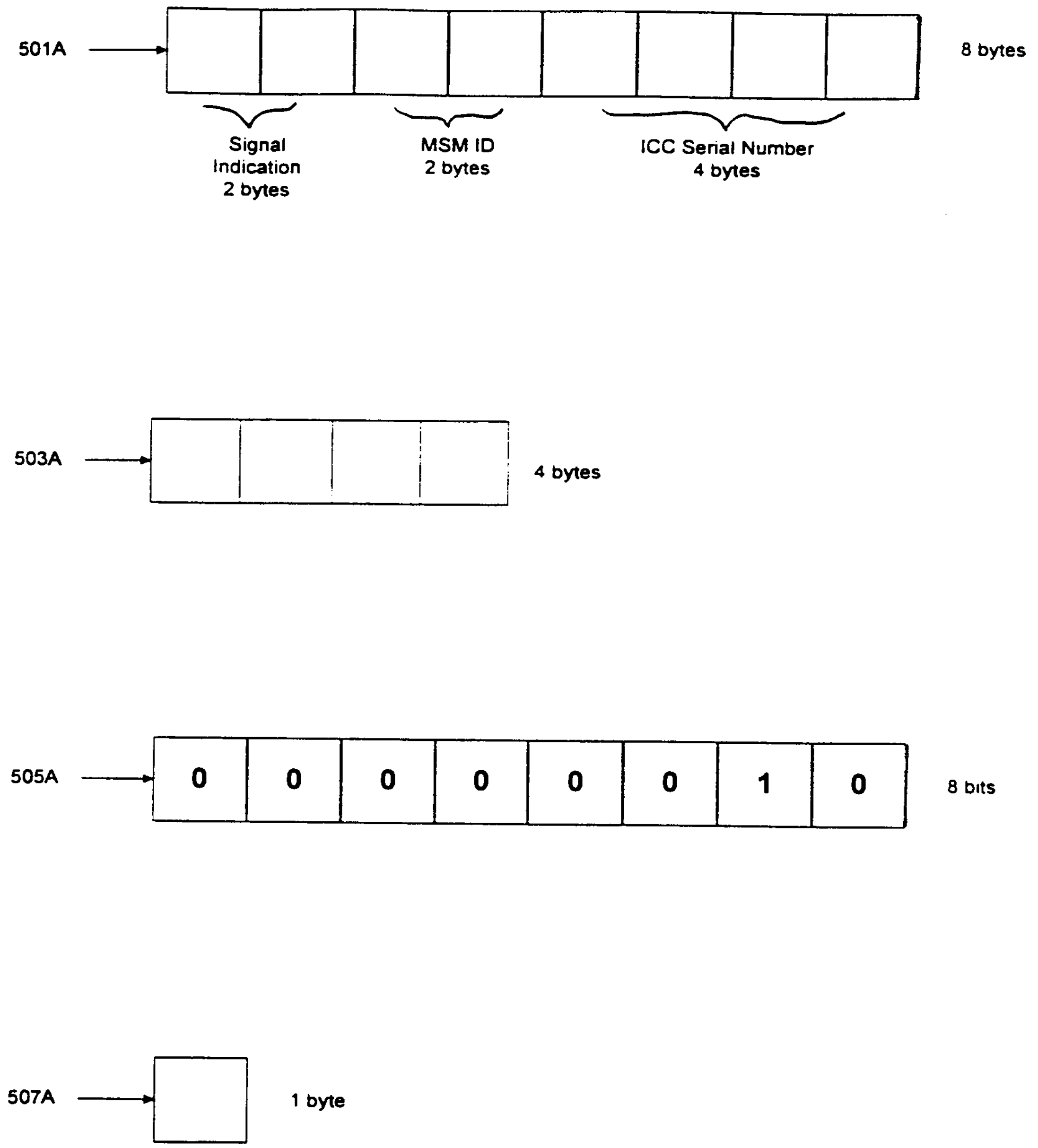

501A
8 bytes
Signal Indication 2 bytes
MSM ID 2 bytes
ICC Serial Number 4 bytes
503A
4 bytes
505A
0 0 0 0 0 0 1 0
8 bits
507A
1 byte

FIG. 5A

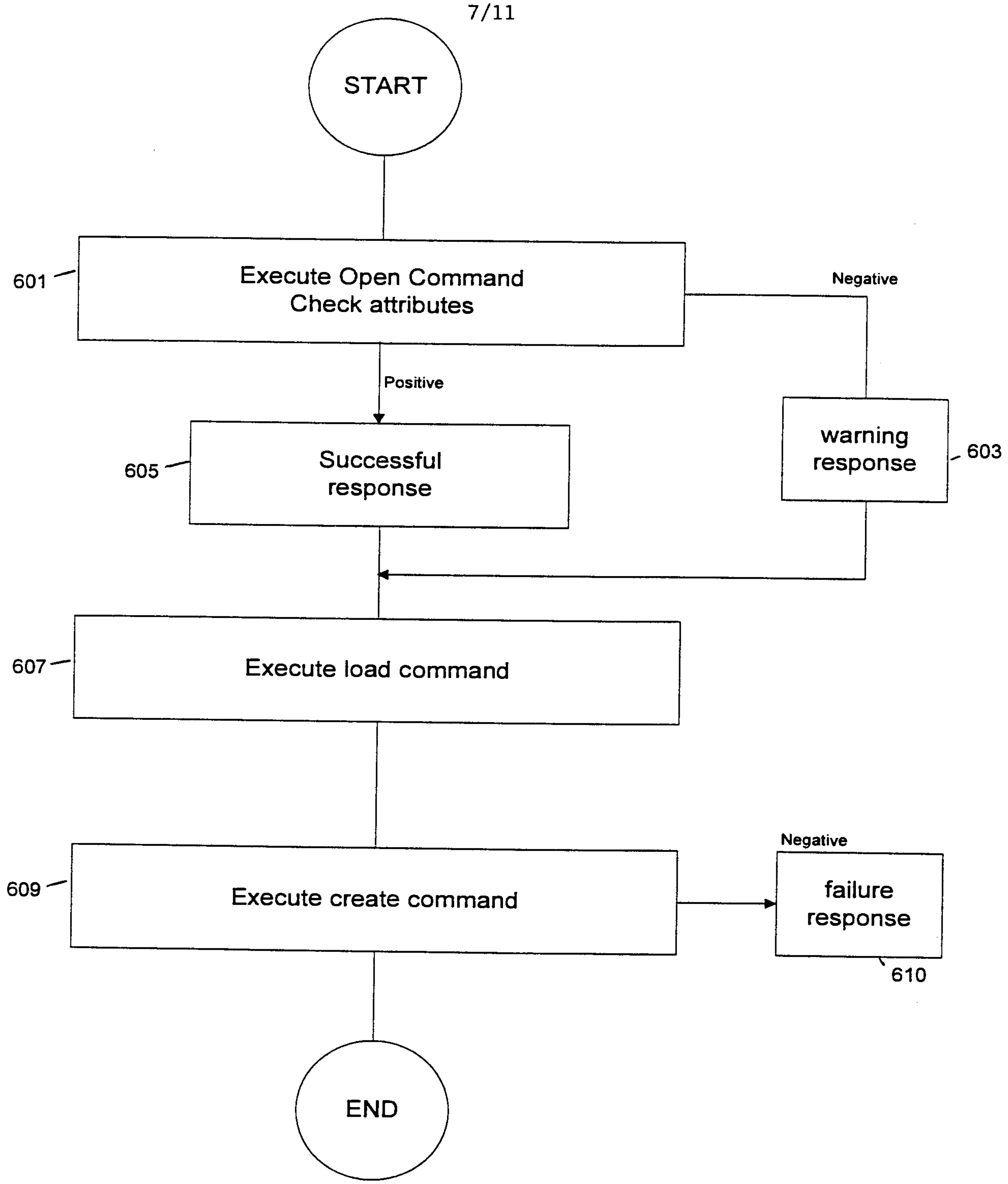

START
601
Execute Open Command
Check attributes
Negative
Positive
605
Successful
response
warning
response
603
607
Execute load command
609
Execute create command
Negative
failure
response
610
END

FIG. 6

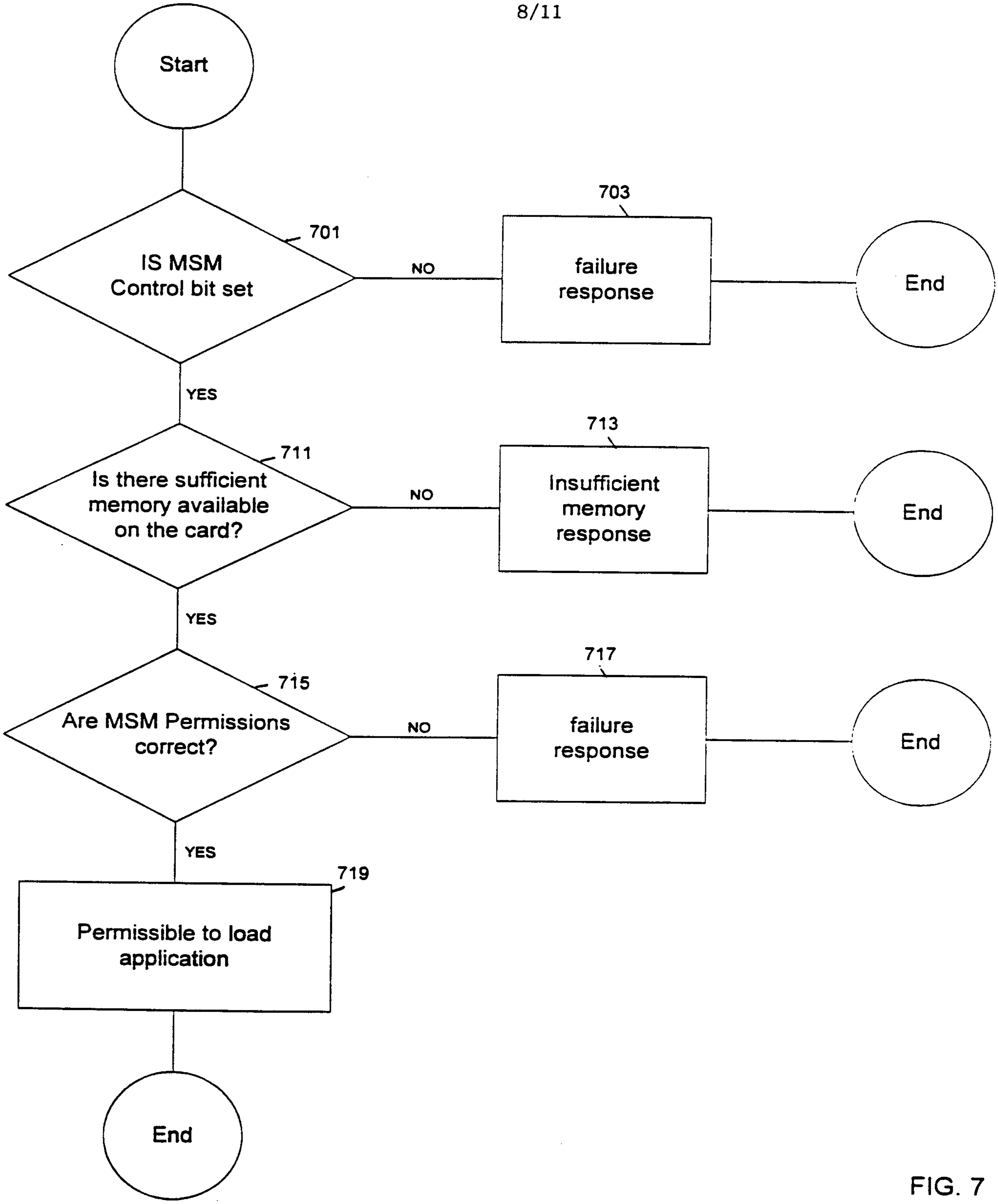

Start
IS MSM Control bit set
701
NO
703
failure response
End
YES
Is there sufficient memory available on the card?
711
NO
713
Insufficient memory response
End
YES
Are MSM Permissions correct?
715
NO
717
failure response
End
YES
719
Permissible to load application
End

FIG. 7

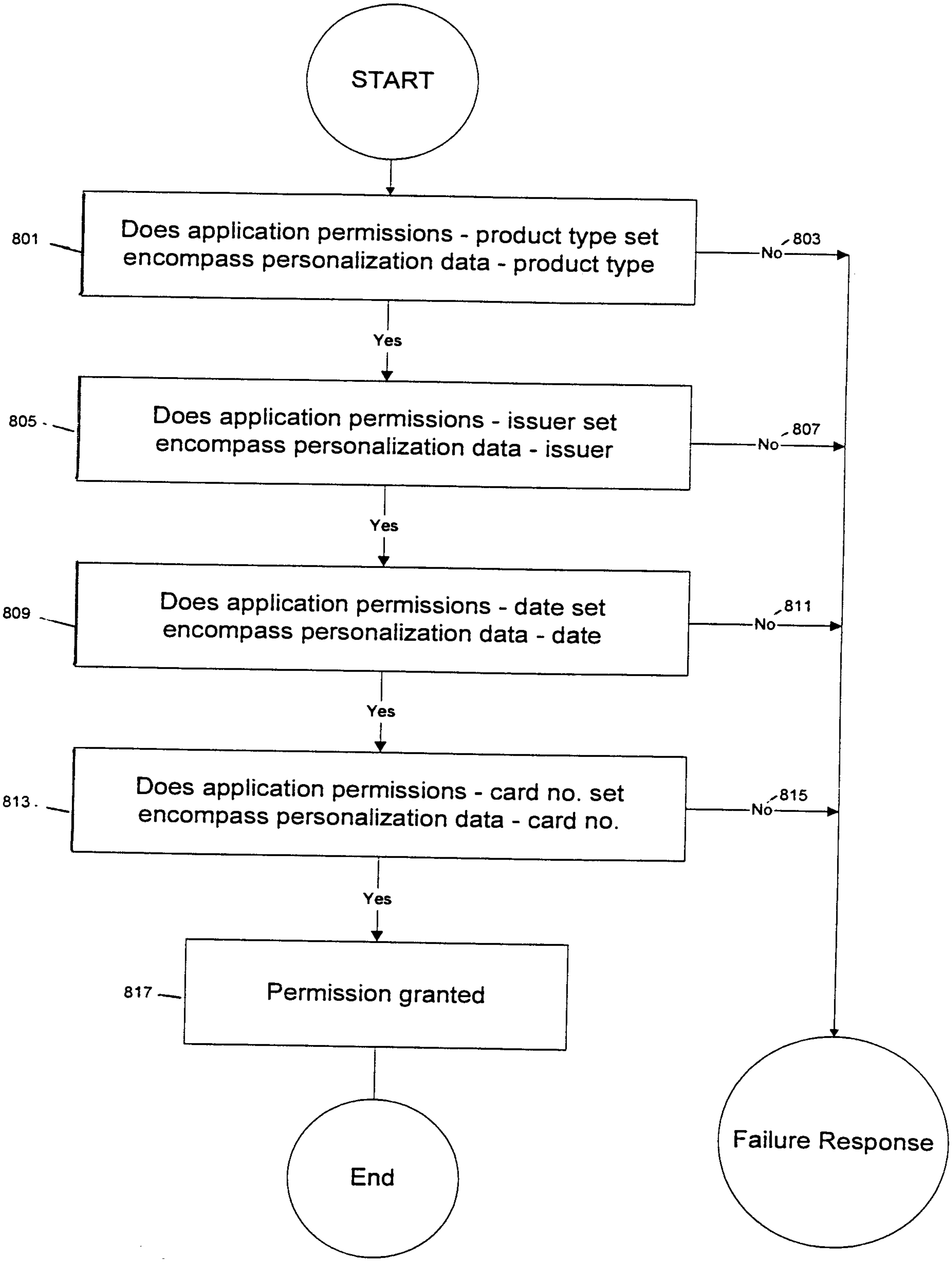

START
801
Does application permissions - product type set encompass personalization data - product type
No
803
Yes
805
Does application permissions - issuer set encompass personalization data - issuer
No
807
Yes
809
Does application permissions - date set encompass personalization data - date
No
811
Yes
813
Does application permissions - card no. set encompass personalization data - card no.
No
815
Yes
817
Permission granted
End
Failure Response

FIG. 8

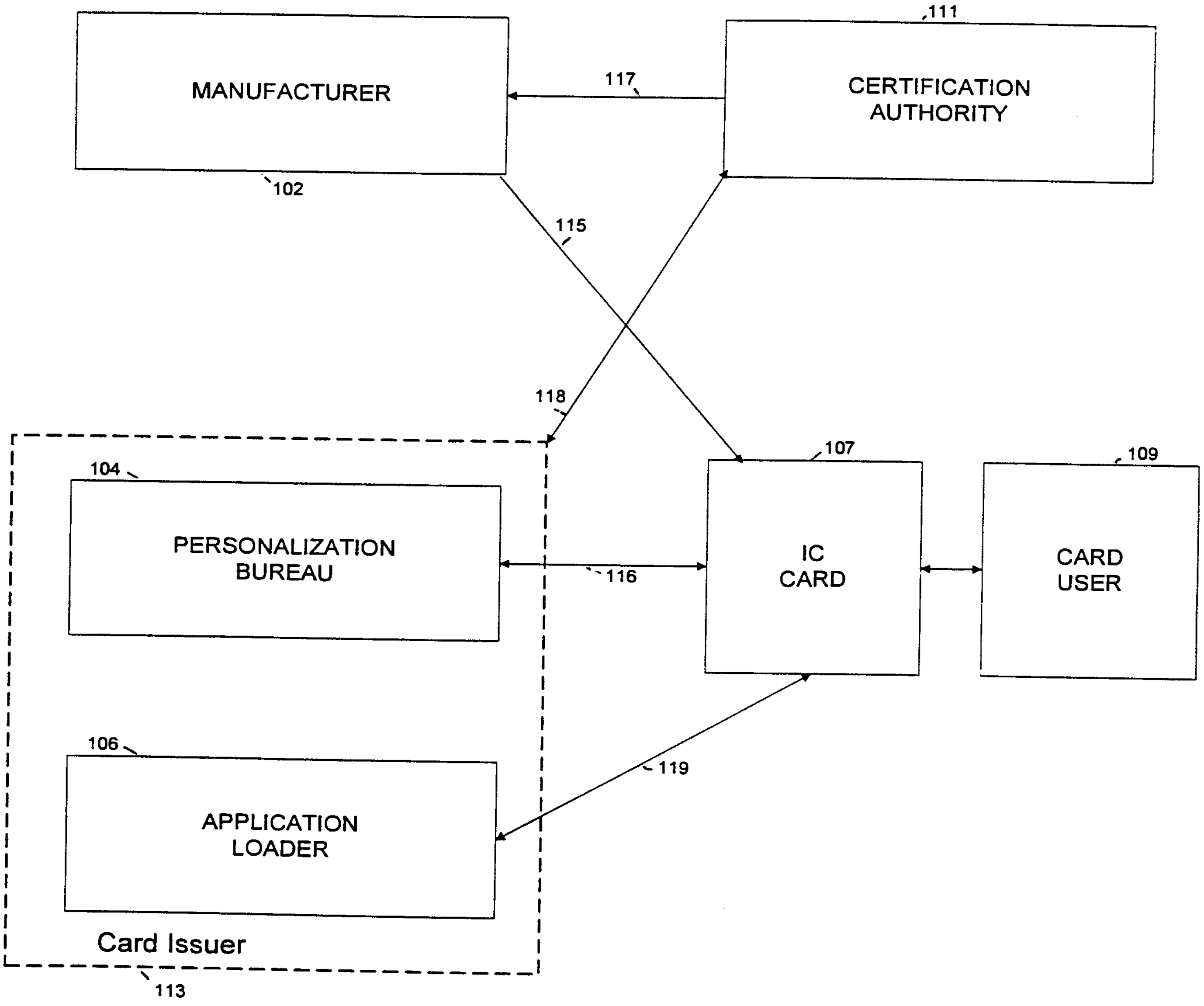

MANUFACTURER
102
117
CERTIFICATION AUTHORITY
111
115
118
104
PERSONALIZATION BUREAU
116
IC CARD
107
CARD USER
109
106
APPLICATION LOADER
119
Card Issuer
113

FIG. 9

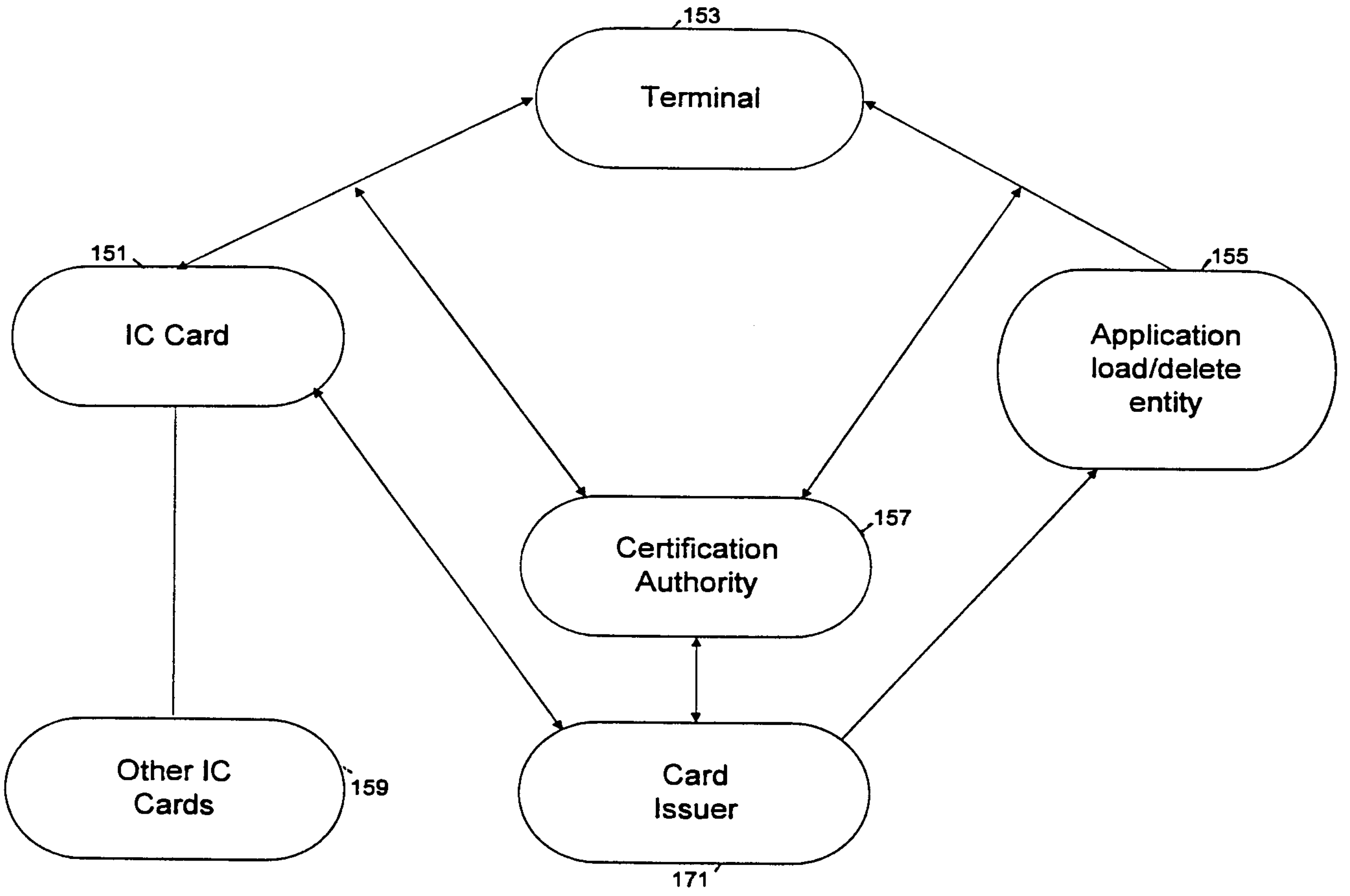

153
Terminal
151
IC Card
155
Application load/delete entity
Certification Authority
157
Other IC Cards
159
Card Issuer
171

FIG. 10

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G07F 7/10, G06K 19/073	A1	(11) International Publication Number: WO 98/43212 (43) International Publication Date: 1 October 1998 (01.10.98)

(21) International Application Number: PCT/US98/05674

(22) International Filing Date: 24 March 1998 (24.03.98)

(30) Priority Data:

60/041,468	24 March 1997 (24.03.97)	US
60/061,763	14 October 1997 (14.10.97)	US

(71) Applicant *(for all designated States except US)*: VISA INTERNATIONAL SERVICE ASSOCIATION [US/US]; 900 Metro Center Boulevard, Foster City, CA 94404 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: CHAN, Alfred [US/US]; 298 Cerro Drive, Daly City, CA 94015 (US). KEKICHEFF, Marc, B. [FR/US]; 2901 Simkins Court, Palo Alto, CA 94303 (US). WEISE, Joel, M. [US/US]; 3 Kenmar Way, Burlingame, CA 94010 (US). WENTKER, David, C. [US/US]; 933 Stanyan Street, San Francisco, CA 94117 (US).

(74) Agent: YI, Susan, C.; Beyer & Weaver, LLP, P.O. Box 61059, Palo Alto, CA 94306 (US).

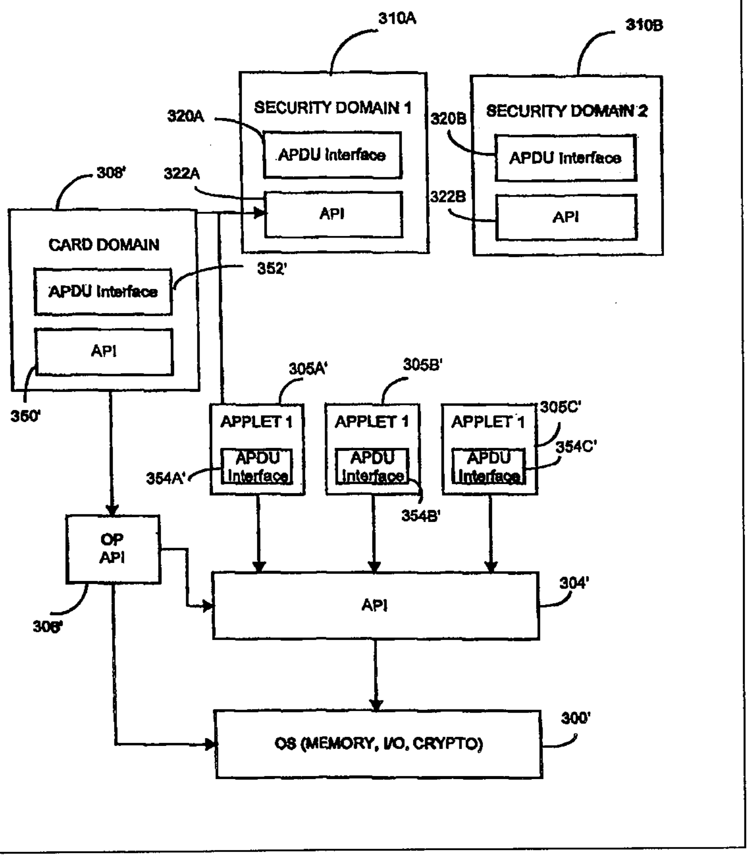
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU,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

Published
With international search report.

(54) Title: A SYSTEM AND METHOD FOR A MULTI–APPLICATION SMART CARD WHICH CAN FACILITATE A POST–ISSUANCE DOWNLOAD OF AN APPLICATION ONTO THE SMART CARD

(57) Abstract

The embodiments of the present invention teaches a system and method which allows card issuers to securely add applications (305A–305C) during the lifetime of the card (304) after the card has already been issued (post issuance). The system and method according to embodiments of the present invention allows the loading of an application and/or objects from an application server via a card acceptance device and its supporting system infrastructure delivery mechanism, onto a card post issuance in a secure and confidential manner.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

A SYSTEM AND METHOD FOR A MULTI- APPLICATION SMART CARD WHICH CAN FACILITATE A POST-ISSUANCE DOWNLOAD OF AN APPLICATION ONTO THE SMART CARD

FIELD OF THE INVENTION

The present invention relates to smart cards. In particular, the present invention relates to a system and method for providing a multi-application smart card which can facilitate a post-issuance download of an application onto the smart card.

BACKGROUND OF THE INVENTION

A smart card is typically a credit card-sized plastic card that includes a semiconductor chip capable of holding data supporting multiple applications.

Physically, a smart card often resembles a traditional “credit” card having one or more semiconductor devices attached to a module embedded in the card, providing contacts to the outside world. The card can interface with a point-of-sale terminal, an ATM, or a card reader integrated into a telephone, a computer, a vending machine, or any other appliance.

A micro-controller semiconductor device embedded in a “processor” smart card allows the card to undertake a range of computational operations, protected storage, encryption and decision making. Such a micro-controller typically includes a microprocessor, memory, and other functional hardware elements. Various types of cards are described in “The Advanced Card Report: Smart Card Primer”, Kenneth R. Ayer and Joseph F. Schuler, The Schuler Consultancy, 1993.

One example of a smart card implemented as a processor card is illustrated in FIG. 1. Of course, a smart card may be implemented in many ways, and need not necessarily include a microprocessor or other features. The smart card may be programmed with various types of functionality, including applications such as stored-value; credit/debit; loyalty programs, etc.

In some embodiments, smart card 5 has an embedded micro-controller 10 that includes a microprocessor 12, random access memory (RAM) 14, read-only memory (ROM) 16, non-volatile memory 18, a cryptographic module 22, and a card reader interface 24. Other features of the micro-controller may be present but are not shown, such as a clock, a random number generator, interrupt control, control logic, a charge pump, power connections, and interface contacts that allow the card to communicate with the outside world.

Microprocessor 12 is any suitable central processing unit for executing commands and controlling the device. RAM 14 serves as storage for calculated results and as stack memory. ROM 16 stores the operating system, fixed data, standard routines, and look up tables. Non-volatile memory 18 (such as EPROM or EEPROM) serves to store information that must not be lost when the card is disconnected from a power source but that must also be alterable to accommodate data specific to individual cards or any changes possible over the card lifetime. This information might include a card identification number, a personal identification number, authorization levels, cash balances, credit limits, etc. Cryptographic module 22 is an optional hardware module used for performing a variety of crptographic algorithms. Card reader interface 24 includes the software and hardware necessary for communication with the outside world. A wide variety of interfaces are possible. By way of example, interface 24 may provide a contact interface, a close-coupled interface, a remote-coupled interface, or a variety of other interfaces. With a contact interface, signals from the micro-controller are routed to a number of metal contacts on the outside of the card which come in physical contact with similar contacts of a card reader device.

Various mechanical and electrical characteristics of smart card 5 and aspects of its interaction with a card reading device are defined by the following specifications, all of which are herein incorporated by reference.

Visa Integrated Circuit Card Specification, (Visa International Service Association 1996).

EMV Integrated Circuit Card Specification for Payment Systems, (Visa International Service Association 1996).

EMV Integrated Circuit Card Terminal Specification for Payment Systems, (Visa International Service Association 1996).

EMV Integrated Circuit Card Application Specification for Payment Systems, (Visa International Service Association 1996).

International Standard; Identification Cards - Integrated Circuit(s) Cards with Contacts, Parts 1-6 (International Standards Organization 1987-1995).

Prior to issuance of a smart card to a card user, the smart card is initialized such that some data is placed in the card. For example, during initialization, the smart card may be loaded with at least one application, such as credit or stored cash value, a file structure initialized with default values, and some initial cryptographic keys for transport security. Once a card is initialized, it is typically personalized. During personalization, the smart card is loaded with data which uniquely identifies the card. For example, the personalization data can include a maximum value of the card, a personal identification number (PIN), the currency in which the card is valid, the expiration date of the card, and cryptographic keys for the card.

A limitation of conventional smart cards is that new applications typically can not be added to an issued smart card. Smart cards are traditionally issued with one or more applications predefined and installed during the manufacturing process of the card. As a result, with traditional smart card implementation, once a card has been issued to a card user, the smart card becomes a fixed application card. If a new application is desired, the smart card is typically discarded and a new smart card, which includes the new application, is issued.

It would be desirable to provide a smart card which would allow applications to be loaded after the card is issued. Further, it is desirable to provide a mechanism to manage the loading of an application as well as general management of the applications on the smart card. Additionally, it is desirable to allow an application provider to keep cryptographic keys confidential from the issuer of the smart card and to securely allow application from different entities to coexist on a card.

Summary of the Invention

Embodiments of the present invention teach a system and method which allow card issuers to add applications during the lifetime of the card after the card has already been issued (referred to herein as post issuance loading). Downloading an application after the card has been issued to the card holder will be referred to herein as a "secure install" process.

The system and method according to embodiments of the present invention allow the loading of an application and/or objects from an application server via a card acceptance device and its supporting system infrastructure delivery mechanism, onto a card, post issuance in a secure and confidential manner.

An embodiment of the present invention provides a system and method for controlling at least one function associated with an issued smart card. In a multi-application smart card, a privileged application, herein referred to as a card domain, manages multiple functions related to the smart card. Examples of these functions include card initialization, global card data, card life cycle, and secure installation of smart card applications.

A method according to an embodiment of the present invention for providing a first application onto an issued smart card comprises the steps of forwarding the first application to the issued smart card; and loading the first application onto the issued smart card, wherein the loading of the first application is managed by a second application.

In another aspect of the invention, a system according to an embodiment of the present invention for controlling at least one function associated with an issued smart card is disclosed. The system comprises a first application associated with the issued smart card; and a second application associated with the issued smart card, the second application being in communication with the first application, wherein the second application manages at least one function associated with the first application.

Furthermore, an embodiment of the present invention provides a system and method for providing confidential information to an application in a smart card. In a multi-application smart card, a privileged application, herein referred to as a security domain, is utilized as a confidential representative of an application provider. The security domain can contain cryptographic keys which can be kept confidential from the smart card issuer, thus allowing separation of cryptographic security between the issuer and the application provider. When a new application is loaded onto a smart card, the newly loaded application can utilize its associated security domain's cryptographic service. A privileged application representing the issuer, herein referred to as a card domain, can approve of commands, such as commands for initialization and personalization, by invoking the security domain's cryptographic service. In this manner, a post issuance download of an application onto the issued smart card can be accomplished.

A method according to an embodiment of the present invention for providing confidential information to an application in a smart card is presented. The method comprises the steps of providing a first application in the smart card, the first application including a cryptographic service; loading a second application onto the smart card; and installing the second application, wherein the cryptographic service of the first application is utilized to install the second application.

In another aspect of the invention, a system according to an embodiment of the present invention for providing confidential information to an application in a smart card is presented. The system comprises a first application associated with the issued smart card, wherein the first application includes cryptographic service; and a second application associated with the issued smart card, the second application being in communication with the first application, wherein the cryptographic service included in the first application is utilized for at least one function related to the second application.

In yet another aspect of the invention, a method according to an embodiment of the present invention for providing an application to a smart card is presented. The method comprising the steps of issuing a smart card; loading a first application onto the issued smart card; and initializing the first application.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a smart card system suitable for implementing the present invention.

Figure 2 is an example of a block diagram of software layers which can be utilized in a smart card.

Figures 3A - 3B are block diagrams of examples of software layers according to embodiments of the present invention.

Figure 4 is a flow diagram of an example of a method according to an embodiment of the present invention for installing an application onto an issued smart card utilizing a card domain.

Figure 5 is a flow diagram of a method according to an embodiment of the present invention for providing confidential information to an application in a smart card using security domains.

Figure 6 is a flow diagram of an example of a method according to an embodiment of the present invention for installing an application onto an issued smart card utilizing a card domain.

Figure 7A is a flow diagram illustrating a sequence of card life states.

Figure 7B is a flow diagram illustrating a sequence of card life states.

Figure 8 is an illustration of an example of a card life cycle.

Figure 9 is a flow diagram of an example of a method according to an embodiment of the present invention for blocking a card utilizing a card domain.

Figure 10 is a block diagram illustrating interactions between a card domain and a security domain on a smart card according to an embodiment of the present invention.

Figures 11A and 11B are flow diagrams of an example of a method according to an embodiment of the present invention for loading an application by using a security domain after the smart card has issued.

Figures 12A-12B are flow diagrams of an example of a method according to an alternate embodiment of the present invention for loading an application using a security domain after the smart card has issued.

Figure 13 is a block diagram illustrating an example of key management and key dependencies for post issuance download of applications onto the smart card.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The following description is presented to enable one of ordinary skill in the art to make and to use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiments will be readily apparent to those skilled in the art and the generic principles herein may be applied to other embodiments. Thus, the present invention is not intended to be limited to the embodiment shown but is to be accorded the widest scope consistent with the principles and features described herein.

Figure 2 is a block diagram of an example of software layers which can be utilized in a smart card. The smart card shown in Figure 2 includes an operating system 200, a card application programming interface (API) 204, and applications 206A-206B. Operating system 200 can include functionality to control the cards, memory management, input/output (I/O), and cryptographic features. Card API 204 utilizes the instructions from operating system 200 and writes these instructions into blocks which can be reused for common routines in multiple applications. Applications 206A and 206B can run on the smart card via instructions from API 204. These applications can include any application which can run on a smart card, such as stored value, credit, debit, transit, and loyalty.

One embodiment of the present invention is based upon the Java Card standard. In this case applications are referred to as 'Applets' and they are written to link to a Java Card API which is the application programming interface present on smart cards built to the Java Card standard.

Although the conventional software system shown in Figure 2 allows for multiple applications, it does not solve the problem of how to load, securely, an application after issuance of the smart card to a user. If an application is to be loaded post issuance, a mechanism is needed to manage the loading of an application as well as general management of the applications on the smart card. Additionally, an application provider may wish to keep cryptographic keys confidential from the issuer of the smart card. Accordingly, a mechanism is needed to provide for the separation of confidential information between an application provider and an issuer of a smart card. Embodiments of the present invention address such a need.

Figures 3A - 3B are block diagrams showing software components of a smart card according to embodiments of the present invention. The arrows indicate dependencies between components. Figure 3A shows an embodiment of a smart card utilizing a card domain, while Figure 3B shows an embodiment of a smart card utilizing a security domain, as well as a card domain.

The example shown in Figure 3A includes an operating system 300, a card API 304, applications 305A-305C, a card domain 308, and open platform (OP) API 306. The system shown in Figure 3 allows for a secure and managed post issuance download of an application onto a smart card.

Open platform API 306 classifies instructions into card domain 308 and security domains 310A-310B (shown in Figure 3B). Accordingly, OP API 306 facilitates the formation of instructions into sets which can be identified as being included as part of card domain 308 and security domains 310A-310B.

Applications 305A-305C can include any application which can be supported by a smart card. Examples of these applications include credit, debit, stored value, transit, and loyalty. Applications 305A-305C are shown to include command interfaces, such as APDU interfaces 354A-354C which facilitate communication with the external environment.

Applications 305A and 305B can run on the smart card via instructions from card API 304. Card API 304 is implemented using the instructions from the card operating system and writes these instructions into blocks which can be reused for common routines for multiple applications. Those skilled in the art will recognize that a translation layer or interpreter may reside between API 304 and operating system 300. An interpreter interprets the diverse hardware chip instructions from vendor specific operating system 300 into a form which can be readily utilized by card API 304.

Card domain 308 can be a "privileged" application which represents the interests of the smart card issuer. As a "privileged" application, card domain 308 may be configured to perform multiple functions to manage various aspects of the smart card. For instance, card domain 308 can perform functions such as installing an application on the smart card, installing security domains 310A-310B (shown on Figure 3B), personalization and reading of card global data, managing card life cycle states (including card blocking), performing auditing of a blocked card, maintaining a mapping of card applications 305A-305C to security domains 310A-310B, and performing security domain functions for applications 305A-305C which are not associated with a security domain 310.

Card domain 308 is shown to include an API interface 350 and a command interface, such as Application Protocol Data Unit (APDU) interface 352. APDU interface 352 facilitates interfacing with the external environment. In compliance with, e.g., International Standards Organization (ISO) Standard 7816-4, entitled "Identification Cards - Integrated circuit(s) cards with contacts - Part 4, Inter-industry commands for interchange," which is herein incorporated by reference.

For example, APDU interface 352 can be used during post issuance installation of an application or during loading of card global data. An application load and install option is performed via a set of appropriate APDU commands received by card domain 308. API interface 350 facilitates interfacing with the internal smart card environment. For example, API interface 350 can by used if card domain 308 is being utilized as a default in place of a security domain 310, or if an application requires information such as card global data, key derivation data, or information regarding card life cycle.

Memory allocations have been performed by the time an application is in an install state. An application is also personalized after loading and installing. A personalized application includes card holder specific data and other required data which allows the application to run. In addition to managing the installation and personalization of the application, card domain 308 can also manage global card information. Global card information includes information that several applications may need to perform their functions, such as card holder name and card unique data utilized in cryptographic key derivations. Card domain 308 can be a repository for the global card information to avoid storing the same data multiple times.

Card domain 308 can also manage card life cycle states including card blocking. The smart card will typically move through several states during its life cycle. Card domain 308 keeps track of what state the card is in during its life cycle. Card domain 308 may also manage a block request to block virtually all functions of the card. Further details of card domain 308 management of a block request will be discussed in conjunction with Figure 6. Card domain 308 may also keep track of the state of an application during an application's life cycle. This kind of information regarding an application can be utilized during an auditing of a card. Auditing can be performed at any time during a card's lifetime. For instance, auditing may be performed after a card has been blocked or prior to installing a new application to validate the card contents. Although virtually all card functions are no longer functioning when a card is blocked, an issuer may be able to query card domain 308 for information regarding a state of an application or the life cycle state of the card. In this manner, the issuer of a card may still access a profile of the blocked card and its applications.

Figure 3B shows an embodiment of the present invention utilizing a security domain 310, as well as card domain 308. The example shown in Figure 3B includes a operating system 300', a card API 304', applications 305A-305C', security domains 310A-310B', a card domain 308', and open platform (OP) API 306'. The system shown in Figure 3B also allows for a secure and managed post issuance download of an application onto a smart card.

Card domain 308' can work in conjunction with a security domain 310. Security domain 310 is a logical construct that can be implemented as an application to provide security related functions to card domain 308' and to applications associated with security domain 310. Security domains 310A-310B can assist in secure post issuance loading of an application onto the smart card. Security domains 310A-310B provide for a mechanism which keeps the application provider's confidential information, such as cryptographic keys, from being disclosed to the issuer of the smart card.

There may be multiple security domains 310 on a smart card, each represented by a unique cryptographic relationship. A security domain 310 is responsible for the management and sharing of cryptographic keys and the associated cryptographic methods which make up the security domain's cryptographic relationship. An application which is loaded to the smart card post issuance can be associated with a security domain, preferably with only one security domain. However, multiple applications may be associated with the same security domain 310. Applications installed on a smart card during the pre-issuance phase may optionally be associated with a security domain 310 on the smart card for purposes of loading confidential personalization data to those applications using security domain 310 keys.

The software for security domain 310 may be installed by the card manufacturer at the time of card manufacturing (e.g., when the ROM is masked), or may be added during initialization or personalization stages. Security domains 310 can be implemented as selectable applications which are isolated from one another and the rest of the system. If security domain 310 is implemented in a Java card as an application, standard Java card security can be relied upon to ensure isolation of security domain 310. In addition, or alternatively, other security mechanisms such as hardware security which can be utilized through OP API 306 implementation. OP API 306 may utilize special security features to enforce isolation of security domain 310. An example of such a security feature is the utilization of chip hardware security routines which may be employed by OP API 306.

Each security domain 310A-310B provides a command interface, such as an Application Protocol Data Unit (APDU) interface 320A-320B, for communication off card and an on card API interface 322A-322B.

The APDU interface 320A-320B consists of personalization commands and is intended to allow the initial loading of security domain keys and to support key rotation if desired during the life of the security domain. API interfaces 322A-322B may include a signature verification method and decryption method which are shared with card domain 308' for post issuance loading of applications. Additionally, applications may utilize API interfaces 322A-322B for decrypting application confidential data. Note that card domain 308' may always function as a security domain and does so as the default.

Security domain 310 manages signing and decrypting keys and provides cryptographic services using those keys. Security domain 310 processes APDU's for numerous functions. These functions can include key management functions e.g., functions to load or update keys. During Secure Installation of an application, security domain 310 can provide services to card domain 308' to decrypt an application install file and check the signature of an application file. For an application associated with a security domain 310, that application's security domain 310 provides decrypt and signature functions, such as MACing on an update key APDU command during the personalization phase of a newly installed application. Thereafter, the application can use the updated key to decrypt and check signatures on subsequent key updates.

The smart card issuer may decide whether security domain 310 utilizes a static key or a session key for transactions. A static key is a cryptographic key which exists prior to processing APDUs and which exist during and after the processing of APDUs. A session key is a cryptographic key which can be generated for a particular transaction and is typically no longer used for APDU processing after the transaction. If a session key is utilized, security domain 310 preferably derives its own session key for processing APDUs.

Figure 4 is a flow diagram of a method accordingly to an embodiment of the present invention for providing an application onto a smart card. The example illustrated in Figure 4 also applies to installing a security domain 310 onto a smart card. Note that all of the flow diagrams in this application are merely examples. Accordingly, the illustrated steps of this and any other flow diagram herein, can occur in various orders and in varying manners in order to accomplish virtually the same goal.

A smart card is issued (step 400), and an application is forwarded to the issued smart card (step 402). The forwarding of the application can occur through any electronic media which can interface with a smart card and connect to an appropriate network. For example, devices such as an automatic teller machine (ATM), a display phone, or a home computer, can be used to forward an application to the issued smart card. The forwarded application is then loaded onto the smart card, wherein the loading of the application is managed by card domain 308 (step 404).

Figure 5 is another flow diagram of a method according to an embodiment of the present invention for providing an application onto an issued smart card. A smart card is created and provided with a first application, the first application including a cryptographic service (step 1002). A second application is loaded onto the smart card (step 1004). Thereafter, the second application is installed, wherein the cryptographic service of the first application is utilized to install the second application (step 1006).

Figure 6 is another flow diagram of an example of a method according to an embodiment of the present invention for providing an application onto an issued smart card. This method for providing an application also applies to providing a security domain 310 onto the smart card. In the example shown in Figure 6, a card issuer deploys smart cards to customers (step 500). A decision is made to install vendor A's application onto the issued smart card (step 502). When a dialogue between the issuer and the smart card is initiated, a pre-signed copy of the application is forwarded to the smart card (step 504). As previously stated, the dialogue between the issuer and the smart card can occur via any electronic device which can interface with a smart card and connect to an appropriate network. The application can be pre-signed with a key equivalent to that which already exists on the card so that each application has a unique signature that can be verified by the card.

Card domain 308 can then take the steps to load the application. Card domain 308 decrypts the forwarded application and checks the signature of the application (step 508). Card domain 308 can decrypt the application with the issuer's secret key. An appropriate cryptography method, such as Data Encryption Standard (DES) or 3DES, can be utilized to decrypt at least a portion of the application. Those skilled in the art will recognize that a number of cryptographic techniques may be used to implement embodiments of the present invention. For the purpose of illustration, symmetric key techniques are addressed herein, although asymmetric techniques are also contemplated. A good general cryptography reference is Schneier, Applied Cryptography, 2d Ed. (John Wiley, 1996), the contents of which are incorporated herein by reference.

It is then determined whether the signature on the application is valid (step 510). If the signature associated with the application is not valid, then the application is not loaded onto the card and the process ends (step 520). If, however, the signature associated with the application is valid the application is then installed and available for personalization. During personalization the application receives personalization data (step 512). Personalization data includes data which is unique to the smart card user. For instance, in a airline loyalty application, personalization data can include the smart card user's seating preference, meal preference, and eligibility for various possible perks. This personalization data can also be signed and encrypted.

The application then invokes card domain's 308 decryption service (step 513). Card domain 308 can then performs a signature check (step 514). Methods of decrypting personalization data and performing signature checks are well known in the art. Finally, the application can then be activated (step 518).

A new application which as been downloaded onto a smart card post-issuance can be stored in a variety of ways. One example is to store the application into a file. Another example is to maintain a pointer to the application object.

Figure 7A is a flow diagram illustrating an example of a sequence of card life states. The sequence is preferably considered irreversible. The first card life state is when the smart card is Masked (700). During the Masked state (700), the smart card obtains its operating system, card identification, and preferably at least one application. The Masked state (700) is achieved as soon as all of the necessary components for card initialization are made available. An example of when necessary components are made available is when card domain 308 and OP API 306 are enabled, as well as the Java card environment being enabled, such as Java card virtual machine 302 and Java card API 304 (both of Figure 3).

After the Masked state, the next state is the Initialized (step 702) state. The Initialized state is achieved once all card activity requiring an initialization key is complete. As part of card initialization, if not already available, the card domain 308 application must be installed and registered. In addition , one or more security domains may also be installed and registered. These installed domains must then be selected and personalized. An initialization key is a secret key which is typically used by a smart card manufacturer during loading of data onto the smart card prior to issuance.

The next state is Load Secured (step 704). The Load Secured state is achieved after a secure install (post-issuance download) mechanism for loading of applications through the remainder of the card lifetime has been established.

The final card life state is when the card is either expired or blocked (step 706). The blocked state is achieved as soon as an authorized smart card application has received a command to block the card.

The card life cycle is preferably an irreversible sequence of states with increasing security. Initialization and all subsequent card life cycle states and their transition are preferably under the control of card domain 308. Card domain 308 executes and responds to commands that result in a transition in a card life cycle from one state to the next. These commands are preferably Application Protocol Data Unit (APDU) commands. Card domain 308 is also responsible for the installation of applications on the card, but preferably has no control over the applications' life cycle states. Each application is preferably responsible for its own application life cycle state management but it preferably allows card domain 308 to have access to its life cycle states for auditing purposes.

The Card Life cycle is designed in such a way to increase the level of security enforced by the card at each successive state. As stated above, the cycle is also established as a process which can only ratchet forward to ensure that once the card begins a life cycle state with associated security policies, the only option is to cycle forward to the next state in the life cycle with a higher level of security. The Card Domain as the system security manager of the card maintains the current life cycle state, enforces the associated security policies, and controls the state transitions in the Card life cycle.

Figure 7B is a flow diagram illustrating an example of a sequence of an application life cycle. The application is initially unavailable (step 750). The next state is a loaded state (step 752). The application reaches the loaded state once the application has been loaded onto the smart card. The application is then installed (step 754), and registered (step 756). Once the application is registered, it can be deleted at any time thereafter. The next state is the personalized state, wherein personalized information is included in the application (step 758). Finally, the application may expire or be blocked (step 760).

Figure 8 is an illustration of an example of multi-application card life time line. This time line starts with a Masked ROM stage 800 and ends with a card blocked/expired stage 802. At Masked ROM stage 800, applications A, B, C and D are shown to be installed. This example shows applications A and B being installed at a masking stage of the card, applications C and D being installed at initialization stage, and applications D and F being installed post issuance.

In this example, application A can be installed in ROM and used during the complete life of the card from Masked ROM stage 800 to card blocked/expired stage 802. Application B is also in ROM and utilized during a first portion of the life of the smart card. The life of application B is ended at stage 804A. Application C is located in non-volatile memory, such as EEPROM, which is loaded during initialization. Application C is shown to expire at stage 804B. Application D is also located in EEPROM and is used for the complete life of the card until card blocked/expired stage 802. Application E is installed at stage 806A, sometime after issuance of the smart card. Application E is located in EEPROM and used until the end of the card life at card blocked/expired stage 802. Application F is also installed post issuance at stage 806B, and expires sometime before the end of the card life at stage 804C.

Figure 9 is a flow diagram of a method according to an embodiment of the present invention for blocking a card. A card be can be blocked if a breach of security is detected by an application. According to an embodiment of the present invention, a smart card can be blocked while an application is in use. A blocked card will no longer operate so that a suspect user cannot utilize any of the applications on the smart card. Blocking is merely one example of the many functions card domain 308 can perform in managing the other applications on the smart card. Examples of other functions include installing an application on the smart card, installing security domains 310A-310B, personalization and reading of card global data, managing card life cycle states including card blocking, performing auditing of a block card, maintaining a mapping of card applications to security domains, and performing security domain functions for applications which are not associated with a security domain.

In the example shown in Figure 9, an application is currently in use (step 600). The application detects a problem which triggers a card block request from the application (step 602). The application then sends a card block request to card domain 308 (step 604). Card domain 308 determines whether the card block request is valid (step 606). A card block request can be valid if the request originates from a predetermined application. If the card block request is not valid, the card domain 308 does not block the smart card (step 608). However, if the card block request is valid, then card domain 808 authorizes the card blocking (step 610), and card domain 308 blocks the smart card (step 612) such that the smart card will reject any attempted transactions for any of the applications on the card.

Figure 10 is a block diagram illustrating the use of security domain 310 by the card domain 308. The method and system according to an embodiment of the present invention allows for multiple application providers to be represented on a smart card in a secure and confidential manner. This security and confidentiality can be achieved through the use of security domain 310A-310B shown in Figure 3.

Figure 10 illustrates an example of a smart card which contains two security domains 310A-310B. In this example, it is assumed that a masked application 305A from the smart card is associated with a security domain, such as security domain 310A, and an additional application 305B will be added post issuance and be associated with a second security domain, such as security domain 310B. The arrows indicate key relationships between the various smart card entities. Masked application 305A uses key services from security domain 310A for decrypting confidential data and optionally for full personalization. Card domain 308 uses key services from security domain 310B for decrypting and checking the signature of an application loaded post issuance, such as post issuance loaded application 305B. Post issuance loaded application 305B uses key services from security domain 310B for decrypting confidential data and optionally for full personalization.

Figures 11A and 11B are further flow diagrams of an example for a method according to an embodiment of the present invention for providing an application onto an issued smart card. The card issuer decides to include a security domain 310 onto a smart card (step 1100). The issuer assigns security domain 310 to vendor A (step 1102). Vendor A, or an application developer on behalf of vendor A, generates cryptographic keys such as those used in symmetric or asymmetric cryptography operations (step 1104). Examples of these cryptography operations include encryption, decryption, MACing, Hashing, and digital signatures. Examples of cryptographic methods which utilize such keys and are suitable for implementation for the embodiment of the method and system of the present invention include Data Encryption Standard (DES) and 3DES. The card personalization agent receives the keys and loads security domain keys associated with a specific security domain 310 for each smart card (1106). The card personalization agent receives smart cards and collects other data, such as application and card holder specific data, and places data on the smart card (step 1108).

The card issuer then deploys the smart card to customers (step 1110). A decision is then made to install vendor A's application on the smart card (step 1112). When a dialogue between the smart card issuer and the smart card is initiated, a signed copy of the application is forwarded to the smart card (step 1114). The application can be signed with a key equivalent to that which already exists on the smart card so that each application has a unique signature that can be verified by the smart card.

The smart card's card domain 308 then takes steps to load the application. Card domain 308 invokes an associated security domain's cryptographic service to decrypt the application and check the signature (step 1118). It is then determined if the signature is valid (step 1120). If the signature is not valid, the process ends (step 1122). If, however, the signature is found to be valid, then the application receives personalization data which can be signed and optionally encrypted (step 1124). The loaded application then invokes its associated security domain's decryption service and signature check (step 1126). Secret keys required to run or operate the application on the smart card are used to activate the application by authentication (step 1130).

Figures 12A and 12B are flow diagrams of a method according to another embodiment of the present invention for providing confidential information to an application using a security domain 310. The issuer decides to include a security domain 310 on a smart card (step 1200). A trusted party generates secret cryptographic keys and sends the keys to a card personalization agent in a secure manner (step 1201). A trusted party is typically a third party who performs the function of certifying the source of information, such as a signature. A card personalization agent (which may be the same as the trusted party) receives the key and loads a unique secure domain key associated with a specific security domain 310 for each smart card (step 1202).

The card personalization agent receives the smart card and collects other data, such as application and card holder specific data, and places the data on the smart card (step 1204). The issuer then deploys the smart card to its customers (step 1206). A decision is made to install vendor A's application on the issued smart card (step 1208). Vendor A obtains secret keys for security domain 310 from the trusted party (step 1210). Vendor A then sends the smart card issuer a signed copy of Vendor A's application (step 1212).

When a dialogue between the smart card issuer and the smart card is initiated, a signed copy of the application is forwarded to the smart card (step 1214). The application can be signed with a key equivalent to that which already exists on the smart card so that each application has a unique signature that can be verified by the smart card. Card domain 308 invokes security domain's cryptographic service to decrypt the associated application and check its signature (step 1218). It is then determined whether the signature is valid (step 1220). If the signature is not valid, then the process ends (step 1222).

If, however, the signature is valid, then the application receives personalization data, which can be signed and optionally encrypted (step 1224). The loaded application then invokes security domain's decryption service and signature check (step 1226). The cryptographic secret data required to run or operate the application on the card are used to activate the application (step 1230).

Figure 13 is a block diagram illustrating the use of cryptographic keys for post issuance loading of an application onto a smart card. Applications that are not masked and not loaded during card initialization stage or personalization stage need their executables downloaded using a secure installation method, such as the post issuance download described in previous Figures. The applications can be loaded using the card domain cryptographic keys. The applications are then decrypted and can have their signature verified using the key services of the corresponding security domain 310. Therefore, the desired security domain(s) 310 preferably have encryption and signature keys installed prior to the post issuance download of the corresponding application.

In the example shown in Figure 13, only one security domain 310 is shown since security domains 310 for other applications are not relevant to illustrate the downloading of a single application. Note that the result of the secure installation is initially a loaded application, which must then be installed, registered and personalized. After loading, the application is installed, preferably by issuing an install APDU command to card domain 308. An application can be installed when its install method has executed successfully. Memory allocations have been performed by the time an application is in an install state. A loaded application should also be registered. When an application is registered, it is selectable and it is ready to process and respond to APDU commands. Installation and registration may be performed simultaneously by the same APDU command. An application is also personalized after loading. A personalized application includes card holder specific data and other required data which allows the application to run.

In the example shown in Figure 13, the cryptographic key and MAC/Signature key are shown to be included in the functions of card domain 308/security domain 310. If a security domain is associated with the application being loaded, then the security domain will be invoked. However, if no security domain 310 is associated with the application which is being loaded, then the cryptographic key and the signature key of card domain 308 will be utilized. In contrast to the install commands sent to the smart card during the initialization phase, the post issuance install command is not issued in a secured environment, therefore it is preferably protected with a cryptographic key, such as a MAC/Signature key. Card domain 308 manages the post-issuance loading of a new application, while secure domain 310 ensures the validity and integrity of the new application once the new application has been loaded onto the smart card. If a secure domain 310 is not associated with the newly loaded application, then card domain 308 performs secure domain's 310 functions. Once the new application is post-issuance downloaded, various keys, such as an cryptographic key and a signature key, are preferably utilized for installation and personalization of the application.

A method and system for a smart card domain and a security domain has been disclosed. Software written according to the present invention may be stored in some form of computer-readable medium, such as memory or CD-ROM, or transmitted over a network, and executed by a processor.

Although the present invention has been described in accordance with the embodiment shown, one of ordinary skill in the art will readily recognize that there could be variations to the embodiment and these variations would be within the spirit and scope of the present invention. Accordingly, many modifications may be made by one of ordinary skill in the art without departing from the spirit and scope of the appended claims.

CLAIMS

1. A method for providing a first application onto an issued smart card, the method comprising:

forwarding the first application to the issued smart card; and

loading the first application onto the issued smart card, wherein the loading of the first application being managed by a second application.

2. The method of claim 1, further including a step of decrypting at least a portion of the first application.

3. The method of claim 2, wherein the decryption is provided by the second application.

4. The method of claim 1, further including a step of checking a signature associated with the first application.

5. The method of claim 4, wherein the checking of the signature is performed by the second application.

6. The method of claim 1, further including a step of providing personalization data to the first application.

7. The method of claim 6, further including a step of decrypting the personalization data provided to the first application.

8. The method of claim 7, wherein the decryption is provided by the second application.

9. The method of claim 6, further including a step of checking a signature associated with the personalization data.

10. The method of claim 9, wherein the checking of the signature is performed by the second application.

11. The method of claim 1, further comprising a step of providing a cryptographic key related to the first application.

12. The method of claim 1, further comprising a step of invoking a third application's cryptography service to decrypt at least a portion of the first application.

13. The method of claim 12, wherein the invoking is performed by the second application.

14. The method of claim 1, further comprising a step of invoking a third application to check a signature associated with the first application.

15. The method of claim 14, wherein the invoking is performed by the second application.

16. The method of claim 1, further comprising a step of invoking a third application's cryptography service to decrypt at least a portion of personalization data associated with the first application.

17. The method of claim 16, wherein the invoking is performed by the second application.

18. A system for controlling at least one function associated with an issued smart card, the system comprising:

a first application associated with the issued smart card; and

a second application associated with the issued smart card, the second application being in communication with the first application, wherein the second application manages at least one function associated with the first application.

19. The system of claim 18, wherein the at least one function includes personalization of the first application.

20. The system of claim 18, wherein the at least one function includes card life-cycle states.

21. The system of claim 18, wherein the at least one function includes card blocking.

22. The system of claim 18, wherein the at least one function includes auditing of a blocked card.

23. The system of claim 18, wherein the at least one function includes maintaining a mapping of the first application to an associated security domain.

24. The system of claim 18, wherein at least one function includes a cryptographic service associated with the first application.

25. The system of claim 18, wherein the second application also manages global data related to the issued smart card.

26. A system for providing a first application onto an issued smart card, the system comprising:

means for forwarding the first application to the issued smart card; and

means for loading the first application onto the issued smart card, wherein the loading of the first application being managed by a second application.

27. A computer program product for providing a first application onto an issued smart card, comprising:

computer code for forwarding the first application to the issued smart card;

computer code for loading the first application onto the issued smart card, wherein the loading of the first application being managed by a second application; and

a computer readable medium that stores the computer codes.

28. The computer program product of claim 27, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, EEPROM, ROM, and data signal embodied in a carrier wave.

29. A method for providing confidential information to an application in a smart card, the method comprising:

providing a first application in a smart card, the first application including a cryptographic service;

loading a second application onto the smart card; and

installing the second application, wherein the cryptographic service of the first application is utilized to install the second application.

30. The method of claim 29, wherein the step of loading the second application is performed after the smart card has issued.

31. The method of claim 29, wherein an association between the first application and the second application is maintained.

32. The method of claim 29, wherein details of the cryptographic service of the first application is kept confidential from an issuer of the smart card.

33. The method of claim 29, wherein the cryptographic service accessed in the first application is used in addition to a second cryptographic service included in a third application to perform the step of loading the second application.

34. The method of claim 29, wherein an association between the first application and the second application can be determined after the smart card has been issued.

35. A system for providing confidential information to an application in a smart card, the system comprising:

means for accessing a cryptographic service in a first application, the first application being included in the smart card; and

means for loading a second application in the smart card, wherein the cryptographic service of the first application is utilized to load the second application.

36. A system for providing confidential information to an application in a smart card, the system comprising:

a first application associated with the issued smart card, wherein the first application includes cryptographic service; and

a second application associated with the issued smart card, the second application being in communication with the first application, wherein the cryptographic service included in the first application is utilized for at least one function related to the second application.

37. A computer program product for providing confidential information to an application in a smart card, comprising:

computer code for accessing a cryptographic service in a first application, the first application being included in the smart card; and

computer code for loading a second application in the smart card, wherein the cryptographic service of the first application is utilized to load the second application; and

a computer readable medium that stores the computer codes.

38. The computer program product of claim 37, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, EEPROM, ROM, and data signal embodied in a carrier wave.

39. A system for providing confidential information to an application in a smart card, the system comprising:

a first application associated with an issued smart card, wherein the first application includes cryptographic service;

a second application associated with the issued smart card; and

a third application associated with the issued smart card, the first and second applications being in communication with the third application, wherein the cryptographic service included in the first application is utilized for at least a first function related to the second application.

40. The system of claim 39, wherein the second application invokes the cryptographic service of the first application for utilization on the at least first function related to the second application.

41. The system of claim 39, wherein the second application manages at least a second function of the third application.

42. The system of claim 39, wherein the first application includes a command interface.

43. The system of claim 42, wherein the command interface is an APDU interface.

44. The system of claim 39, wherein the first application includes an API interface.

45. A method for providing an application to a smart card, the method comprising:

issuing a smart card;

loading a first application onto the issued smart card; and

initializing the first application.

46. The method of claim 45, wherein the loading of the application is managed by a second application.

47. The method of claim 46, wherein the second application is included in the issued smart card.

48. The method of claim 45, wherein the initializing of the first application includes a substep of utilizing a cryptographic service of a third application.

49. The method of claim 48, wherein the third application is included in the issued smart card.

50. The method of claim 45, wherein the initializing of the first application includes a substep of invoking a cryptographic service by a fourth application for use by the first application, wherein the cryptographic service is included in a fifth application.

51. The method of claim 50, wherein the fourth and fifth applications are included in the issued smart card.

52. The method of claim 45, further including a step of personalizing the first application.

53. The method of claim 52, wherein the personalization of the first application includes a substep of utilizing a cryptographic service of a sixth application.

54. The method of claim 53, wherein the sixth application is included in the issued smart card.

55. The method of claim 52, wherein the personalization of the first application includes a substep of invoking a cryptographic service by a seventh application for use by the first application, wherein the cryptographic service is included in a eighth application.

56. The method of claim 55, wherein the seventh and eighth applications are included in the issued smart card.

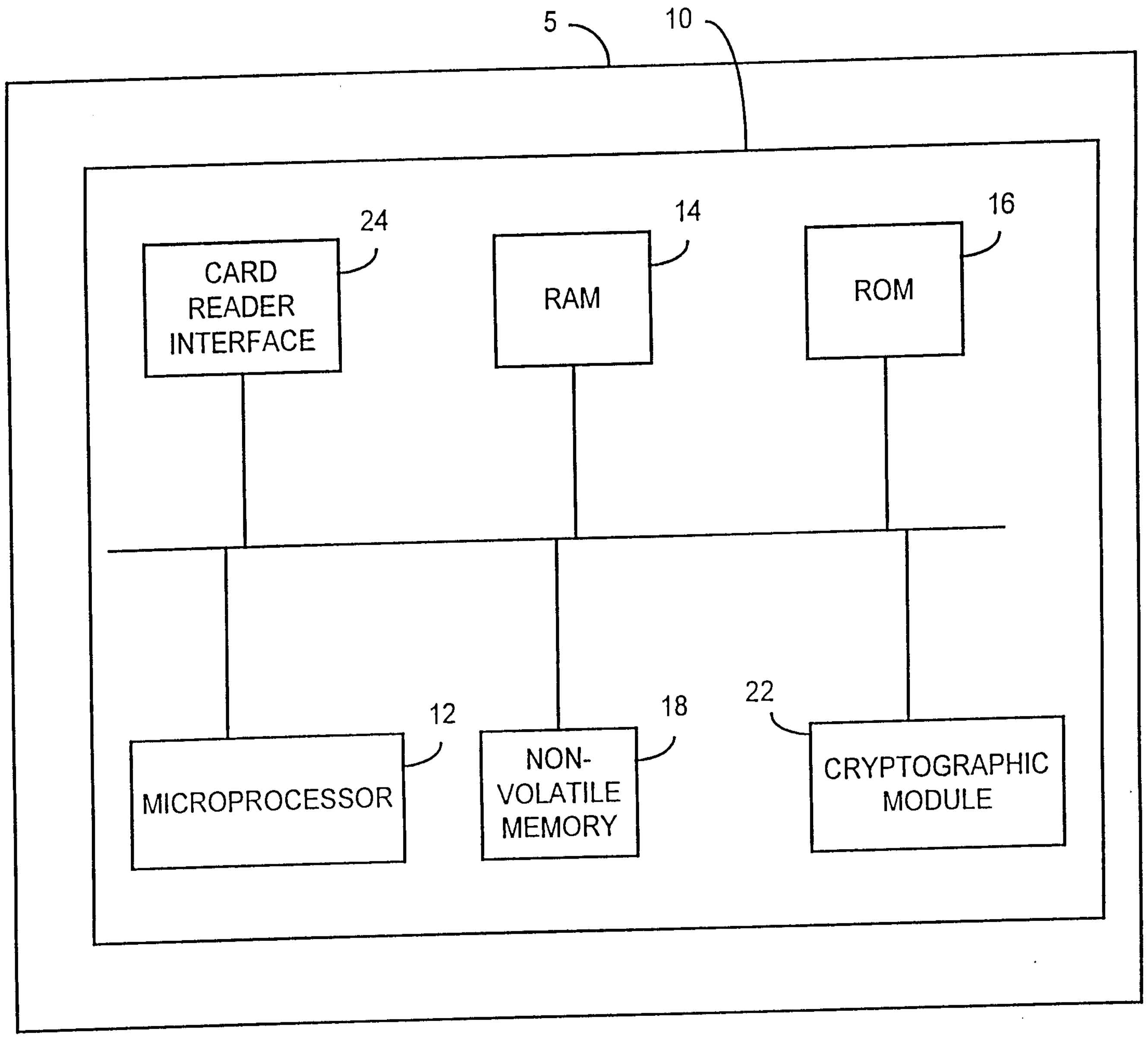

5
10
24
CARD READER INTERFACE
14
RAM
16
ROM
12
MICROPROCESSOR
18
NON-VOLATILE MEMORY
22
CRYPTOGRAPHIC MODULE

FIG 1

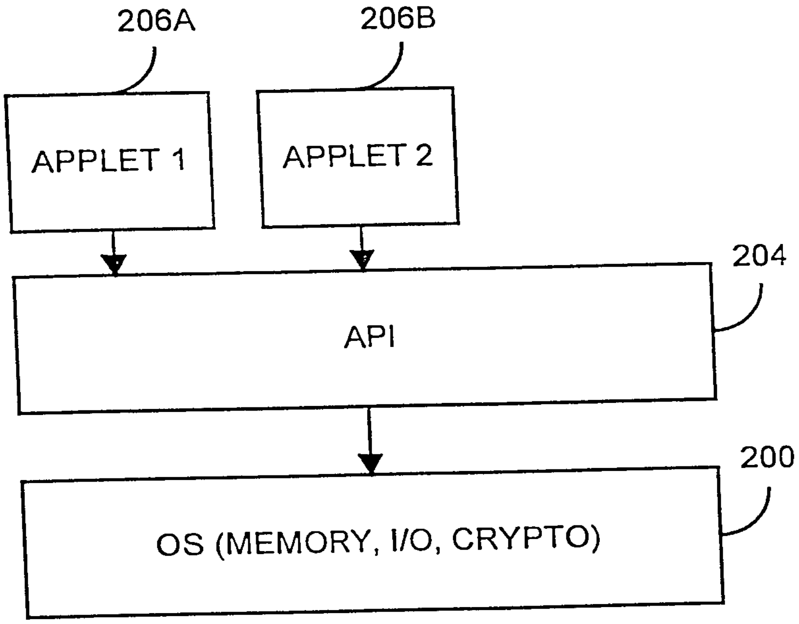


FIG 2

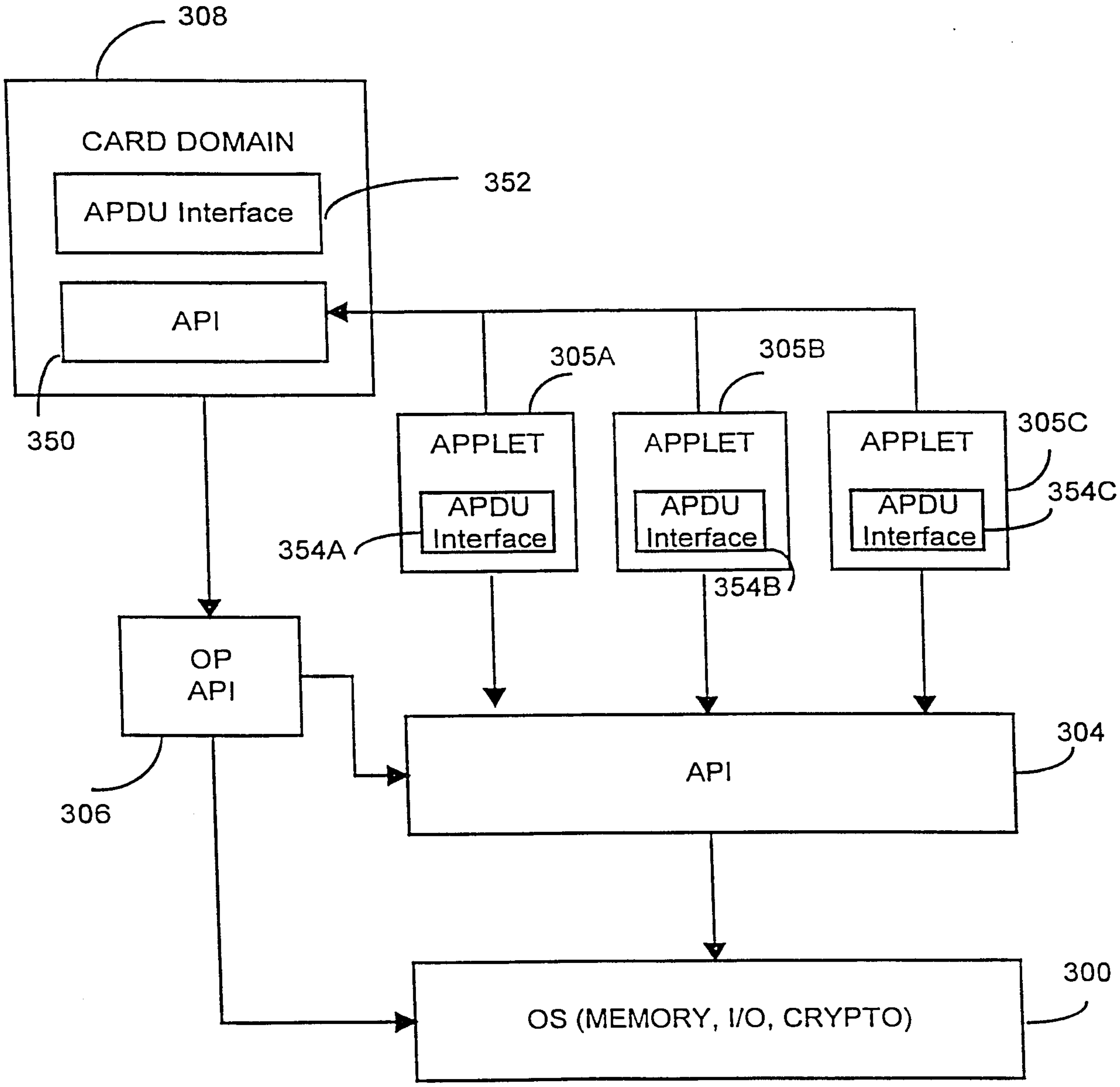

308
CARD DOMAIN
APDU Interface
352
API
350
305A
305B
APPLET
APDU Interface
354A
APPLET
APDU Interface
354B
APPLET
305C
354C
APDU Interface
OP API
306
API
304
OS (MEMORY, I/O, CRYPTO)
300

FIG 3A

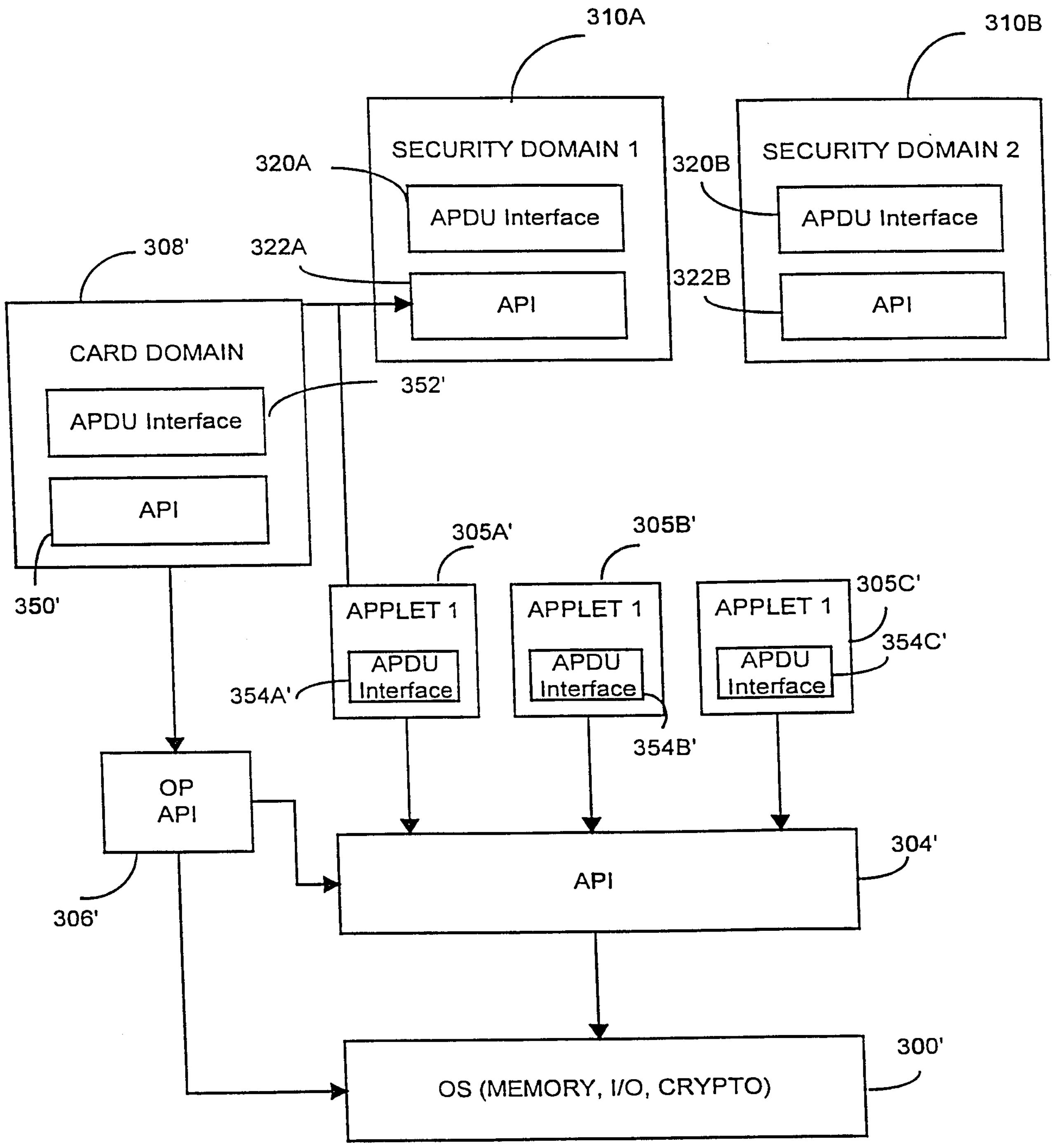

310A
310B
SECURITY DOMAIN 1
SECURITY DOMAIN 2
320A
320B
APDU Interface
APDU Interface
308'
322A
322B
API
API
CARD DOMAIN
APDU Interface
352'
API
305A'
305B'
350'
APPLET 1
APPLET 1
APPLET 1
305C'
354C'
APDU Interface
APDU Interface
APDU Interface
354A'
354B'
OP API
304'
API
306'
300'
OS (MEMORY, I/O, CRYPTO)

FIG 3B

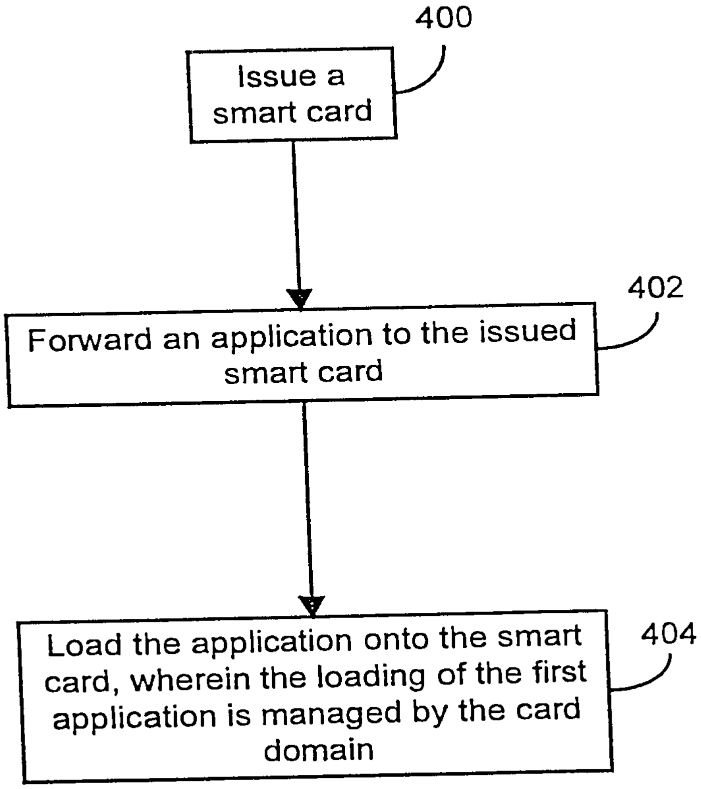

400
Issue a smart card
402
Forward an application to the issued smart card
404
Load the application onto the smart card, wherein the loading of the first application is managed by the card domain

FIG 4

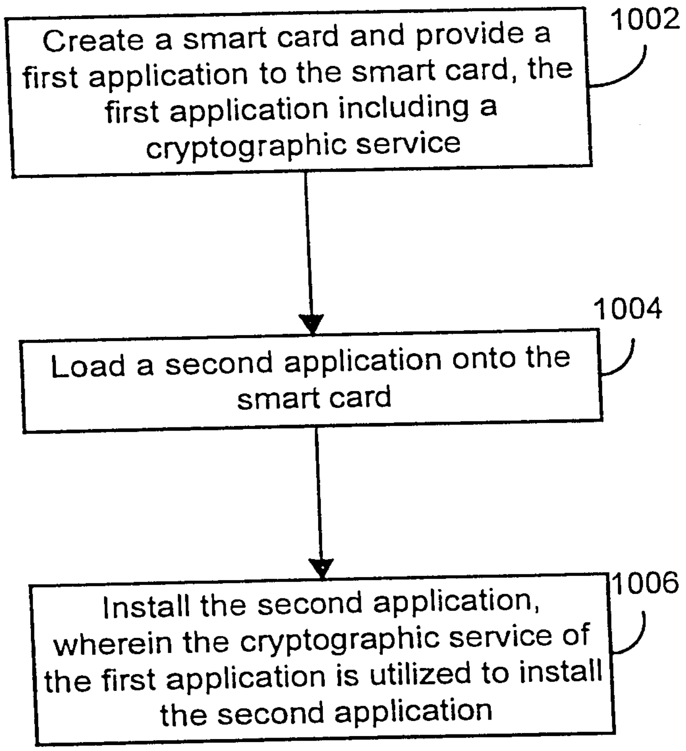

Create a smart card and provide a first application to the smart card, the first application including a cryptographic service
1002
Load a second application onto the smart card
1004
Install the second application, wherein the cryptographic service of the first application is utilized to install the second application
1006

FIG 5

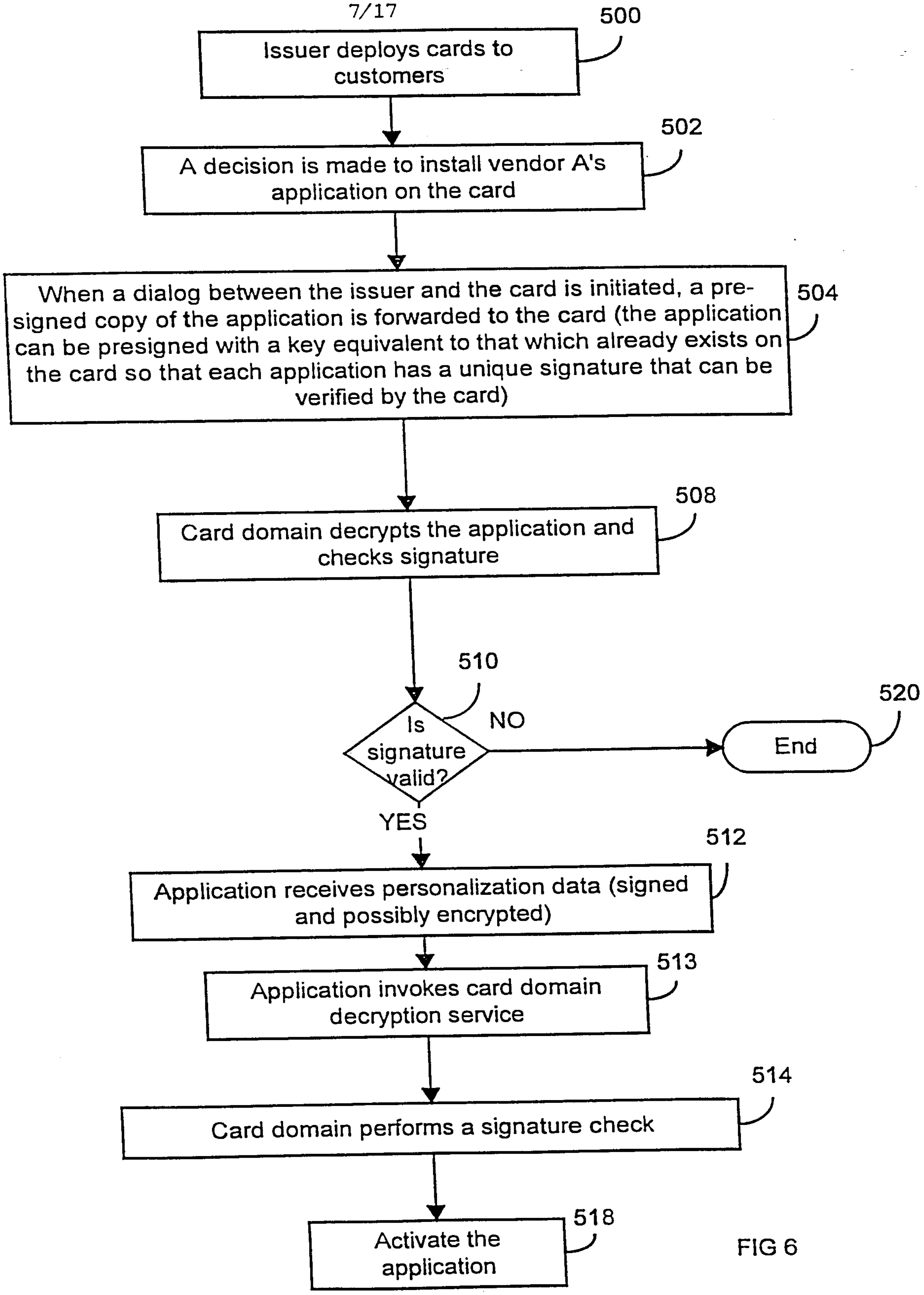

500
Issuer deploys cards to customers
502
A decision is made to install vendor A's application on the card
When a dialog between the issuer and the card is initiated, a pre-signed copy of the application is forwarded to the card (the application can be presigned with a key equivalent to that which already exists on the card so that each application has a unique signature that can be verified by the card)
504
508
Card domain decrypts the application and checks signature
510
Is signature valid?
NO
520
End
YES
512
Application receives personalization data (signed and possibly encrypted)
513
Application invokes card domain decryption service
514
Card domain performs a signature check
518
Activate the application

FIG 6

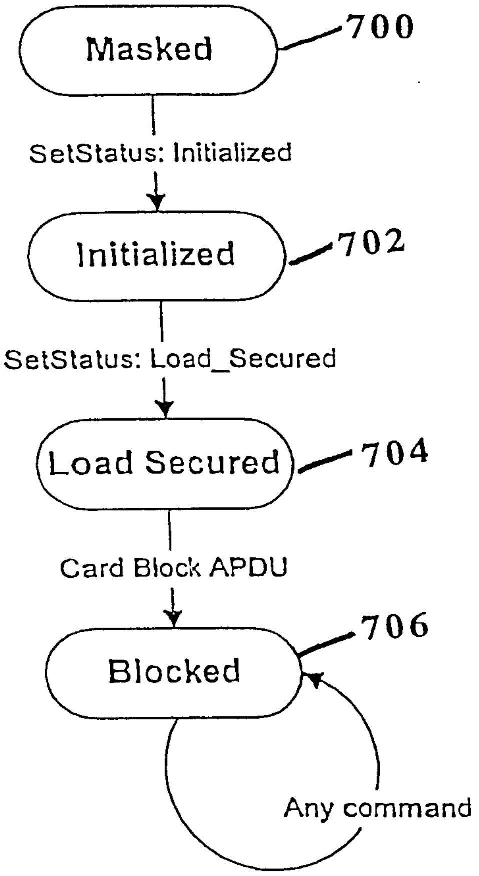

Masked
700
SetStatus: Initialized
Initialized
702
SetStatus: Load_Secured
Load Secured
704
Card Block APDU
Blocked
706
Any command

FIG 7A

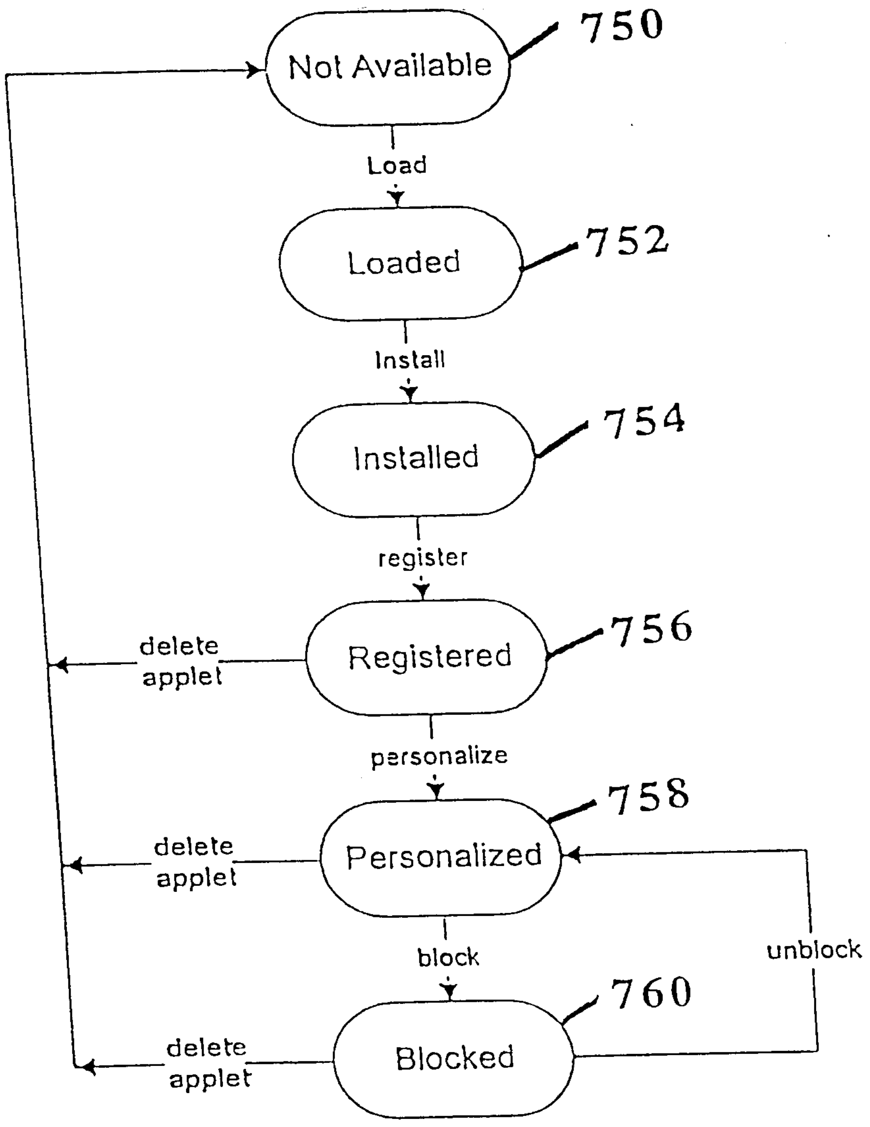

Not Available
750
Load
Loaded
752
Install
Installed
754
register
Registered
756
delete applet
personalize
Personalized
758
delete applet
unblock
block
Blocked
760
delete applet

FIG 7B

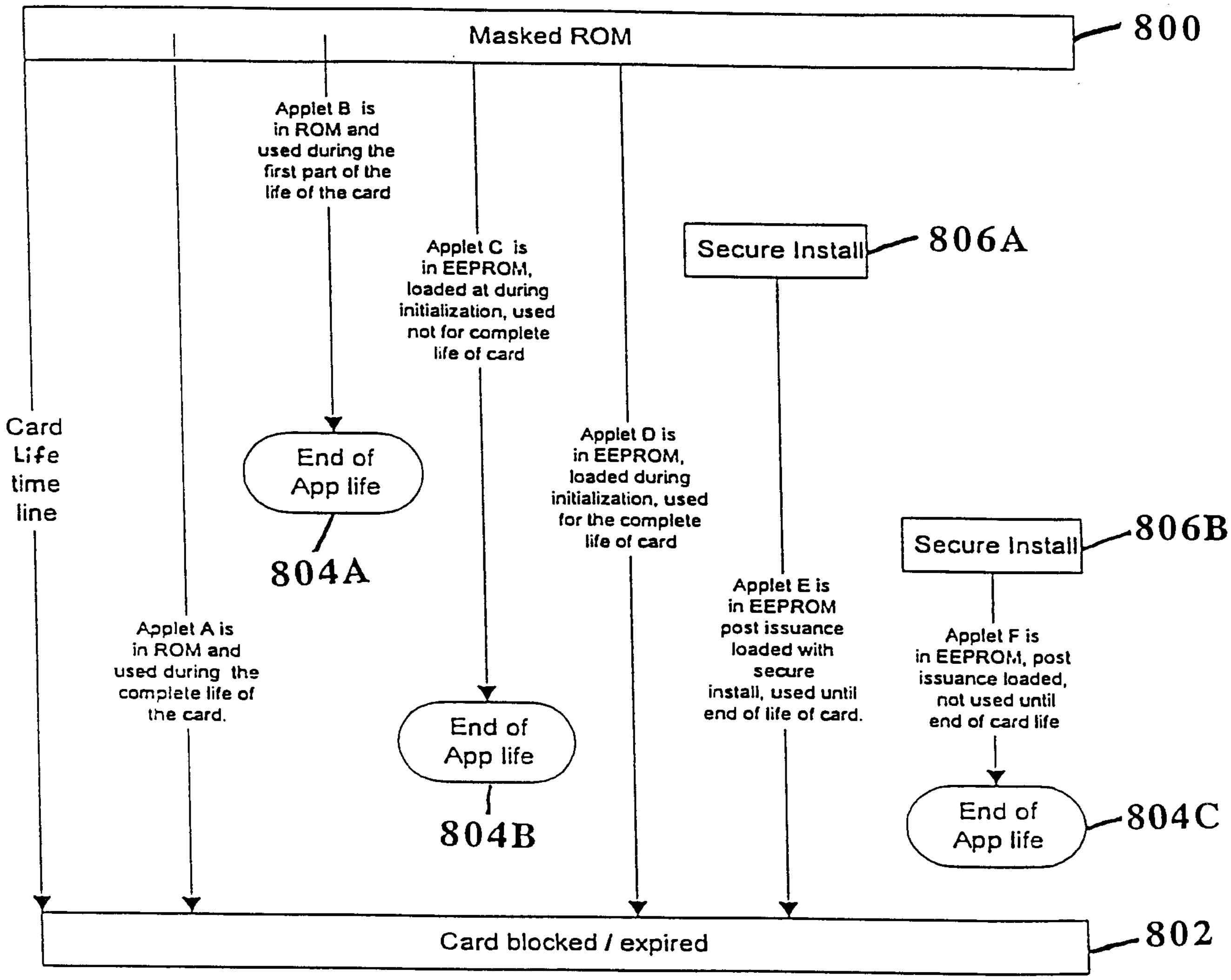

Masked ROM
800
Applet B is in ROM and used during the first part of the life of the card
Applet C is in EEPROM, loaded at during initialization, used not for complete life of card
Secure Install
806A
Card Life time line
End of App life
804A
Applet D is in EEPROM, loaded during initialization, used for the complete life of card
Secure Install
806B
Applet E is in EEPROM post issuance loaded with secure install, used until end of life of card.
Applet A is in ROM and used during the complete life of the card.
Applet F is in EEPROM, post issuance loaded, not used until end of card life
End of App life
804B
End of App life
804C
Card blocked / expired
802

FIG 8

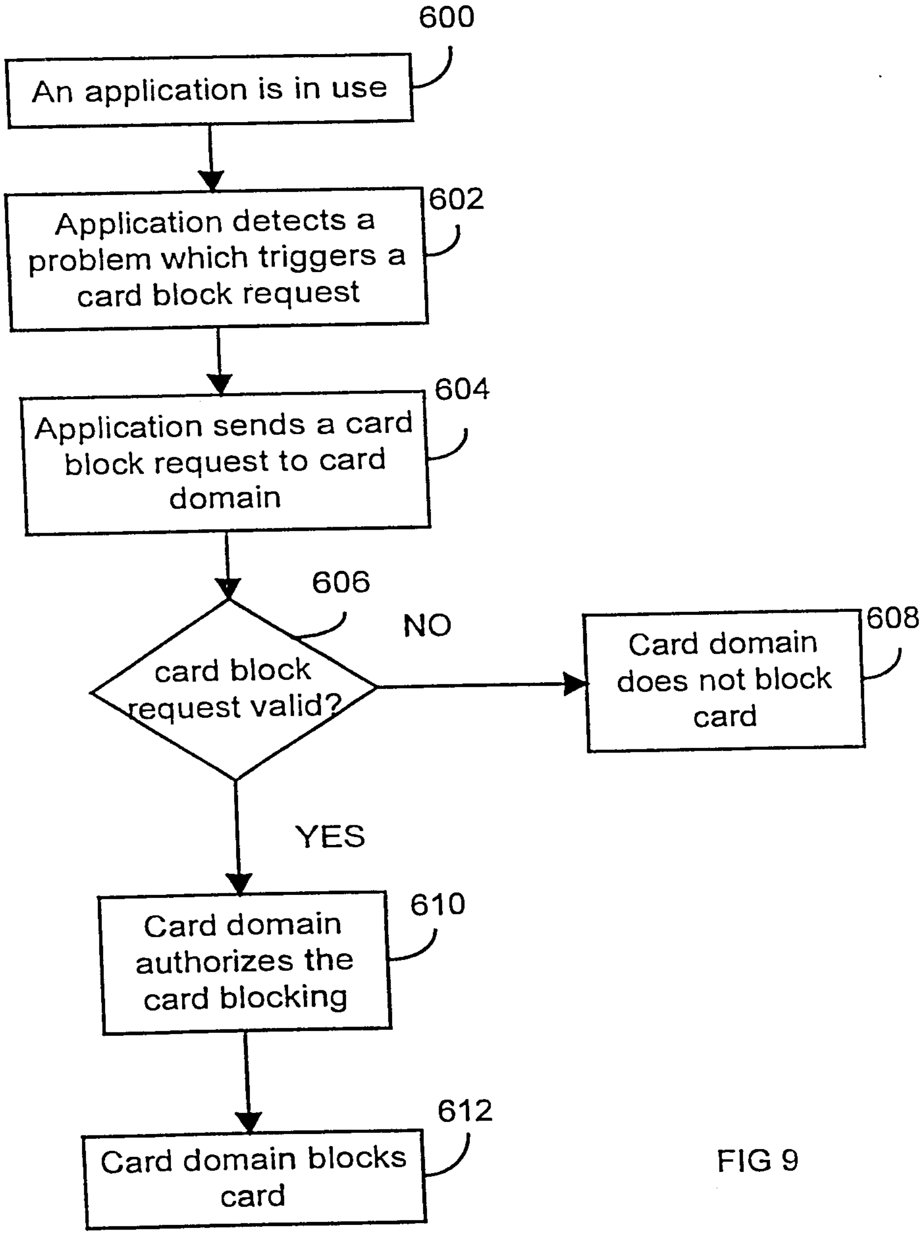

600
An application is in use
602
Application detects a problem which triggers a card block request
604
Application sends a card block request to card domain
606
card block request valid?
NO
608
Card domain does not block card
YES
610
Card domain authorizes the card blocking
612
Card domain blocks card

FIG 9

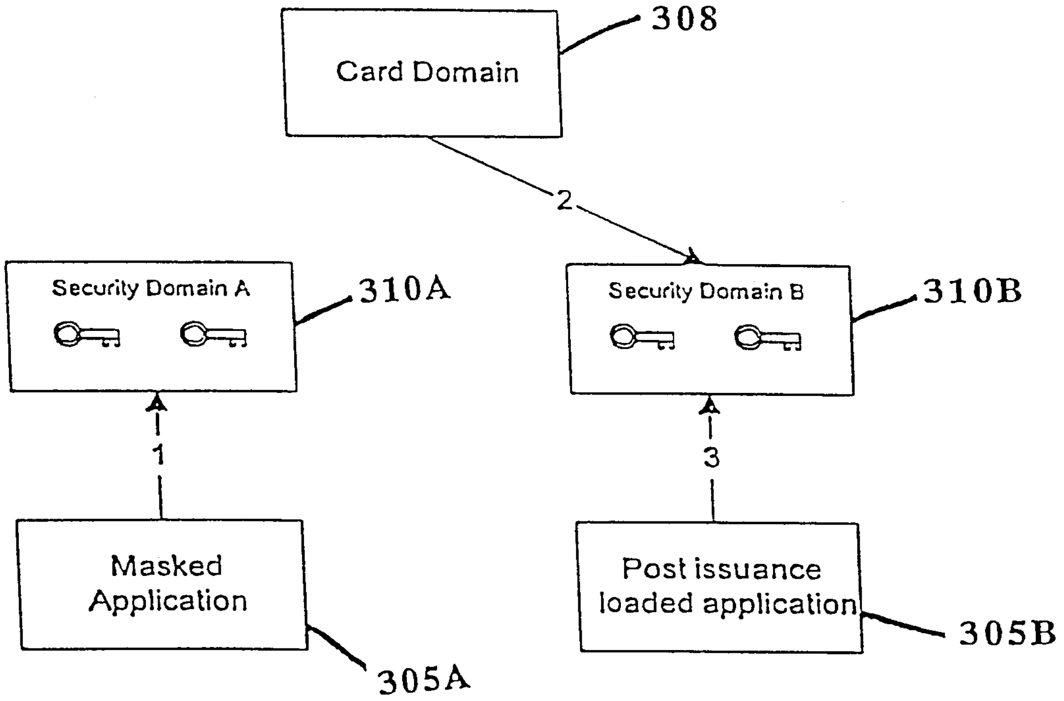

Card Domain
308
2
Security Domain A
310A
Security Domain B
310B
1
3
Masked Application
305A
Post issuance loaded application
305B

FIG 10

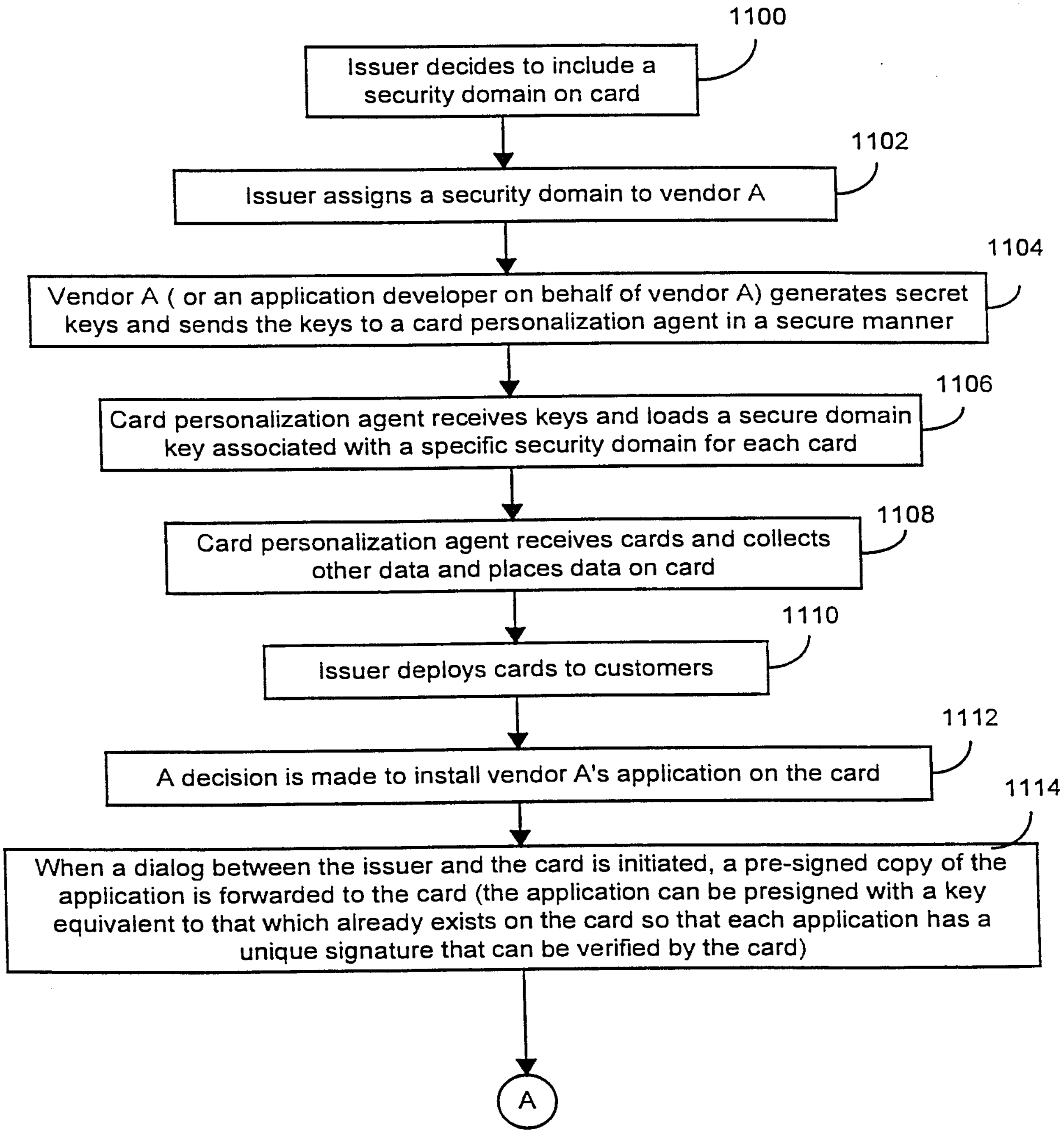

1100
Issuer decides to include a security domain on card
1102
Issuer assigns a security domain to vendor A
1104
Vendor A (or an application developer on behalf of vendor A) generates secret keys and sends the keys to a card personalization agent in a secure manner
1106
Card personalization agent receives keys and loads a secure domain key associated with a specific security domain for each card
1108
Card personalization agent receives cards and collects other data and places data on card
1110
Issuer deploys cards to customers
1112
A decision is made to install vendor A's application on the card
1114
When a dialog between the issuer and the card is initiated, a pre-signed copy of the application is forwarded to the card (the application can be presigned with a key equivalent to that which already exists on the card so that each application has a unique signature that can be verified by the card)
A

FIG 11A

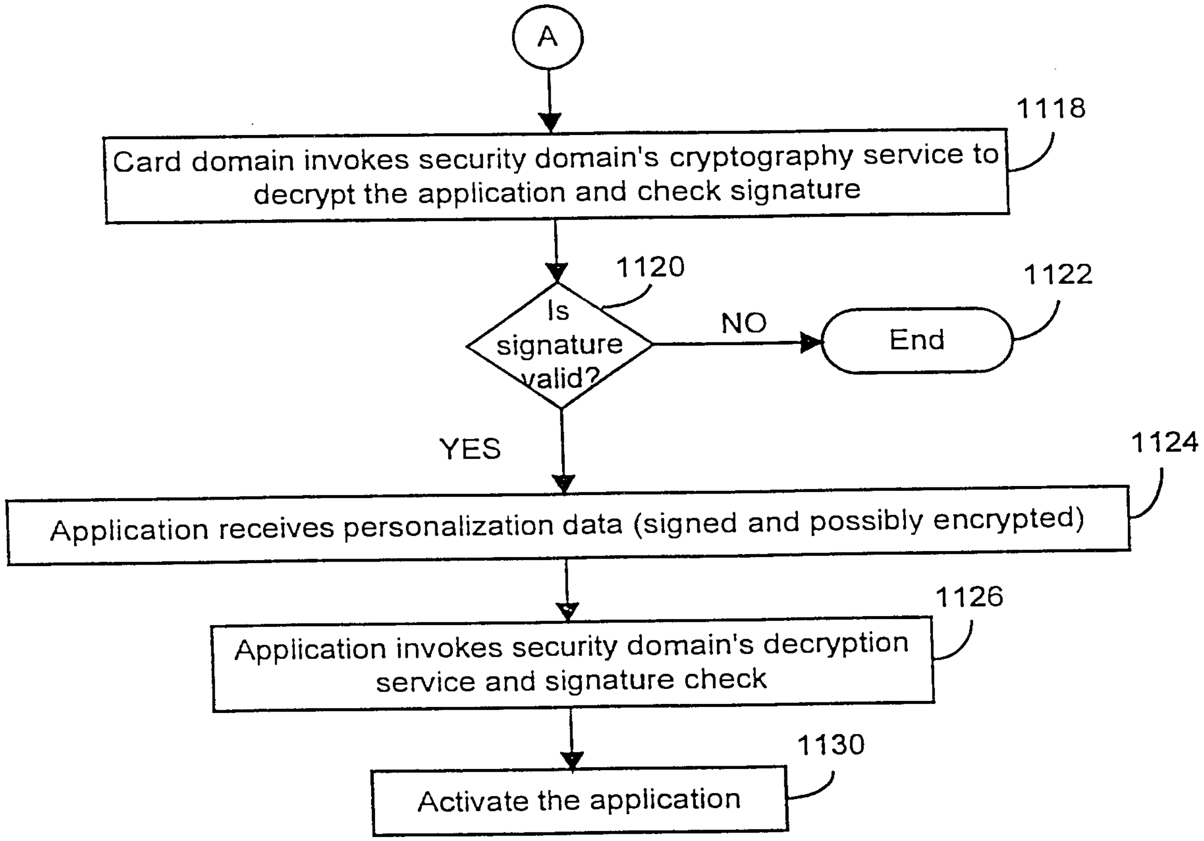

A
1118
Card domain invokes security domain's cryptography service to decrypt the application and check signature
1120
Is signature valid?
NO
End
1122
YES
1124
Application receives personalization data (signed and possibly encrypted)
1126
Application invokes security domain's decryption service and signature check
1130
Activate the application

FIG 11B

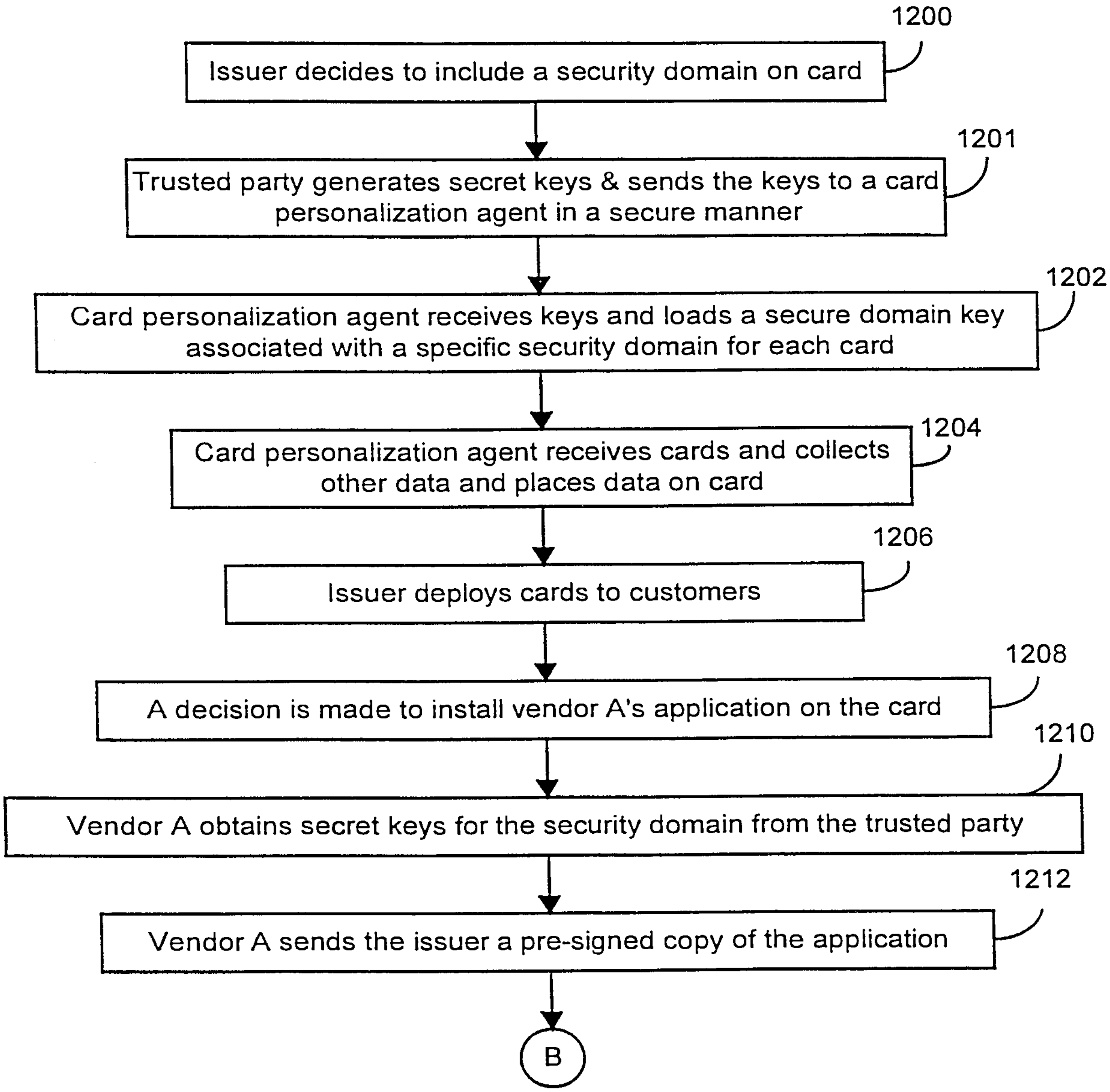

1200
Issuer decides to include a security domain on card
1201
Trusted party generates secret keys & sends the keys to a card personalization agent in a secure manner
1202
Card personalization agent receives keys and loads a secure domain key associated with a specific security domain for each card
1204
Card personalization agent receives cards and collects other data and places data on card
1206
Issuer deploys cards to customers
1208
A decision is made to install vendor A's application on the card
1210
Vendor A obtains secret keys for the security domain from the trusted party
1212
Vendor A sends the issuer a pre-signed copy of the application
B

FIG 12A

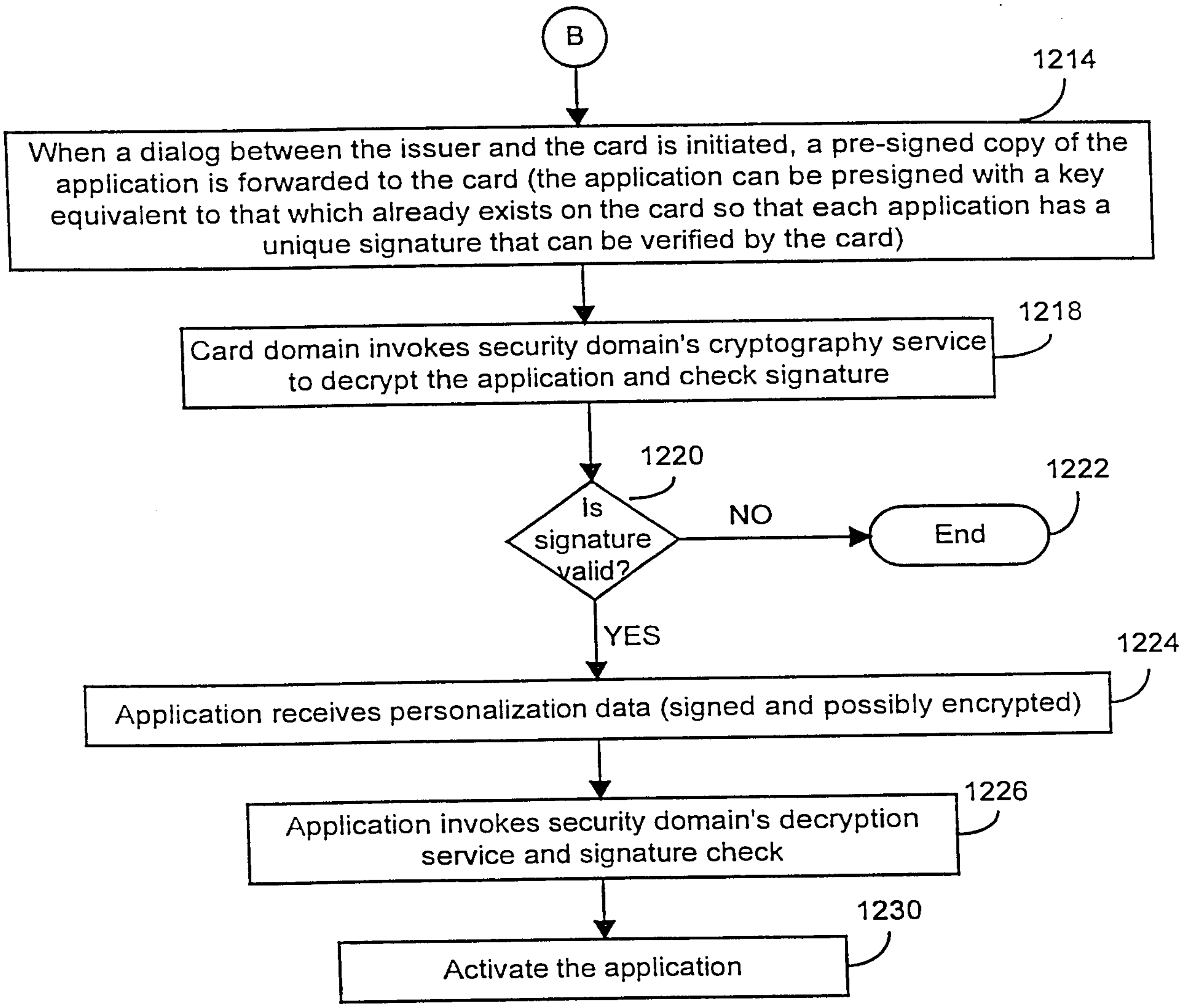

B
1214
When a dialog between the issuer and the card is initiated, a pre-signed copy of the application is forwarded to the card (the application can be presigned with a key equivalent to that which already exists on the card so that each application has a unique signature that can be verified by the card)
1218
Card domain invokes security domain's cryptography service to decrypt the application and check signature
1220
Is signature valid?
NO
1222
End
YES
1224
Application receives personalization data (signed and possibly encrypted)
1226
Application invokes security domain's decryption service and signature check
1230
Activate the application

FIG 12B

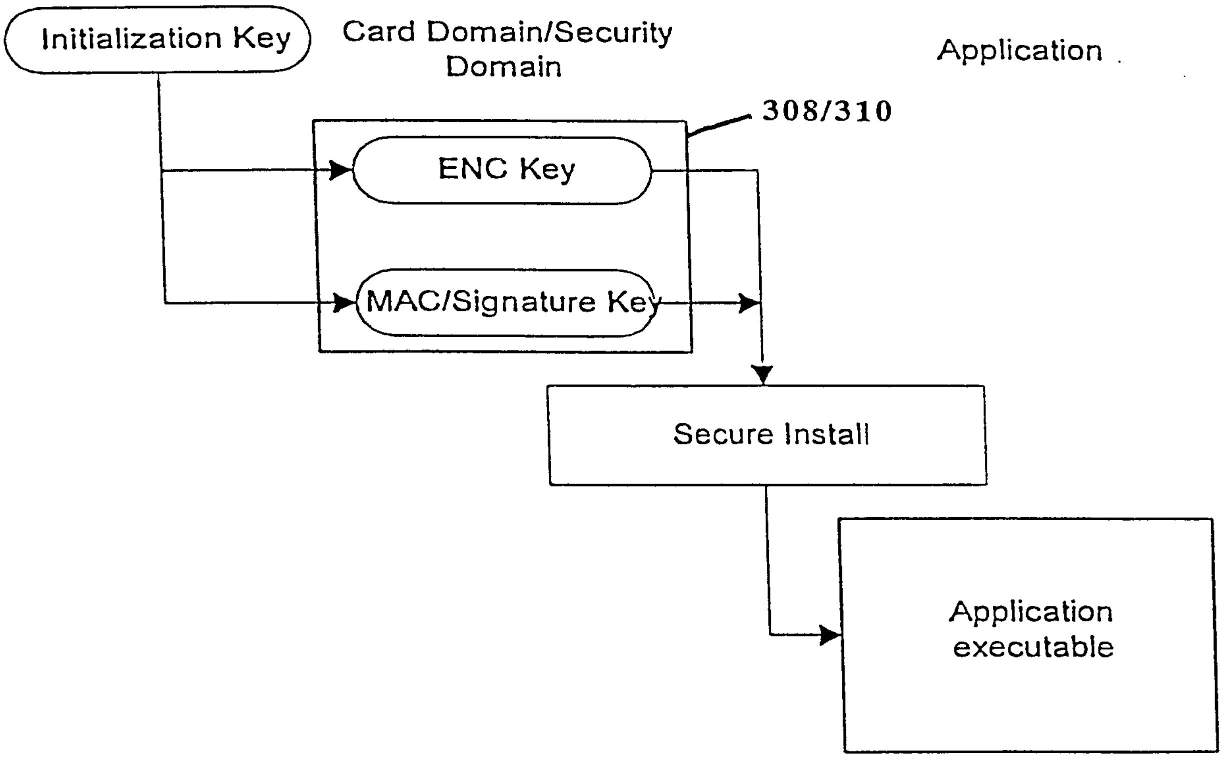

Initialization Key
Card Domain/Security Domain
Application
308/310
ENC Key
MAC/Signature Key
Secure Install
Application executable

FIG 13

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06K 19/07	A2	(11) International Publication Number: WO 98/52152 (43) International Publication Date: 19 November 1998 (19.11.98)

(21) International Application Number: PCT/GB98/01401

(22) International Filing Date: 14 May 1998 (14.05.98)

(30) Priority Data:

60/046,514	15 May 1997 (15.05.97)	US
60/046,543	15 May 1997 (15.05.97)	US
09/078,051	13 May 1998 (13.05.98)	US

(71) Applicant: MONDEX INTERNATIONAL LIMITED [GB/GB]; 47–53 Cannon Street, London EC4M 5SQ (GB).

(72) Inventors: RICHARDS, Timothy, Philip; 32 Craig Mount, Radlett, Herts. WD7 7LW (GB). PEACHAM, David, Anthony; 4 Lynwood, Groombridge, Tunbridge Wells, Kent TN3 9LX (GB).

(74) Agent: POTTER, Julian, Mark; D. Young & Co., 21 New Fetter Lane, London EC4A 1DA (GB).

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

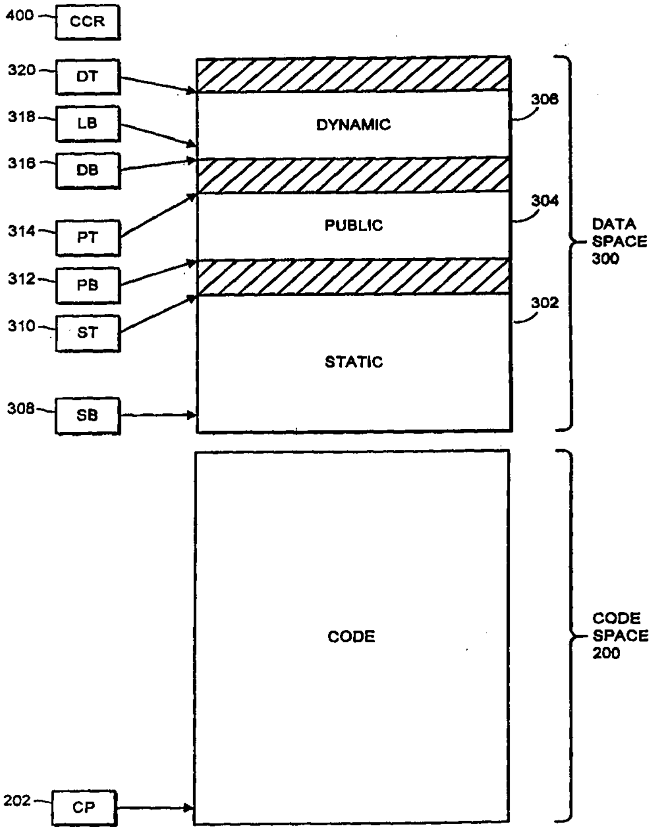
Published

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

(54) Title: COMMUNICATION BETWEEN INTERFACE DEVICE AND IC CARD

(57) Abstract

A multi–application IC card which processes two or more applications using an Application Abstract Machine architecture. The AAM architecture only allows one application to be executed at a time and allows for shared processing by performing a delegation function to a second application. A data space for each application is allocated when the application is selected to be executed. The data space includes a volatile and non–volatile region. The delegation function temporarily interrupts the execution of the first application, saves the temporary data of the first application, shares any data needed with the second application and the second application is executed until the delegated task is completed. The first application then retrieves the saved data and completes its execution. A delegator stack is used to keep track of the delegator's identity when multiple delegations occur. The AAM model allows for a high level of security while transferring data between applications.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

COMMUNICATION BETWEEN INTERFACE DEVICE AND IC CARD

BACKGROUND OF INVENTION

Integrated circuit (IC) cards are becoming increasingly used for many different purposes in the world today, principally because they are ideal tools for the delivery of distributed, secure information processing at a low cost. An IC card, also called a "smart card," is a card typically the size of a conventional credit card, but which contains a computer chip on the card. The computer chip on the IC card typically includes a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), a random access memory (RAM), an input/output (I/O) mechanism, and other circuitry to support the microprocessor in its operations. The computer chip can execute one or more applications stored on the card. Examples of applications that IC cards are being used to store and execute include credit/debit, electronic money/purse, telephone calling card, and loyalty reward applications.

To enable the inter-operability of various IC cards and IC card interface devices, the International Organization for Standardization (ISO) has promulgated a series of standards pertaining to IC cards. For example, ISO 7816-3 is a standard that covers the low-level details of the transmission link between an IC card and an interface device, such as the signal rates, voltage levels, and transmission protocols. At a higher level of detail, the ISO 7816-4 standard covers the format of commands and responses transmitted between an IC card and an interface device.

As defined by ISO 7816-4, commands always originate from an IC card interface device. Once an IC card receives a command, it processes the command and sends back a response. This set of communication between an IC card and interface device is referred to as a "command-response pair." In a command-response pair, the command and/or response may contain associated data, thus producing four possible cases of command-response pairs. These four cases are summarized in Table 1.

Table 1: Command-Response Pair Cases

Case	Command	Response
1	No Data	No Data
2	No Data	Data
3	Data	No Data
4	Data	Data

When an IC card receives a command from an interface device, the operating system present on the IC card may route the command to an application stored on the IC card for processing. Preferably, when a command is sent to an IC card application for processing as part of the regular data exchange specified in the application program, the IC card application should not be required to concern itself with the underlying details and protocol of the transmission link —— it would be desirable for the application to be concerned only with processing the commands it receives. This independence of layers between the transmission layer and the application layer saves programming effort required for the development of an application and enhances the portability of the application between hardware platforms that use different transmission protocols.

To properly process a command it receives, an application is required to know the case of the command ___ i.e., an application is required to know whether the command has any data associated with it or whether it is required to return data. Because of the way certain standards are promulgated, however, it may not be possible to know the case of a command without knowing the details of the underlying transmission protocol. For example, under the T=0 transmission protocol promulgated under ISO 7816-3, it is explicitly assumed that the IC card knows the direction of a data transfer. Such information is usually dependent on the application being executed and the state of the application's program code. Therefore, under the T=0 protocol, it is not usually possible for an IC card operating system to handle all of the low-level details of the transmission layer and shield the application from such details.

It would advantageous if independence of the transmission and application layers could be maintained, even when a transmission protocol requires some intervention by an application.

The foregoing technical problems are addressed by embodiments of the invention providing technical solutions.

SUMMARY OF THE INVENTION

According to a preferred embodiment of the present invention, there is provided a method of responding to a command from an interface device by an integrated circuit card. The integrated circuit card comprises a microprocessor and a memory coupled to the microprocessor. The method includes the steps of selecting an expected case for the command representing whether data is to be transferred between the interface device and the integrated circuit card, determining whether the expected case is applicable to the command, and processing the command if the expected case is applicable to the command. An example of an expected case is one of the four cases defined under ISO 7816-4.

The method in accordance with the preferred embodiment of the present invention may be used where the command is transmitted from the interface device to the integrated circuit card under a transmission protocol requiring the integrated circuit card to have prior information related to the data, if any, to be transferred. For example, the method of the present invention may be used with the T=0 protocol defined by ISO 7816-3. The method of the present invention may also be used when the interface device and the integrated circuit card support a plurality of transmission protocols.

Preferably, the integrated circuit card comprises an application stored in the memory, and the selecting and processing steps are performed by the application. Moreover, before the selecting step, the method of the present invention preferably further includes the step of determining whether the command is recognized by the application.

It is also preferred that the integrated circuit card comprises an operating system stored in the memory and that the determining step is performed by a function of the operating system. Before the step of determining whether the expected case is applicable to the command, the method of the present invention may also include the step of calling by the application a function of the operating system with the expected case. The application may then receive a return value from the function indicative of whether the expected case is applicable to the command.

Preferably, the memory of the integrated circuit card comprises a publicly available memory space and a stack. The method of the present invention may then include the steps of communicating between the operating system and the application using the publicly available memory space or the stack. In addition, the application and the operating system may communicate with each other through a register in the integrated circuit card.

The method of the preferred embodiment of the present invention may also include the step of determining by the function called by the application whether data is to be received from the interface device. If data is to be received, the method may also include the step of receiving such data.

After the application has called the operating system function, the method of the present invention may also include the step of responding by the operating system to subsequent commands by the interface device related to the initial command without interaction between the operating system and the application. For example, if the T=0 protocol defined under ISO 7816-3 is used, the operating system may respond to GET RESPONSE commands without interaction with the application, after the application has called the appropriate operating system function with the expected case of the command.

The method of the preferred embodiment of the present invention may also include the step of communicating response data by the application to the operating system if the return value from the called function is positive. The response data is to be transmitted by the integrated circuit card to the interface device.

In accordance with another aspect of the present invention, there is provided an integrated circuit card for use with an interface device. The integrated circuit card includes a microprocessor, a memory coupled to the microprocessor, means for selecting an expected case for a command transmitted by the interface device, where the expected case represents whether data is to be transferred between the interface device and the integrated circuit card, means for determining whether the expected case is applicable to the command, and means for processing the command if the expected case is applicable to the command.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments in accordance with the invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Fig. 1 is a schematic representation of an IC card in accordance with a preferred embodiment of the present invention;

Fig. 2 is a perspective view of an IC card and terminal in accordance with a preferred embodiment of the present invention;

Fig. 3 is a functional block diagram of an IC card in accordance with a preferred embodiment of the present invention;

Fig. 4 is a block diagram illustrating an exemplary code space, data space, address registers, and control registers for an abstract machine architecture that may be used in accordance with a preferred embodiment of the present invention;

Figs. 5 is a diagram illustrating the cases of commands defined by ISO 7816-4;

Fig. 6 is a diagram illustrating the structure of a message under the T=1 protocol defined by ISO 7816-3;

Fig. 7 is a diagram illustrating the chaining function of the T=1 protocol defined by ISO 7816-3;

Fig. 8 is a flowchart illustrating the steps for processing a command received from an interface device by an IC card in accordance with a preferred embodiment of the present invention;

Fig. 9 is a flowchart illustrating the steps for a routine for checking the case of a command in accordance with a preferred embodiment of the present invention;

Fig. 10 is a flowchart illustrating the steps for a routine for checking the case of a command under the T=0 protocol in accordance with a preferred embodiment of the present invention;

Fig. 11 is a flowchart illustrating the steps for a routine for checking whether a command is consistent with a case 1 command under the T=0 protocol in accordance with a preferred embodiment of the present invention;

Fig. 12 is a flowchart illustrating the steps for a routine for checking whether a command is consistent with a case 2 command under the T=0 protocol in accordance with a preferred embodiment of the present invention;

Fig. 13 is a flowchart illustrating the steps for a routine for checking whether a command is consistent with a case 3 command under the T=0 protocol in accordance with a preferred embodiment of the present invention;

Fig. 14 is a flowchart illustrating the steps for a routine for checking whether a command is consistent with a case 4 command under the T=0 protocol in accordance with a preferred embodiment of the present invention;

Fig. 15 is a flowchart illustrating the steps for a routine for checking the case of a command under the T=1 protocol in accordance with a preferred embodiment of the present invention;

Fig. 16 is a flowchart illustrating the steps for a routine for initializing communications information and for processing the GET RESPONSE command under the T=0 protocol in accordance with a preferred embodiment of the present invention;

Fig. 17 is a flowchart illustrating the steps for a routine for receiving data from an interface device under the T=0 protocol in accordance with a preferred embodiment of the present invention;

Fig. 18 is a diagram illustrating an exemplary communication between an interface device and an IC card under the T=0 protocol;

Fig. 19 is a flowchart illustrating the steps for a routine for transmitting response data and procedure bytes to an interface device in accordance with a preferred embodiment of the present invention; and

Fig. 20 is a flowchart illustrating the steps for a routine for initializing communications information and for processing the GET RESPONSE command under the T=1 protocol in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 1 provides a schematic representation of a typical IC card 10 that can be used with the presently claimed invention. The IC card 10 includes an integrated circuit 12 having one or more electrical contacts 14 connected to the integrated circuit 12.

Fig. 2 shows an example of a device with which the IC card 10 communicates. As used in this specification and the appended claims, the terms "interface device" and "terminal" shall be used to generically describe devices with which an IC card may communicate. A typical terminal 20, as shown in Fig. 2, includes a card reader 22, a keypad 24, and a display 26. The keypad 24 and the display 26 allow a user of the IC card 10 to interact with the terminal. The keypad 24 allows the user to select a transaction, to enter a personal identification number ("PIN"), and to enter transactional information. The display 26 allows the user to receive informational messages and prompts for data entry. Other types of terminals may include IC card-compatible ATM machines and telephones.

Fig. 3 provides a functional block diagram of the integrated circuit 12. At a minimum, the integrated circuit 12 includes a processing unit 100 and a memory unit 110. Preferably, the integrated circuit 12 also includes control logic 150, a timer 160, security circuitry 170, input/output ports 180, and a co-processor 190. The control logic 150 provides, in conjunction with the processing unit 100, the control necessary to handle communications between the memory unit 110 and input/output ports 180. The timer 160 provides a timing reference signal for the processing unit 100 and the control logic 150. The security circuitry 170 preferably provides fusible links that connect the input/output ports 180 to internal circuitry for testing during manufacturing. The fusible links are burned after completion of testing to limit later access to sensitive circuit areas. The co-processor 190 provides the ability to perform complex computations in real time, such as those required by cryptographic algorithms.

The memory unit 110 may include different types of memory, such as volatile and non-volatile memory and read-only and programmable memory. For example, as shown in Fig. 3, the memory unit 110 may include read-only memory (ROM), electrically erasable programmable read-only memory (EEPROM), and random-access memory (RAM).

The memory unit 110 stores IC card data such as secret cryptographic keys and a user PIN. The secret cryptographic keys may be any type of well-known cryptographic keys, such as the private keys of public-key pairs. Preferably, the secret cryptographic keys are stored in a secure area of ROM or

EEPROM that is either not accessible or has very limited accessibility from outside the IC card.

The memory unit 110 also stores the operating system of the IC card. The operating system loads and executes IC card applications and provides file management and other basic card services to the IC card applications. Preferably, the operating system is stored in ROM.

In addition to the basic services provided by the operating system, the memory unit 110 may also include one or more IC card applications. For example, if the IC card is to be used as an electronic cash card, an application called MONDEX™ PURSE (from Mondex International Limited) might be included on the IC card, which loads an electronic value of a certain currency from a user's account in a financial institution onto the IC card. Preferably, the operating system of the IC card 10 supports multiple applications, such as the MULTOS™ operating system from Mondex International Limited.

An IC card application may include both program and data files, which are typically stored in EEPROM. The application program may be written either in the native programming code of the processing unit 100 or it may be written in a higher level language that must be translated before it is executed on the processing unit 100. An example of such a higher level language for use on IC cards is the MULTOS™ Executable Language (MEL). Advantageously, by using a higher level language such as MEL, an application program is capable of running on multiple hardware platforms without any need for re-writing.

Because IC cards typically have limited memory capacity due to the size and cost restraints of placing memory on the IC cards, an IC card may also have primitives stored in ROM, which are subroutines that perform frequently used functions or procedures, such as mathematical functions. The primitives are usually written in the native language of the processing unit 100 so that they can be executed very quickly.

Preferably, the operating system of the IC card 10 imposes a layer of abstraction between an application and the underlying hardware of the IC card 10. This abstraction layer permits the application to be hardware independent and to run on multiple IC cards. From the standpoint of the application, it is executing on an "abstract machine" defined by the operating system.

Fig. 4 illustrates an exemplary architecture of an operating system "abstract machine." The abstract machine contains a memory space for each of the program code and data of an application, referred to as the code space 200 and the data space 300, respectively. An exemplary size for each of the code and data spaces is 64K bytes.

The program code in the code space 200 is stored in non-volatile memory such as EEPROM and is addressed by the Code Pointer (CP) register 202 which must be at least sixteen (16) bits to address the entire 64K bytes of the code space.

The data space 300 is divided into three segments: the static segment 302, the dynamic segment 304, and the public segment 306. The static segment 302 contains non-volatile data, which can be stored in EEPROM, while the dynamic segment 304 and the public segment 306 contain volatile data, which can be stored in RAM. As shown in Fig. 4, there may be gaps between the segments, so that not all of the 64K of data space is used.

The static segment 302 contains the application's non-volatile data. Static data includes cardholder data, such as a card user's name, address, account number, and PIN. Static data also includes variable transactional data, such as the electronic value of a purse or the available credit limit of a credit/debit application.

The dynamic segment 306 contains the application's volatile or temporary data. Dynamic data includes data which is temporarily used during the execution of an application such as intermediate values used in calculations or working variables. For example, a purse application may temporarily store the value of a transaction in order to reduce the amount of the value in the purse. The dynamic segment is used in much the same way as a conventional computer program uses RAM to perform its assigned operations. The dynamic segment preferably is divided into two parts, the session data portion and the stack data portion. The size of the session data portion is a constant for each application and is determined when the application is loaded. The data in the session data portion retains its value for the duration of a session in which the application is involved. A typical use for the session data portion is to support the use of a session PIN. The stack data portion holds variable data which is unique to the particular transaction being executed. The stack data portion stores data in a last-in-first-out manner. The stack is initially empty, but expands and contracts during execution of the application. The data in the dynamic segment is private to an application and cannot be read by other applications.

The public segment is used to store commands and response data passed between an interface device and an application. Applications may also use the public segment for temporary working storage of data, but any data written into the public segment may be available to the interface device.

The data space 300 is preferably addressed using seven address registers: Static Base (SB) 308, Static Top (ST) 310, Public Base (PB) 312, Public Top (PT) 314, Dynamic Base (DB) 316, Local Base (LB) 318, and Dynamic Top (DT) 320. Each of these registers is preferably a sixteen-bit register. These registers define the boundaries of the static, dynamic, and public segments of the data space 300. Each base register contains the address of the first byte in each segment, and each top register contains the address of the byte immediately after the last byte in each segment. The LB register 318 acts as a stack pointer for the stack data portion of the dynamic segment. The address registers can contain physical memory addresses but preferably contain offset addresses in order to be hardware independent.

The abstract machine architecture also contains a Condition Code Register (CCR) 400, which contains bits that are set or cleared based on the result of an operating system or primitive instruction call. An exemplary eight-bit CCR is defined in Table 2.

Table 2: Exemplary Code Condition Register

Bit(s)	Function	Description
7-4		Not defined.
3	Carry (C)	This bit indicates a carry or borrow at the most significant bit.
2	Overflow (V)	This bit is set to 1 when an arithmetic overflow occurs and cleared to 0 at other times.
1	Negative (N)	This bit indicates the most significant bit (sign bit) of the result of an instruction.
0	Zero (Z)	This bit is set to one to indicate a zero result and cleared to 0 to indicate a non-zero result.

A more complete description of an implementation of an abstract machine architecture for an IC card operating system is set forth in the U.S. patent application entitled "Multi-Application IC Card with Delegation Feature" of Everett et al., filed April 23, 1998, which is incorporated herein by reference to Annex A attached hereto.

The low-level communications handler of the operating system may use the public segment 304 to communicate with an application. Table 3 shows an exemplary mapping of the communications buffer of the low-level communication handler onto the public segment. The communication information mapped onto the public segment in Table 3 is consistent with ISO standards 7816-3 and 7816-4 and will be better understood in light of the following discussion.

Table 3: Public Communications Map

Public Address	Use
PB[0]	Data field (either command or response)
PT[-17]	Protocol Flags, where: bit 0: P3Valid bit 1: LcValid bit 2: LeValid bit 3: CmdDataRxed bit 4: Expected_GR
PT[-16]	ProtocolType
PT[-15]	GetResponseCLA
PT[-14]	GetResponseSW1
PT[-13]	CLA
PT[-12]	INS
PT[-11]	P1
PT[-10]	P2
PT[-9]	P3
PT[-8]	Lc (upper byte)
PT[-7]	Lc (lower byte)
PT[-6]	Le (upper byte)
PT[-5]	Le (lower byte)
PT[-4]	La (upper byte)
PT[-3]	La (lower byte)
PT[-2]	SW1
PT[-1]	SW2

To enable the inter-operability of IC cards, IC cards typically follow conventional, industry-wide standards, such as ISO 7816-3 and 7816-4. ISO 7816-3 defines the low-level details of the transmission link between an IC card and an interface device, such as the transmission protocol. ISO 7816-4 defines the format of commands exchanged between interface devices and IC cards.

Beginning with the ISO 7816-4 standard, the ISO 7816-4 standard defines a command having a mandatory header of four (4) bytes and an optional body of variable length. Table 4 sets forth the definition of the command header under ISO 7816-4. The CLA class byte of the command header specifies the extent to which the command and response comply with ISO 7816-4. The INS instruction byte specifies the command function. For example, ISO 7816-4 defines a hexadecimal value of "A4" for INS as a SELECT FILE command. The P1 and P2 parameter bytes provide qualifying information for a command. If no qualification is necessary, then P1 and P2 are set to zero.

Table 4: ISO 7816-4 Command Header

Byte	Code	Description
1	CLA	Class of the Command
2	INS	Instruction Code of the Command
3	P1	Parameter for the Command
4	P2	Parameter for the Command

The command body is conditional and depends on whether there is any data associated with the command or expected with the response. For each case of a command, the structure of the command is illustrated in Figs. 5A to 5D. In the figures, L_c represents the length of the data associated with the command, and L_e represents the maximum length of the data expected to be returned with the response. L_c and L_e may be either one, two, or three bytes. When L_c and L_e are one byte, the cases are referred to as "short." In short cases, L_c may be a number between 1 and 255 (zero is not permitted), and L_e may be a number between 1 and 256 (zero is interpreted as 256). When L_c and L_e are more than one byte, the cases are referred to as "extended." In extended cases, L_c is coded on three bytes, with the first byte being zero and the subsequent two bytes being non-zero. Thus, L_c may take on a value from 1 to 65,535. For an extended case 2, L_e is also coded on three bytes with the first byte being zero. For an extended case 4, L_e is coded on only 2 bytes. In both an extended case 2 and an extended case 4, L_e may take on a value from 1 to 65,536 (a zero value representing the number 65,536).

Table 5 shows the decoding of the case of a received command using the length of the command body (L), the value of the first byte of the command body (B_1), and the value of the second and third bytes of the command body ($B_{2,3}$).

Table 5: Decoding of Command Cases

Condition	Case
L=0	1
L=1	2 Short
L=2	3 Short (B_1 must be greater than zero)
L=3 and B_1=0	2 Extended
L=3 and B_1=1	4 Short
L=3 and B_1=2	3 Short
L=4 and B_1=0	3 Extended
L=4 and B_1=1	4 Short
L=4 and B_1=2	3 Short
L≥5 and B_1=0 and $B_{2,3}$=L-3	3 Extended
L≥5 and B_1=0 and $B_{2,3}$=L-5	4 Extended
L≥5 and B_1=L-1	3 Short
L≥5 and B_1=L-2	4 Short

Turning now to the ISO 7816-3 standard, ISO 7816-3 defines two transmission protocols, referred to as the “T=0” and “T=1” protocols. The T=0 is an asynchronous, half-duplex, character-oriented protocol, while the T=1 protocol is an asynchronous, half-duplex, block-oriented protocol.

Under the T=0 protocol, an interface device initiates the processing of a command by transmitting 5 bytes, designated CLA, INS, P1, P2, and P3. CLA, INS, P1, and P2 correspond to the similarly-named command header bytes defined in ISO 7816-4. P3 is defined as shown in Table 6. In Table 6, B_3 refers to the third byte of the command body.

Table 6: Definition of P3 for T=0 Protocol

Command Case	Hexadecimal Value of P3
1	"00"
2 Short	L_e
2 Extended and $L_e \leq 256$	B_3 (L_e)
2 Extended and $L_e > 256$	"00"
3 Short	L_c
3 Extended and $1 \leq L_c \leq 255$	B_3 (L_c)
3 Extended and $L_c \geq 256$	Split command data into segments of length less than 256 and transmit segments using ENVELOPE command
4 Short	L_c
4 Extended and $L_c < 256$	B_3 (L_c)
4 Extended and $L_c \geq 256$	Split command data into segments of length less than 256 and transmit segments using ENVELOPE command

Under the T=0 protocol, after transmission of the five (5) bytes, the interface device waits for a procedure byte from the IC card. The IC card may respond with three types of procedure bytes: an ACK byte, a NULL byte, or a SW1 byte. An ACK byte permits the subsequent exchange of data to or from the IC card. A NULL byte resets the waiting time of the interface device. The NULL byte is used when the IC card needs more time to process a command. A SW1 byte instructs the interface device to wait for an SW2 byte from the IC card. Together, the SW1 and SW2 bytes provide the interface device with status information. The particular coding of these procedure bytes is not relevant for the purposes of this specification.

As shown in Table 6, when the case of a command is an extended case 2 (with L_e greater than 256) or case 4 (either short or extended), L_e is either partially or completely missing from the five-byte header sent from the interface device to the IC card. In these cases, to receive the appropriate amount of data, the interface device is required to send one or more subsequent GET RESPONSE commands with P3 equal to the number of bytes to receive (256 or less).

Fig. 6 shows the structure of messages transmitted between an interface device and an IC card using the T=1 protocol. The protocol defines a block having mandatory prologue and epilogue fields and an optional information field. The prologue field contains three bytes: a node address (NAD) byte, a protocol control byte (PCB), and a length (LEN) byte. The NAD byte is used to identify the source and destination of the block. The PCB is used to convey control information regarding the block. The LEN byte indicates the number of bytes in the information (INF) field, which may be 0 to 254 bytes. The presence of the INF field is optional. When present, it conveys either application-related or status information. The epilogue field contains an error detection code (EDC), which may be either LRC (longitudinal redundancy check) or CRC (cyclical redundancy check).

As illustrated in Fig. 7, the T=1 protocol supports chaining of blocks. Chaining refers to the segmentation of data and the transmission of the segmented data over several blocks. Chaining is used when the data to be transmitted is greater than 254 bytes, which is the maximum number of bytes supported by the INF field. Chaining is supported through the PCB, which includes a "More Data bit" (or M-bit). The M-bit is set when chained data follows in subsequent blocks and is cleared to indicate the last (or only) block in a chain.

When the T=1 protocol is used, the complete command header and command body of a command defined under ISO 7816-4 are transmitted in the INF field of a block. Thus, without knowing anything about the function of a command, an operating system may determine the case of the command using the decoding rules in Table 5.

Under the T=0 protocol, however, an IC card cannot decode the case of a command simply from the five-byte header sent from the interface device to the IC card because the entire command and data may not be transferred in those initial five bytes. Because of the truncation of information under the T=0 protocol, the same five bytes may be transmitted for different cases. For example, for any given command, an interface device will transmit exactly the same five CLA, INS, P1, P2, and P3 bytes for a short case 2 with L_e=256 and an extended case 2 with

L_e>256. (In both cases, P3=0.) Moreover, an IC card cannot distinguish between a case 3 and a case 4 command from the five-byte header sent from an interface device because L_e is not part of that five-byte header.

Indeed, under the T=0 protocol, ISO 7816-3 explicitly assumes that the IC card and the interface device have information, prior to the transmission of a command, regarding the direction of data, to distinguish between instructions for data transfers into and out of the IC card. It is also implicitly assumed that the IC card has information regarding the number of bytes of data to be transferred.

Such information is, of course, application dependent. The operating system cannot itself know this information. Thus, to properly process a command under the T=0 protocol, the intervention of the currently selected application is necessary. Such intervention typically requires the application to delve into the details of the transmission protocol. Such intervention is undesirable, however, because it destroys the independence of the transmission and application layers. Such layer independence is advantageous because it saves programming effort on the part of an application developer and does not require an application to be updated each time a protocol is changed or a new protocol is promulgated.

According to preferred embodiments of the present invention, however, an application need not delve into the details of the transmission protocol to determine the case of a command. Instead, the operating system passes to the application the CLA, INS, P1, and P2 bytes when they are received, and the application merely checks these bytes to determine whether the command is one that the application recognizes and supports. The application then calls an operating system function or primitive, referred to as *Check_Case*, with the expected case of the command.

It is expected that because most applications on IC cards are written in conjunction with corresponding applications on interface devices, the applications will know which case of a command to expect. For example, in an electronic purse application, the application would probably know that after it has been selected, it will be required to transfer the value in the purse. Otherwise, if an application does not expect any particular case for a command, an application may guess a case. In either instance, the application calls the *Check_Case* function or primitive, which determines whether the command is consistent with the case the application has passed to it. The *Check_Case* function or primitive returns a "true" or "false" value to the application, depending on its determination.

Specifically, with regard to the abstract machine architecture discussed above, an application may PUSH the expected case onto the top of the stack data portion of the dynamic segment, and CALL the *Check_Case* function or primitive. The *Check_Case* function or primitive may return an answer through the Z bit of the CCR register.

Advantageously, the use of a *Check_Case* function or primitive permits layer independence between the transmission protocol layer and the application layer. By using the *Check_Case* command, the application is not required to know which protocol is being used by the IC card or the details of that protocol. Therefore, as new protocols are defined, the application need not be re-written to function with those protocols. In addition, the operating system is not required to know the context in which the command is sent to perform the *Check_Case* function or primitive. The operating system is only required to determine whether the expected case is consistent with the format of the command received. Thus, the *Check_Case* function or primitive accomplishes true independence of transmission and application layers even when the transmission protocol does not explicitly support such independence.

Fig. 8 is an exemplary flowchart illustrating the steps for processing a command received under the transmission protocol T=0 and using the *Check_Case* function or primitive. For purposes of illustration, reference will be made to the communication exchange shown in Fig. 18, which shows an exchange involving an INTERNAL AUTHENTICATE command. An INTERNAL AUTHENTICATE command sends challenge data to an IC card, which receives the data, encrypts it using a secret key, and returns the encrypted data to the interface device. For the purposes of this illustration, it is assumed that an application is involved in processing the INTERNAL AUTHENTICATE command.

As shown in Fig. 18, an interface device (IFD) initiates the communication exchange by sending the command header 1810, consisting of the five hexadecimal bytes 00, 88, 00, 00, and 03. These bytes correspond to the CLA, INS, P1, P2, and P3 bytes defined by the T=0 protocol. In this case the value of P3 signifies that L_c=3 ___ i.e., there are three bytes of challenge data that the interface device desires to transmit to the IC card.

With reference to Fig. 8, the low-level communications handler of the operating system of the IC card receives the T=0 command header from the interface device (IFD) in step 801. In steps 803 and 805, the communications handler stores these bytes in a communications buffer, called *comm_buffer*, and calls the subroutine *Receive_Command_T0*.

The *Receive_Command_T0* subroutine initializes communications variables and processes GET RESPONSE commands under the T=0 protocol. With reference to Fig. 16, the *Receive_Command_T0* subroutine first checks in step 1610 the variable *public.protocol_flags.expecting_gr* to determine whether a GET RESPONSE command is expected by the IC card. If the *public.protocol_flags.expecting_gr* variable is "false," which will be the case when a command header is first received, the *Receive_Command_T0* subroutine initializes the public segment according to the values shown in step 1630. The variables initialized in step 1630 correspond to those shown in Table 3. Once the variables in step 1630 are initialized, the *Receive_Command_T0* subroutine exits.

With reference to Fig. 8, once the *Receive_Command_T0* subroutine has initialized the public segment, the currently selected application is notified of the received command header in step 807 by the use of any conventional means, such as by the use of an interrupt or by setting a bit in the public segment or a control register that the application can poll.

In step 809, the application checks the bytes *public.cla*, *public.ins*, *public.p1*, and *public.p2* in the public segment. Although other protocol-specific information is available in the public segment, such as *public.p3*, an application need not —— and, indeed, should not —— check this information in order to maintain layer independence.

In step 809, if the application recognizes the command defined by the bytes *public.cla*, *public.ins*, *public.p1*, and *public.p2*, the application determines the expected case of the command. Using the INTERNAL AUTHENTICATE command as an example, the application would expect a case 4, since it expects to receive and send data. The application pushes this expected case onto the stack portion of the dynamic segment, and in step 813, calls *Check_Case*.

In step 815, *Check_Case* checks the consistency of the command header received from the interface device against the expected case provided by the application. Depending on whether the expected case is consistent with the command header, *Check_Case* sets the status variable *check_case_response.status* equal to "success" or "failed." In step 817, if the expected case is 3 or 4 (command data is expected to be received), *Check_Case* calls the *Cmd_Data_Rxed* subroutine, which handles receiving the command data.

With reference to Fig. 17, the *Cmd_Data_Rxed* subroutine checks in step 1710 whether the flag *public.protocol_flags.cmd_data_rxd* is "false." If it is not "false," indicating data has already been received, the subroutine exits. If it is "false," in step 1720, the subroutine transmits an ACK byte to the interface device, which signals the interface device to send command data. In steps 1730 and 1740, the subroutine receives the command data from the interface device and sets the command data in the data field of the public segment, *public.data_field*. The *Cmd_Data_Rxed* subroutine then sets the flag *public.protocol_flags.cmd_data_rxd* to "true," to indicate that the command data has been received.

Returning to the example of Fig. 18, the *Cmd_Data_Rxed* subroutine transmits the ACK byte 1820, which consists of the INS byte of the command header, hexadecimal value 88. Three bytes of data 1830 are then transmitted by the interface device and received by the IC card, for storage in the public segment by the *Cmd_Data_Rxed* subroutine.

Returning once more to Fig. 8, once the *Cmd_Data_Rxed* subroutine is finished receiving data and returns control to the operating system, the operating system of the IC card in step 821 sets or clears the Z-bit of the CCR register based on the value of *check_case_response.status*. Control is then returned to the application, which checks the Z-bit of the CCR register to determine whether *Check_Case* successfully verified its expected case. In step 823, if *Check_Case* successfully verified the expected case of the command, the command is processed. Otherwise, an error routine is called.

In step 825, if the case of the command is either 2 or 4 (data is to be sent to the interface device), the application sets the data and the length of the data in the public segment, in *public.data_field* and *public.la*, respectively. The application then returns control to the operating system by a system call.

In step 827, the operating system calls the *Transmit_Response* subroutine, which transmits the appropriate procedure bytes to the interface device, depending on the status of the protocol flags set in the public segment. The *Transmit_Response* subroutine also sets the flag *public.protocol_flags.expecting_gr* to "true" under appropriate circumstances. For example, in a case 4 command, as in the INTERNAL AUTHENTICATE example above, a GET RESPONSE command is expected from the interface device. Thus, the *Transmit_Response* subroutine would set the flag *public.protocol_flags.expecting_gr* to "true."

Fig. 19 is a flowchart illustrating the steps of the *Transmit_Response* subroutine. In step 1910, the subroutine checks whether *public.la* is equal to zero, indicating that there is no data to be transmitted from the IC card to the interface device. If *public.la* is zero, in step 1920, the *response_tpdu* is set simply to the procedure bytes *public.sw1* and *public.sw2.*

If *public.la* is greater than zero, in step 1930, it is checked whether *public.la* is greater than the size of the public segment less the size of the communications parameters stored in the public segment. If this is the case, then *public.la* is greater than the data field size. Accordingly, in step 1940, the response is set to hexadecimal value "6F00," indicating a fatal error.

If *public.la* is within the bounds of the size of the data field in the public segment, in step 1950, it is checked whether *public.protocol_flags.le_valid* is "false," indicating that the expected length of the response data is not yet known. It is also checked whether *public.le* is greater than *public.la*, indicating that the interface device requested more data than is actually available from the currently selected application. If either of these conditions is met, in step 1960, the first procedure byte of the response is set to *public.get_response.sw1*, indicating that the IC card has data available for the interface device. The second procedure byte is set to the length of the data available, either *public.la* if *public.la* is less than 256 bytes or hexadecimal "00" if *public.la* is equal to or greater than 256 bytes. In the case that *public.la* is equal to or greater than 256 bytes, more than one data transmission from the IC card to the interface device will be needed. In step 1970, *public.protocol_flags.expecting_gr* is set to "true," since it is expected the interface device will send a GET RESPONSE command to the IC card in response to the procedure bytes indicating that data is available.

If the expected length of the response data is known and is less than the actual length of the response data, in step 1980, it is checked whether *public.le* is greater than zero. If *public.le* is greater than zero, in step 1990, the variable *data*, which is initialized to null by the operating system, is set to the data in the data field of the public segment. In step 1995, the response is set to: (1) an ACK procedure byte (*public.ins*), indicating that data may follow; (2) the response data (either null or the data in the data field of the public segment); and (3) the procedure bytes *public.sw1* and *public.sw2* (which will normally indicate the completion of the command).

With reference to the example of Fig. 18, the *Transmit_Response* subroutine transmits the procedure bytes SW1 and SW2 with hexadecimal values of "61" and "04," respectively. The hexadecimal value "61" for SW1 informs the interface device that the IC card has data to transfer, and the hexadecimal value

"04" for SW2 informs the interface device of the number of bytes to be transferred (in this case, four). These bytes are transferred in step 1960 of Fig. 19 because it is not known what the expected length of the data is (since L_e is not transmitted with the T=0 header). In step 1970 of Fig. 19, *public.protocol_flags.expecting_gr* is set to "true."

With reference to Fig. 18, when the interface device receives the SW1 and SW2 procedure bytes 1840 from the IC card, the interface device sends a GET RESPONSE command 1850 with L_e (P3) equal to the number of bytes to be transferred (in this case, four).

Returning to Fig. 8, the low-level communications handler once again receives a T=0 command header in step 801. Again, in steps 803 and 805, the communications handler stores the header in the *comm_buffer* and calls the *Receive_Command_T0* subroutine.

Referring once more to Fig. 16, the *Receive_Command_T0* subroutine again checks the status of the flag *public.protocol_flag.expecting_gr*. Since *Transmit_Response* has set the flag to "true," *Receive_Command_T0* proceeds to step 1620. In step 1620, the *Receive_Command_T0* subroutine first determines if P3 is equal to zero. If it is, then *public.le* is set to 256. (Since GET RESPONSE is defined as a case 2 command by ISO 7816-4, then P3=L_e as set forth in Table 6. Moreover, if L_e=0, 256 bytes of data are expected.) The *Receive_Command_T0* then processes the GET RESPONSE command, which involves transmitting the data in the data field of the public segment, *public.data_field*, followed by the appropriate procedure bytes.

With reference to Fig. 18, *Receive_Command_T0* transmits response 1860, which consists of an ACK byte (set to the INS of the GET RESPONSE command), the data to be transferred to the interface device, and procedure bytes SW1 and SW2 (set to Hex "9000," indicating the completion of the command).

The processing of a command received under the T=1 protocol is similar to the processing shown in Fig. 8 with regard to the T=0 protocol. It is noted that *Check_Case* is used for processing the command, even though under the T=1 protocol the operating system may determine the case of a command without any intervention by an application. Nonetheless, to maintain a consistent interface and layer independence, all protocols must be supported by *Check_Case*.

Under the T=1 protocol, when the low-level communications handler of the operating system receives a T=1 block, the low-level communications handler extracts the information contained in the INF field using the control information in the PCB byte. The low-level communications handler also checks the error detection code (EDC) to ensure that a communications error has not occurred. If the PCB indicates that chained blocks follow, the low-level communications handler waits for the chained blocks. As it receives the blocks, the low-level communications handler chains the data in the blocks. After the last block is received, the low-level communications handler calls the *Receive_Command_T1* subroutine, which is the T=1 counterpart of the *Receive_Command_T0* subroutine.

Fig. 20 is a flowchart illustrating the steps for the

Receive_Command_T1 subroutine. In step 2010, the subroutine checks whether *public.protocol_flags.expecting_gr* is "true," indicating the operating system is expecting a GET RESPONSE command from the interface device. If the flag is set to "true" and the command is a GET RESPONSE command, in step 2020, the GET RESPONSE command is processed. If the *public.protocol_flags.expecting_gr* is "false," in step 2030, various communications variables are initialized. In step 2040, the communications variables *public.protocol_flags.le_valid*, *public.protocol_flags.lc_valid*, *public.protocol_flags.cmd_data_rxd*, *public.le*, and *public.lc* are set according to body of the command received by the low-level communications handler and stored in *comm_buffer.T1_body*, using the decoding rules set forth in Table 5.

When the *Receive_Command_T1* has completed its processing, the processing of the received command proceeds in the same way as described with regard to Fig. 8 for a T=0 command after step 805.

Figs. 9 to 15 are flowcharts setting forth exemplary, detailed steps of the *Check_Case* function or primitive. In Fig. 9, steps 910 and 930, *Check_Case* checks if the protocol type is T=0 or T=1. If the protocol type is one of these protocols, the appropriate subroutine, *Check_Case_T0* or *Check_Case_T1*, is called in either of steps 920 and 940. If the protocol type is unrecognized by *Check_Case*, then in step 950, *check_case_response.status* is set to "failed."

Fig. 10 illustrates an exemplary embodiment of the *Check_Case_T0* subroutine. In step 1010, a default value of "success" is assigned to *check_case_response.status*. In step 1020, *Check_Case_T0* checks whether the flags *public.protocol_flags.p3_valid*, *public.protocol_flags.lc_valid*, and *public.protocol_flags.le_valid* are in the initialized states set by *Receive_Command_T0*. If they are not, *check_case_response.status* is set to "failed." If the flags contain proper values, in steps 1030 to 1060, *Check_Case_T0* checks whether the application expects case 1, 2, 3, or 4. In step 1070, if the application has passed an expected case that is not 1 to 4, *check_case_response.status* is set to "failed." In the last step, step 1080, if the subroutine has been successful, *public.protocol_flags.p3_valid* is set to "false" (indicating that *Check_Case* has checked the command header) and *public.protocol_flags.lc_valid* is set to "true" (since, *Check_Case* will set the correct *public.lc*).

Fig. 11 is a flowchart illustrating the steps for the case 1 logic of the *Check_Case_T0* subroutine. In step 1110, if the T=0 protocol byte P3 is valid and greater than zero, then the command is inconsistent with case 1 (see Table 6). Thus, *check_case_response.status* is set to "failed." In steps 1120 and 1130, if either of the L_c or L_e bytes are valid and greater than zero, then the command header is inconsistent with case 1 (which requires no command or response data). Thus, *check_case_response.status* is set to "failed." Otherwise, *public.lc* and *public.le* are set to zero and the *public.protocol_flags.le_valid* is set to "true." (The flag *public.protocol_flags.lc_valid* is set by default in step 1080 of Fig. 10.)

Fig. 12 is a flowchart illustrating the steps for the case 2 logic of the

Check_Case_T0 subroutine. In step 1210, if L_c is valid and greater than zero, *Check_Case* fails (because case 2 does not expect command data). In step 1220, if L_e is valid and equal to zero, *Check_Case* fails (because case 2 expects response data). In step 1230, *public.lc* is set to zero (indicating no command data is present). In steps 1240 and 1250, if P3 is valid and greater than zero, the flag *public.protocol_flags.le_valid* is set to "true" and *public.le* is set to *public.p3* (see Table 6). If P3=0, the expected length of the response data is not known (because P3=0 is consistent with an expected data length of equal to or greater than 256 bytes). Thus, *public.le_valid* is not set to "true" (it remains "false"). The state of the *public.le_valid* variable is used in the *Transmit_Response* subroutine to determine the proper procedural bytes to send (see step 1950 of Fig. 19).

Fig. 13 is a flowchart illustrating the steps for the case 3 logic of the *Check_Case_T0* subroutine. In step 1310, if P3 is valid and greater than zero (which is required for case 3), *public.lc* is set to *public.p3* (see Table 6). Otherwise, in step 1320, other conditions inconsistent with case 3 are checked. In step 1330, the flag *public.protocol_flags.le_valid* is set to "true" and *public.le* is set to zero (because no response data is to be sent for case 3). In step 1340, the *Cmd_Data_Rxed* subroutine, previously described with reference to Figs. 8 and 17, is called, to receive the command data from the interface device.

Fig. 14 is a flowchart illustrating the steps for the case 4 logic of the *Check_Case_T0* subroutine. In step 1410, if P3 is valid and greater than zero (which is required for case 4), *public.lc* is set to *public.p3* (see Table 6).

Otherwise, in step 1420, other conditions inconsistent with case 4 are checked. In step 1430, the *Cmd_Data_Rxed* subroutine, previously described with reference to Figs. 8 and 17, is called, to receive command data from the interface device.

Fig. 15 is a flowchart illustrating the steps for the *Check_Case_T1* subroutine. It is noted again that, under the T=1 protocol, the operating system does not require the intervention of an application to determine the case of a command. Nonetheless, to maintain a consistent interface and layer independence, all protocols must be supported by *Check_Case.*

In step 1510, a default value of "success" is assigned to *check_case_response.status.* In steps 1520, 1540, 1560, and 1580, *Check_Case_T1* determines whether the expected case is 1, 2, 3, or 4, respectively. If the expected case is not one of these cases, in step 1595, a value of "failed" is assigned to *check_case_response.status.* In each of steps 1530, 1550, 1570, and 1590, *Check_Case_T1* checks for conditions that are inconsistent with the cases 1, 2, 3, and 4, respectively, using the communications variables set by the *Receive_Command_T1* subroutine. If inconsistent conditions are found, the value of "failed" is assigned to *check_case_response.status.* Otherwise, *Check_Case_T1* exits (with the default value of "success" assigned to *check_case_response.status*).

Although the present invention has been described with reference to certain preferred embodiments, various modifications, alterations, and substitutions will be known or obvious to those skilled in the art without departing from the spirit and scope of the invention, as defined by the appended claims.

The scope of the present disclosure includes any novel feature or combination of features disclosed therein either explicitly or implicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or mitigates any or all of the problems addressed by the present invention. The application hereby gives notice that new claims may be formulated to such features during the prosecution of this application or of any such further application derived therefrom. In particular, with reference to the appended claims, features from dependant claims may be combined with those of the independent claims in any appropriate manner and not merely in the specific combinations enumerated in the claims.

ANNEX A TO THE DESCRIPTION

ANNEX A

MULTI-APPLICATION IC CARD WITH DELEGATION FEATURE

ANNEX A TO THE DESCRIPTION

BACKGROUND OF INVENTION

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), a random access memory (RAM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. The multiple application operating system present on the IC card allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application card and only executes that one application when inserted into a terminal. For example, a telephone card could only be used to charge a telephone call and could not be used as a credit/debit card. If a card user desires a variety of application functions to be performed by single application IC cards issued to him or her, such as both an electronic purse and a credit/debit function, the card user would be required to carry multiple physical cards on his or her person, which would be quite cumbersome and inconvenient. If an application developer or card user desired two different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal during the transaction, making the transaction difficult, lengthy and inconvenient.

Therefore, it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card.

The increased flexibility and power of storing multiple applications on a single card create new challenges to be overcome concerning the integrity and security of the information (including application code and associated data) exchanged between the individual card and the application provider as well as within the entire system when communicating information between applications.

For instance, the existence of multiple applications on the same card allows for the exchange of data between two applications, while one of the applications is being executed. As stated above, a frequent flyer loyalty program may need to be accessed during the execution of an electronic purse application. If data is passed between applications in an insecure manner, it may be possible for a third party monitoring the transaction to determine the contents of the transferred data or even other private data associated with one or both of the applications. Thus, it would be beneficial to provide an application architecture and memory organization which protects an application's data from being discovered by a third party when it is exchanged with other applications present on the IC card.

Accordingly, it is an object of the invention to provide an application architecture and memory organization which provides for a secure data interaction between applications and allows multiple applications to be accessed while performing a desired task or function.

SUMMARY OF THE INVENTION

The present invention provides for a multiple application architecture for an IC card called an application abstract machine (AAM) and a method for implementing that architecture. The processing of multiple applications is accomplished by generating for at least one application (the "first application") a data memory space including at least two segments, a volatile memory segment and a non-volatile memory segment, commencing the execution of the first application's instructions; delegating or switching execution from the first application to the delegated application and in so doing, saving any data generated by the first application in the logical data memory space associated with the first application; executing the second application's instructions; retrieving the saved data and completing with this data the execution of the first application's instructions.

ANNEX A TO THE DESCRIPTION

Additional delegation commands can be issued by the second application or other subsequent applications. The command delegated is interpreted by a delegated application in the same manner as a selection command being issued directly by a terminal and therefore each application performs the security functions at the same level as if a terminal is issuing the command.

The volatile memory segment can further be separated into public ("Public") and dynamic ("Dynamic") portions. Data can be exchanged between a plurality of applications and/or a terminal when stored in the Public region of the data memory. The Dynamic memory region can be used solely as temporary work space for the specific application being executed.

BRIEF DESCRIPTION OF THE DRAWINGS

Further objects, features and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

Fig. 1 is block diagram illustrating the data memory space segment and associated registers for an IC card application using the AAM organization;

Fig. 2 is a block diagram illustrating the code memory and the data memory spaces for an IC card application using the AAM architecture;

Fig. 3 is a flow diagram illustrating the steps of performing a request for a delegation function by one application to another;

Fig. 4 is a flow diagram illustrating the steps of performing a return

ANNEX A TO THE DESCRIPTION

delegation control function for a delegate application to a delegator application;

Fig. 5 is a flow diagram illustrating the steps of performing an inquire delegator ID request of a delegation function;

Fig. 6 is a block diagram of an IC card chip which can be used as a platform in accordance with the invention; and

Figures 7A, 7B and 7C illustrate multiple delegation calls made between three applications.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

ANNEX A TO THE DESCRIPTION

DETAILED DESCRIPTION OF THE INVENTION

The present invention provides for a method and apparatus for processing multiple application programs with associated data stored on an IC card which can be accessed and executed. An application stored on the card can be selected by a terminal, or other interface device, or another application. Each application program which is stored on the IC card when executed is allocated a memory space organized by the program's software code (instructions which are executed by a processor located on the IC card) and the associated data which the application stores and uses during execution of the program.

For example, a multi-application card may store a purse application, or an electronic money application, and a specific loyalty application such as a frequent flyer awards application. Each application has software code and associated data to support the execution of that software code. Each application is allocated a memory space when executed. In this example, there is interaction between the two applications stored on the card. For each dollar electronically spent to make a purchase, the user may be entitled to one frequent flyer mile which is stored and processed by the frequent flyer program. The purse application need not be aware of the specific loyalty program stored on the card, but instead may contain an instruction to communicate with any loyalty program stored on the card. The loyalty program will require input data representative of the amount of a particular electronic value so that it can update its own stored data of current frequent flyer miles for the user of the card.

ANNEX A TO THE DESCRIPTION

When two applications need to communicate during the same transaction, a system architecture is required to process both applications in an efficient and secure manner. One approach could be a windows type model where both applications could be running at the same time. Presently, however, IC card platforms are not powerful enough to simultaneously operate multiple programs efficiently. Also, transferred data may be exposed to unwanted third party access. The solution to this problem, provided by the current invention, which is described in greater detail below, is to selectively interrupt the execution of applications in a secure manner. This allows the integrity of the applications' data to be maintained and allows the best utilization of the available memory space in the IC card.

An efficient architecture for processing multi applications in an IC card is termed an Application Abstract Machine (AAM) architecture and is described herein. The AAM Architecture applies to any platform independent of its hardware and enables developers to write applications to store on the IC cards which are portable across many different types of platforms (e.g., IC cards built by different manufacturers with different processor configurations) without the need for knowledge about the specific hardware of the platform.

An application abstract machine (AAM), a term for the memory allocation and organization for the data stored and used by each application, is created for each application stored on the IC card which is executed by the processor on the card. In order to ensure data integrity and security when data is transferred between applications which are executed on the IC card, only one application on the IC card is allowed to be executed at a time. Each application has

ANNEX A TO THE DESCRIPTION

a data memory space which is virtually allocated and mapped onto the physical memory addresses available in the IC card memories. Data is then passed between two or more applications within a specified memory location and in a manner consistent with transferring data to an external terminal or device with which the IC card is securely interacting. At a general level, each AAM space created for each application being executed includes two separate address spaces, one for the program code itself and one for the program data which is stored and/or used by the application. The program data address space is effectively divided into three segments: a Static segment, a Dynamic segment and a Public segment which are described in more detail in conjunction with Figure 1. As stated above, the Static, Dynamic and Public segments are logically mapped to the physical memory; they are virtual memory segments as opposed to physical memory segments. The AAM data address space is preferably addressed and processed using seven different address registers and two control registers.

Figure 1 shows an illustrative diagram of a logical data space allocation 101 created for an application used in conjunction with the present invention. The AAM data portion 101 includes a Static data space 103, a Public data space 105 and a Dynamic data space 107. Also shown are a series of address registers: the Static base address register 109, the Static top address register 111, the Public base address register 113, the Public top address register 115, the Dynamic base address register 117, the Dynamic top address register 121 and local base address register 119 which serves as a local stack frame pointer in the Dynamic data space when the application is being executed. The address registers can contain physical memory addresses but preferably contain offset addresses for the various data address spaces in order to be hardware independent. An example of the overall address space is 64K bytes, although the size varies with the applicable platform and the available memory size. The registers can also be considered pointers or can be any other conventional addressing mechanism.

Within the allocated AAM data space 101, the Static portion of the memory is non-volatile which is not erased after power is removed from the IC card (such as EEPROM), the Dynamic space is volatile (such as RAM) which may be erased after power is removed from the card and the Public space is also volatile (such as RAM). An IC card can receive power from a terminal after it is interfaced into the terminal. Although an IC card may contain a battery to maintain some power for memory and circuitry, volatile memory will typically be erased after the IC card is removed from its power source.

The defined AAM data space has bytes in each segment which are contiguous, so that applications can perform pointer and offset arithmetic. For example, if the segment addresses "1515" and "1516," or any other pair of sequential numbers, are both valid and are present within the same segment, then they address adjacent bytes. This allows offset values stored in registers to determine the location of a desired memory address. The segment address of the first byte of the Static segment is zero, so that the segment address of a given location within the Static region is equal to its offset.

Pointers to other specific regions of the Static data area can be stored in the Static data because the Static region is non-volatile. For example, if the card user's name is stored in the Static memory of a credit/debit application, the application will know the card user's name will always be stored in the 5th memory location above the starting point for the Static portion of memory. The location can be noted as SB[5] or the 5th byte above the Static Bottom. Since the Static memory is non-volatile, it will not be erased after each transaction and the application will always know of its location relative to the Static segments' address registers.

On the other hand, the segment address of any location in the Dynamic or Public segments is not always equal to a particular offset from the beginning of the respective segment because the contents of those segments change for each operation. The fourth location in the Dynamic segment will be different for each operation performed by the application. The address of a memory location of Dynamic or Public segment is fixed preferably only for the duration of one command-response pair operation. Because segment addresses in Dynamic or Public are not fixed, MULTOS Executable Language (MEL)™ instructions (or any other program instructions) cannot refer to data using only segment addresses. Instead, a tagged address preferably is used to identify data which is to be retrieved, manipulated, transferred and/or stored with the IC card system.

A tagged address is a nineteen bit value consisting of a three bit tag (address register number) and a sixteen bit offset. Each of the seven address registers for the AAM data space contain a segment physical address. For instance, the address registers SB 109 and ST 111 point to the boundaries of the Static, the address registers PB 113 and PT 115 point to the boundaries of the Public and the address registers DB 117 and DT 121 point to the boundaries of the Dynamic. For each segment, the top register points to the byte immediately after the last valid byte. For example, the last valid byte of the Static is ST[-1]. Register LB functions as a stack frame pointer. It points to a location in the Dynamic segment to indicate a specific byte of local data for the currently executing application.

Referring to Figure 1, the allocated Static segment 103 contains the application's non-volatile data. Static data includes data which is associated with each application for every transaction such as the card user's name, account number, PIN value and address. Static data also includes variable data which is stored for use in future transactions using the application. For example, in a purse transaction, the electronic value data would be read from the Static segment and later saved in the Static segment at the end of the transaction. Additionally, transaction information data or available credit limits in the case of a credit/debit application would be stored in Static data.

The Static data is addressed using register SB (Static Base) and the register ST (Static Top) as offset registers. These registers contain the offset value from a physical address in a memory on the IC card. The individual memory location is then further offset from these starting points such as SB[3] or ST[-5]. SB is defined as zero and ST is equal to the size of the application's Static data which is set when the application is loaded onto the IC card. The multiple application operating system ensures that no other application can read or write the data stored in the Static segment of a particular application. Using current technology, the Static segment is preferably mapped onto an EEPROM (Electrically Erasable Programmable Read-Only Memory) which is non-volatile.

The Dynamic segment 107 contains the application's volatile or temporary data. Dynamic data includes data which is temporarily used during the execution of an application such as intermediate values used in calculations or working variables. For example, a purse application may temporarily store the value of a transaction in order to reduce the amount of the value in the purse. The temporary data is used much like conventional computer programs use RAM to perform their assigned operations. The Dynamic segment preferably is divided into two parts, the session data portion and the stack data portion. The size of the session data is a constant for each application and is determined when the application is loaded. The stack holds variable data which is unique to the particular transaction being executed. The stack data portion stores data in a last-in-first-out manner. The stack is initially empty, but expands and contracts during execution of the application.

The Dynamic data is addressed from the register DB 117 to register DT 121. Register LB 119 serves as a local stack frame pointer to particular memory locations in the Dynamic segment for delegate commands or function calls. Register LB 119 is used to address the topmost frame, that of the currently executing function's session data. Register DT 121 serves as an address offset for the stack pointer. A one byte data item at the top of the stack is addressed as DT[-1], the next byte below is addressed by DT[-2], and so on. A push operation increments the relative value of DT for each item on the stack and a pop operation decrements the relative value of DT for each item on the stack. For example, a data element located at DT[-5] will be located at DT[-6] after an additional data

ANNEX A TO THE DESCRIPTION

item is placed on the stack.

When an application is being executed, the Dynamic segment created for that application also contains the application's session data which is used in performing the assigned task(s) or operation(s). The multiple application operating system ensures that no other application can read or write the data stored in the Dynamic segment of a particular application. The session data is set to zero upon the start of the execution of the application. Stack data will be saved in the stack if the application delegates a task or operation to another application.

A delegation function occurs when one application selects another application to process a command instead of processing the command itself. An example of a delegation function occurs when a delegator application receives a command that it does not recognize or is not programmed to process. The selected application should not reject the command and provide an error response to the interface device (IFD), but instead should pass the command to the appropriate receiver, or delegated application. In order to perform a delegation, the delegator calls the Delegate primitive. The Delegate primitive is a subroutine recognized by the multiple application operating system which is executed when the operating system interprets the Delegate instruction. Primitives can be stored as part of the operating system itself, loaded as a separate routine when the operating system is installed. Primitives are preferably written in machine executable language so that they can be executed quickly although they could be written in a higher level language. When a Delegate command is executed, execution of the delegating application is suspended, and the delegated application is executed instead. The

ANNEX A TO THE DESCRIPTION

delegated application then generates its own data memory space according to the AAM architecture. The data stored in the Public memory space of the first application (stored in RAM) is sent to the Public memory space of the second application (which could be physically the same memory but is allocated separately for each application) so that data can be passed between the applications. The Dynamic memory space is also shared although data is saved in a stack for the delegator and the other portions initialized before the delegated application is executed because the Dynamic data is secret.

In most cases, the delegated application processes the command exactly as though the command has arrived directly from an interface device. When the delegated application has finished processing the command, and has written a response into the allocated Public memory segment, it exits as normal. The delegator then resumes execution at the instruction address following the executed instruction which called the Delegate primitive. The response generated by the delegated application is retrieved or accessed from the allocated Public memory space. The delegator application may simply exit in turn, thus sending the response to the IFD, or may carry out further processing before exiting.

Another example of a delegation operation occurs when two applications need to share data. If an application A always returns a data item N when processing a command B, then another application which also returns data item N in response to a command can delegate the function B to application A in order to reduce the need for duplicate codes stored on the IC card. For example, if a PIN needs to be checked before an application is executed, an application stored

ANNEX A TO THE DESCRIPTION

on the card can delegate the "retrieve PIN function" to a PIN application which returns a stored universal PIN for the card.

Preferably, a new session begins whenever the IFD, e.g. a terminal, successfully selects an application, even if the application has been previously selected during the transaction. For example, if a card user goes to a terminal and transfers twenty dollars of electronic cash using a purse application, charges thirty dollars using a credit/debit application and then transfers ten dollars using the purse application again, three separate sessions will have occurred even though only two applications were used during the entire transaction. Each time an application delegates a task or function to another application, the delegated application treats the delegate function as if the IFD devices had selected the application to perform the task or function. However, performing a delegation function as described below has a different effect on session data.

The following examples will help explain when the session data is initialized (i.e., erased) versus when it is saved to be used in further operations. If application A is selected by an IFD device, and receives commands X, Y and Z from the terminal, application A may delegate all three commands to application B. For example, delegations may occur in response to delegation commands in the program code. Both applications A and B will have their session and stack data in their respective Dynamic segments initialized (set to zero) when they receive command X, but the stack will not be initialized when they receive the subsequent commands Y and Z.

In a second example, application A is selected, and receives

ANNEX A TO THE DESCRIPTION

commands X, Y and Z from the terminal. Application A processes X itself, but delegates Y and Z to application B. Application A will have its session and stack data initialized when it receives X, but not when it receives the subsequent commands Y and Z. Application B will have its session and stack data initialized when it receives Y, but not Z.

One example of a use of session data is to support the use of a session Personal Identification Number (PIN). The application could reserve one byte of session data to support the PIN-receiving flag. On receiving the PIN check command, the selected delegated application could update the flag as follows: if the PIN command is received and the inputted PIN is equal to the stored pin, then it will set the session data DB[0] to 1. If not, the application will check if the PIN flag is already set by checking the value in DB[0]. In either of the above cases, the application will process the rest of the commands in the session because the PIN has been verified. If neither of the cases is true, then the application will not process the command because the PIN is not proper. The PIN checking function could be a delegated function from the selected application to a PIN checking application.

The Public segment 105 is used for command and response data being passed between an IFD and an application. During a delegate command, the Public segment contains the data passed between two applications, the delegator (the application initiating the delegation) and the delegated application (the application which performs the delegated function). An application may also use the Public segment as a further temporary working storage space if required. The

ANNEX A TO THE DESCRIPTION

Public data is addressed using offsets stored in register PB 113 as a starting address, to register PT 115 as an ending address. Register PB 113 and Register PT 115 are fixed for the duration of a command-response pair being initiated by the IFD or delegator. Public data can include data inputted into or supplied by a terminal such as a transaction amount, vendor identification data, terminal information, transmission format or other data required or used by an application resident on the IC card. Public data can also include data which is to be transmitted to an IFD device or other application such as an electronic dollar value, card user information transmission format or other data required or used by the terminal or other delegated application.

The multiple application operating system ensures that the data stored in the Public segment remains private to the application until the application exits or delegates. Preferably, the data in the Public segment is then made available to other entities as follows: (1) if the application delegates, the whole of the Public segment becomes available to the delegated application; (2) if the application exits, and is itself delegated by another, the whole of the Public segment becomes available to the delegator; or (3) if the application exits, and is not itself delegated, then a portion of the Public segment containing the I/O response parameters and data are made available to the IFD.

An application may write secret data into the Public memory segment during execution of the application, but the application must make sure it overwrites the secret portion of the Public segment before delegating or exiting. If the application abnormally ends (abends), then the operating system on the IC card

ANNEX A TO THE DESCRIPTION

preferably overwrites all of the data in the Public segment automatically so that no unwanted entities can have access to the secret data. If the MULTOS carrier device (MCD) is reset, the operating system overwrites data in the Public segment automatically, so that no secret data is revealed. A portion of the Public memory segment is also used as a communications buffer. The I/O protocol data and parameters are preferably stored at the top of the Public memory space. In another preferred embodiment, the top seventeen bytes are reserved for the communications protocol between the IFD device and the IC card application. However, additional or less bytes can also be used depending upon the particular application and operating system being utilized.

The spaces shown between the memory segments in Figure 1 will vary depending upon the specific application and commands being processed. There could be no memory space between the memory segments so that the memory segments are contiguous.

Figure 2 shows an extended illustration of the AAM implemented architecture. Data memory space 201 includes the three segments Static, Public and Dynamic as previously described. Code memory space 203 contains the program instructions for an application stored on the IC card. The application instructions are preferably stored in an executable form which can be interpreted by the resident operating system but can also be stored in machine executable form. Instruction 205 is stored at one location in the code memory space 203. Additional instructions are stored in other locations of memory space 203. Two additional registers 207 and 209 are used in the AAM architecture. A code pointer (CP) register 207

ANNEX A TO THE DESCRIPTION

indicates the particular code instruction to be next executed. In the figure, the register indicates, e.g., through an offset or pointer means, that instruction 205 is the next to be executed. Condition Control Register 209 contains eight bits, four of which are for use by the individual application and four of which are set or cleared depending upon the results of the execution of an instruction. These condition codes can be used by conditional instructions such as Branch, Call or Jump. The condition codes can include a carry bit, an overflow bit, a negative bit and a zero bit.

All address and control registers are set to defined values prior to executing the selected or delegated application. The values are set either when the application is first loaded onto the card and the size of the code and non-volatile data can be ascertained or at the moment when the application passes control to the application. When the application is loaded, SB is set to zero and ST is equal to the number of bytes in the application's Static database. The other address registers are initialized when the application is given control. CP 207 is set to zero and all eight bits in CCR 209 are cleared at the start of executing the application.

A communications interface mechanism is present between the IFD and an application which includes the use of the Public data segment as a communications buffer for command-response parameters. A command-response parameter means an application is given a command to perform and returns a response to the entity issuing the command. Applications interact with an IFD by receiving commands, processing them and returning responses across the IFD-Application Interface. When an application has completed executing a command, the application will place the response into the Public segment starting at PB[0] which can be read by the IFD device and will set the proper interface parameters in the reserved Public space relative to PT[0].

While an application can be called directly from an IFD and return a response directly to an IFD, it can also delegate a request to another application where appropriate. The subsequently-called application will then process the request on behalf of the first application. The delegation can be directly in response to a received command in which the delegator acts as a controller for delegating commands or subcommands to other appropriate applications. Alternatively, the delegated command can be embedded in an application's code which delegates control of the processor when the first application needs to interact with another application during its execution, such as updating frequent flyer miles or verifying a PIN.

Figure 3 shows a flow chart of the steps which are performed when a delegate request is executed. Step 301 sets the parameter named delegator_application_id (delegator ID) to be equal to the selected_file.application_id (selected ID). The selected ID indicates the current application which is selected and which is currently being executed. The delegator ID indicates the application which delegates a function to another delegated application stored on the IC card. Step 303 then pushes (stores) the delegator ID onto the top of the delegate_id_stack (delegate stack). The data referenced in the Dynamic portion of allocated memory is saved so that the current application can complete its execution after the delegated function is complete. Data which is to be

ANNEX A TO THE DESCRIPTION

shared with the delegated application is referenced in the Public portion of allocated memory. The delegate stack is preferably stored outside of an application's AAM memory space and keeps track of which applications have delegated functions. Each application is suspended when it delegates a function so the delegate stack can act in a Last-In-First-Out (LIFO) manner so that if a number of applications are suspended due to delegation requests, the proper application is started in the right order. The delegate stack thus keeps track of which application was the last delegator when multiple layered delegation functions are performed. The delegate stack preferably operates in a LIFO manner although different stack schemes could be used as appropriate.

Step 305 then sets the selected ID to the delegate_request.delegate_application_id (delegate ID) value. This step selects the application which will be called to perform the delegated function or functions. The identities of the delegated application can be specifically called by the delegator application or a particular function can be matched up with an application in a look up table. For example, a PIN match operation may be delegated to different applications depending upon which applications are present on the card. Step 307 then sets the application_command parameter to the value stored in the delegate_request.application_command parameter. This step specifies the command to be delegated to the delegate application. Applications typically have the ability to process many different commands. Alternatively, the entire application could be executed to perform one or more functions. The delegator application can choose which command it is delegating to another application. Step 309 then sends the application_command to the AAM operating system for execution by the delegatee application. The delegator application is then suspended (or interrupted). Any data that is required to pass between the applications is transferred via the Public memory space.

Figure 4 is a flow chart of the steps for performing a "return delegation control" command by the delegatee application. This command is executed by the operating system when a delegated application has completed its delegated function. Step 401 gets application_responses from the Public memory space of the delegated AAM. The response data is passed in the Public memory segment of the delegatee AAM. Step 403 then sets the delegate_response.status variable to a success condition. This means that a delegation operation has been successfully completed. Step 405 sets the delegate_ response.application_responses parameter to the application_responses values which were stored in the Public segment of the delegatee application.

Step 407 sets the delegate_response.delegate_application_id parameter to selected_file.application_id (the delegatee application ID). Step 409 pops the top (i.e., reads the last data stored in the stack) delegate_application_id from the delegate_id_stack. This information indicates the identity of the delegator application for the command which was just delegated and completed by the delegated application. Step 411 sets the select_file.application_id value to the delegator_application_id value. This selects the delegator application which was identified from the delegate ID stack as the current application which will resume running. The Dynamic data for the delegator application will be retrieved for the delegator application from its stored location so that the application will continue to execute where it left off with all data intact but will also have the response information from the delegated function. In step 413, the delegate_response data is sent to the current application for further processing. The response data is passed through the Public data space which could be the same physical RAM memory location because all applications share the physical volatile memory space.

Figure 5 shows a flow chart of the steps involved for inquiring about a delegator ID when a delegate command is received by a delegated application. The delegated application may need to know the identity of the delegator because it may perform operations differently for different delegator applications. For example, an airline loyalty program may need to know if awarded frequent flyers will be based on actual dollars processed or a lump sum award for some other activity such as performing a bill payment operation. This information could be passed to the delegated application as a variable or could be ascertained using an inquiry. The delegator inquiry operation could be implemented as a primitive as previously described.

Step 501 receives the delegator_id_enq_request from the AAM operating system. The request is used to identify the identity of the delegator. Step 503 checks if the delegate_id_stack is empty. If the stack is empty, then no delegation operations have occurred and no applications have been suspended. Thus step 511 sets the delegator_id_enq_response.status parameter to a failure indicator. Step 513 then sets the value of delegator_is_enq_request.error_cause to a value indicating "no delegator application." There is no delegator application. The process then continues with step 509.

If the delegate_id_stack is not empty, than one or more delegations have occurred. In that case, step 505 sets the delegator_id_enq_response.status parameter to a value indicating "success". Step 507 then sets the delegator_id_enq_response.delegator_ application_id parameter to the value stored in delegate_id_stack.delegator_ application_id. This sets the inquiry response to indicate the delegator application ID at the top of the stack. As explained above, the stored data at the top of the stack indicates the last delegator application to call a delegate function. Step 509 then sends the delegator_id_enq_ response back to the AAM operator system which delivers the information to the application or IFD entity requesting the information.

Figure 6 shows an example of a block diagram of an integrated circuit located on an IC card chip which can be used in conjunction with the invention. The integrated circuit chip is located on a chip on the card. The IC chip preferably includes a central processing unit 601, a RAM 603, a EEPROM 605, a ROM 607, a timer 609, control logic 611, I/O ports 613 and security circuitry 615, which are connected together by a conventional data bus 617 or other conventional means.

Control logic 611 in the smart card provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports 612. CPU 601 in conjunction with control logic 611 can perform many different functions including performing calculations, accessing memory locations, modifying memory contents, and managing input/output ports. Some IC cards also include a coprocessor for handling complex computations like cryptographic algorithms. Input/output ports 613 are used for communication between the card and an IFD which transfers information to and from the card. Timer 609 (which generates and/or provides a clock pulse) drives the control logic 611, CPU 601 and other components requiring a clock signal through the sequence of steps that accomplish functions including memory access, memory reading and/or writing, processing, and data communication. Security circuitry 615 (which is optional) preferably includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed upon completion of testing to prevent later access. The Static memory space is preferably mapped to memory locations in EEPROM 605 which is non-volatile. The Dynamic memory space is preferably mapped to RAM 603 which is volatile memory which has quick access. The Public memory space is also preferably mapped to RAM 603 which is volatile memory. The Dynamic data and Public data will be stored in different portions of RAM 603, while RAM is identified as a preferred non-volatile memory and EEPROM is identified as a preferred volatile memory. Other types of memory could also be used with the same characteristics.

Figures 7A, 7B and 7C illustrate an example of a delegation function being performed in order to process multiple applications on an IC card. Figure 7A shows a first application being executed as denoted with a double ringed circle 701. At some point during the execution of the first application, a delegation function 702 is called to delegate an operation to the second application which is indicated by circle 703. Also shown in Figure 7A is an empty delegator ID stack 705. Since the stack is empty, there is no data associated with it and it is shown only for illustrative purposes.

The multiple application operating system receives the delegate command and interrupts the execution of the first application 701 and gives control of the integrated circuit to application 703 as shown in Figure 7B. The execution of the second application 703 is illustrated with a double ringed circle. The term "gives control" means that the microprocessor and other circuitry on the card will process the instructions and allocate memory space for the application which is delegated. When the delegate command is processed, the delegator ID 707 is placed on top of the stack 705. The delegator ID stack is operated in a LIFO manner. Also shown in Figure 7B is a third application 709 resident on the card. At some point during the execution of the second application, a delegate function 711 is called to delegate the operation to the third application.

The multiple application operating system receives the delegate command 711 shown in Figure 7B interrupts the execution of the second application 703 and gives control of the integrated circuit to the third application 709 as shown in Figure 7C. When the delegate command is processed, the delegator ID 713 of the second application is pushed onto the delegator ID stack 705. The delegator ID 707 of the first application whose execution is still interrupted is pushed down in the stack consistent with a LIFO stack management. Thus when the third application has finished its execution, the delegator ID at the top of the stack is popped to indicate that execution of the second application should be resumed first. The delegator ID 707 from the first application will then be at the top of the stack so that when the second application is finished executing, the first application will resume its execution.

Additional applications can be managed by the delegator ID stack in a similar manner. By interrupting the execution of the applications when a delegate command is processed and keeping track of the order of delegations, the security and integrity of the data for each individual application can be maintained which is important because IC cards will store data for applications which is private to the card user such as account numbers, social security number, address and other personal information.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous apparatus, systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

ANNEX A TO THE DESCRIPTION

WE CLAIM:

1. An integrated circuit card comprising:

a microprocessor; a volatile memory coupled to said microprocessor; a non-volatile memory coupled to said microprocessor; and a plurality of applications stored in said non-volatile memory, wherein upon execution of each said application, said microprocessor allocates for each said executing application an associated data memory space comprising at least a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data; and further comprising means for delegating the performance of a function from a first executing application to a second executing application.

2. The integrated circuit card of claim 1, wherein said non-volatile memory segment is divided into at least two regions, including a public region and a dynamic region.

3. The integrated circuit card of claim 2, wherein said public region is used to share data between said first and second applications.

4. The integrated circuit card of claim 2, wherein said dynamic region is used to reference temporary data utilized during an application's execution.

ANNEX A TO THE DESCRIPTION

5. The integrated circuit card of claim 1, further comprising at least one register coupled to said microprocessor which is used to determine the starting locations of each of said segments.

6. The integrated circuit card of claim 5, further comprising at least one register coupled to said microprocessor which is used to determine the top locations of each of said segments.

7. The integrated circuit card of claim 6, further comprising at least one register coupled to said microprocessor which is used as a local dynamic pointer.

8. The integrated circuit card system of claim 1, wherein each said application comprise a plurality of program instructions and wherein at least one of said program instructions when executed causes said memory referenced by said volatile memory segment to be accessed.

9. The integrated circuit card of claim 1, wherein said volatile memory segment references RAM and said non-volatile memory segment references EEPROM.

10. A method for processing a plurality of applications stored in a memory of an integrated circuit:

selecting a first application for execution;

allocating a data space for said first application including at least two memory segments comprising a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data;

executing said first application, interrupting execution of said first application and saving data referenced by said volatile memory segment;

executing a second application;

utilizing said saved data from said volatile memory segment for execution of said first application; and

completing said execution of said first application.

11. The method of claim 10, wherein said first application's identity is stored in a data stack during said delegation step.

12. The method of claim 11, wherein said data stack is accessed following said completion of said second application.

13. The method of claim 12, further including the step of inquiring said first application's identity by accessing said delegator stack.

14. The method of claim 10, wherein said non-volatile memory segment is divided into at least two regions, including a public region and a dynamic region.

15. The method of claim 14, wherein said public region is used to share data between said first application and said second application.

16. The method of claim 14, wherein data referenced by said dynamic region is utilized during the execution of said first application.

17. The method of claim 10, further including the step of allocating a second data space including at least two memory segments for said second application.

18. The method of claim 17, wherein said second data space's segments comprise a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data.

19. The method of claim 18, wherein said second application's non-volatile segment is divided into at least two regions, including a public region and a dynamic region.

20. The method of claim 19, wherein said second application's public region is used to share data between said first and second applications.

21. The method of claim 19, wherein said data referenced by second application's dynamic region is utilized during said execution of said second application.

22. The method of claim 10, further including the step of delegating use of said microprocessor from said second application to a third application stored on said IC card.

23. The method of claim 22, wherein a third data space for said third application is allocated which includes a volatile memory segment for referencing temporary data and non-volatile memory segment for referencing static data, wherein said third application's volatile segment includes a public and dynamic portion.

24 An apparatus for processing a plurality of applications stored in a memory of a single integrated circuit card comprising:

means for allocating a data space comprising at least a non-volatile memory segment for referencing static data and a volatile memory segment for referencing temporary data; means for executing a first application; means for interrupting execution of said first application, means for saving data from at least a portion of said volatile memory segment; and means for executing a second application; means for retrieving said saved data; and means for completing said execution of said first application.

25. The apparatus of claim 24, further including means for storing said first application's identity on a data stack.

26. The apparatus of claim 25, further including means for inquiring of said first application's identity.

27. The apparatus of claim 24, wherein said first application's non-volatile memory segment is divided into at least two regions, including a public region and a dynamic region.

28. The apparatus of claim 27, wherein said public region references random access memory.

29. The apparatus of claim 27, wherein said dynamic region references random access memory.

30. The apparatus of claim 24, further including means for allocating a second data space including at least two segments for said second application.

31. The apparatus of claim 30, wherein said second data space includes a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data.

ANNEX A TO THE DESCRIPTION

32. The apparatus of claim 31, wherein said second data space's non-volatile segment is divided into at least two regions, including a public region and a dynamic region.

33. The apparatus of claim 32, wherein said public region references random access memory.

34. The apparatus of claim 32, wherein said dynamic region references random access memory.

35. The apparatus of claim 24, further including means for delegating operation of said IC card from said second application to a third application stored on said IC card.

36. The apparatus of claim 35, wherein a third data space for said third application is allocated which includes a volatile memory segment for referencing temporary data and non-volatile memory segment for referencing temporary data, wherein said third application's volatile memory segment includes a public and dynamic portion.

37. A system for processing a plurality of applications stored on an IC card comprising:

a non-volatile memory coupled to a databus;

a volatile memory coupled to said databus;

a first and second application program stored in said non-volatile memory, wherein each application has an associated identifier;

a data stack accessible by said databus for storing said applications' identifier if said application is interrupted during its execution;

processor means for executing instructions from said application programs wherein said processor means allocates a data memory space for said application which is being executed and said data memory space is mapped to at least one address in said non-volatile memory and at least one address in said volatile memory; and

wherein said processor means interrupts said first application at least once during its execution to execute said second application.

38. The system of claim 37, wherein data memory space comprises at least a volatile memory segment for referencing temporary data stored in said volatile memory and a non-volatile memory segment for referencing static data stored in said non-volatile memory.

39. The system of claim 37, further including means for storing said first application's identity on a data stack.

40. The system of claim 39, further including means for inquiring of said first application's identity.

41. The system of claim 38, wherein said first application's non-volatile memory segment is divided into at least two regions, including a public region and a dynamic region.

42. The system of claim 41, wherein said public region references random access memory.

43. The system of claim 41, wherein said dynamic region references random access memory.

44. The system of claim 37, further including means for allocating a second data space including at least two segments for said second application.

45. The system of claim 44, wherein said second data space comprises at least a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data.

46. The system of claim 45, wherein said second data space's non-volatile segment is divided into at least two regions, including a public region and a dynamic region.

47. The system of claim 46, wherein said public region references random access memory.

ANNEX A TO THE DESCRIPTION

48. The system of claim 46, wherein said dynamic region references random access memory.

49. The system of claim 37, further including means for delegating use of said processor means from said second application to a third application stored on said IC card.

50. The system of claim 49, wherein a third data space for said third application is allocated which includes a volatile memory segment for referencing temporary data and non-volatile memory segment for referencing temporary data, wherein said third application's volatile memory segment includes a public and dynamic portion.

51. An integrated circuit card comprising:

a plurality of applications and a microprocessor for controlling execution of said applications wherein execution of at least one first application is interrupted and execution is transferred to another second application, further comprising means for sharing data by said first and second applications and means for resuming execution of said first application at the appropriate location at least after completion of execution of said second application.

52. The integrated circuit card of claim 51, further comprising means for allocating a data memory space comprises at least a volatile memory segment for referencing temporary data stored in said volatile memory and a non-volatile memory segment for referencing static data stored in said non-volatile memory.

53. The integrated circuit card of claim 51, further including means for storing said first application's identity on a data stack.

54. The integrated circuit card of claim 53 further including means for inquiring of said first application's identity.

55. The integrated circuit card of claim 52, wherein said first application's non-volatile memory segment is divided into at least two regions, including a public region and a dynamic region.

56. The integrated circuit card of claim 55, wherein said public region references random access memory.

57. The integrated circuit card of claim 55, wherein said dynamic region references random access memory.

ANNEX A TO THE DESCRIPTION

58. The integrated circuit card of claim 52, further including means for allocating a second data space including at least two segments for said second application.

59. The integrated circuit card of claim 58, wherein said second data space comprises at least a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data.

60. The integrated circuit card of claim 58, wherein said second data space's non-volatile segment is divided into at least two regions, including a public region and a dynamic region.

61. The integrated circuit card of claim 58, wherein said public region references random access memory.

62. The integrated circuit card of claim 60, wherein said dynamic region references random access memory.

63. The integrated circuit card of claim 51, further including means for delegating use of said processor means from said second application to a third application stored on said IC card.

ANNEX A TO THE DESCRIPTION

ABSTRACT OF THE DISCLOSURE

A multi-application IC card which processes two or more applications using an Application Abstract Machine architecture. The AAM architecture only allows one application to be executed at a time and allows for shared processing by performing a delegation function to a second application. A data space for each application is allocated when the application is selected to be executed. The data space includes a volatile and non-volatile region. The delegation function temporarily interrupts the execution of the first application, saves the temporary data of the first application, shares any data needed with the second application and the second application is executed until the delegated task is competed. The first application then retrieves the saved data and completes its execution. A delegator stack is used to keep track of the delegator's identity when multiple delegations occur. The AAM model allows for a high level of security while transferring data between applications.

CLAIMS

I CLAIM:

1. A method of responding to a command from an interface device by an integrated circuit card, said integrated circuit card comprising a microprocessor and a memory coupled to said microprocessor, said method comprising the steps of:

selecting an expected case for said command representing whether data is to be transferred between said interface device and said integrated circuit card;

determining whether said expected case is applicable to said command; and

processing said command if said expected case is applicable to said command.

2. The method of claim 1, wherein said command is transmitted from said interface device to said integrated circuit card under a transmission protocol requiring said integrated circuit card to have prior information related to the data, if any, to be transferred.

3. The method of claim 2, wherein said prior information is related to the direction of the data to be transferred.

4. The method of any of claims 1 to 3, wherein said interface device and said integrated circuit card support a plurality of transmission protocols.

5. The method of any of claims 1 to 4, wherein said integrated circuit card comprises an application stored in said memory, and wherein said selecting step is performed by said application.

6. The method of any of claims 1 to 5, wherein said integrated circuit card comprises an application stored in said memory, and wherein said processing step is performed by said application.

7. The method of any of claims 1 to 6, wherein said integrated circuit card comprises an application stored in said memory, and further comprising the step of determining whether said command is recognized by said application before the selecting step.

8. The method of any of claims 1 to 7, wherein said integrated circuit card comprises an operating system stored in said memory, and wherein said determining step is performed by a function of said operating system.

9. The method of any of claims 1 to 8, wherein said integrated circuit card comprises an operating system and an application stored in said memory, and further comprising the step of calling by said application a function of said operating system with said expected case before said determining step.

10. The method of claim 9, further comprising the step of receiving by said application a return value from said function of said operating system indicative of whether said expected case is applicable to said command.

11. The method of claim 9 or claim 10, wherein said memory comprises a publicly available memory space, and further comprising the step of communicating between said operating system and said application using said publicly available memory space.

12. The method of any of claims 9 to 11, wherein said integrated circuit card comprises a register, and further comprising the step of communicating between said operating system and said application using said register.

13. The method of any of claims 9 to 12, wherein said memory comprises a stack, and further comprising the step of communicating between said operating system and said application using said stack.

14. The method of any of claims 9 to 13, further comprising the step of determining by said function of said operating system whether data is to be received from said interface device.

15. The method of claim 14, further comprising the step of receiving data from said interface device if said step of determining whether data is to be received from said interface device is positive.

16. The method of any of claims 9 to 15, further comprising the step of responding by said operating system to subsequent commands by said interface device related to said command without interaction with said application after the step of calling said function by said application.

17. The method of claim 10, further comprising the step of communicating response data by said application to said operating system if said return value is positive, said response data being data to be transmitted by said integrated circuit card to said interface device.

18. An integrated circuit card for use with an interface device, comprising:

a microprocessor;

a memory coupled to said microprocessor;

means for selecting an expected case for a command transmitted by said interface device, said expected case representing whether data is to be transferred between said interface device and said integrated circuit card;

means for determining whether said expected case is applicable to said command; and

means for processing said command if said expected case is applicable to said command.

19. The integrated circuit card of claim 18, further comprising means for receiving said command from said interface device under a transmission protocol requiring said integrated circuit card to have prior information related to the data, if any, to be transferred with or in response to said command.

20. The integrated circuit card of claim 19, wherein said prior information is related to the direction of the data to be transferred.

21. The integrated circuit card of any of claims 18 to 20, wherein said integrated circuit card supports a plurality of transmission protocols.

22. The integrated circuit card of any of claims 18 to 21, further comprising an application stored in said memory, said application comprising said means for selecting an expected case.

23. The integrated circuit card of any of claims 18 to 22, further comprising an application stored in said memory, said application comprising said means for processing said command.

24. The integrated circuit card of any of claims 18 to 23, further comprising an application stored in said memory, said application comprising means for determining whether said command is recognized by said application.

25. The integrated circuit card of any of claims 18 to 24, further comprising an operating system stored in said memory, said operating system comprising said means for determining whether said expected case is applicable to said command.

26. The integrated circuit card of any of claims 18 to 25, further comprising:

an operating system stored in said memory;

an application stored in said memory; and

means for calling by said application a function of said operating system with said expected case.

27. The integrated circuit card of claim 26, further comprising means for receiving by said application a return value from said function of said operating system indicative of whether said expected case is applicable to said command.

28. The integrated circuit card of claim 26 or claim 27, wherein said memory comprises a publicly available memory space, and further comprising means for communicating between said operating system and said application using said publicly available memory space.

29. The integrated circuit card of any of claims 26 to 28, further comprising a register and means for communicating between said operating system and said application using said register.

30. The integrated circuit card of any of claims 26 to 29, wherein said memory comprises a stack, and further comprising means for communicating between said operating system and said application using said stack.

31. The integrated circuit card of any of claims 26 to 30, further comprising means for determining by said function of said operating system whether data is to be received from said interface device.

32. The integrated circuit card of claim 31, further comprising means for receiving data from said interface device responsive to said means for determining whether data is to be received from said interface device.

33. The integrated circuit card of any of claims 26 to 32, further comprising means for responding by said operating system to subsequent commands by said interface device related to said command without interaction with said application.

34. The integrated circuit card of claim 27 or any claim dependent thereon, further comprising means for communicating response data by said application to said operating system if said return value is positive, said response data being data to be transmitted by said integrated circuit card to said interface device.

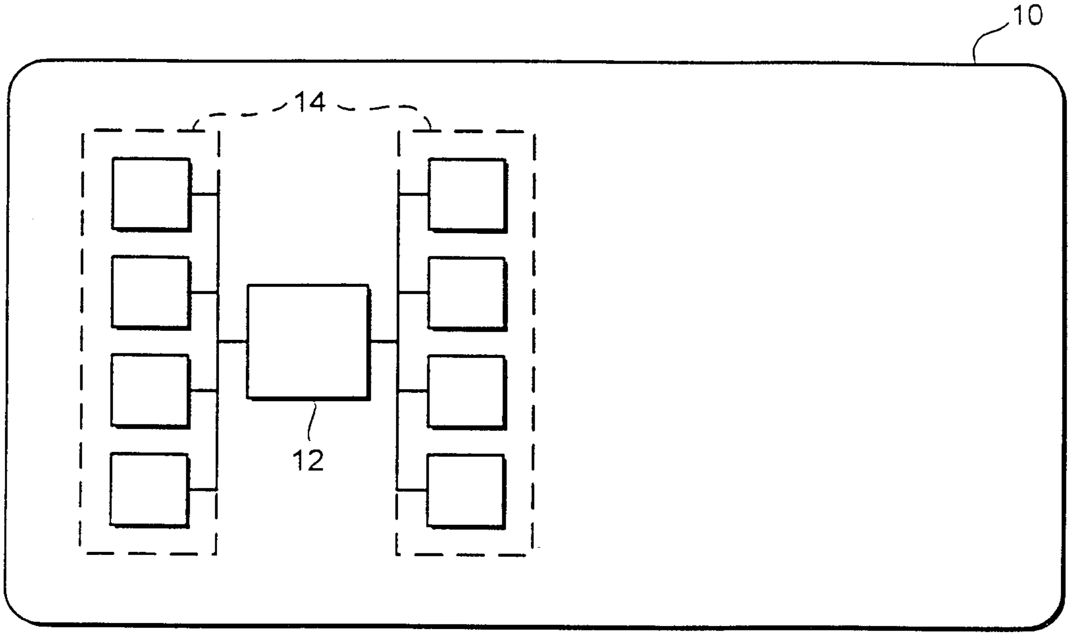


FIG. 1

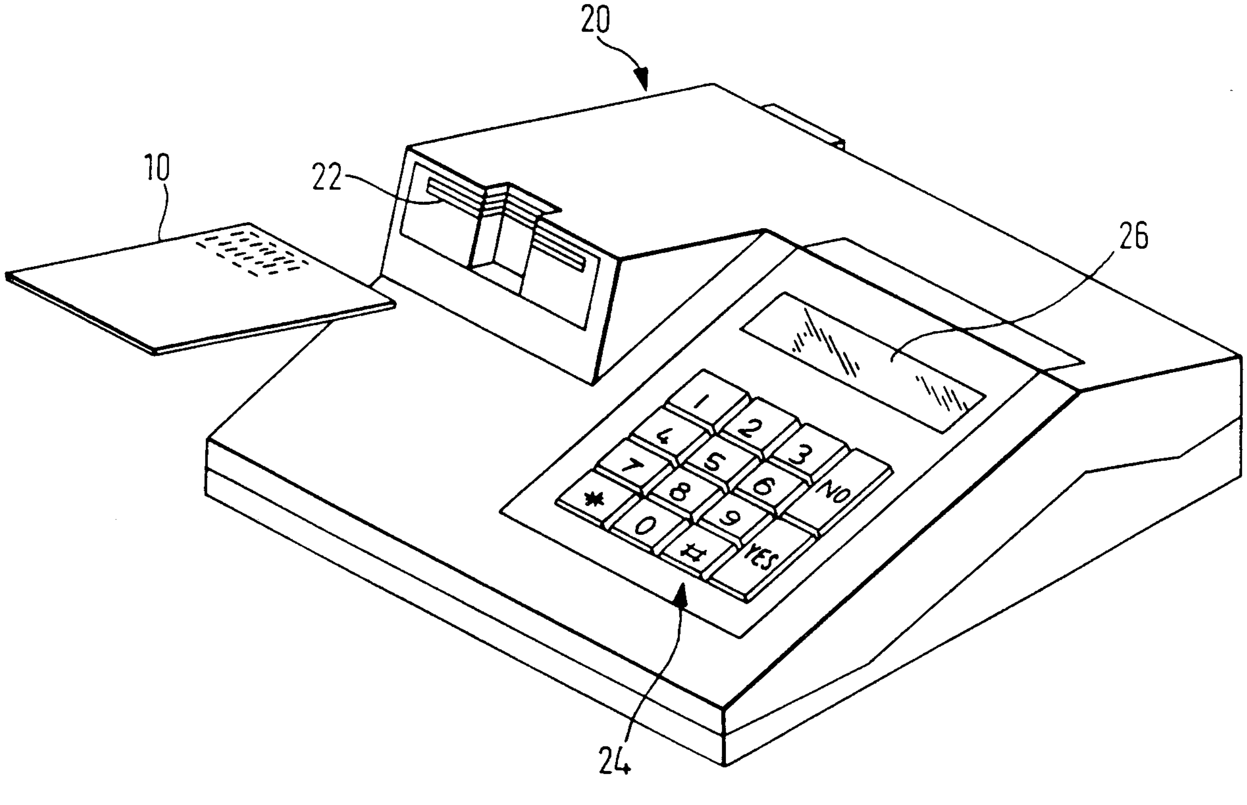


FIG. 2

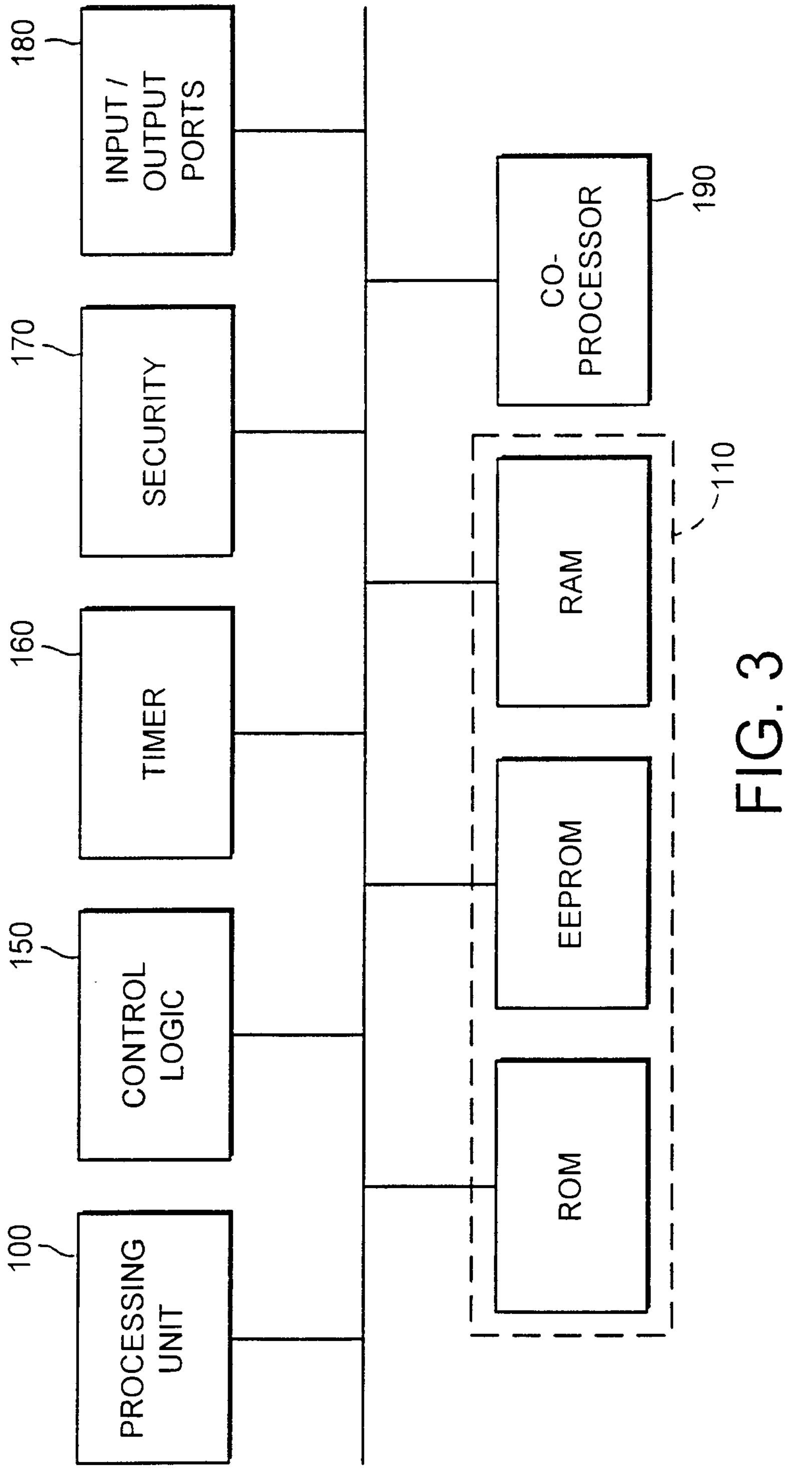
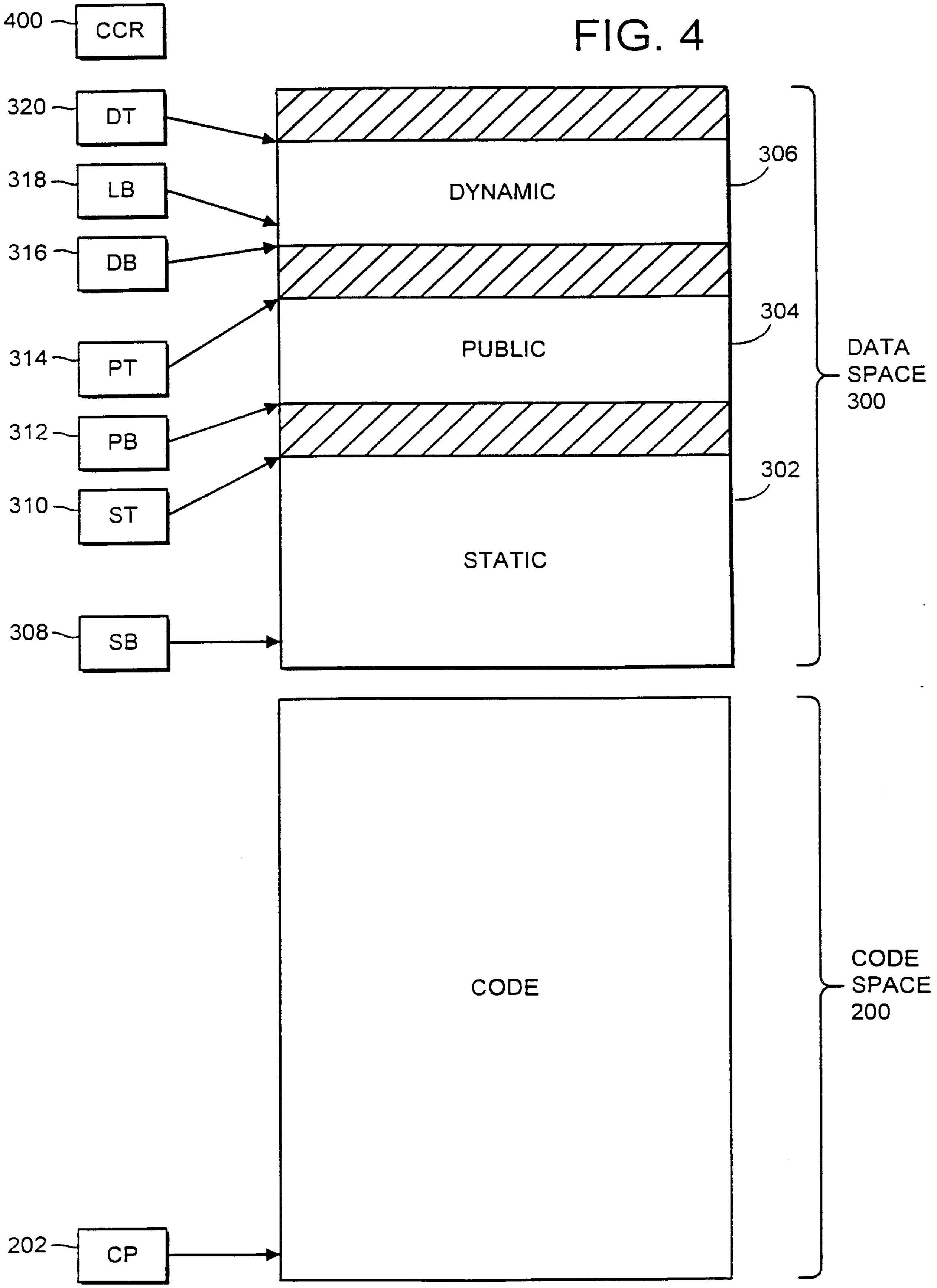

100
PROCESSING UNIT
150
CONTROL LOGIC
160
TIMER
170
SECURITY
180
INPUT / OUTPUT PORTS
ROM
EEPROM
RAM
110
CO-PROCESSOR
190

FIG. 3

FIG. 4


400
CCR
320
DT
318
LB
316
DB
314
PT
312
PB
310
ST
308
SB
202
CP
DYNAMIC
306
PUBLIC
304
STATIC
302
DATA SPACE 300
CODE
CODE SPACE 200

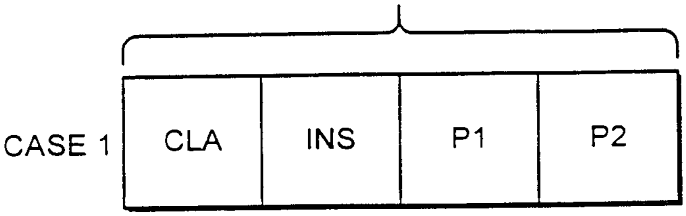


FIG. 5A

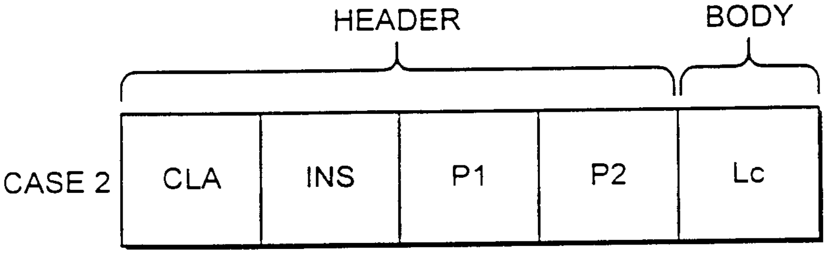


FIG. 5B

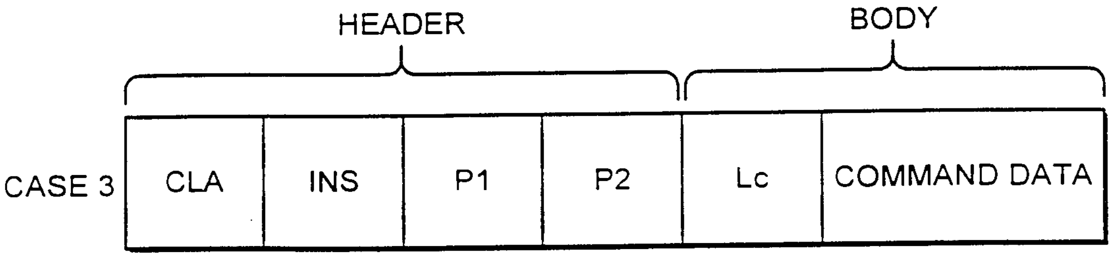


FIG. 5C

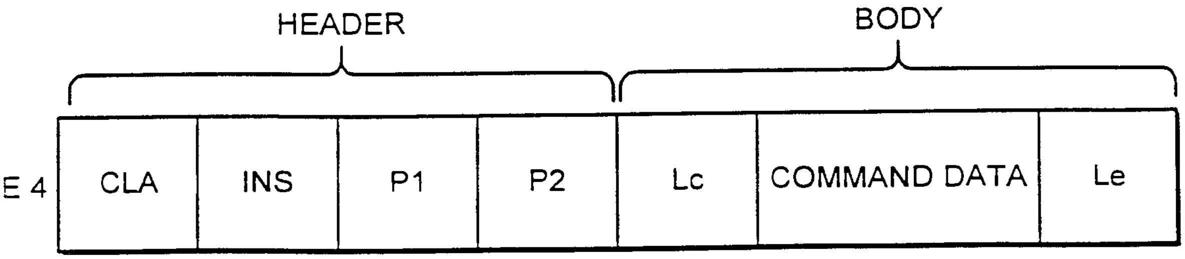


FIG. 5D

T = 1 TRANSMISSION PROTOCOL

PROLOGUE FIELD			INFORMATION FIELD	EPILOGUE FIELD
NOISE ADDRESS NAD	PROTOCOL CONTROL BYTE PCB	LENGTH LEN	INF	ERROR DETECTION CODE (LRC OR CRC) EDC
1 BYTE	1 BYTE	1 BYTE	0 - 254 BYTES	1 - 2 BYTES

FIG. 6

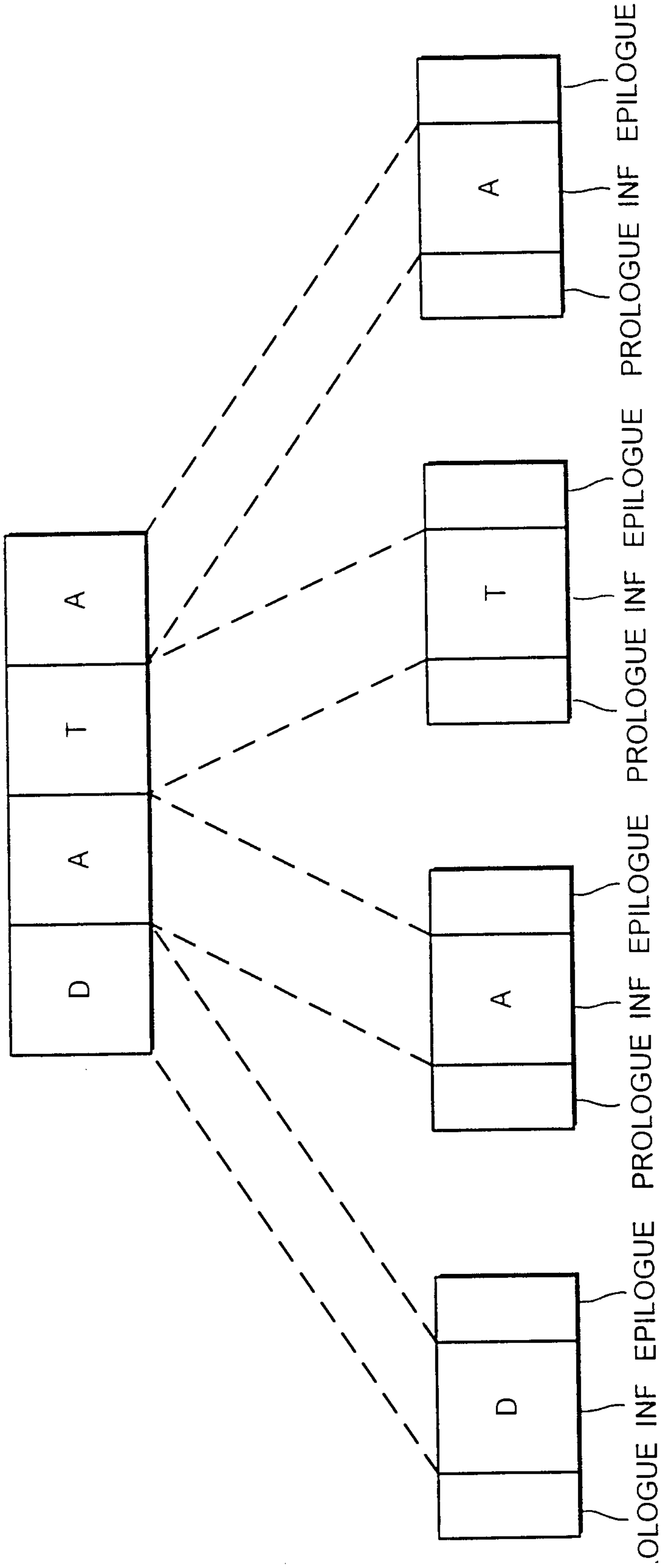

D
A
T
A
D
OLOGUE INF EPILOGUE
A
PROLOGUE INF EPILOGUE
T
PROLOGUE INF EPILOGUE
A
PROLOGUE INF EPILOGUE

FIG. 7

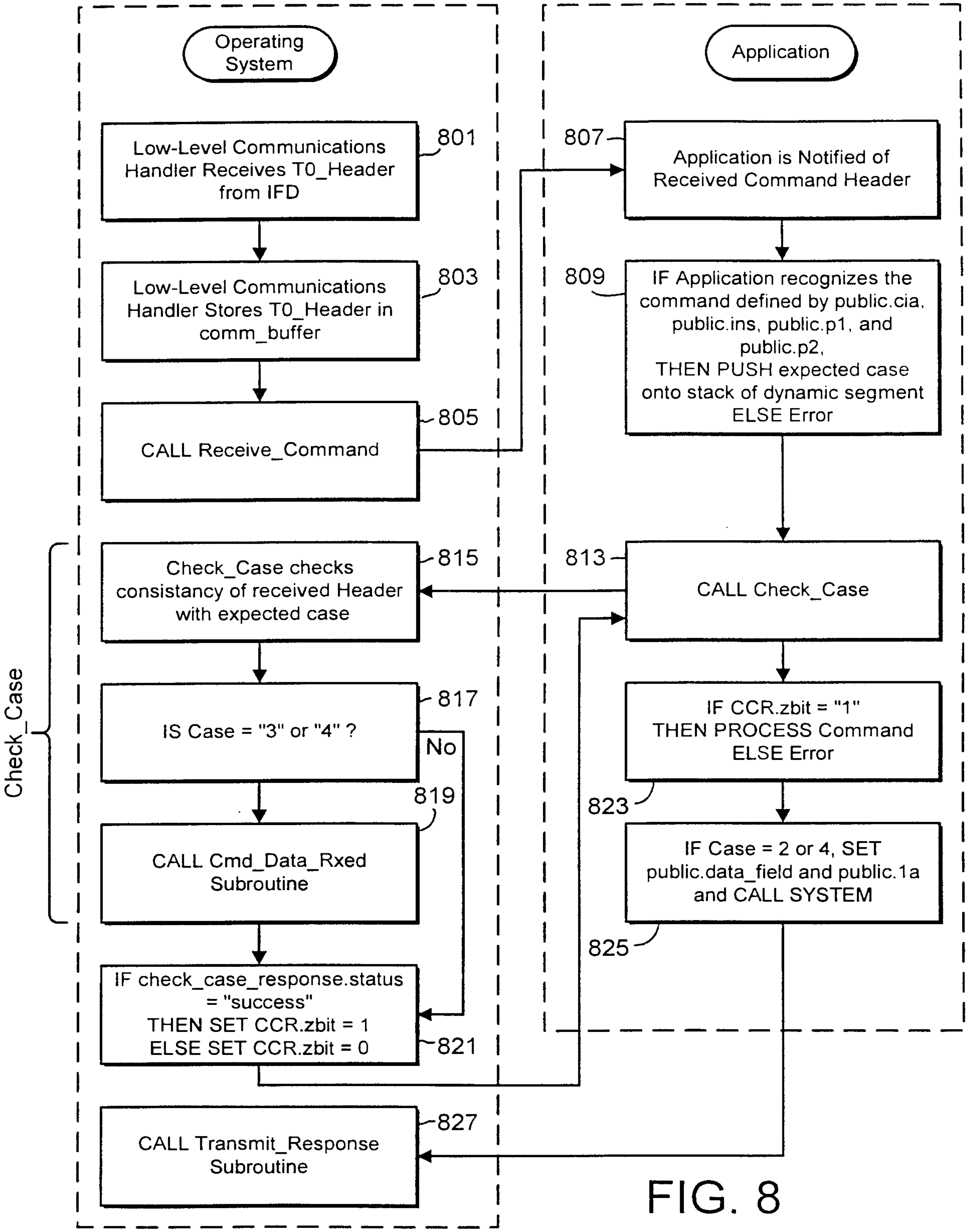

Operating System
Application
Low-Level Communications Handler Receives T0_Header from IFD
801
Low-Level Communications Handler Stores T0_Header in comm_buffer
803
CALL Receive_Command
805
807
Application is Notified of Received Command Header
809
IF Application recognizes the command defined by public.cia, public.ins, public.p1, and public.p2, THEN PUSH expected case onto stack of dynamic segment ELSE Error
Check_Case
Check_Case checks consistancy of received Header with expected case
815
813
CALL Check_Case
IS Case = "3" or "4" ?
817
No
819
CALL Cmd_Data_Rxed Subroutine
IF CCR.zbit = "1" THEN PROCESS Command ELSE Error
823
IF Case = 2 or 4, SET public.data_field and public.1a and CALL SYSTEM
825
IF check_case_response.status = "success" THEN SET CCR.zbit = 1 ELSE SET CCR.zbit = 0
821
CALL Transmit_Response Subroutine
827

FIG. 8

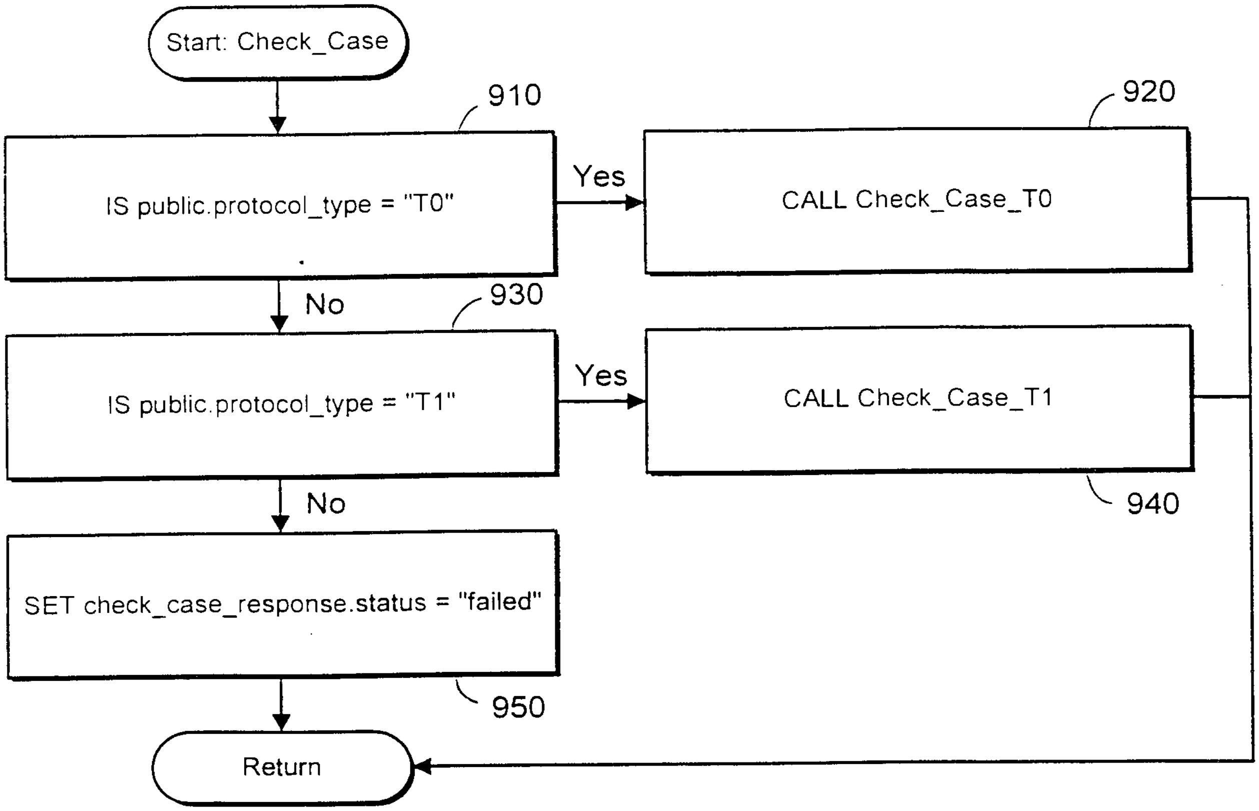

Start: Check_Case
910
IS public.protocol_type = "T0"
Yes
920
CALL Check_Case_T0
No
930
IS public.protocol_type = "T1"
Yes
CALL Check_Case_T1
940
No
SET check_case_response.status = "failed"
950
Return

FIG. 9

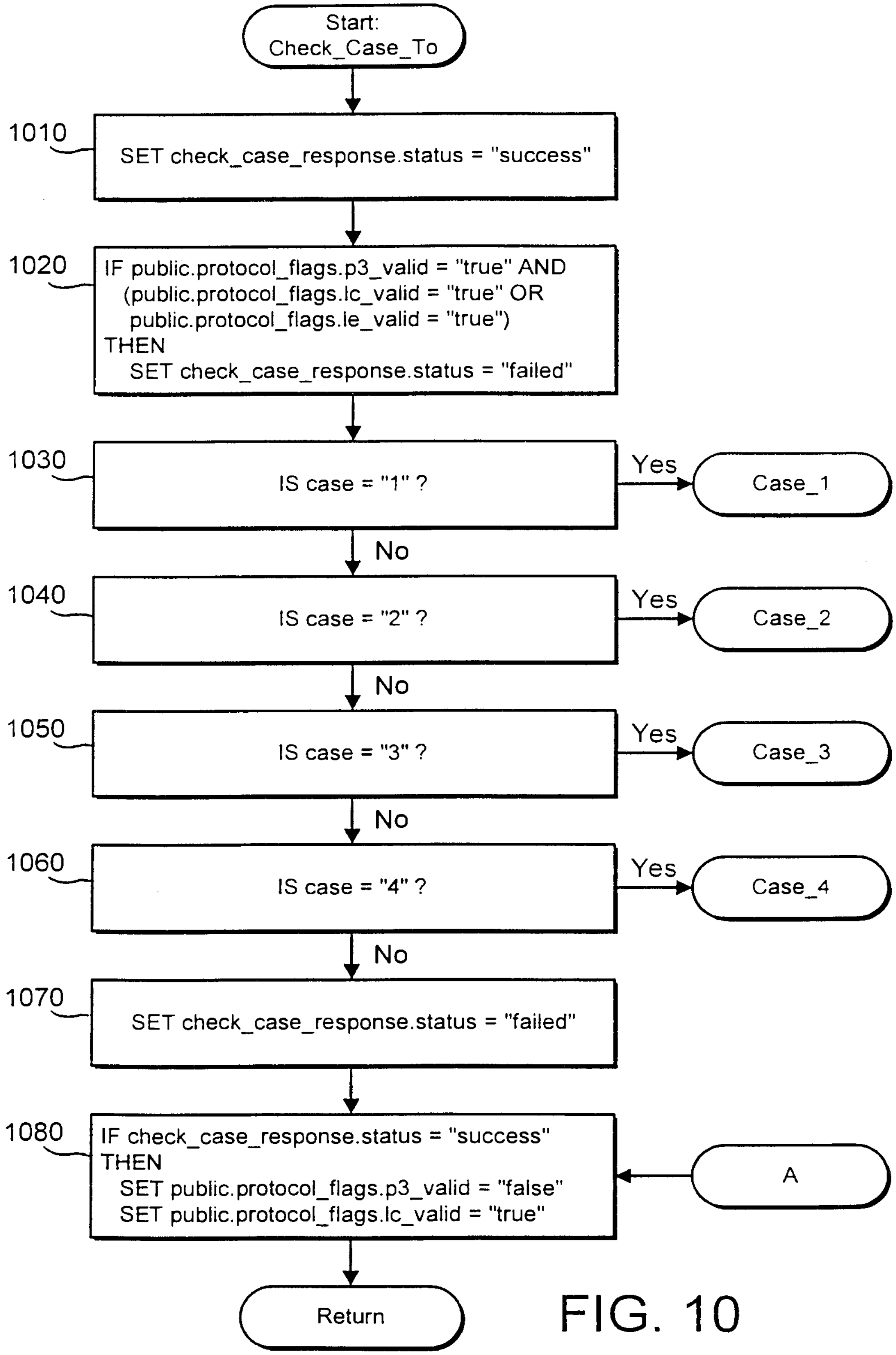

Start:
Check_Case_To
1010
SET check_case_response.status = "success"
1020
IF public.protocol_flags.p3_valid = "true" AND
(public.protocol_flags.lc_valid = "true" OR
public.protocol_flags.le_valid = "true")
THEN
SET check_case_response.status = "failed"
1030
IS case = "1" ?
Yes
Case_1
No
1040
IS case = "2" ?
Yes
Case_2
No
1050
IS case = "3" ?
Yes
Case_3
No
1060
IS case = "4" ?
Yes
Case_4
No
1070
SET check_case_response.status = "failed"
1080
IF check_case_response.status = "success"
THEN
SET public.protocol_flags.p3_valid = "false"
SET public.protocol_flags.lc_valid = "true"
A
Return

FIG. 10

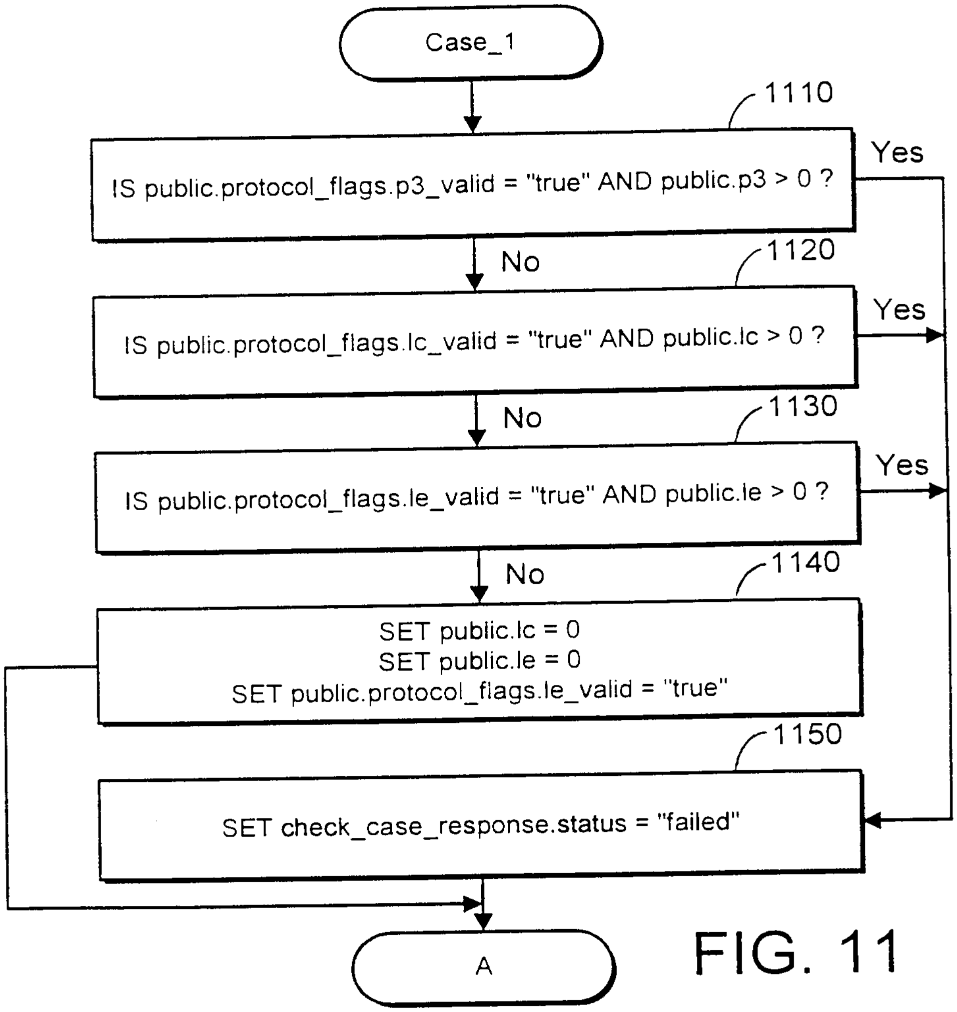

Case_1
1110
IS public.protocol_flags.p3_valid = "true" AND public.p3 > 0 ?
Yes
No
1120
IS public.protocol_flags.lc_valid = "true" AND public.lc > 0 ?
Yes
No
1130
IS public.protocol_flags.le_valid = "true" AND public.le > 0 ?
Yes
No
1140
SET public.lc = 0
SET public.le = 0
SET public.protocol_flags.le_valid = "true"
1150
SET check_case_response.status = "failed"
A

FIG. 11

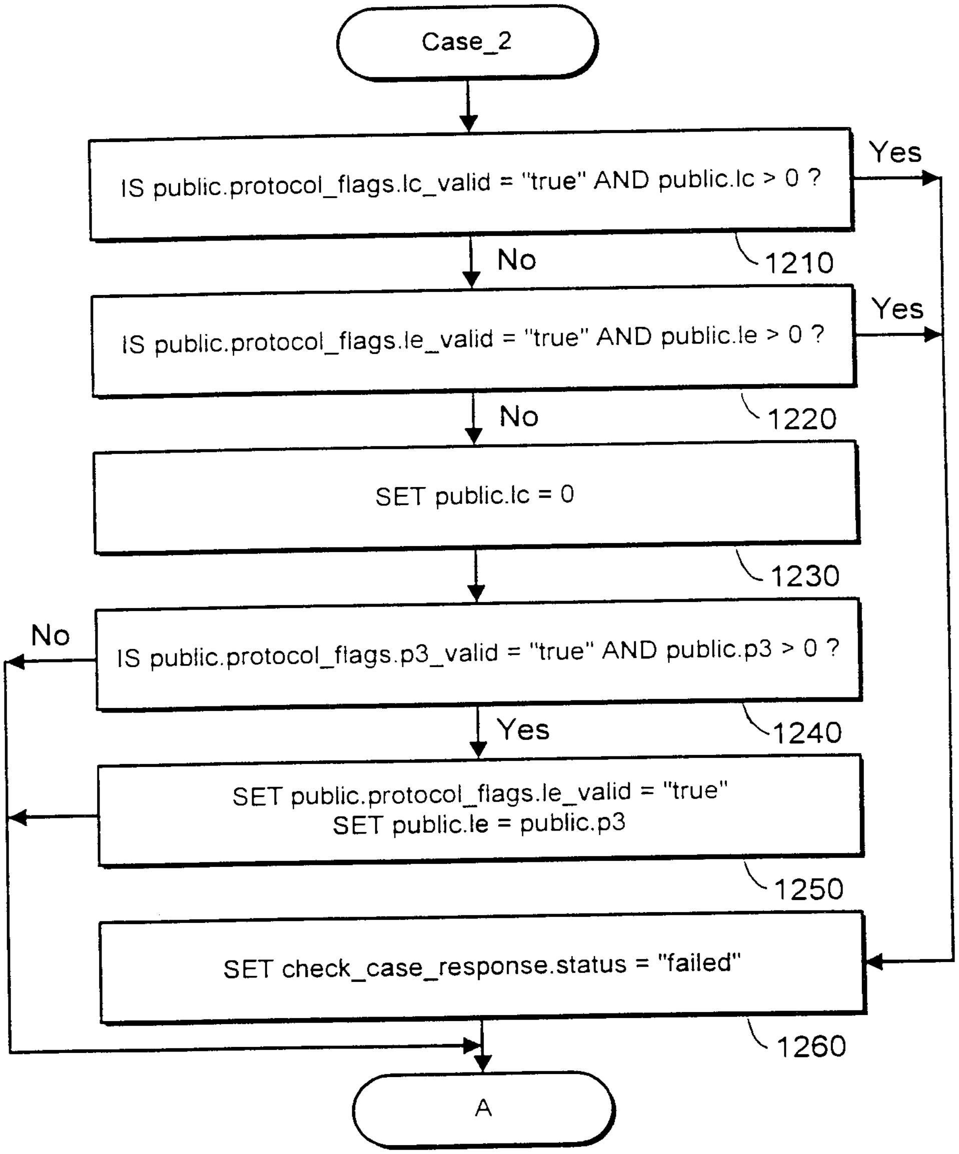

Case_2
IS public.protocol_flags.lc_valid = "true" AND public.lc > 0 ?
Yes
No
1210
IS public.protocol_flags.le_valid = "true" AND public.le > 0 ?
Yes
No
1220
SET public.lc = 0
1230
IS public.protocol_flags.p3_valid = "true" AND public.p3 > 0 ?
No
Yes
1240
SET public.protocol_flags.le_valid = "true"
SET public.le = public.p3
1250
SET check_case_response.status = "failed"
1260
A

FIG. 12

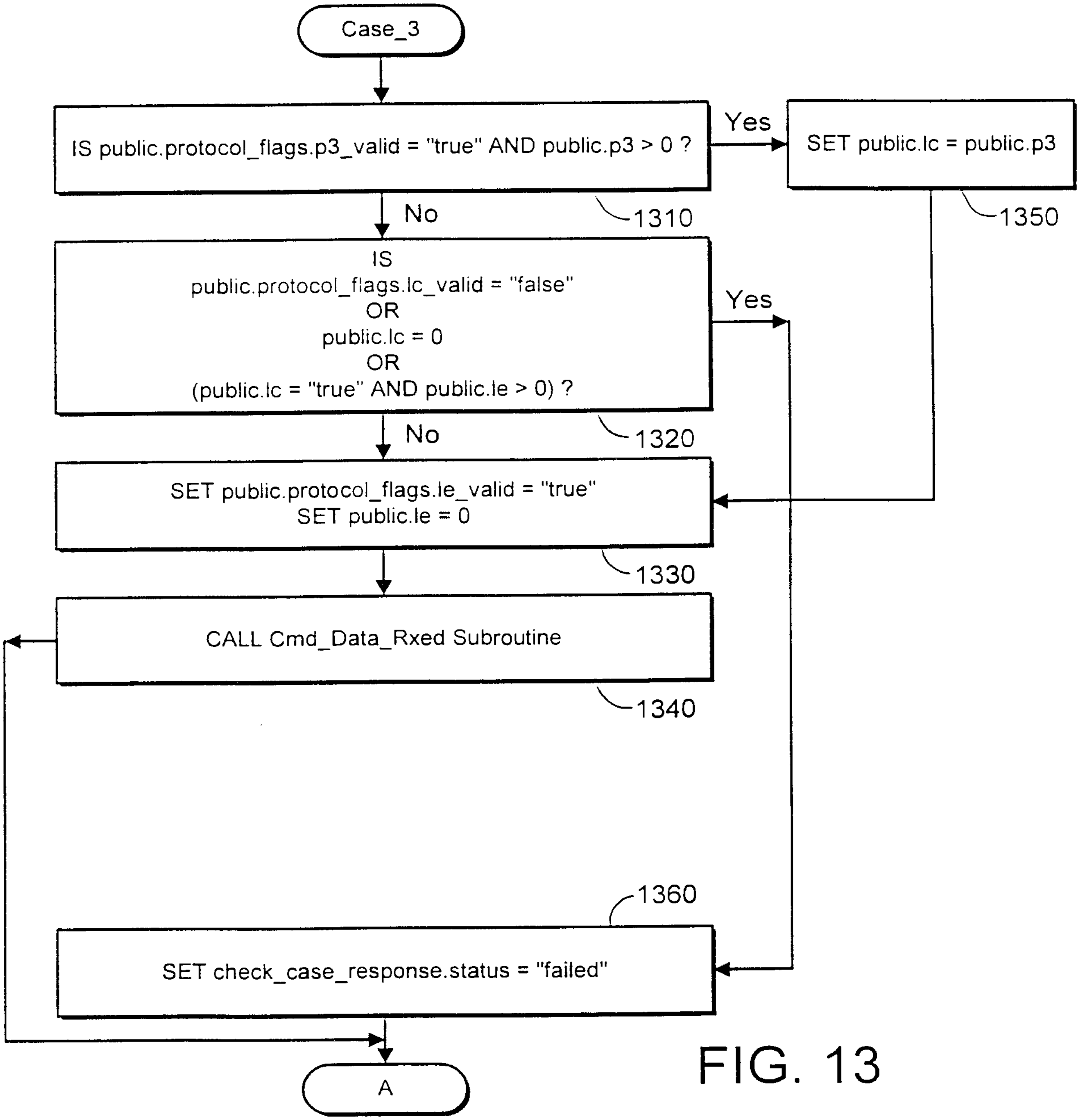

Case_3
IS public.protocol_flags.p3_valid = "true" AND public.p3 > 0 ?
Yes
SET public.lc = public.p3
1310
1350
No
IS
public.protocol_flags.lc_valid = "false"
OR
public.lc = 0
OR
(public.lc = "true" AND public.le > 0) ?
Yes
No
1320
SET public.protocol_flags.le_valid = "true"
SET public.le = 0
1330
CALL Cmd_Data_Rxed Subroutine
1340
1360
SET check_case_response.status = "failed"
A

FIG. 13

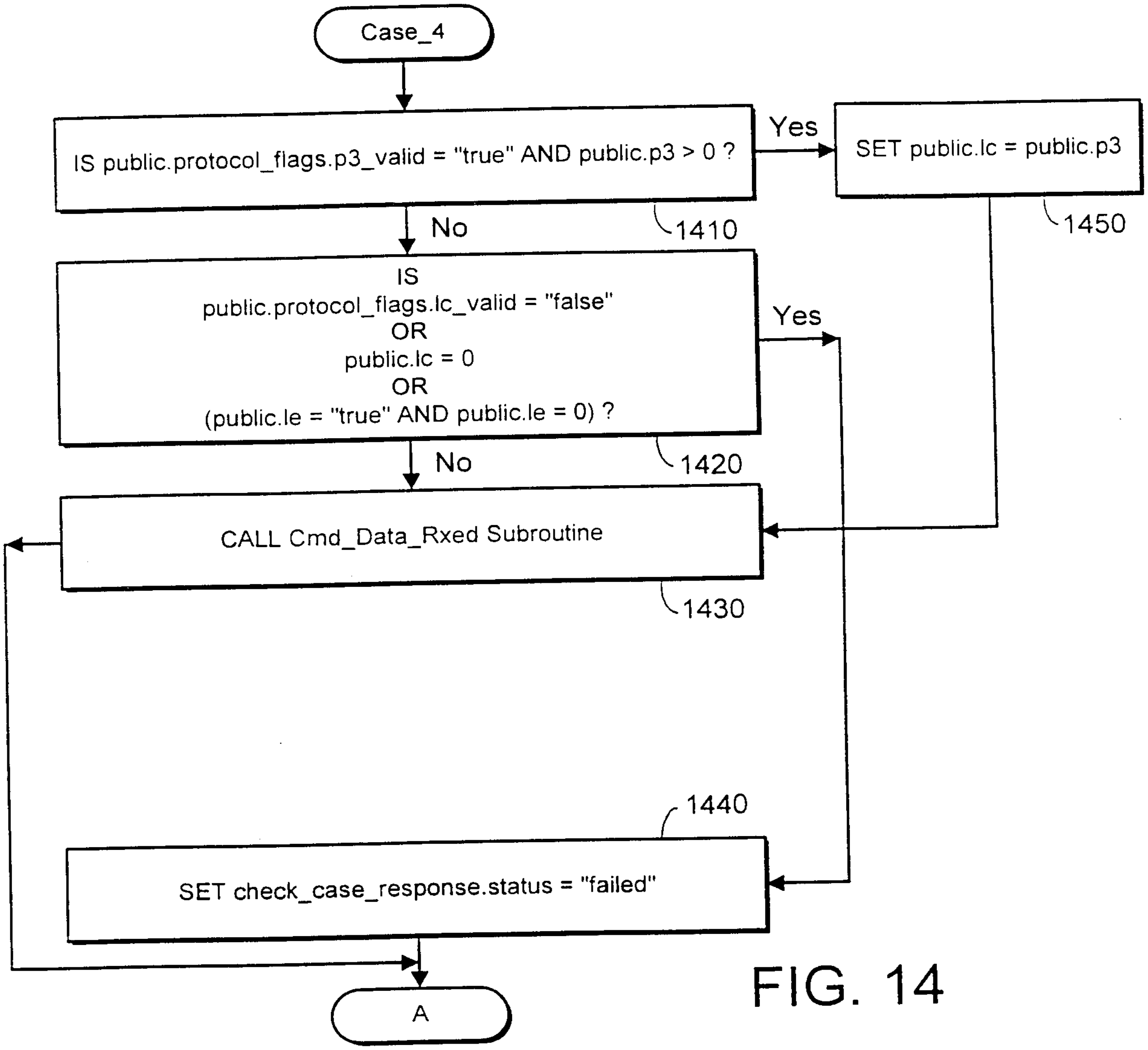

Case_4
IS public.protocol_flags.p3_valid = "true" AND public.p3 > 0 ?
Yes
SET public.lc = public.p3
1450
No
1410
IS
public.protocol_flags.lc_valid = "false"
OR
public.lc = 0
OR
(public.le = "true" AND public.le = 0) ?
Yes
No
1420
CALL Cmd_Data_Rxed Subroutine
1430
1440
SET check_case_response.status = "failed"
A

FIG. 14

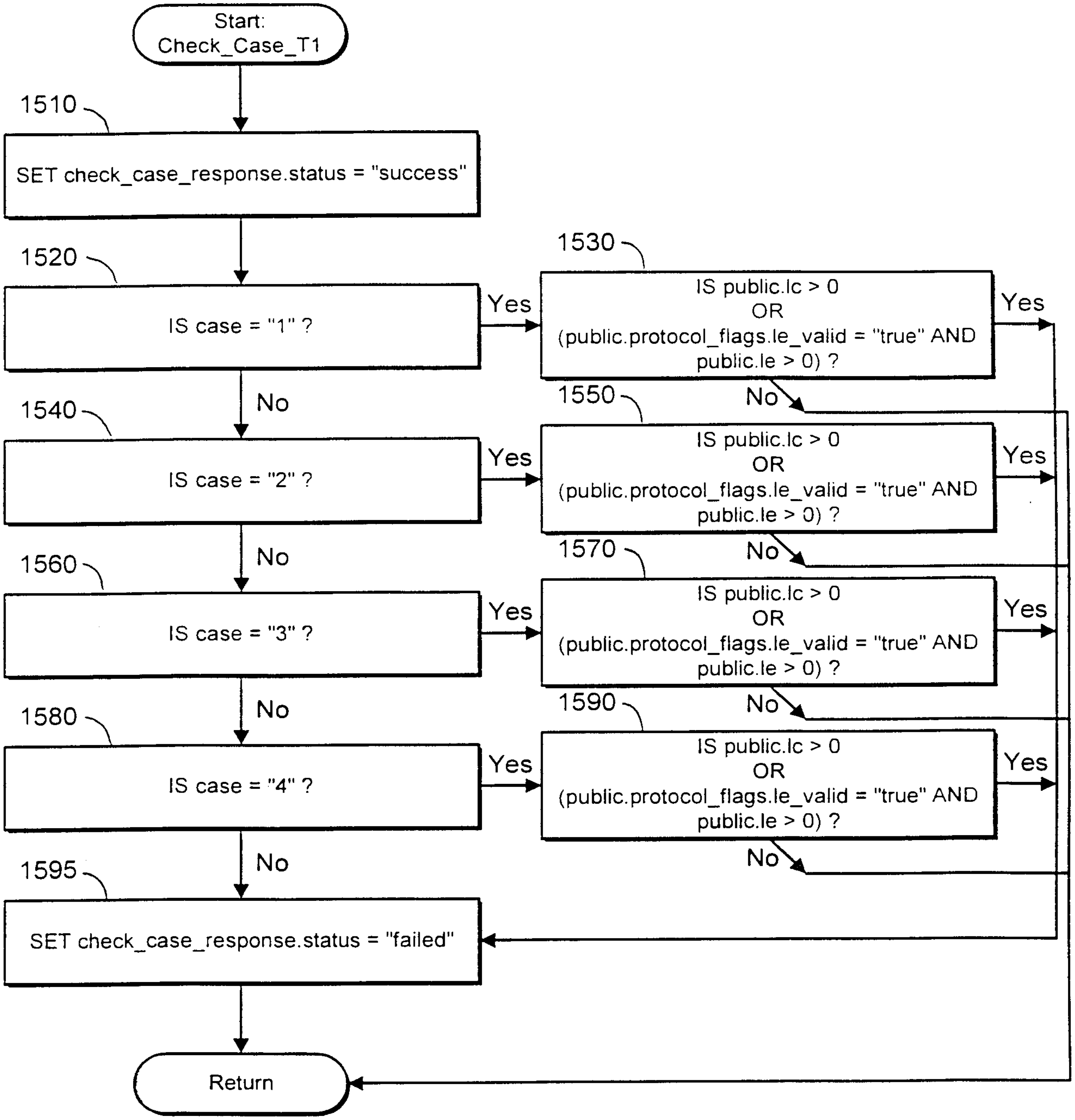

Start:
Check_Case_T1
1510
SET check_case_response.status = "success"
1520
IS case = "1" ?
Yes
1530
IS public.lc > 0
OR
(public.protocol_flags.le_valid = "true" AND
public.le > 0) ?
Yes
No
1540
IS case = "2" ?
Yes
1550
IS public.lc > 0
OR
(public.protocol_flags.le_valid = "true" AND
public.le > 0) ?
Yes
No
1560
IS case = "3" ?
Yes
1570
IS public.lc > 0
OR
(public.protocol_flags.le_valid = "true" AND
public.le > 0) ?
Yes
No
1580
IS case = "4" ?
Yes
1590
IS public.lc > 0
OR
(public.protocol_flags.le_valid = "true" AND
public.le > 0) ?
Yes
No
1595
SET check_case_response.status = "failed"
Return

FIG. 15

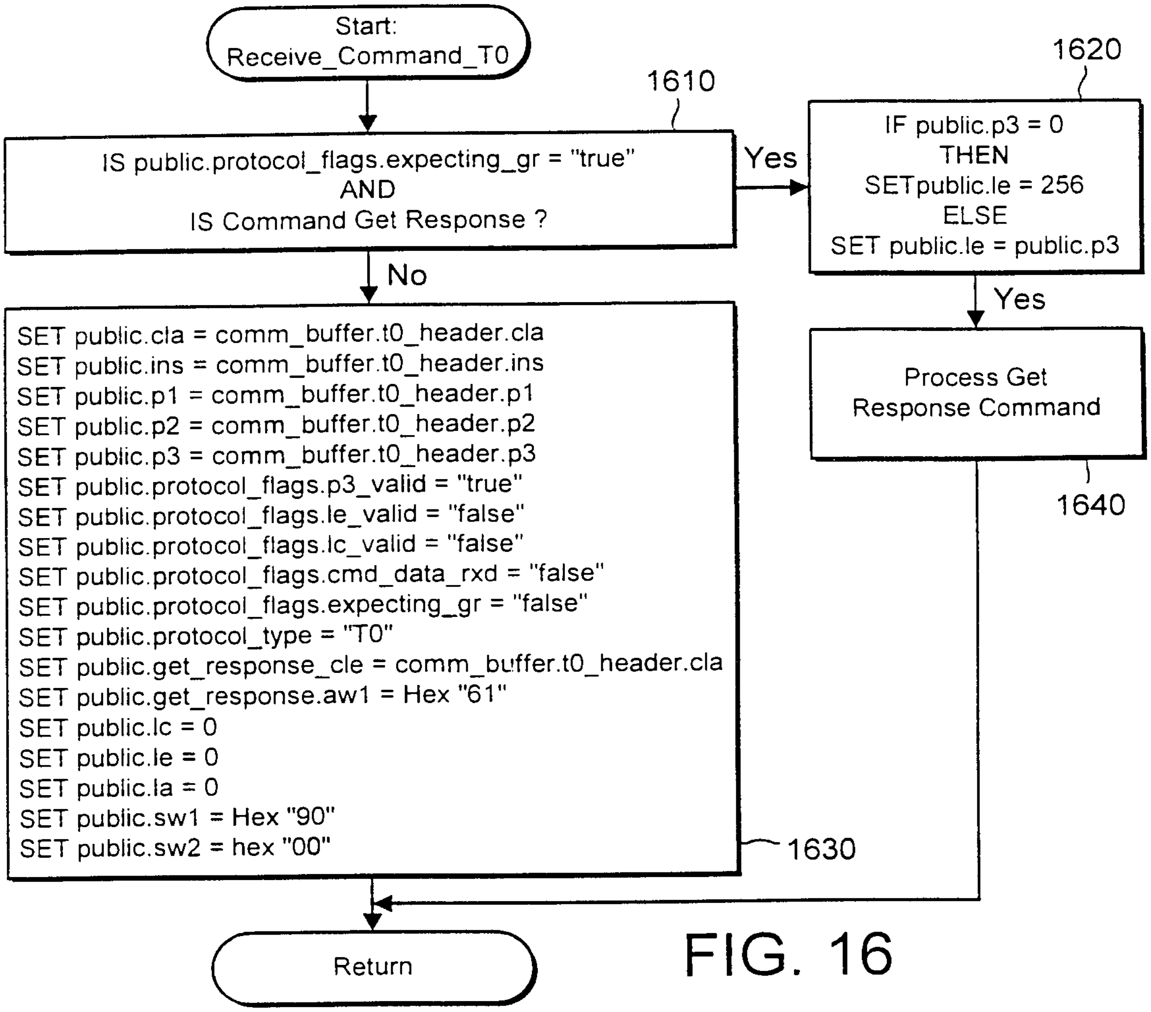

Start:
Receive_Command_T0
1610
IS public.protocol_flags.expecting_gr = "true"
AND
IS Command Get Response ?
Yes
1620
IF public.p3 = 0
THEN
SETpublic.le = 256
ELSE
SET public.le = public.p3
No
SET public.cla = comm_buffer.t0_header.cla
SET public.ins = comm_buffer.t0_header.ins
SET public.p1 = comm_buffer.t0_header.p1
SET public.p2 = comm_buffer.t0_header.p2
SET public.p3 = comm_buffer.t0_header.p3
SET public.protocol_flags.p3_valid = "true"
SET public.protocol_flags.le_valid = "false"
SET public.protocol_flags.lc_valid = "false"
SET public.protocol_flags.cmd_data_rxd = "false"
SET public.protocol_flags.expecting_gr = "false"
SET public.protocol_type = "T0"
SET public.get_response_cle = comm_buffer.t0_header.cla
SET public.get_response.aw1 = Hex "61"
SET public.lc = 0
SET public.le = 0
SET public.la = 0
SET public.sw1 = Hex "90"
SET public.sw2 = hex "00"
1630
Yes
Process Get
Response Command
1640
Return

FIG. 16

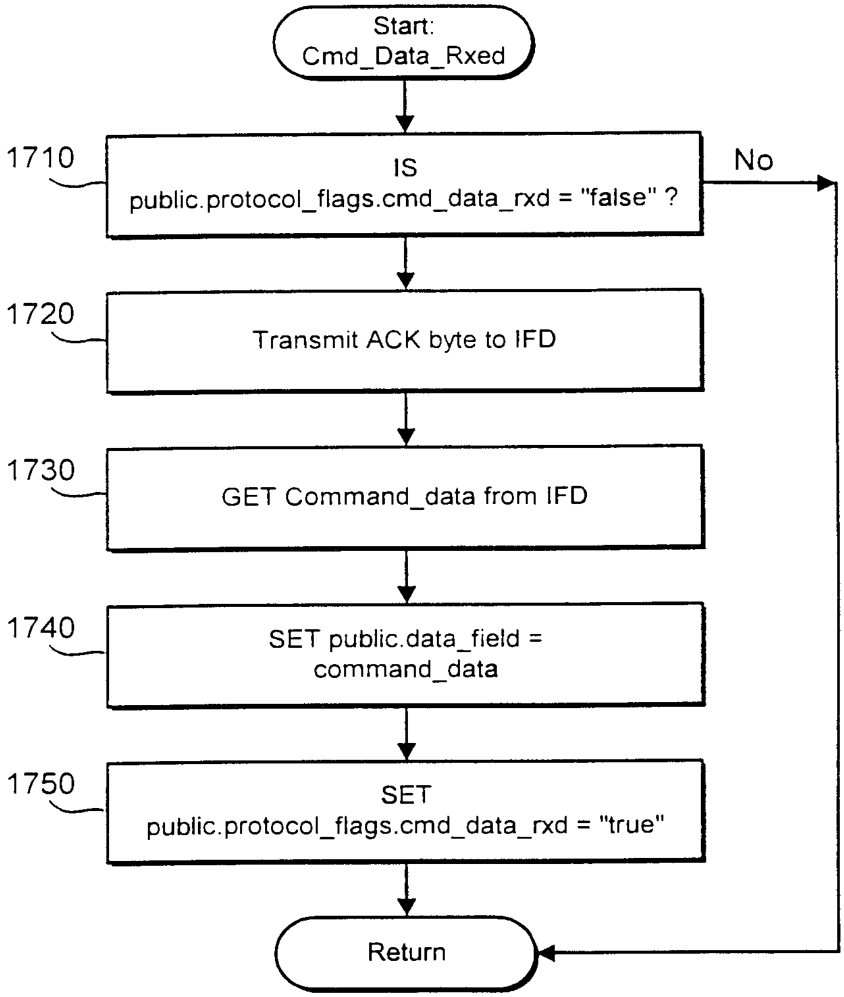
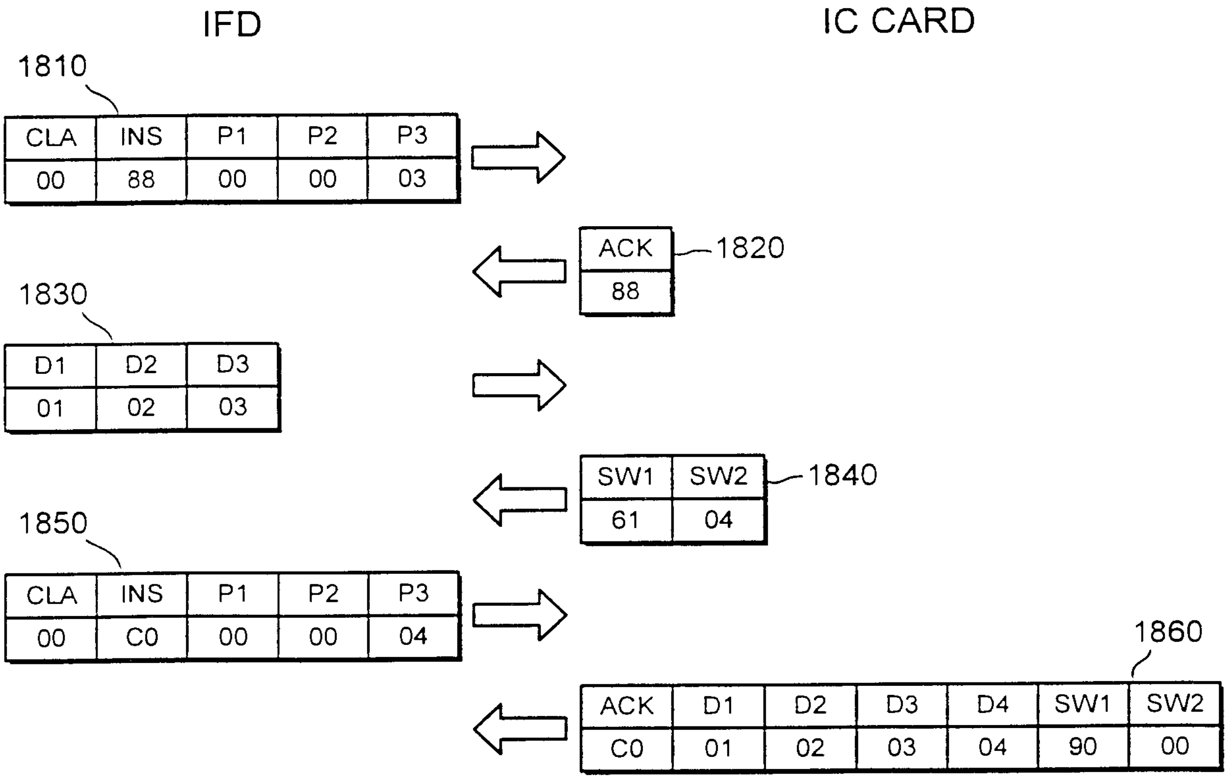

Start:
Cmd_Data_Rxed
1710
IS
public.protocol_flags.cmd_data_rxd = "false" ?
No
1720
Transmit ACK byte to IFD
1730
GET Command_data from IFD
1740
SET public.data_field =
command_data
1750
SET
public.protocol_flags.cmd_data_rxd = "true"
Return

FIG. 17



CLA	INS	P1	P2	P3
00	88	00	00	03

ACK
88

D1	D2	D3
01	02	03

SW1	SW2
61	04

CLA	INS	P1	P2	P3
00	C0	00	00	04

ACK	D1	D2	D3	D4	SW1	SW2
C0	01	02	03	04	90	00

FIG. 18

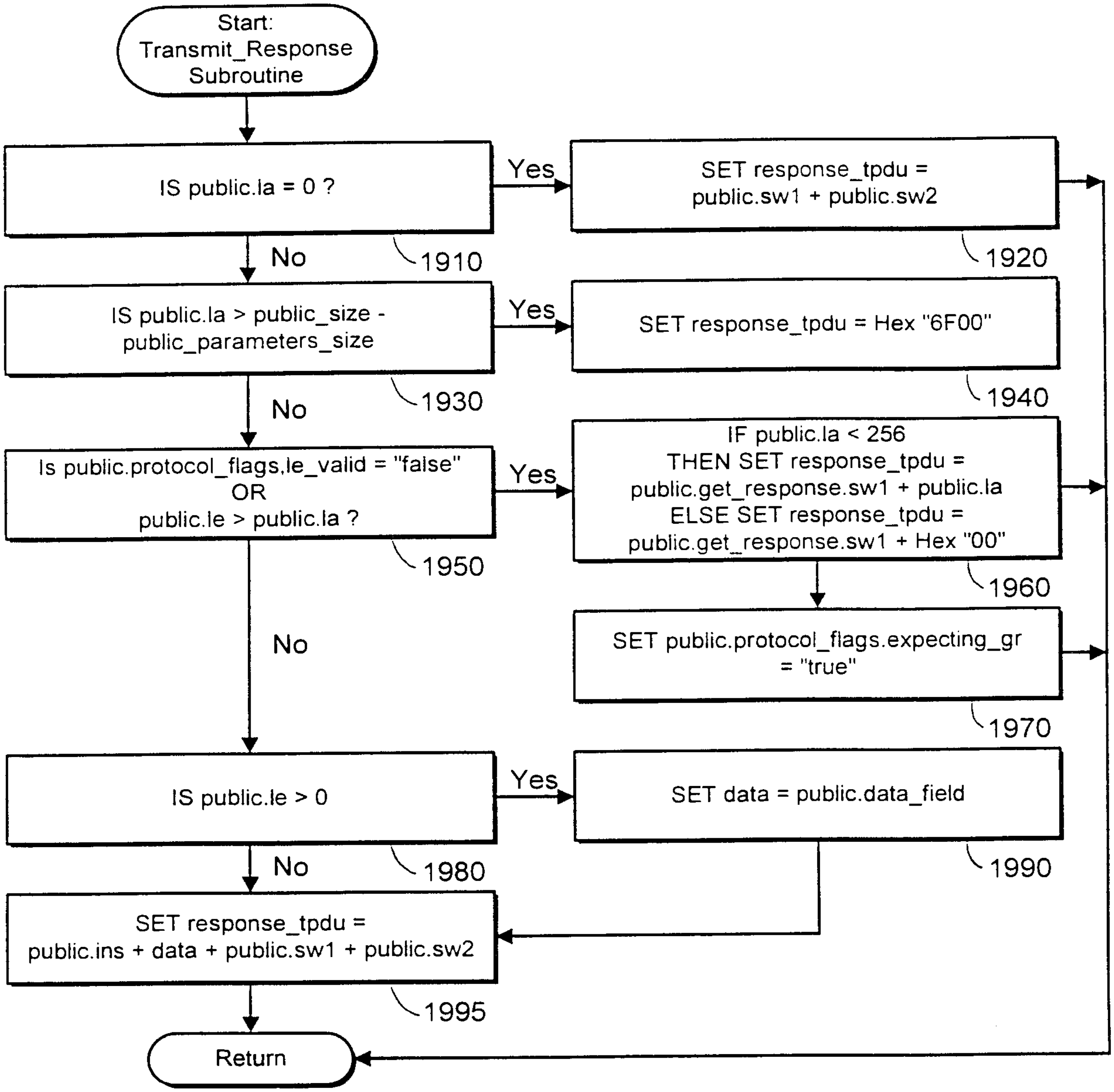

Start:
Transmit_Response
Subroutine
IS public.la = 0 ?
Yes
SET response_tpdu =
public.sw1 + public.sw2
No
1910
1920
IS public.la > public_size -
public_parameters_size
Yes
SET response_tpdu = Hex "6F00"
1930
1940
No
Is public.protocol_flags,le_valid = "false"
OR
public.le > public.la ?
Yes
IF public.la < 256
THEN SET response_tpdu =
public.get_response.sw1 + public.la
ELSE SET response_tpdu =
public.get_response.sw1 + Hex "00"
1950
1960
No
SET public.protocol_flags.expecting_gr
= "true"
1970
IS public.le > 0
Yes
SET data = public.data_field
No
1980
1990
SET response_tpdu =
public.ins + data + public.sw1 + public.sw2
1995
Return

FIG. 19

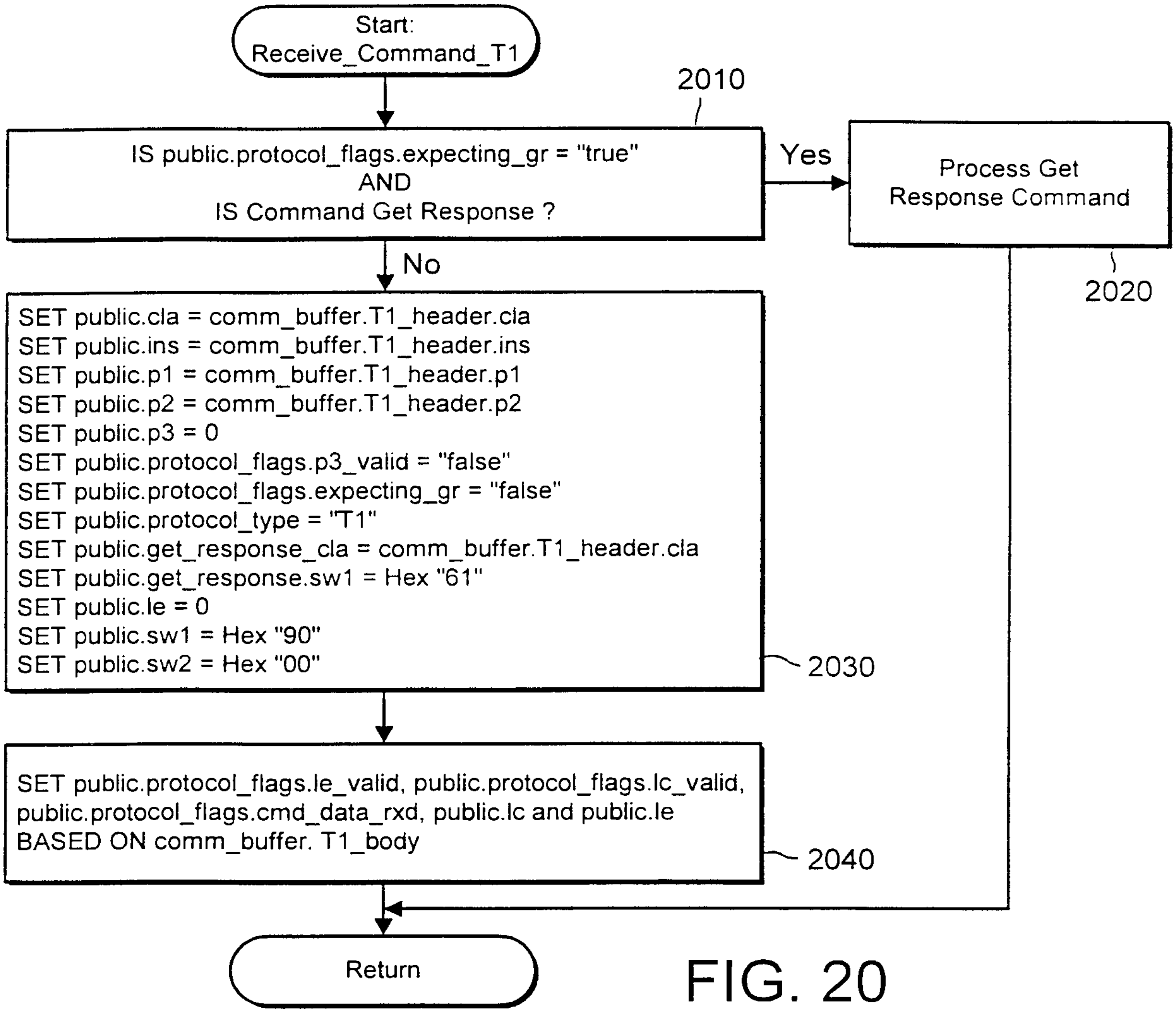

Start:
Receive_Command_T1
2010
IS public.protocol_flags.expecting_gr = "true"
AND
IS Command Get Response ?
Yes
Process Get
Response Command
2020
No
SET public.cla = comm_buffer.T1_header.cla
SET public.ins = comm_buffer.T1_header.ins
SET public.p1 = comm_buffer.T1_header.p1
SET public.p2 = comm_buffer.T1_header.p2
SET public.p3 = 0
SET public.protocol_flags.p3_valid = "false"
SET public.protocol_flags.expecting_gr = "false"
SET public.protocol_type = "T1"
SET public.get_response_cla = comm_buffer.T1_header.cla
SET public.get_response.sw1 = Hex "61"
SET public.le = 0
SET public.sw1 = Hex "90"
SET public.sw2 = Hex "00"
2030
SET public.protocol_flags.le_valid, public.protocol_flags.lc_valid,
public.protocol_flags.cmd_data_rxd, public.lc and public.le
BASED ON comm_buffer. T1_body
2040
Return

FIG. 20

ANNEX A TO THE DRAWINGS

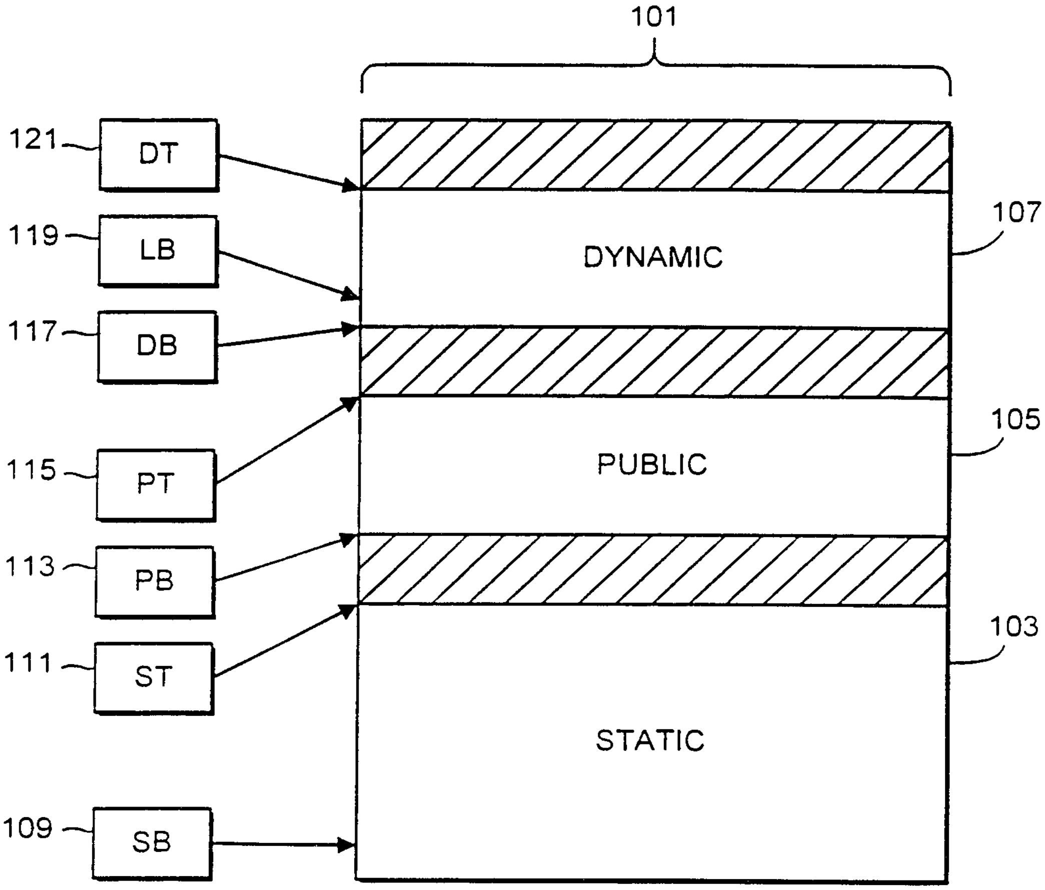


FIG. 1

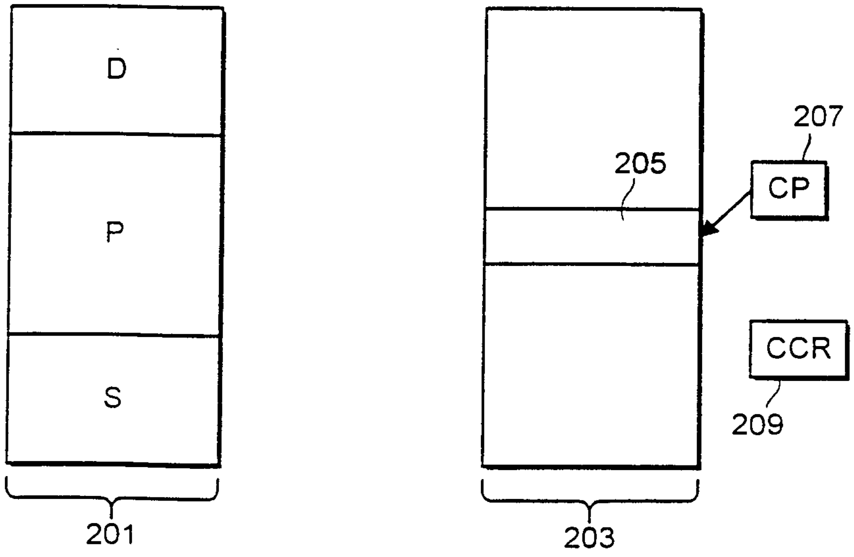


FIG. 2

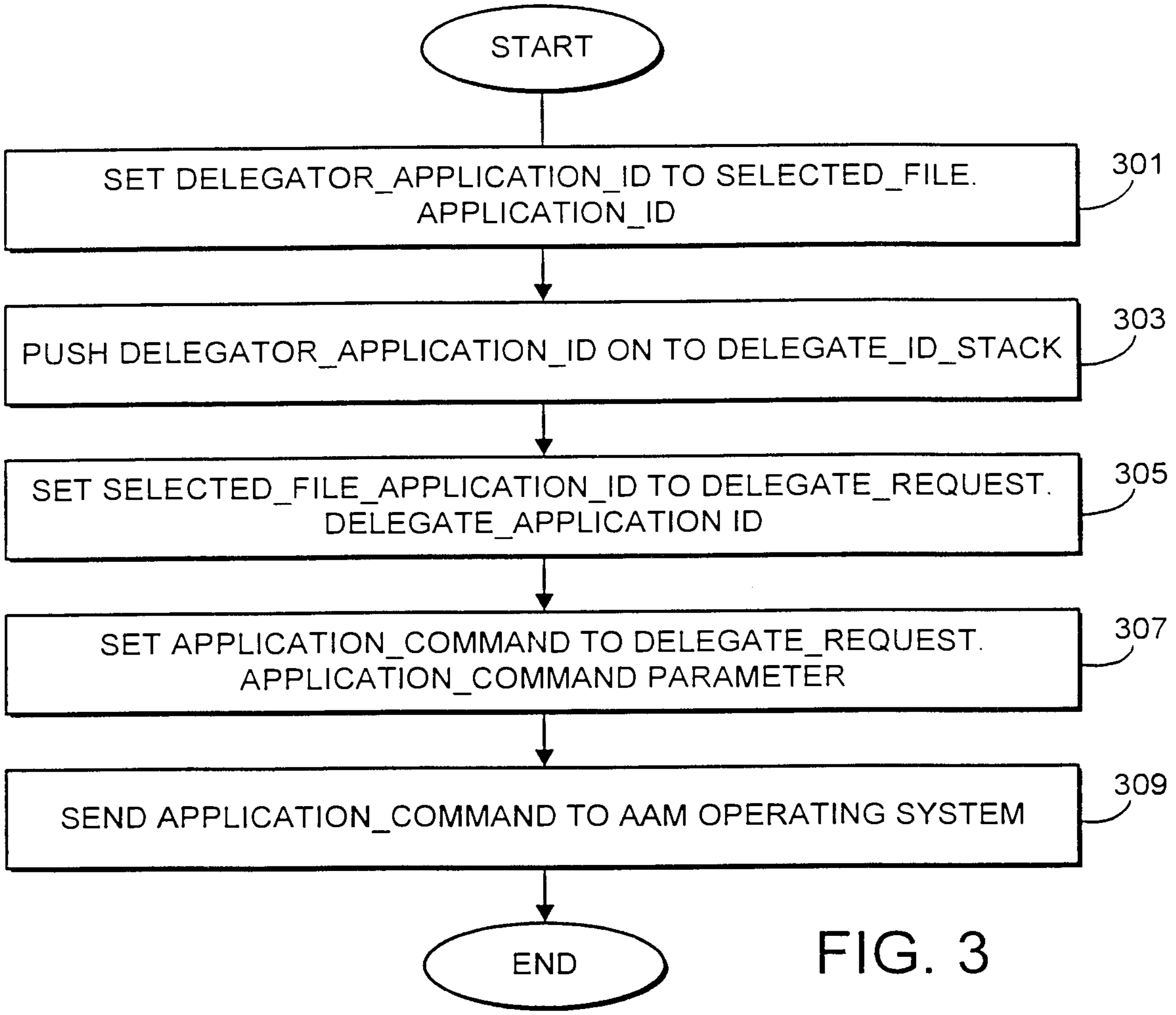

START
SET DELEGATOR_APPLICATION_ID TO SELECTED_FILE. APPLICATION_ID
301
PUSH DELEGATOR_APPLICATION_ID ON TO DELEGATE_ID_STACK
303
SET SELECTED_FILE_APPLICATION_ID TO DELEGATE_REQUEST. DELEGATE_APPLICATION ID
305
SET APPLICATION_COMMAND TO DELEGATE_REQUEST. APPLICATION_COMMAND PARAMETER
307
SEND APPLICATION_COMMAND TO AAM OPERATING SYSTEM
309
END

FIG. 3

ANNEX A TO THE DRAWINGS

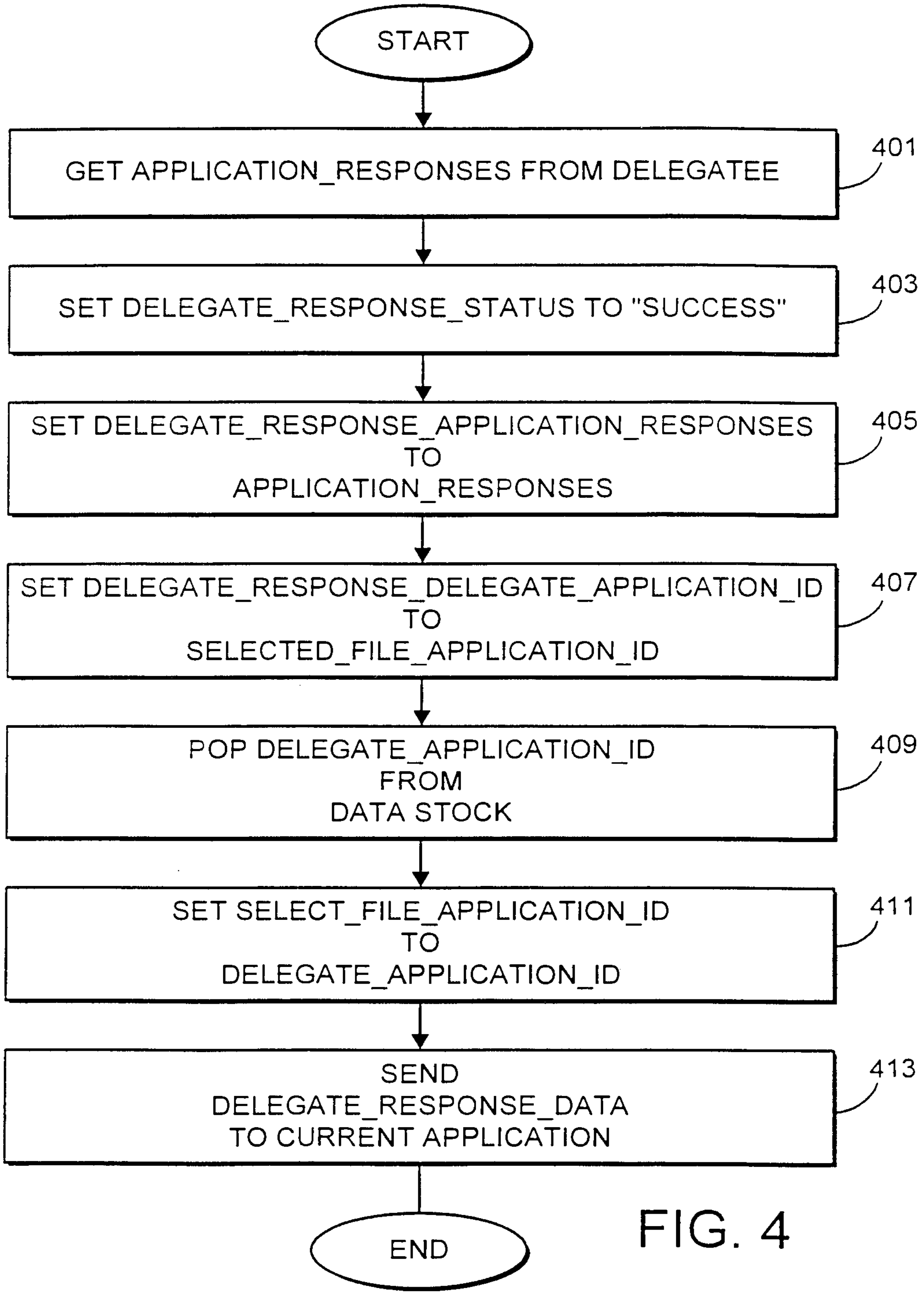

START
GET APPLICATION_RESPONSES FROM DELEGATEE
401
SET DELEGATE_RESPONSE_STATUS TO "SUCCESS"
403
SET DELEGATE_RESPONSE_APPLICATION_RESPONSES TO APPLICATION_RESPONSES
405
SET DELEGATE_RESPONSE_DELEGATE_APPLICATION_ID TO SELECTED_FILE_APPLICATION_ID
407
POP DELEGATE_APPLICATION_ID FROM DATA STOCK
409
SET SELECT_FILE_APPLICATION_ID TO DELEGATE_APPLICATION_ID
411
SEND DELEGATE_RESPONSE_DATA TO CURRENT APPLICATION
413
END

FIG. 4

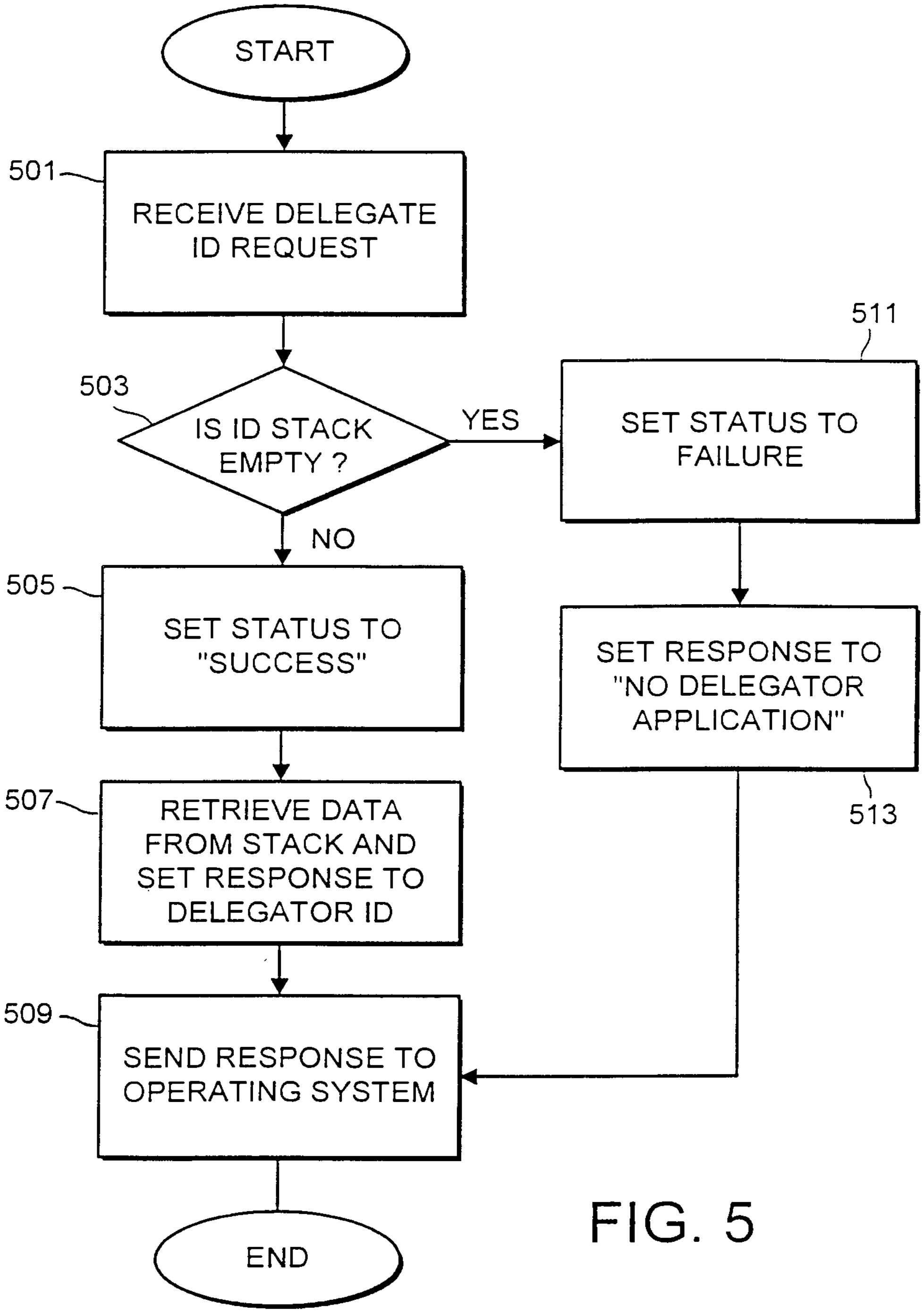

START
501
RECEIVE DELEGATE ID REQUEST
503
IS ID STACK EMPTY ?
YES
NO
511
SET STATUS TO FAILURE
505
SET STATUS TO "SUCCESS"
SET RESPONSE TO "NO DELEGATOR APPLICATION"
513
507
RETRIEVE DATA FROM STACK AND SET RESPONSE TO DELEGATOR ID
509
SEND RESPONSE TO OPERATING SYSTEM
END

FIG. 5

ANNEX A TO THE DRAWINGS

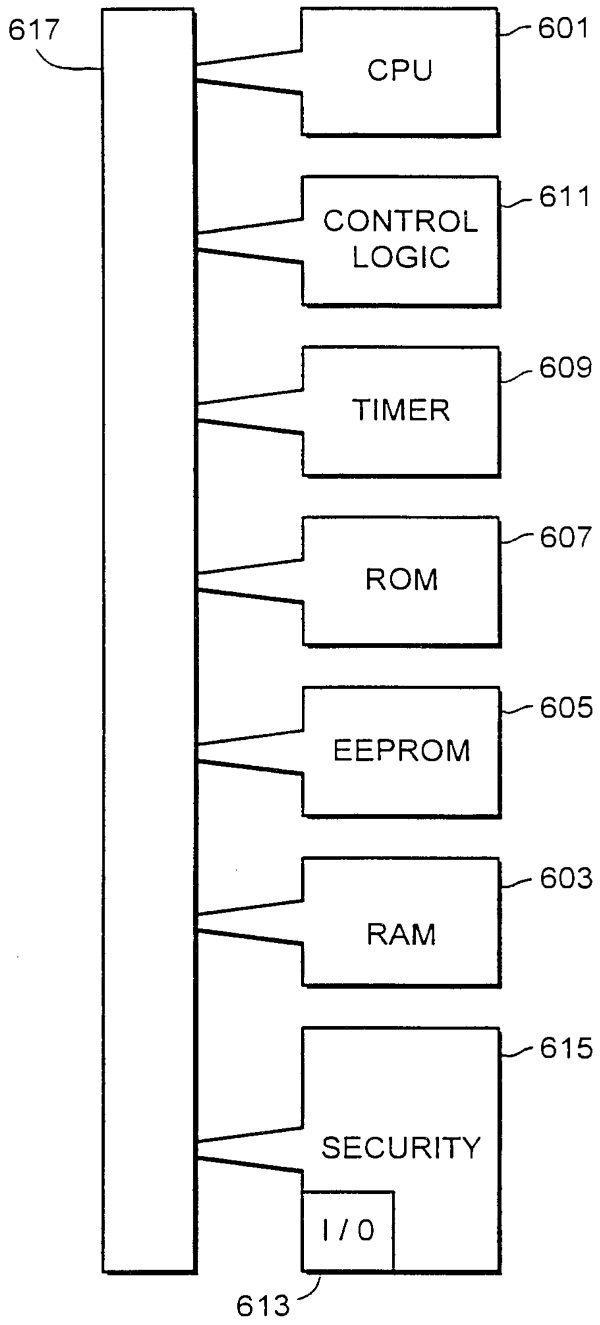


FIG. 6

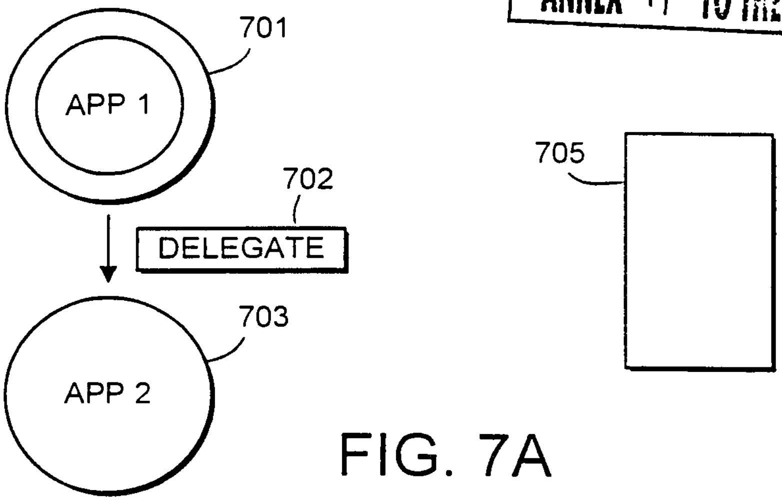


FIG. 7A

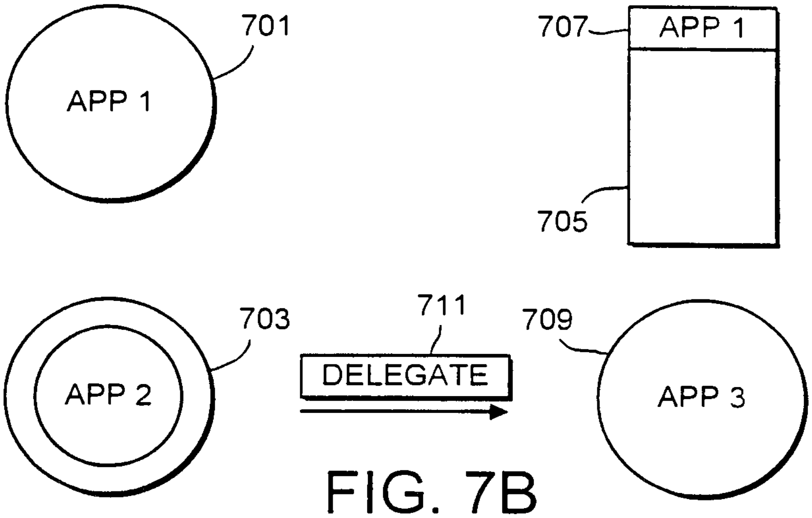


FIG. 7B

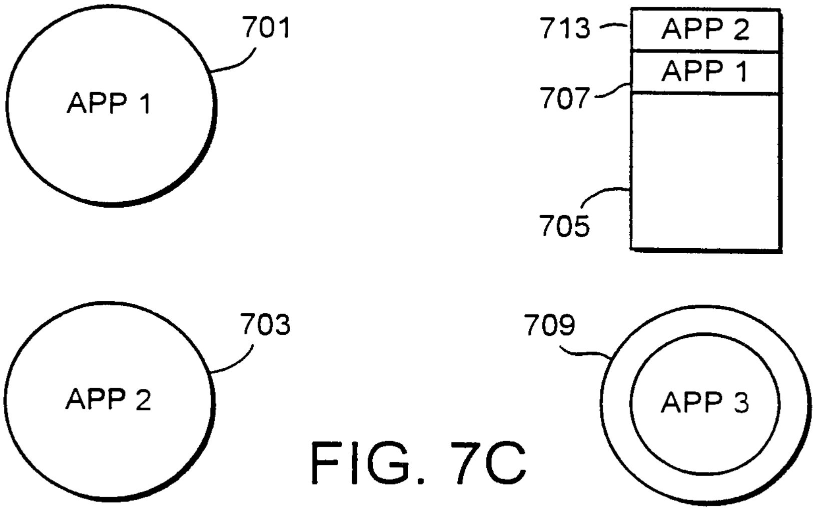


FIG. 7C

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : **G06K 19/07**	**A2**	**(11) International Publication Number:** **WO 98/52153** **(43) International Publication Date:** 19 November 1998 (19.11.98)

(21) International Application Number: PCT/GB98/01411

(22) International Filing Date: 14 May 1998 (14.05.98)

(30) Priority Data:

60/046,514	15 May 1997 (15.05.97)	US
60/046,543	15 May 1997 (15.05.97)	US
09/075,975	11 May 1998 (11.05.98)	US

(71) Applicant: MONDEX INTERNATIONAL LIMITED [GB/GB]; 47–53 Cannon Street, London EC4M 5SQ (GB).

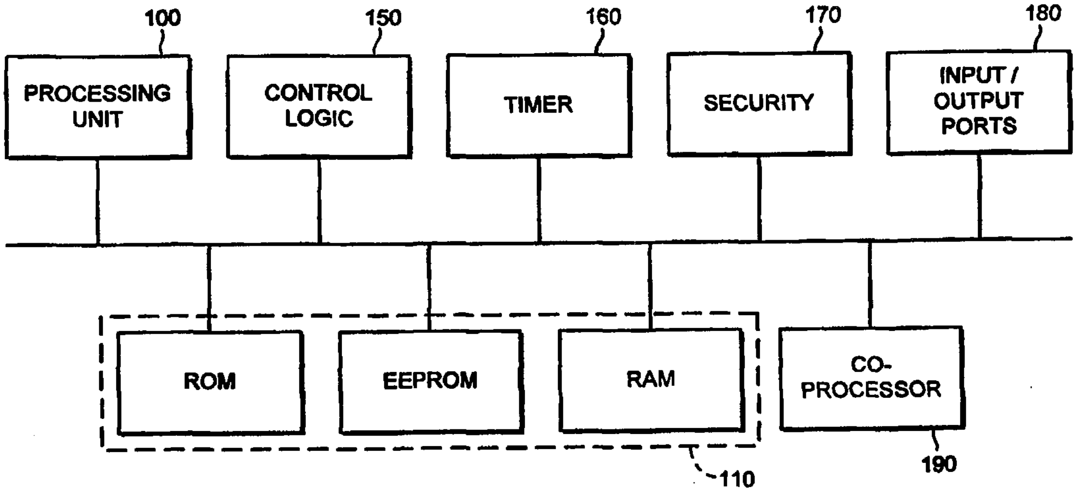
(72) Inventor: RICHARDS, Timothy, Philip; 32 Craig Mount, Radlett, Herts WD7 7LW (GB).

(74) Agent: POTTER, Julian, Mark; D. Young & Co., 21 New Fetter Lane, London EC4A 1DA (GB).

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

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

(54) Title: IC CARD WITH SHELL FEATURE



(57) Abstract

There is provided an integrated circuit card having an associated operating mode. The integrated circuit card includes: a microprocessor; a memory coupled to the microprocessor; data stored in the memory representative of the operating mode; an operating system stored in the memory for processing selected information in a first IC card format; a shell application stored in the memory for processing the selected information in a second IC card format; and means for routing the selected information to either the operating system or the shell application responsive to the operating mode. The selected information may be a command, such as a file access command.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

IC CARD WITH SHELL FEATURE

BACKGROUND OF INVENTION

Integrated circuit (IC) cards are becoming increasingly used for many different purposes in the world today, principally because they are ideal tools for the delivery of distributed, secure information processing at a low cost. An IC card, also called a “smart card,” is a card typically the size of a conventional credit card, but which contains a computer chip on the card. The computer chip on the IC card typically includes a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), a random access memory (RAM), an input/output (I/O) mechanism, and other circuitry to support the microprocessor in its operations. The computer chip can execute one or more applications stored on the card. Examples of applications that IC cards are being used to store and execute include credit/debit, electronic money/purse, telephone calling card, and loyalty reward applications.

As the use and application of IC cards has increased, IC card standards have been promulgated. For example, the International Organization for Standardization (ISO) and the International Engineering Consortium (IEC) have promulgated several industry-wide standards for IC cards, ISO/IEC 7816-1 through ISO 7816-8. The ISO/IEC standards provide, for example, general guidelines for file structures and referencing methods so that various applications and IC card operating systems can understand one another and work in a cohesive manner. Additionally, in the field of payment systems (such as credit and debit card systems), the EMV ‘96 Integrated Circuit Card Specification for Payment Systems, Version 3.0, June 30, 1996, available from MasterCard International Incorporated®, specifies file structures and file referencing methods that are generally compliant with ISO/IEC standards 7816-4 and 7816-5. Nonetheless, proprietary IC card standards exist that are not compliant with ISO/IEC standards.

The existence of multiple IC card standards is problematic to the IC card manufacturer, who is required to produce different versions of its IC cards, with different operating systems that are compatible with the different standards. Moreover, since operating systems are typically loaded into the ROM of an IC card when it is initially produced, each time a standard is updated or a new standard is adopted, an IC card manufacturer may be required to distribute new IC cards with an updated operating system compatible with the new or updated standard.

It would advantageous to the card manufacturer, card issuer, application provider, and card user if the operating system of an IC card was not required to be updated each time a new or updated IC card standard was promulgated. These and other technical problems are addressed by embodiments of the present invention.

SUMMARY OF THE INVENTION

The present invention addresses the aforementioned technical problems by introducing a "shell" application that executes "on top" of the operating system and that handles the implementation of IC card standards that are not compatible with the initially loaded operating system of the IC card. Advantageously, the shell application supplements the IC card standards with which the IC card is compatible. Thus, as standards change or new standards are adopted, an IC card needs to be updated only with a new shell application, rather than having to be updated with a new operating system.

According to a preferred embodiment of the present invention, there is provided an integrated circuit card having an associated operating mode. The integrated circuit card includes: a microprocessor; a memory coupled to the microprocessor; data stored in the memory representative of the operating mode; an operating system stored in the memory for processing selected information in a first IC card format; a shell application stored in the memory for processing the selected information in a second IC card format; and means for routing the selected information to either the operating system or the shell application responsive to the operating mode. The selected information may be a command, such as a file access command. In addition, the selected information may be associated with a file structure format.

In accordance with a further preferred embodiment of the present invention, there is also provided a method of loading an application onto an IC card, wherein the application has an associated file mode type and the IC card has an associated operating mode. The method includes the steps of determining whether the file mode type of the application is a predetermined file mode type, and changing the operating mode of the IC card if the file mode type corresponds to the predetermined file mode type. The predetermined file mode type is, for example, a "shell" file mode type, and the operating mode of the IC card is, for example, either "OS" or "shell." Thus, when an application has an associated file mode type of "shell," the operating mode of the IC card is changed from "OS" to "shell."

Preferably, a shell application is not loaded unless it is the first application loaded. In this way, operability of the non-shell applications loaded onto the IC card may be guaranteed. Thus, the method of loading an application according to a further embodiment of the present invention preferably further includes the steps of: determining whether any other applications have already been loaded onto the IC card; loading the application onto the IC card if the file mode type of the application corresponds to the predetermined file mode type and no other applications have already been loaded onto the IC card; and changing the operating mode of the IC card if the file mode type corresponds to the predetermined file mode type and no other applications have already been loaded onto the IC card.

In accordance with another preferred embodiment of the present invention, there is also provided a method of routing a command by an operating system of an IC card, wherein the IC card has an associated operating mode. The method includes the steps of determining whether the operating mode of the IC card is a predetermined operating mode; and routing the command directly to an application if the operating mode of the IC card corresponds to the predetermined operating mode. For example, assuming a SELECT FILE command is received by an IC card from a terminal and the IC card has a shell application loaded thereon, if the operating mode of the IC card and the predetermined operating mode are both "shell," the operating system would route the SELECT FILE command to the shell application.

Preferably, the method of routing further includes the steps of: if the operating mode of the IC card does not correspond to the predetermined operating mode, determining whether the command is a select file command supported by the operating system; and routing the command to an operating system routine responsible for the select file command if the command is a select file command supported by the operating system.

Preferably, the IC card further comprises a currently selected file having an associated file type and the method of routing further comprises the steps of: if the operating mode of the IC card does not correspond to the predetermined operating mode, determining whether the file type of the currently selected file is supported by the operating system; and routing the command to an operating system routine responsible for the file type if the file type of the currently selected file is supported by the operating system. If the file type of the currently selected file is not supported by operating system, the method further comprises the step of routing the command to an application.

In accordance with another preferred embodiment of the present invention, there is also provided a method of delegating control between applications by an operating system of an IC card, wherein the IC card is for use with a defined IC card format and has an associated operating mode. The method includes the steps of storing a shell application in the IC card for communicating with the operating system and for processing information in a format compliant with the defined IC card format; receiving a request by the operating system from a first application for delegating control to a second application; determining whether the operating mode of the IC card is a predetermined operating mode; determining whether the second application corresponds to the shell application; and failing the request for delegating control if the operating mode of the IC card corresponds to the predetermined operating mode and the second application corresponds to the shell application.

In accordance with another preferred embodiment of the present invention, there is also provided a method of initiating communication between an IC card and a terminal, wherein the IC card comprises a microprocessor and a memory, the memory having stored therein an operating system, a shell application, and data representative of an operating mode of the IC card, the operating mode representing whether selected information is to be routed to the operating system or the shell application. The method of initiating includes the steps of receiving a reset signal by the IC card from the terminal; and returning an answer-to-reset from the IC card to the terminal based on the operating mode of the IC card.

Preferably, a plurality of answer-to-reset files are stored in the memory of the IC card, and the step of returning an answer-to-reset comprises selecting one of the answer-to-reset files based on the operating mode. The selected information may be a command, such as a file access command. In addition, the selected information may be associated with a file structure format.

BRIEF DESCRIPTION OF THE DRAWINGS

Preferred embodiments in accordance with the invention will now be described by way of example only, with reference to the accompanying drawings, in which:

Fig. 1 is a schematic representation of an IC card in accordance with a preferred embodiment of the present invention;

Fig. 2 is a perspective view of an IC card and terminal in accordance with a preferred embodiment of the present invention;

Fig. 3 is a functional block diagram of an IC card in accordance with a preferred embodiment of the present invention;

Fig. 4 is an exemplary hierarchical file structure according to the EMV Specification;

Figs. 5A and 5B are flowcharts illustrating the steps for a load_file command used in accordance with a preferred embodiment of the present invention;

Fig. 6 is a flowchart illustrating the steps for a delete_file command used in accordance with a preferred embodiment of the present invention;

Fig. 7 is a flowchart illustrating the steps for a route command used in accordance with a preferred embodiment of the present invention;

Fig. 8 is a flowchart illustrating the steps for a delegate_request command used in accordance with a preferred embodiment of the present invention; and

Fig. 9 is a flowchart illustrating the steps for a determine_ATR_status command used in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 1 provides a schematic representation of a typical IC card 10 that can be used with the presently claimed invention. The IC card 10 includes an integrated circuit 12 and one or more electrical contacts 14, connected to the integrated circuit 12, for communication between the integrated circuit 12 and devices outside the IC card 10.

Fig. 2 shows an example of a device with which the IC card 10 communicates. As used in this specification and the appended claims, the term "terminal" shall be used to generically describe devices with which an IC card may communicate. A typical terminal 20, as shown in Fig. 2, includes a card reader 22, a keypad 24, and a display 26. The keypad 24 and the display 26 allow a user of the IC card 10 to interact with the terminal. The keypad 24 allows the user to select a transaction, to enter a personal identification number ("PIN"), and to enter transactional information. The display 26 allows the user to receive informational messages and prompts for data entry. Other types of terminals may include IC card compatible ATM machines and telephones.

Fig. 3 provides a functional block diagram of the integrated circuit 12. At a minimum, the integrated circuit 12 includes a processing unit 100 and a memory unit 110. Preferably, the integrated circuit 12 also includes control logic 150, a timer 160, security circuitry 170, input/output ports 180, and a co-processor 190. The control logic 150 provides, in conjunction with the processing unit 100, the control necessary to handle communications between the memory unit 110 and input/output ports 180. The timer 160 provides a timing reference signal for the processing unit 100 and the control logic 150. The security circuitry 170 preferably provides fusible links that connect the input/output ports 180 to internal circuitry for testing during manufacturing. The fusible links are burned after completion of testing to limit later access to sensitive circuit areas. The co-processor 190 provides the ability to perform complex computations in real time, such as those required by cryptographic algorithms.

The memory unit 110 may include different types of memory, such as volatile and non-volatile memory and read-only and programmable memory. For example, as shown in Fig. 3, the memory unit 110 may include read-only memory (ROM), electrically erasable programmable read-only memory (EEPROM), and random-access memory (RAM).

The memory unit 110 stores IC card data such as secret cryptographic keys and a user PIN. The secret cryptographic keys may be any type of well-known cryptographic keys, such as the private keys of public-key pairs. Preferably, the secret cryptographic keys are stored in a secure area of ROM or EEPROM that is either not accessible or has very limited accessibility from outside the IC card.

The memory unit 110 also stores the operating system of the IC card. The operating system loads and executes IC card applications and provides file management and other basic card services to the IC card applications. Preferably, the operating system is stored in ROM.

In addition to the basic services provided by the operating system, the memory unit 110 may also include one or more IC card applications. For example, if the IC card is to be used as an electronic cash card, an application called MONDEX™ PURSE might be included on the IC card, which loads an electronic value of a certain currency from a user's account in a financial institution onto the IC card. An application may include both program and data files, which may be stored in either ROM or EEPROM.

To enable the inter-operability of different terminals with different IC cards and applications, standards have been promulgated with respect to the organization of files stored on an IC card. For example, in the payment systems industry, the EMV '96 Integrated Circuit Card Specification for Payment Systems, Version 3.0, June 30, 1996, available from MasterCard International Incorporated® (hereinafter the "EMV Specification"), incorporated herein by reference in its entirety, sets forth a hierarchical tree structure for accessing files, which is generally compliant with the ISO/IEC 7816-4 and 7816-5 standards. An illustrative example of such a hierarchical tree structure is provided in Fig. 4.

In Fig. 4, there are shown four types of file categories: the Directory Definition File (DDF), the Directory File (DIR), the Application Definition File (ADF), and the Application Elementary File (AEF). According to the EMV Specification, each DDF contains one DIR. Each DIR may contain one or more ADF and/or DDF. Each ADF contains one or more AEF, which are files containing data related to a particular application.

According to the EMV Specification, files are referenced either by a unique name or by a short file identifier (SFI). A DDF or ADF is referenced by its unique name using a SELECT command. Once a particular DDF or ADF is selected, a corresponding DIR or AEF is referenced with an SFI using a READ RECORD command. In the case of a DIR, the SFI is in the range of 1 to 10. In the case of an AEF, the SFI is in the range 1 to 30. The EMV Specification sets forth at least one mandatory DDF with a unique name of "1PAY.SYS.DDF01."

The format for a SELECT command for selecting a DDF or ADF according to the EMV Specification is shown in Table 1. In response to a SELECT command for a DDF, an IC card returns the SFI of the DIR attached to the DDF. When an ADF is selected, an IC card returns information that the terminal may use, in conjunction with other commands, to retrieve the SFI of AEFs related to the ADF.

Once the SFI of a DIR or AEF is known, a terminal may use the READ RECORD command to read the records of the DIR or AEF. The format of the READ RECORD command according to the EMV Specification is shown in Table 2.

TABLE 1: SELECT Command Format

Byte Number	Value
1	Hexadecimal “00”
2	Hexadecimal “A4”
3	Hexadecimal “04”
4	Hexadecimal “00”
5	Length of File Name (Hexadecimal “05” - “10”)
6-21	File Name (number of bytes variable depending on length of file name)
Last	Hexadecimal “00”

TABLE 2: READ RECORD Command Format

Byte Number	Value
1	Hexadecimal “00”
2	Hexadecimal “B2”
3	Record Number
4	SFI
5	Hexadecimal “00”

Although the EMV Specification sets a standard for file organization within the payment systems industry, other IC card file organization standards may exist in other industries. Some may be proprietary and may not be generally compatible with the EMV Specification or ISO/IEC 7816-4 or 7816-5.

Typically, an IC manufacturer who desires to produce IC cards compatible with the EMV Specification and other proprietary specifications must produce IC cards with different operating systems to implement the different file structures and different file referencing and access methods defined by the various specifications. According to embodiments of the presently claimed invention, however, a manufacturer may produce an IC card with a single operating system and execute different shell applications to implement the different standards.

Figs. 5A to 9 are flowcharts illustrating a preferred embodiment of IC card operating system routines capable of supporting a shell application. In the embodiment of Figs. 5A to 9, the operating system is a multiple application operating system that runs on IC cards, such as the MULTOS™ operating system from Mondex International Limited. Such an operating system includes routines for loading and deleting applications, routines for routing commands to appropriate operating system processes or applications, routines for handling delegation of processing between applications, and routines for handling the answer-to-reset (ATR) message.

In the embodiment of Figs. 5A to 9, only one shell application can be loaded onto an IC card at any one time. Once the shell application is loaded, it is valid for all applications loaded on the IC card. Preferably, the operating system has a delegation feature, such as the delegation feature described in the United States patent application entitled “Multi-Application IC Card with Delegation

Feature," by Everett et al., filed April 23, 1998, which is hereby incorporated by reference to Annex A attached hereto. When the shell application receives a command from the operating system, it interprets the command and/or delegates control to the application associated with the command. If control is delegated to an application, when the application is finished, it returns control to the shell application. The shell application then returns any response to the operating system in the proper format for transmission to the terminal.

Although for the sake of simplicity the preferred embodiment loads only a single shell application at a time, the present invention is not limited to such an embodiment. It is within the scope of embodiments of the present invention for multiple shell applications to be loaded onto an IC card and to be used with different sets of applications.

As a matter of notation, the data elements referred to in the flowcharts of Figs. 5A to 9 follow a dot notation convention where the data element following the dot (".") is a component of the data element preceding the dot. For example, the data element *file_mode* includes two components: *file_mode_type* and *application_id*. In the dot notation used, the first component data element is referred to as *file_mode.file_mode_type* and the second component data element is referred to as *file_mode.application_id.*

Figs. 5A and 5B are flowcharts illustrating the implementation of a file loading routine by an operating system capable of supporting a shell application. In step 510, the routine receives the file loading command *load_file_command* from the security manager of the operating system, *OS_Security_Manager*. In step 520, after receiving the command, the routine checks whether the application identification number associated with the command, *load_file_command.application_id*, is present in the operating system control information, *os_control_info.application_id*. If the application identification number is already present, in step 521, the routine sets the response status *load_file_response.status* to "failed" and sets the error description *load_file_response.error_cause* to "duplicate application id." This error response indicates that the application is already loaded and cannot be loaded again. The error response *load_file_response* is then returned to the *OS_Security_Manager*.

If the application identification number of the application to be loaded is not present, in step 530, the routine checks the file mode type of *load_file_command*. The file mode type may be, for example, "shell" or "non-shell." A "shell" file mode type indicates that the application to be loaded is a shell application, while a "non-shell" file mode type indicates that the application to be loaded is not a shell application.

If the application to be loaded is a shell application, the routine further checks whether *os_control_info* is empty. If *os_control_info* is not empty, then one or more applications have already been loaded onto the IC card. If this is the case, in step 531, the routine sets the response status *load_file_response.status* to "failed" and sets the error description *load_file_response.error_cause* to "application already loaded." This error response is a result of the restriction that the shell application is to be valid for all applications loaded onto the IC card. To ensure that all applications will operate correctly with the shell application, the shell application must be the first application loaded onto the IC card.

Assuming that an error condition has not been triggered in steps 520 and 530, the directory file and *os_control_info* are updated with the appropriate application information in steps 540 and 550.

With reference to Fig. 5B, in step 560, the file mode type of *load_file_command* is checked once again. If the file mode type is "shell," then in step 570, the *file_mode* and the *selected_file* data elements are updated. The *file_mode* data element contains both the *file_mode_type* of the IC card and the *application_id* of the shell application. The *file_mode.file_mode_type* variable represents the operating mode of the IC card and, thus, may also be referred to as the "operating mode." The operating mode of the IC card may be, for example, either "OS" or "shell." "OS" mode indicates that a shell is not loaded, while "shell" mode indicates that a shell is loaded. The *selected_file* data element contains the *application_id* and the *file_type* of the currently selected file.

In step 570, *file_mode.file_mode_type* is set to "shell." The *file_mode.file_mode_type* represents the operating mode of the IC card and, thus, is also referred to as the "operating mode." In addition, the application identification number of the currently selected file is set to the application identification number of the shell application. The *file_type* of the selected file is set to "dedicated file," indicating that file commands are not to be handled by the operating system.

In step 580, the response status *load_file_response.status* is set to "success" and is returned to the *OS_Security_Manager*.

Fig. 6 is a flowchart illustrating the implementation of a file deleting routine by an operating system capable of supporting a shell application. In step 610, a *delete_file_command* is received from the *OS_Security_Manager*. In step 620, checking is performed to verify that the application being deleted exists in *os_control_info* —— i.e., that the application is loaded on the IC card. If the application identification number is not in *os_control_info*, then in step 670, the response status *delete_file_response.status* is set to "failed" and the error description *delete_file_response.error_cause* is set to "application not loaded."

If the application is loaded on the IC card, in step 630 checking is performed to determine whether the file mode type of the application being deleted, *delete_file_command.file_mode_type,* is equal to "shell." Checking is also performed to determine whether the application identification number of the application being deleted, *delete_file_command.application_id,* is equal to the application identification number assigned to the file mode of the IC card, *file_mode.application_id.* In short, checking is performed to determine whether a loaded shell application is being deleted.

If a loaded shell application is being deleted, in step 680, *file_mode.file_mode_type* is set to "OS" and *selected_file.file_type* is set to the default file type for the IC card, i.e., "master file."

In step 640, the directory file record corresponding to the application is deleted from the directory in which it is stored. In step 650, the application identification number of the application is deleted from *os_control_info*. In step 660, *delete_file_response.status* is set to "success" and the response status is returned to the *OS_Security_Manager*.

Fig. 7 is a flowchart illustrating the implementation of a command routing routine by an operating system capable of supporting a shell application. In step 710, the route routine receives a command from the cardholder ___ i.e., a command from outside of the IC card. In step 720, checking is performed to determine the operating mode of the IC card. If *file_mode.file_mode_type* is not equal to "OS," a shell application has been loaded onto the IC card. Thus, the command from the cardholder is sent directly to the currently selected application or applications. In the typical case, the currently selected application will be the shell application. It may be the case, however, that the shell application has delegated control to another application and that that application receives and processes the command directly.

If the operating mode of the IC card is equal to "OS," the various conditions defined in steps 730 to 750 are checked. In step 730, if the command is a *select_file* command, the command is sent to the *select_file* routine of the operating system. In step 740, if the file type of the currently selected file is "master file," the command is sent to the *provide_card_facilities* routine of the operating system, which handles commands associated with the master file type. Similarly, in step 750, if the file type is "directory file," the command is sent to the *read_card-level_data_files* routine of the operating system, which handles commands associated with the directory file type. If none of the conditions in steps 730 to 750 are satisfied, then the selected file must be an application. Therefore, the command is sent to the currently selected applications.

Fig. 8 is a flowchart illustrating a delegate request checking routine that is necessary if an operating system supports both a shell application and a delegate feature. In step 810, a *delegate_request* is received from an application. In step 820, checking is performed to determine whether the operating mode of the IC card is "shell" and whether the application identification number of the delegated application (the application to which control is being sought to be transferred) is the same as the application identification of the shell application of the IC card. If both conditions are true, then an application is attempting to delegate control to the shell application. Since the shell application is the first application loaded and selected, and thus delegates control to all other applications, such a delegation would be recursive. Recursive delegation is not allowed. In step 830, therefore, *delegate_response.status* is set to "failed" and *delegate_response.error_cause* is set to "recursive shell delegation." The delegate response is returned to the delegator applications. In step 820, if it is determined that the delegator application has submitted a proper, non-recursive delegate request, the request is processed in accordance with the operating system's delegate handling procedures.

When an IC card is inserted into a terminal, it receives a reset signal. To initiate communication with the terminal, the IC card must respond to the reset signal with an appropriate answer-to-reset (ATR) message. Fig. 9 is a flowchart illustrating an ATR routine for an IC card operating system that supports a shell application.

In step 910, the operating mode of the IC card is checked. If the *file_mode.file_mode_type* is equal to "OS," in step 920, the file type of *selected_file* is set to the default "master file" and *s_ATR_status* is set to "default ATR." Otherwise, if the operating mode of the IC card is "shell," in step 930, the file type and application identification number of the selected file are set to "dedicated file" and *file_mode.application_id*, respectively. *s_ATR_status* is set to "shell ATR." In both cases, *s_ATR_status* is returned to the *control_ATR* routine of the operating system. Using *s_ATR_status*, the *control_ATR* routine responds with the appropriate ATR to the reset signal from the terminal. The appropriate ATR may be stored in different files on the IC card, which are selected based on *s_ATR_status*.

Although the present invention has been described with reference to certain preferred embodiments, various modifications, alterations, and substitutions will be known or obvious to those skilled in the art without departing from the spirit and scope of the invention, as defined by the appended claims.

The scope of the present disclosure includes any novel feature or combination of features disclosed therein either explicitly or implicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or mitigates any or all of the problems addressed by the present invention.

The application hereby gives notice that new claims may be formulated to such features during the prosecution of this application or of any such further application derived therefrom. In particular, with reference to the appended claims, features from dependant claims may be combined with those of the independent claims in any appropriate manner and not merely in the specific combinations enumerated in the claims.

ANNEX A

MULTI-APPLICATION IC CARD WITH DELEGATION FEATURE

ANNEX A TO THE DESCRIPTION

BACKGROUND OF INVENTION

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), a random access memory (RAM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. The multiple application operating system present on the IC card allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application card and only executes that one application when inserted into a terminal. For example, a telephone card could only be used to charge a telephone call and could not be used as a credit/debit card. If a card user desires a variety of application functions to be performed by single application IC cards issued to him or her, such as both an electronic purse and a credit/debit function, the card user would be required to carry

ANNEX A TO THE DESCRIPTION

multiple physical cards on his or her person, which would be quite cumbersome and inconvenient. If an application developer or card user desired two different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal during the transaction, making the transaction difficult, lengthy and inconvenient.

Therefore, it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card.

The increased flexibility and power of storing multiple applications on a single card create new challenges to be overcome concerning the integrity and security of the information (including application code and associated data) exchanged between the individual card and the application provider as well as within the entire system when communicating information between applications.

For instance, the existence of multiple applications on the same card allows for the exchange of data between two applications, while one of the applications is being executed. As stated above, a frequent flyer loyalty program may need to be accessed during the execution of an electronic purse application. If data is passed between applications in an insecure manner, it may be possible for a third party monitoring the transaction to determine the contents of the transferred data or even other private data associated with one or both of the applications. Thus, it would be beneficial to provide an application architecture and memory organization which protects an application's data from being discovered by a third party when it is exchanged with other applications present on the IC card.

Accordingly, it is an object of the invention to provide an application architecture and memory organization which provides for a secure data interaction between applications and allows multiple applications to be accessed while performing a desired task or function.

SUMMARY OF THE INVENTION

The present invention provides for a multiple application architecture for an IC card called an application abstract machine (AAM) and a method for implementing that architecture. The processing of multiple applications is accomplished by generating for at least one application (the "first application") a data memory space including at least two segments, a volatile memory segment and a non-volatile memory segment, commencing the execution of the first application's instructions; delegating or switching execution from the first application to the delegated application and in so doing, saving any data generated by the first application in the logical data memory space associated with the first application; executing the second application's instructions; retrieving the saved data and completing with this data the execution of the first application's instructions.

ANNEX A TO THE DESCRIPTION

Additional delegation commands can be issued by the second application or other subsequent applications. The command delegated is interpreted by a delegated application in the same manner as a selection command being issued directly by a terminal and therefore each application performs the security functions at the same level as if a terminal is issuing the command.

The volatile memory segment can further be separated into public ("Public") and dynamic ("Dynamic") portions. Data can be exchanged between a plurality of applications and/or a terminal when stored in the Public region of the data memory. The Dynamic memory region can be used solely as temporary work space for the specific application being executed.

BRIEF DESCRIPTION OF THE DRAWINGS

Further objects, features and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

Fig. 1 is block diagram illustrating the data memory space segment and associated registers for an IC card application using the AAM organization;

Fig. 2 is a block diagram illustrating the code memory and the data memory spaces for an IC card application using the AAM architecture;

Fig. 3 is a flow diagram illustrating the steps of performing a request for a delegation function by one application to another;

Fig. 4 is a flow diagram illustrating the steps of performing a return delegation control function for a delegate application to a delegator application;

Fig. 5 is a flow diagram illustrating the steps of performing an inquire delegator ID request of a delegation function;

Fig. 6 is a block diagram of an IC card chip which can be used as a platform in accordance with the invention; and

Figures 7A, 7B and 7C illustrate multiple delegation calls made between three applications.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

DETAILED DESCRIPTION OF THE INVENTION

The present invention provides for a method and apparatus for processing multiple application programs with associated data stored on an IC card which can be accessed and executed. An application stored on the card can be selected by a terminal, or other interface device, or another application. Each application program which is stored on the IC card when executed is allocated a memory space organized by the program's software code (instructions which are executed by a processor located on the IC card) and the associated data which the application stores and uses during execution of the program.

For example, a multi-application card may store a purse application, or an electronic money application, and a specific loyalty application such as a frequent flyer awards application. Each application has software code and associated data to support the execution of that software code. Each application is allocated a memory space when executed. In this example, there is interaction between the two applications stored on the card. For each dollar electronically spent to make a purchase, the user may be entitled to one frequent flyer mile which is stored and processed by the frequent flyer program. The purse application need not be aware of the specific loyalty program stored on the card, but instead may contain an instruction to communicate with any loyalty program stored on the card. The loyalty program will require input data representative of the amount of a particular electronic value so that it can update its own stored data of current frequent flyer miles for the user of the card.

ANNEX A TO THE DESCRIPTION

When two applications need to communicate during the same transaction, a system architecture is required to process both applications in an efficient and secure manner. One approach could be a windows type model where both applications could be running at the same time. Presently, however, IC card platforms are not powerful enough to simultaneously operate multiple programs efficiently. Also, transferred data may be exposed to unwanted third party access. The solution to this problem, provided by the current invention, which is described in greater detail below, is to selectively interrupt the execution of applications in a secure manner. This allows the integrity of the applications' data to be maintained and allows the best utilization of the available memory space in the IC card.

An efficient architecture for processing multi applications in an IC card is termed an Application Abstract Machine (AAM) architecture and is described herein. The AAM Architecture applies to any platform independent of its hardware and enables developers to write applications to store on the IC cards which are portable across many different types of platforms (e.g., IC cards built by different manufacturers with different processor configurations) without the need for knowledge about the specific hardware of the platform.

An application abstract machine (AAM), a term for the memory allocation and organization for the data stored and used by each application, is created for each application stored on the IC card which is executed by the processor on the card. In order to ensure data integrity and security when data is transferred between applications which are executed on the IC card, only one application on the IC card is allowed to be executed at a time. Each application has a data memory space which is virtually allocated and mapped onto the physical memory addresses available in the IC card memories. Data is then passed between two or more applications within a specified memory location and in a manner consistent with transferring data to an external terminal or device with which the IC card is securely interacting. At a general level, each AAM space created for each application being executed includes two separate address spaces, one for the program code itself and one for the program data which is stored and/or used by the application. The program data address space is effectively divided into three segments: a Static segment, a Dynamic segment and a Public segment which are described in more detail in conjunction with Figure 1. As stated above, the Static, Dynamic and Public segments are logically mapped to the physical memory; they are virtual memory segments as opposed to physical memory segments. The AAM data address space is preferably addressed and processed using seven different address registers and two control registers.

Figure 1 shows an illustrative diagram of a logical data space allocation 101 created for an application used in conjunction with the present invention. The AAM data portion 101 includes a Static data space 103, a Public data space 105 and a Dynamic data space 107. Also shown are a series of address registers: the Static base address register 109, the Static top address register 111, the Public base address register 113, the Public top address register 115, the Dynamic base address register 117, the Dynamic top address register 121 and local base address register 119 which serves as a local stack frame pointer in the Dynamic data space when the application is being executed. The address registers can contain physical memory addresses but preferably contain offset addresses for the various data address spaces in order to be hardware independent. An example of the overall address space is 64K bytes, although the size varies with the applicable platform and the available memory size. The registers can also be considered pointers or can be any other conventional addressing mechanism.

Within the allocated AAM data space 101, the Static portion of the memory is non-volatile which is not erased after power is removed from the IC card (such as EEPROM), the Dynamic space is volatile (such as RAM) which may be erased after power is removed from the card and the Public space is also volatile (such as RAM). An IC card can receive power from a terminal after it is interfaced into the terminal. Although an IC card may contain a battery to maintain some power for memory and circuitry, volatile memory will typically be erased after the IC card is removed from its power source.

The defined AAM data space has bytes in each segment which are contiguous, so that applications can perform pointer and offset arithmetic. For example, if the segment addresses "1515" and "1516," or any other pair of sequential numbers, are both valid and are present within the same segment, then they address adjacent bytes. This allows offset values stored in registers to determine the location of a desired memory address. The segment address of the first byte of the Static segment is zero, so that the segment address of a given location within the Static region is equal to its offset.

Pointers to other specific regions of the Static data area can be stored in the Static data because the Static region is non-volatile. For example, if the card

ANNEX A TO THE DESCRIPTION

user's name is stored in the Static memory of a credit/debit application, the application will know the card user's name will always be stored in the 5th memory location above the starting point for the Static portion of memory. The location can be noted as SB[5] or the 5th byte above the Static Bottom. Since the Static memory is non-volatile, it will not be erased after each transaction and the application will always know of its location relative to the Static segments' address registers.

On the other hand, the segment address of any location in the Dynamic or Public segments is not always equal to a particular offset from the beginning of the respective segment because the contents of those segments change for each operation. The fourth location in the Dynamic segment will be different for each operation performed by the application. The address of a memory location of Dynamic or Public segment is fixed preferably only for the duration of one command-response pair operation. Because segment addresses in Dynamic or Public are not fixed, MULTOS Executable Language (MEL)™ instructions (or any other program instructions) cannot refer to data using only segment addresses. Instead, a tagged address preferably is used to identify data which is to be retrieved, manipulated, transferred and/or stored with the IC card system.

A tagged address is a nineteen bit value consisting of a three bit tag (address register number) and a sixteen bit offset. Each of the seven address registers for the AAM data space contain a segment physical address. For instance, the address registers SB 109 and ST 111 point to the boundaries of the Static, the address registers PB 113 and PT 115 point to the boundaries of the Public and the address registers DB 117 and DT 121 point to the boundaries of the Dynamic. For each segment, the top register points to the byte immediately after the last valid byte. For example, the last valid byte of the Static is ST[-1]. Register LB functions as a stack frame pointer. It points to a location in the Dynamic segment to indicate a specific byte of local data for the currently executing application.

Referring to Figure 1, the allocated Static segment 103 contains the application's non-volatile data. Static data includes data which is associated with each application for every transaction such as the card user's name, account number, PIN value and address. Static data also includes variable data which is stored for use in future transactions using the application. For example, in a purse transaction, the electronic value data would be read from the Static segment and later saved in the Static segment at the end of the transaction. Additionally, transaction information data or available credit limits in the case of a credit/debit application would be stored in Static data.

The Static data is addressed using register SB (Static Base) and the register ST (Static Top) as offset registers. These registers contain the offset value from a physical address in a memory on the IC card. The individual memory location is then further offset from these starting points such as SB[3] or ST[-5]. SB is defined as zero and ST is equal to the size of the application's Static data which is set when the application is loaded onto the IC card. The multiple application operating system ensures that no other application can read or write the data stored in the Static segment of a particular application. Using current technology, the Static segment is preferably mapped onto an EEPROM (Electrically Erasable Programmable Read-Only Memory) which is non-volatile.

ANNEX A TO THE DESCRIPTION

The Dynamic segment 107 contains the application's volatile or temporary data. Dynamic data includes data which is temporarily used during the execution of an application such as intermediate values used in calculations or working variables. For example, a purse application may temporarily store the value of a transaction in order to reduce the amount of the value in the purse. The temporary data is used much like conventional computer programs use RAM to perform their assigned operations. The Dynamic segment preferably is divided into two parts, the session data portion and the stack data portion. The size of the session data is a constant for each application and is determined when the application is loaded. The stack holds variable data which is unique to the particular transaction being executed. The stack data portion stores data in a last-in-first-out manner. The stack is initially empty, but expands and contracts during execution of the application.

The Dynamic data is addressed from the register DB 117 to register DT 121. Register LB 119 serves as a local stack frame pointer to particular memory locations in the Dynamic segment for delegate commands or function calls. Register LB 119 is used to address the topmost frame, that of the currently executing function's session data. Register DT 121 serves as an address offset for the stack pointer. A one byte data item at the top of the stack is addressed as DT[-1], the next byte below is addressed by DT[-2], and so on. A push operation increments the relative value of DT for each item on the stack and a pop operation decrements the relative value of DT for each item on the stack. For example, a data element located at DT[-5] will be located at DT[-6] after an additional data

ANNEX A TO THE DESCRIPTION

item is placed on the stack.

When an application is being executed, the Dynamic segment created for that application also contains the application's session data which is used in performing the assigned task(s) or operation(s). The multiple application operating system ensures that no other application can read or write the data stored in the Dynamic segment of a particular application. The session data is set to zero upon the start of the execution of the application. Stack data will be saved in the stack if the application delegates a task or operation to another application.

A delegation function occurs when one application selects another application to process a command instead of processing the command itself. An example of a delegation function occurs when a delegator application receives a command that it does not recognize or is not programmed to process. The selected application should not reject the command and provide an error response to the interface device (IFD), but instead should pass the command to the appropriate receiver, or delegated application. In order to perform a delegation, the delegator calls the Delegate primitive. The Delegate primitive is a subroutine recognized by the multiple application operating system which is executed when the operating system interprets the Delegate instruction. Primitives can be stored as part of the operating system itself, loaded as a separate routine when the operating system is installed. Primitives are preferably written in machine executable language so that they can be executed quickly although they could be written in a higher level language. When a Delegate command is executed, execution of the delegating application is suspended, and the delegated application is executed instead. The delegated application then generates its own data memory space according to the AAM architecture. The data stored in the Public memory space of the first application (stored in RAM) is sent to the Public memory space of the second application (which could be physically the same memory but is allocated separately for each application) so that data can be passed between the applications. The Dynamic memory space is also shared although data is saved in a stack for the delegator and the other portions initialized before the delegated application is executed because the Dynamic data is secret.

In most cases, the delegated application processes the command exactly as though the command has arrived directly from an interface device. When the delegated application has finished processing the command, and has written a response into the allocated Public memory segment, it exits as normal. The delegator then resumes execution at the instruction address following the executed instruction which called the Delegate primitive. The response generated by the delegated application is retrieved or accessed from the allocated Public memory space. The delegator application may simply exit in turn, thus sending the response to the IFD, or may carry out further processing before exiting.

Another example of a delegation operation occurs when two applications need to share data. If an application A always returns a data item N when processing a command B, then another application which also returns data item N in response to a command can delegate the function B to application A in order to reduce the need for duplicate codes stored on the IC card. For example, if a PIN needs to be checked before an application is executed, an application stored

ANNEX A TO THE DESCRIPTION

on the card can delegate the "retrieve PIN function" to a PIN application which returns a stored universal PIN for the card.

Preferably, a new session begins whenever the IFD, e.g. a terminal, successfully selects an application, even if the application has been previously selected during the transaction. For example, if a card user goes to a terminal and transfers twenty dollars of electronic cash using a purse application, charges thirty dollars using a credit/debit application and then transfers ten dollars using the purse application again, three separate sessions will have occurred even though only two applications were used during the entire transaction. Each time an application delegates a task or function to another application, the delegated application treats the delegate function as if the IFD devices had selected the application to perform the task or function. However, performing a delegation function as described below has a different effect on session data.

The following examples will help explain when the session data is initialized (i.e., erased) versus when it is saved to be used in further operations. If application A is selected by an IFD device, and receives commands X, Y and Z from the terminal, application A may delegate all three commands to application B. For example, delegations may occur in response to delegation commands in the program code. Both applications A and B will have their session and stack data in their respective Dynamic segments initialized (set to zero) when they receive command X, but the stack will not be initialized when they receive the subsequent commands Y and Z.

In a second example, application A is selected, and receives commands X, Y and Z from the terminal. Application A processes X itself, but delegates Y and Z to application B. Application A will have its session and stack data initialized when it receives X, but not when it receives the subsequent commands Y and Z. Application B will have its session and stack data initialized when it receives Y, but not Z.

One example of a use of session data is to support the use of a session Personal Identification Number (PIN). The application could reserve one byte of session data to support the PIN-receiving flag. On receiving the PIN check command, the selected delegated application could update the flag as follows: if the PIN command is received and the inputted PIN is equal to the stored pin, then it will set the session data DB[0] to 1. If not, the application will check if the PIN flag is already set by checking the value in DB[0]. In either of the above cases, the application will process the rest of the commands in the session because the PIN has been verified. If neither of the cases is true, then the application will not process the command because the PIN is not proper. The PIN checking function could be a delegated function from the selected application to a PIN checking application.

The Public segment 105 is used for command and response data being passed between an IFD and an application. During a delegate command, the Public segment contains the data passed between two applications, the delegator (the application initiating the delegation) and the delegated application (the application which performs the delegated function). An application may also use the Public segment as a further temporary working storage space if required. The

ANNEX A TO THE DESCRIPTION

Public data is addressed using offsets stored in register PB 113 as a starting address, to register PT 115 as an ending address. Register PB 113 and Register PT 115 are fixed for the duration of a command-response pair being initiated by the IFD or delegator. Public data can include data inputted into or supplied by a terminal such as a transaction amount, vendor identification data, terminal information, transmission format or other data required or used by an application resident on the IC card. Public data can also include data which is to be transmitted to an IFD device or other application such as an electronic dollar value, card user information transmission format or other data required or used by the terminal or other delegated application.

The multiple application operating system ensures that the data stored in the Public segment remains private to the application until the application exits or delegates. Preferably, the data in the Public segment is then made available to other entities as follows: (1) if the application delegates, the whole of the Public segment becomes available to the delegated application; (2) if the application exits, and is itself delegated by another, the whole of the Public segment becomes available to the delegator; or (3) if the application exits, and is not itself delegated, then a portion of the Public segment containing the I/O response parameters and data are made available to the IFD.

An application may write secret data into the Public memory segment during execution of the application, but the application must make sure it overwrites the secret portion of the Public segment before delegating or exiting. If the application abnormally ends (abends), then the operating system on the IC card

ANNEX A TO THE DESCRIPTION

preferably overwrites all of the data in the Public segment automatically so that no unwanted entities can have access to the secret data. If the MULTOS carrier device (MCD) is reset, the operating system overwrites data in the Public segment automatically, so that no secret data is revealed. A portion of the Public memory segment is also used as a communications buffer. The I/O protocol data and parameters are preferably stored at the top of the Public memory space. In another preferred embodiment, the top seventeen bytes are reserved for the communications protocol between the IFD device and the IC card application. However, additional or less bytes can also be used depending upon the particular application and operating system being utilized.

The spaces shown between the memory segments in Figure 1 will vary depending upon the specific application and commands being processed. There could be no memory space between the memory segments so that the memory segments are contiguous.

Figure 2 shows an extended illustration of the AAM implemented architecture. Data memory space 201 includes the three segments Static, Public and Dynamic as previously described. Code memory space 203 contains the program instructions for an application stored on the IC card. The application instructions are preferably stored in an executable form which can be interpreted by the resident operating system but can also be stored in machine executable form. Instruction 205 is stored at one location in the code memory space 203. Additional instructions are stored in other locations of memory space 203. Two additional registers 207 and 209 are used in the AAM architecture. A code pointer (CP) register 207

ANNEX A TO THE DESCRIPTION

indicates the particular code instruction to be next executed. In the figure, the register indicates, e.g., through an offset or pointer means, that instruction 205 is the next to be executed. Condition Control Register 209 contains eight bits, four of which are for use by the individual application and four of which are set or cleared depending upon the results of the execution of an instruction. These condition codes can be used by conditional instructions such as Branch, Call or Jump. The condition codes can include a carry bit, an overflow bit, a negative bit and a zero bit.

All address and control registers are set to defined values prior to executing the selected or delegated application. The values are set either when the application is first loaded onto the card and the size of the code and non-volatile data can be ascertained or at the moment when the application passes control to the application. When the application is loaded, SB is set to zero and ST is equal to the number of bytes in the application's Static database. The other address registers are initialized when the application is given control. CP 207 is set to zero and all eight bits in CCR 209 are cleared at the start of executing the application.

A communications interface mechanism is present between the IFD and an application which includes the use of the Public data segment as a communications buffer for command-response parameters. A command-response parameter means an application is given a command to perform and returns a response to the entity issuing the command. Applications interact with an IFD by receiving commands, processing them and returning responses across the IFD-Application Interface. When an application has completed executing a command, the application will place the response into the Public segment starting at PB[0] which can be read by the IFD device and will set the proper interface parameters in the reserved Public space relative to PT[0].

While an application can be called directly from an IFD and return a response directly to an IFD, it can also delegate a request to another application where appropriate. The subsequently-called application will then process the request on behalf of the first application. The delegation can be directly in response to a received command in which the delegator acts as a controller for delegating commands or subcommands to other appropriate applications. Alternatively, the delegated command can be embedded in an application's code which delegates control of the processor when the first application needs to interact with another application during its execution, such as updating frequent flyer miles or verifying a PIN.

Figure 3 shows a flow chart of the steps which are performed when a delegate request is executed. Step 301 sets the parameter named delegator_application_id (delegator ID) to be equal to the selected_file.application_id (selected ID). The selected ID indicates the current application which is selected and which is currently being executed. The delegator ID indicates the application which delegates a function to another delegated application stored on the IC card. Step 303 then pushes (stores) the delegator ID onto the top of the delegate_id_stack (delegate stack). The data referenced in the Dynamic portion of allocated memory is saved so that the current application can complete its execution after the delegated function is complete. Data which is to be

ANNEX A TO THE DESCRIPTION

shared with the delegated application is referenced in the Public portion of allocated memory. The delegate stack is preferably stored outside of an application's AAM memory space and keeps track of which applications have delegated functions. Each application is suspended when it delegates a function so the delegate stack can act in a Last-In-First-Out (LIFO) manner so that if a number of applications are suspended due to delegation requests, the proper application is started in the right order. The delegate stack thus keeps track of which application was the last delegator when multiple layered delegation functions are performed. The delegate stack preferably operates in a LIFO manner although different stack schemes could be used as appropriate.

Step 305 then sets the selected ID to the delegate_request.delegate_ application_id (delegate ID) value. This step selects the application which will be called to perform the delegated function or functions. The identities of the delegated application can be specifically called by the delegator application or a particular function can be matched up with an application in a look up table. For example, a PIN match operation may be delegated to different applications depending upon which applications are present on the card. Step 307 then sets the application_command parameter to the value stored in the delegate_request.application_command parameter. This step specifies the command to be delegated to the delegate application. Applications typically have the ability to process many different commands. Alternatively, the entire application could be executed to perform one or more functions. The delegator application can choose which command it is delegating to another application. Step 309 then sends the application_command to the AAM operating system for execution by the delegatee application. The delegator application is then suspended (or interrupted). Any data that is required to pass between the applications is transferred via the Public memory space.

Figure 4 is a flow chart of the steps for performing a "return delegation control" command by the delegatee application. This command is executed by the operating system when a delegated application has completed its delegated function. Step 401 gets application_responses from the Public memory space of the delegated AAM. The response data is passed in the Public memory segment of the delegatee AAM. Step 403 then sets the delegate_response.status variable to a success condition. This means that a delegation operation has been successfully completed. Step 405 sets the delegate_ response.application_responses parameter to the application_responses values which were stored in the Public segment of the delegatee application.

Step 407 sets the delegate_response.delegate_application_id parameter to selected_file.application_id (the delegatee application ID). Step 409 pops the top (i.e., reads the last data stored in the stack) delegate_application_id from the delegate_id_stack. This information indicates the identity of the delegator application for the command which was just delegated and completed by the delegated application. Step 411 sets the select_file.application_id value to the delegator_application_id value. This selects the delegator application which was identified from the delegate ID stack as the current application which will resume running. The Dynamic data for the delegator application will be retrieved for the

ANNEX A TO THE DESCRIPTION

delegator application from its stored location so that the application will continue to execute where it left off with all data intact but will also have the response information from the delegated function. In step 413, the delegate_response data is sent to the current application for further processing. The response data is passed through the Public data space which could be the same physical RAM memory location because all applications share the physical volatile memory space.

Figure 5 shows a flow chart of the steps involved for inquiring about a delegator ID when a delegate command is received by a delegated application. The delegated application may need to know the identity of the delegator because it may perform operations differently for different delegator applications. For example, an airline loyalty program may need to know if awarded frequent flyers will be based on actual dollars processed or a lump sum award for some other activity such as performing a bill payment operation. This information could be passed to the delegated application as a variable or could be ascertained using an inquiry. The delegator inquiry operation could be implemented as a primitive as previously described.

Step 501 receives the delegator_id_enq_request from the AAM operating system. The request is used to identify the identity of the delegator. Step 503 checks if the delegate_id_stack is empty. If the stack is empty, then no delegation operations have occurred and no applications have been suspended. Thus step 511 sets the delegator_id_enq_response.status parameter to a failure indicator. Step 513 then sets the value of delegator_is_enq_request.error_cause to a value indicating "no delegator application." There is no delegator application. The process then continues with step 509.

If the delegate_id_stack is not empty, than one or more delegations have occurred. In that case, step 505 sets the delegator_id_enq_response.status parameter to a value indicating "success". Step 507 then sets the delegator_id_enq_response.delegator_ application_id parameter to the value stored in delegate_id_stack.delegator_ application_id. This sets the inquiry response to indicate the delegator application ID at the top of the stack. As explained above, the stored data at the top of the stack indicates the last delegator application to call a delegate function. Step 509 then sends the delegator_id_enq_ response back to the AAM operator system which delivers the information to the application or IFD entity requesting the information.

Figure 6 shows an example of a block diagram of an integrated circuit located on an IC card chip which can be used in conjunction with the invention. The integrated circuit chip is located on a chip on the card. The IC chip preferably includes a central processing unit 601, a RAM 603, a EEPROM 605, a ROM 607, a timer 609, control logic 611, I/O ports 613 and security circuitry 615, which are connected together by a conventional data bus 617 or other conventional means.

Control logic 611 in the smart card provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports 612. CPU 601 in conjunction with control logic 611 can perform many different functions including performing calculations, accessing memory locations, modifying memory contents, and managing input/output ports. Some IC cards also include a coprocessor for handling complex computations like cryptographic algorithms. Input/output ports 613 are used for communication between the card and an IFD which transfers information to and from the card. Timer 609 (which generates and/or provides a clock pulse) drives the control logic 611, CPU 601 and other components requiring a clock signal through the sequence of steps that accomplish functions including memory access, memory reading and/or writing, processing, and data communication. Security circuitry 615 (which is optional) preferably includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed upon completion of testing to prevent later access. The Static memory space is preferably mapped to memory locations in EEPROM 605 which is non-volatile. The Dynamic memory space is preferably mapped to RAM 603 which is volatile memory which has quick access. The Public memory space is also preferably mapped to RAM 603 which is volatile memory. The Dynamic data and Public data will be stored in different portions of RAM 603, while RAM is identified as a preferred non-volatile memory and EEPROM is identified as a preferred volatile memory. Other types of memory could also be used with the same characteristics.

Figures 7A, 7B and 7C illustrate an example of a delegation function being performed in order to process multiple applications on an IC card. Figure 7A shows a first application being executed as denoted with a double ringed circle 701. At some point during the execution of the first application, a delegation function 702 is called to delegate an operation to the second application which is indicated

ANNEX A TO THE DESCRIPTION

by circle 703. Also shown in Figure 7A is an empty delegator ID stack 705. Since the stack is empty, there is no data associated with it and it is shown only for illustrative purposes.

The multiple application operating system receives the delegate command and interrupts the execution of the first application 701 and gives control of the integrated circuit to application 703 as shown in Figure 7B. The execution of the second application 703 is illustrated with a double ringed circle. The term "gives control" means that the microprocessor and other circuitry on the card will process the instructions and allocate memory space for the application which is delegated. When the delegate command is processed, the delegator ID 707 is placed on top of the stack 705. The delegator ID stack is operated in a LIFO manner. Also shown in Figure 7B is a third application 709 resident on the card. At some point during the execution of the second application, a delegate function 711 is called to delegate the operation to the third application.

The multiple application operating system receives the delegate command 711 shown in Figure 7B interrupts the execution of the second application 703 and gives control of the integrated circuit to the third application 709 as shown in Figure 7C. When the delegate command is processed, the delegator ID 713 of the second application is pushed onto the delegator ID stack 705. The delegator ID 707 of the first application whose execution is still interrupted is pushed down in the stack consistent with a LIFO stack management. Thus when the third application has finished its execution, the delegator ID at the top of the stack is popped to indicate that execution of the second application should be resumed first. The delegator ID 707 from the first application will then be at the top of the stack so that when the second application is finished executing, the first application will resume its execution.

Additional applications can be managed by the delegator ID stack in a similar manner. By interrupting the execution of the applications when a delegate command is processed and keeping track of the order of delegations, the security and integrity of the data for each individual application can be maintained which is important because IC cards will store data for applications which is private to the card user such as account numbers, social security number, address and other personal information.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous apparatus, systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

ANNEX A TO THE DESCRIPTION

WE CLAIM:

1. An integrated circuit card comprising:

a microprocessor; a volatile memory coupled to said microprocessor; a non-volatile memory coupled to said microprocessor; and a plurality of applications stored in said non-volatile memory, wherein upon execution of each said application, said microprocessor allocates for each said executing application an associated data memory space comprising at least a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data; and further comprising means for delegating the performance of a function from a first executing application to a second executing application.

2. The integrated circuit card of claim 1, wherein said non-volatile memory segment is divided into at least two regions, including a public region and a dynamic region.

3. The integrated circuit card of claim 2, wherein said public region is used to share data between said first and second applications.

4. The integrated circuit card of claim 2, wherein said dynamic region is used to reference temporary data utilized during an application's execution.

5. The integrated circuit card of claim 1, further comprising at least one register coupled to said microprocessor which is used to determine the starting locations of each of said segments.

6. The integrated circuit card of claim 5, further comprising at least one register coupled to said microprocessor which is used to determine the top locations of each of said segments.

7. The integrated circuit card of claim 6, further comprising at least one register coupled to said microprocessor which is used as a local dynamic pointer.

8. The integrated circuit card system of claim 1, wherein each said application comprise a plurality of program instructions and wherein at least one of said program instructions when executed causes said memory referenced by said volatile memory segment to be accessed.

9. The integrated circuit card of claim 1, wherein said volatile memory segment references RAM and said non-volatile memory segment references EEPROM.

10. A method for processing a plurality of applications stored in a memory of an integrated circuit:

selecting a first application for execution;

allocating a data space for said first application including at least two memory segments comprising a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data;

executing said first application, interrupting execution of said first application and saving data referenced by said volatile memory segment;

executing a second application;

utilizing said saved data from said volatile memory segment for execution of said first application; and

completing said execution of said first application.

11. The method of claim 10, wherein said first application's identity is stored in a data stack during said delegation step.

12. The method of claim 11, wherein said data stack is accessed following said completion of said second application.

13. The method of claim 12, further including the step of inquiring said first application's identity by accessing said delegator stack.

14. The method of claim 10, wherein said non-volatile memory segment is divided into at least two regions, including a public region and a dynamic region.

15. The method of claim 14, wherein said public region is used to share data between said first application and said second application.

16. The method of claim 14, wherein data referenced by said dynamic region is utilized during the execution of said first application.

17. The method of claim 10, further including the step of allocating a second data space including at least two memory segments for said second application.

18. The method of claim 17, wherein said second data space's segments comprise a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data.

19. The method of claim 18, wherein said second application's non-volatile segment is divided into at least two regions, including a public region and a dynamic region.

20. The method of claim 19, wherein said second application's public region is used to share data between said first and second applications.

ANNEX A TO THE DESCRIPTION

21. The method of claim 19, wherein said data referenced by second application's dynamic region is utilized during said execution of said second application.

22. The method of claim 10, further including the step of delegating use of said microprocessor from said second application to a third application stored on said IC card.

23. The method of claim 22, wherein a third data space for said third application is allocated which includes a volatile memory segment for referencing temporary data and non-volatile memory segment for referencing static data, wherein said third application's volatile segment includes a public and dynamic portion.

24 An apparatus for processing a plurality of applications stored in a memory of a single integrated circuit card comprising:

means for allocating a data space comprising at least a non-volatile memory segment for referencing static data and a volatile memory segment for referencing temporary data; means for executing a first application; means for interrupting execution of said first application, means for saving data from at least a portion of said volatile memory segment; and means for executing a second application; means for retrieving said saved data; and means for completing said execution of said first application.

ANNEX A TO THE DESCRIPTION

25. The apparatus of claim 24, further including means for storing said first application's identity on a data stack.

26. The apparatus of claim 25, further including means for inquiring of said first application's identity.

27. The apparatus of claim 24, wherein said first application's non-volatile memory segment is divided into at least two regions, including a public region and a dynamic region.

28. The apparatus of claim 27, wherein said public region references random access memory.

29. The apparatus of claim 27, wherein said dynamic region references random access memory.

30. The apparatus of claim 24, further including means for allocating a second data space including at least two segments for said second application.

31. The apparatus of claim 30, wherein said second data space includes a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data.

32. The apparatus of claim 31, wherein said second data space's non-volatile segment is divided into at least two regions, including a public region and a dynamic region.

33. The apparatus of claim 32, wherein said public region references random access memory.

34. The apparatus of claim 32, wherein said dynamic region references random access memory.

35. The apparatus of claim 24, further including means for delegating operation of said IC card from said second application to a third application stored on said IC card.

36. The apparatus of claim 35, wherein a third data space for said third application is allocated which includes a volatile memory segment for referencing temporary data and non-volatile memory segment for referencing temporary data, wherein said third application's volatile memory segment includes a public and dynamic portion.

37. A system for processing a plurality of applications stored on an IC card comprising:

a non-volatile memory coupled to a databus;

a volatile memory coupled to said databus;

a first and second application program stored in said non-volatile memory, wherein each application has an associated identifier;

a data stack accessible by said databus for storing said applications' identifier if said application is interrupted during its execution;

processor means for executing instructions from said application programs wherein said processor means allocates a data memory space for said application which is being executed and said data memory space is mapped to at least one address in said non-volatile memory and at least one address in said volatile memory; and

wherein said processor means interrupts said first application at least once during its execution to execute said second application.

38. The system of claim 37, wherein data memory space comprises at least a volatile memory segment for referencing temporary data stored in said volatile memory and a non-volatile memory segment for referencing static data stored in said non-volatile memory.

39. The system of claim 37, further including means for storing said first application's identity on a data stack.

40. The system of claim 39, further including means for inquiring of said first application's identity.

41. The system of claim 38, wherein said first application's non-volatile memory segment is divided into at least two regions, including a public region and a dynamic region.

42. The system of claim 41, wherein said public region references random access memory.

43. The system of claim 41, wherein said dynamic region references random access memory.

44. The system of claim 37, further including means for allocating a second data space including at least two segments for said second application.

45. The system of claim 44, wherein said second data space comprises at least a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data.

46. The system of claim 45, wherein said second data space's non-volatile segment is divided into at least two regions, including a public region and a dynamic region.

47. The system of claim 46, wherein said public region references random access memory.

ANNEX A TO THE DESCRIPTION

48. The system of claim 46, wherein said dynamic region references random access memory.

49. The system of claim 37, further including means for delegating use of said processor means from said second application to a third application stored on said IC card.

50. The system of claim 49, wherein a third data space for said third application is allocated which includes a volatile memory segment for referencing temporary data and non-volatile memory segment for referencing temporary data, wherein said third application's volatile memory segment includes a public and dynamic portion.

51. An integrated circuit card comprising:

a plurality of applications and a microprocessor for controlling execution of said applications wherein execution of at least one first application is interrupted and execution is transferred to another second application, further comprising means for sharing data by said first and second applications and means for resuming execution of said first application at the appropriate location at least after completion of execution of said second application.

52. The integrated circuit card of claim 51, further comprising means for allocating a data memory space comprises at least a volatile memory segment for referencing temporary data stored in said volatile memory and a non-volatile memory segment for referencing static data stored in said non-volatile memory.

53. The integrated circuit card of claim 51, further including means for storing said first application's identity on a data stack.

54. The integrated circuit card of claim 53 further including means for inquiring of said first application's identity.

55. The integrated circuit card of claim 52, wherein said first application's non-volatile memory segment is divided into at least two regions, including a public region and a dynamic region.

56. The integrated circuit card of claim 55, wherein said public region references random access memory.

57. The integrated circuit card of claim 55, wherein said dynamic region references random access memory.

58. The integrated circuit card of claim 52, further including means for allocating a second data space including at least two segments for said second application.

59. The integrated circuit card of claim 58, wherein said second data space comprises at least a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data.

60. The integrated circuit card of claim 58, wherein said second data space's non-volatile segment is divided into at least two regions, including a public region and a dynamic region.

61. The integrated circuit card of claim 58, wherein said public region references random access memory.

62. The integrated circuit card of claim 60, wherein said dynamic region references random access memory.

63. The integrated circuit card of claim 51, further including means for delegating use of said processor means from said second application to a third application stored on said IC card.

ANNEX A TO THE DESCRIPTION

ABSTRACT OF THE DISCLOSURE

A multi-application IC card which processes two or more applications using an Application Abstract Machine architecture. The AAM architecture only allows one application to be executed at a time and allows for shared processing by performing a delegation function to a second application. A data space for each application is allocated when the application is selected to be executed. The data space includes a volatile and non-volatile region. The delegation function temporarily interrupts the execution of the first application, saves the temporary data of the first application, shares any data needed with the second application and the second application is executed until the delegated task is competed. The first application then retrieves the saved data and completes its execution. A delegator stack is used to keep track of the delegator's identity when multiple delegations occur. The AAM model allows for a high level of security while transferring data between applications.

CLAIMS

I CLAIM:

1. An integrated circuit card having an associated operating mode, comprising:

a microprocessor;

a memory coupled to said microprocessor;

data stored in said memory representative of said operating mode;

an operating system stored in said memory for processing selected information in a first IC card format;

a shell application stored in said memory for processing said selected information in a second IC card format; and

means responsive to said operating mode for routing said selected information to either said operating system or said shell application.

2. The integrated circuit card of claim 1, wherein said second IC card format is different than said first IC card format.

3. The integrated circuit card of claim 1 or claim 2, wherein said selected information is a command.

4. The integrated circuit card of claim 3, wherein said command is a file access command.

5. The method of any preceding claim, wherein said selected information is associated with a file structure format.

6. The integrated circuit card of any preceding claim, further comprising:

a non-shell application stored in said memory;

means for receiving a request by said operating system from said non-shell application for delegating control to a delegated application;

means for determining whether said operating mode of said IC card is a predetermined operating mode;

means for determining whether said delegated application corresponds to said shell application; and

means for failing the request for delegating control if the operating mode of said IC card corresponds to said predetermined operating mode and said delegated application corresponds to said shell application.

7. A method of loading an application onto an IC card, wherein said application has an associated file mode type and said IC card has an associated operating mode, comprising the steps of:

determining whether the file mode type of said application is a predetermined file mode type; and

changing the operating mode of said IC card if said file mode type corresponds to said predetermined file mode type.

8. The method of claim 7, further comprising the step of determining whether any other applications have already been loaded onto the IC card before the step of changing the operating mode.

9. The method of claim 7 or claim 8, further comprising loading said application onto the IC card if the file mode type of said application corresponds to the predetermined file mode type and no other applications have already been loaded onto the IC card.

10. The method of claim 8, wherein the changing step comprises changing the operating mode of said IC card if said file mode type corresponds to said predetermined file mode type and no other applications have already been loaded onto the IC card.

11. A method of routing a command by an operating system of an IC card, wherein said IC card has an associated operating mode, comprising the steps of:

determining whether the operating mode of said IC card is a predetermined operating mode; and

routing the command directly to an application if the operating mode of said IC card corresponds to the predetermined operating mode.

12. The method of claim 11, further comprising the steps of:

if the operating mode of said IC card does not correspond to the predetermined operating mode, determining whether said command is a select file command supported by said operating system; and

routing said command to an operating system routine responsible for said select file command if said command is a select file command supported by said operating system.

13. The method of claim 11 or claim 12, wherein the IC card further comprises a currently selected file having an associated file type, the method further comprising the steps of:

if the operating mode of said IC card does not correspond to the predetermined operating mode, determining whether the file type of said currently selected file is supported by said operating system; and

routing said command to an operating system routine responsible for said file type if the file type of said currently selected file is supported by said operating system.

14. The method of claim 13, if the file type of said currently selected file is not supported by said operating system, further comprising the step of routing said command to an application.

15. A method of delegating control between applications by an operating system of an IC card, wherein said IC card is for use with a defined IC card format and has an associated operating mode, comprising the steps of:

storing a shell application in said IC card for communicating with said operating system and for processing information in a format compliant with said defined IC card format;

receiving a request by said operating system from a first application for delegating control to a second application;

determining whether the operating mode of said IC card is a predetermined operating mode;

determining whether said second application corresponds to said shell application; and

failing the request for delegating control if the operating mode of said IC card corresponds to said predetermined operating mode and said second application corresponds to said shell application.

16. A method of initiating communication between an IC card and a terminal, wherein said IC card comprises a microprocessor and a memory, said memory having stored therein an operating system, a shell application, and data

representative of an operating mode of said IC card, said operating mode representing whether selected information is to be routed to said operating system or said shell application, said method comprising the steps of:

receiving a reset signal by said IC card from said terminal; and

returning an answer-to-reset from said IC card to said terminal based on said operating mode of said IC card.

17. The method of claim 16, wherein a plurality of answer-to-reset files are stored in said memory of said IC card, and said step of returning an answer-to-reset comprises selecting one of said answer-to-reset files based on said operating mode.

18. The method of claim 16 or claim 17, wherein said selected information is a command.

19. The method of claim 18, wherein said command is a file access command.

20. The method of claim 16, wherein said selected information is associated with a file structure format.

1/14

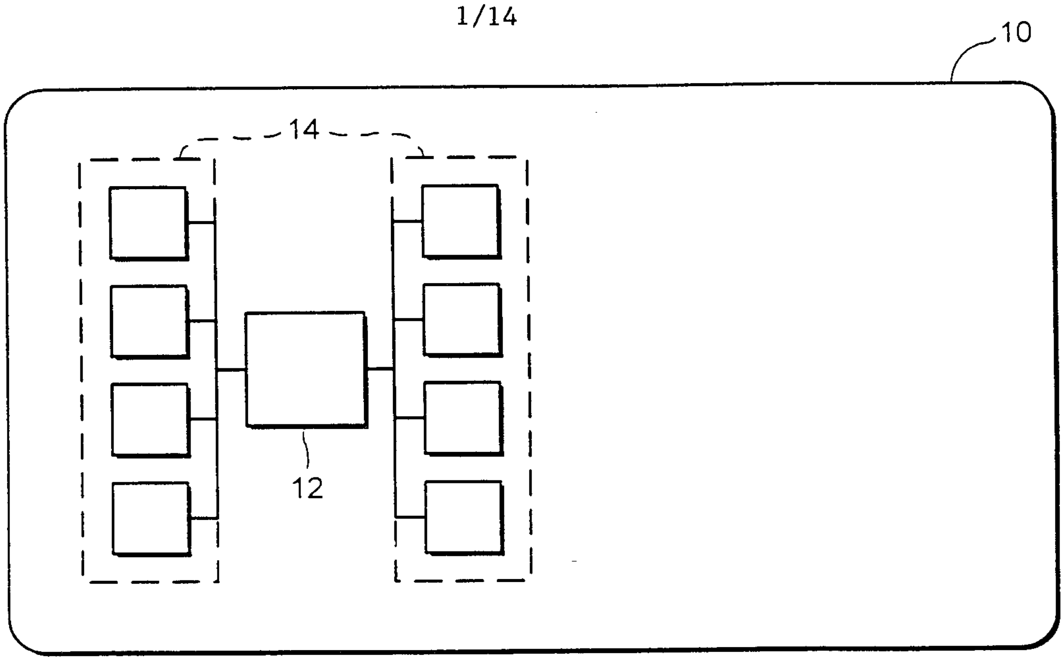


FIG. 1

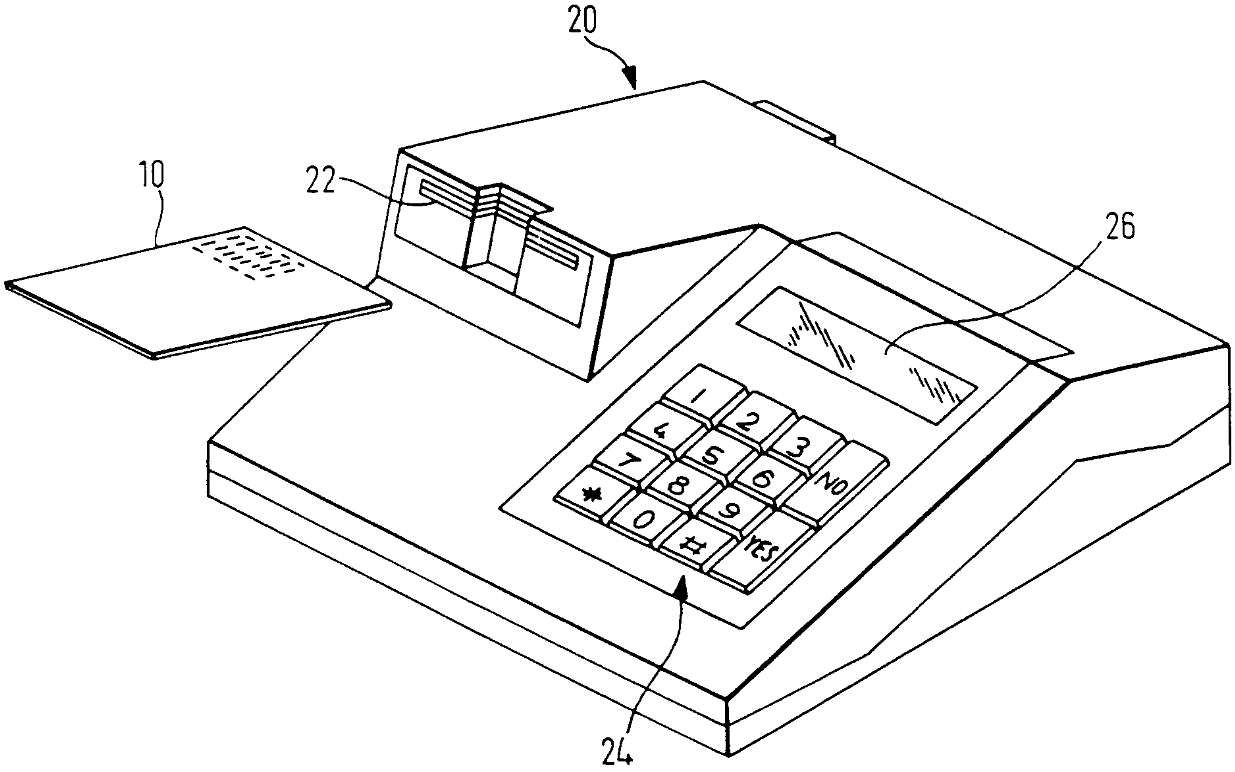


FIG. 2

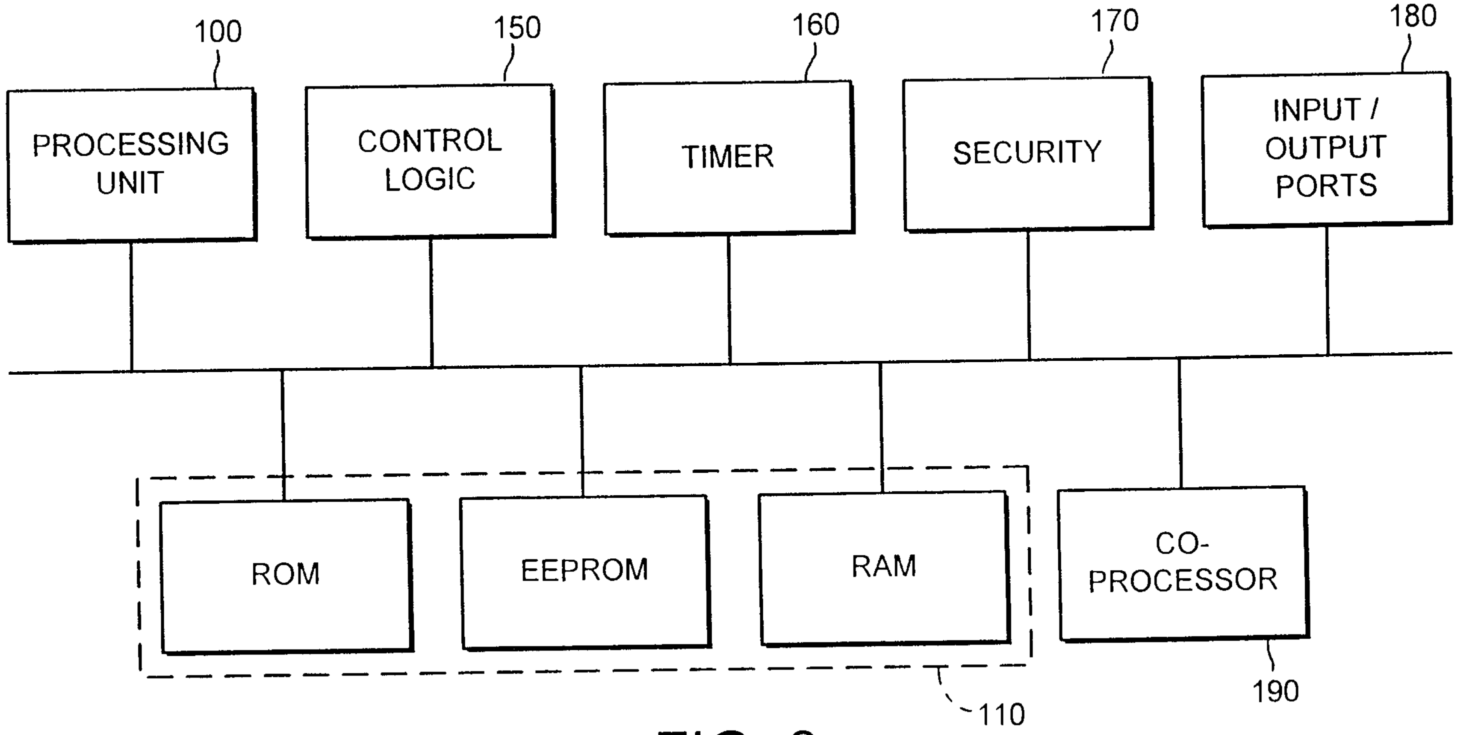

100
PROCESSING UNIT
150
CONTROL LOGIC
160
TIMER
170
SECURITY
180
INPUT / OUTPUT PORTS
ROM
EEPROM
RAM
110
CO-PROCESSOR
190

FIG. 3

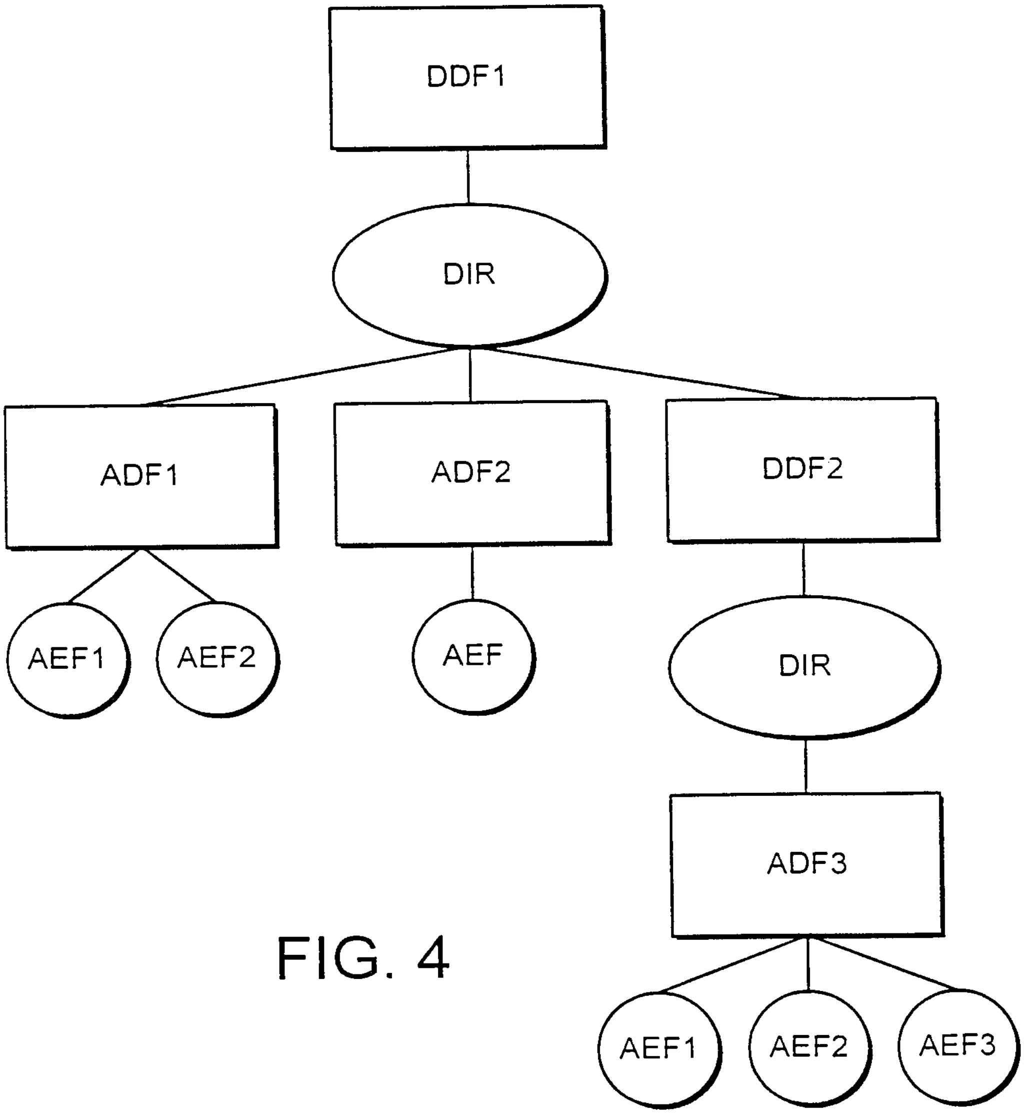


FIG. 4

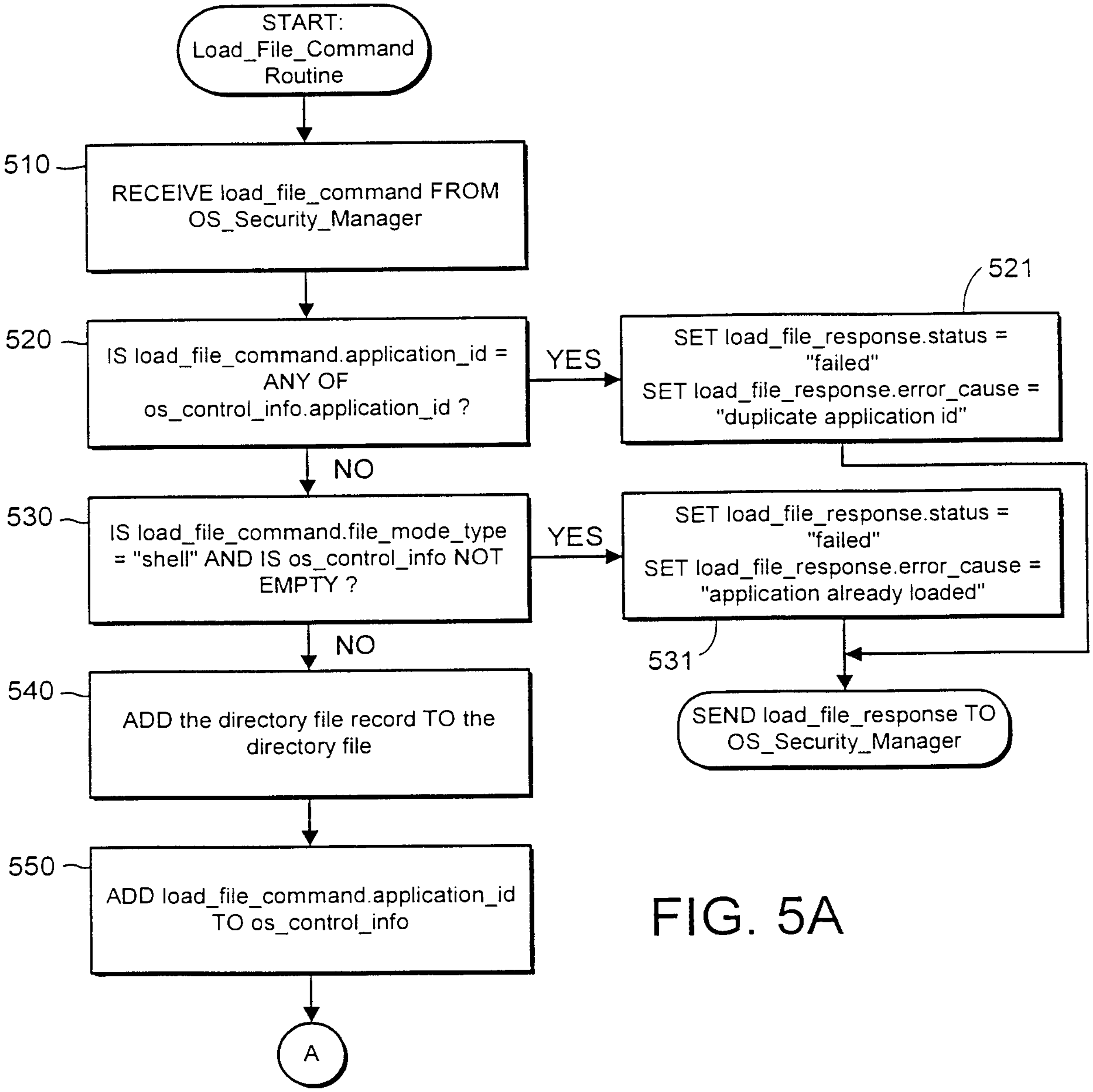

START:
Load_File_Command
Routine
510
RECEIVE load_file_command FROM OS_Security_Manager
520
IS load_file_command.application_id = ANY OF os_control_info.application_id ?
YES
521
SET load_file_response.status = "failed"
SET load_file_response.error_cause = "duplicate application id"
NO
530
IS load_file_command.file_mode_type = "shell" AND IS os_control_info NOT EMPTY ?
YES
SET load_file_response.status = "failed"
SET load_file_response.error_cause = "application already loaded"
531
NO
540
ADD the directory file record TO the directory file
SEND load_file_response TO OS_Security_Manager
550
ADD load_file_command.application_id TO os_control_info
A

FIG. 5A

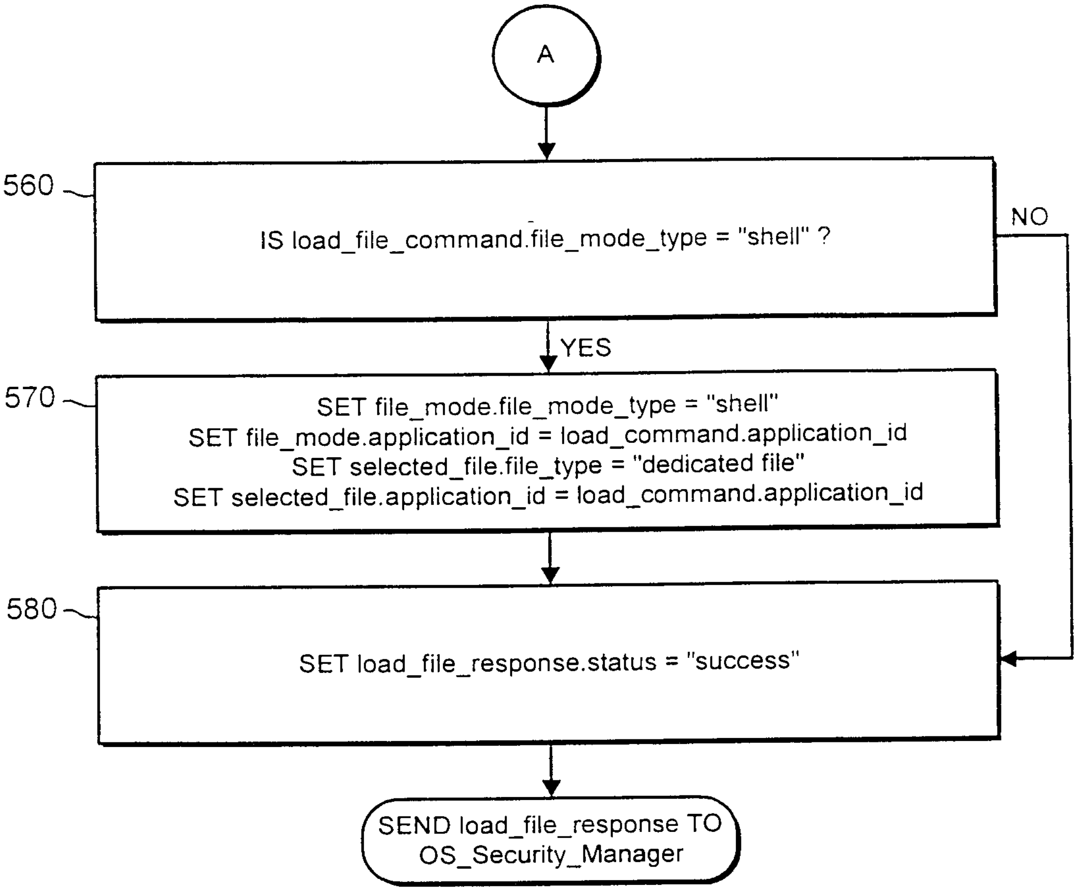

A
560
IS load_file_command.file_mode_type = "shell" ?
NO
YES
570
SET file_mode.file_mode_type = "shell"
SET file_mode.application_id = load_command.application_id
SET selected_file.file_type = "dedicated file"
SET selected_file.application_id = load_command.application_id
580
SET load_file_response.status = "success"
SEND load_file_response TO OS_Security_Manager

FIG. 5B

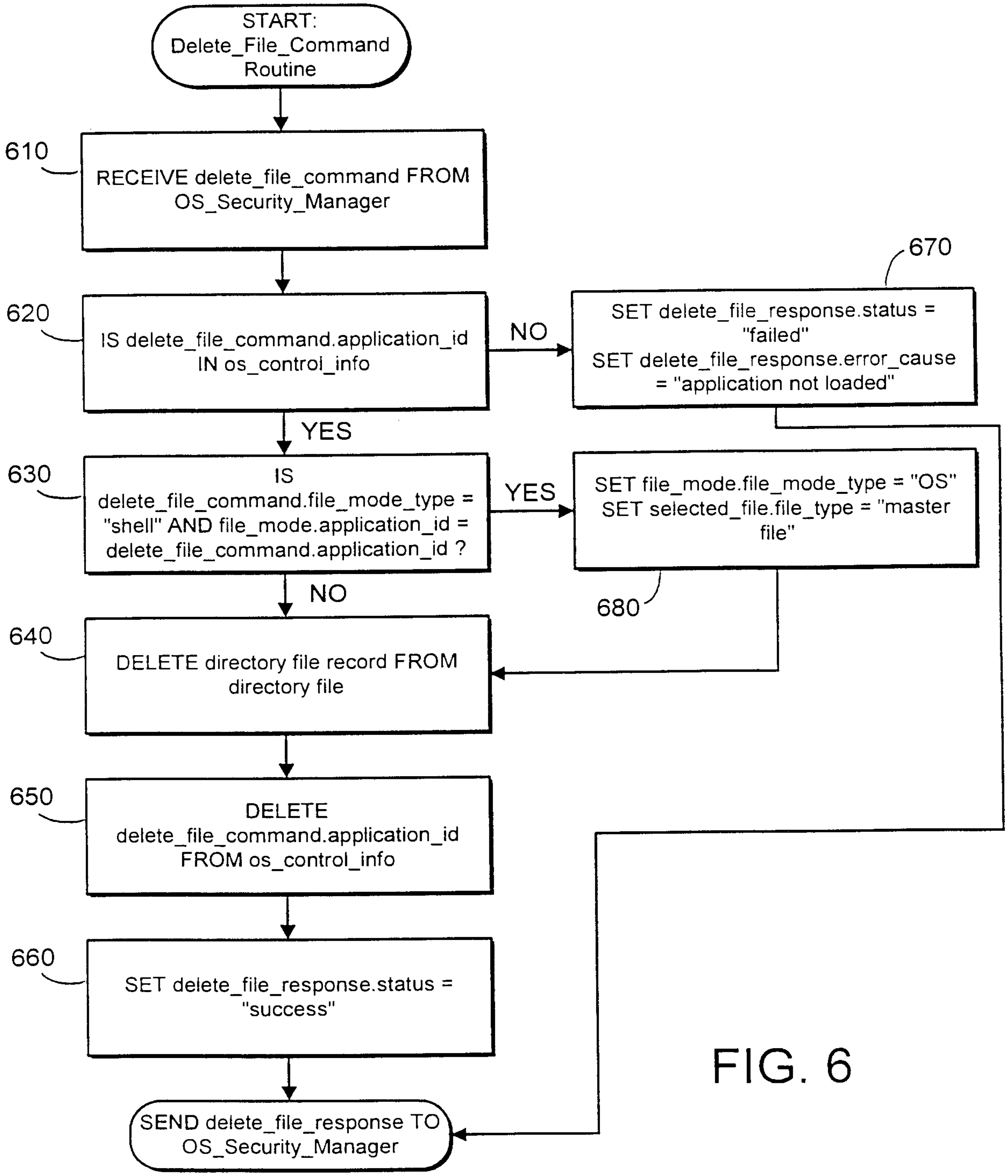

START:
Delete_File_Command
Routine
610
RECEIVE delete_file_command FROM OS_Security_Manager
620
IS delete_file_command.application_id IN os_control_info
NO
670
SET delete_file_response.status = "failed"
SET delete_file_response.error_cause = "application not loaded"
YES
630
IS delete_file_command.file_mode_type = "shell" AND file_mode.application_id = delete_file_command.application_id ?
YES
SET file_mode.file_mode_type = "OS"
SET selected_file.file_type = "master file"
680
NO
640
DELETE directory file record FROM directory file
650
DELETE delete_file_command.application_id FROM os_control_info
660
SET delete_file_response.status = "success"
SEND delete_file_response TO OS_Security_Manager

FIG. 6

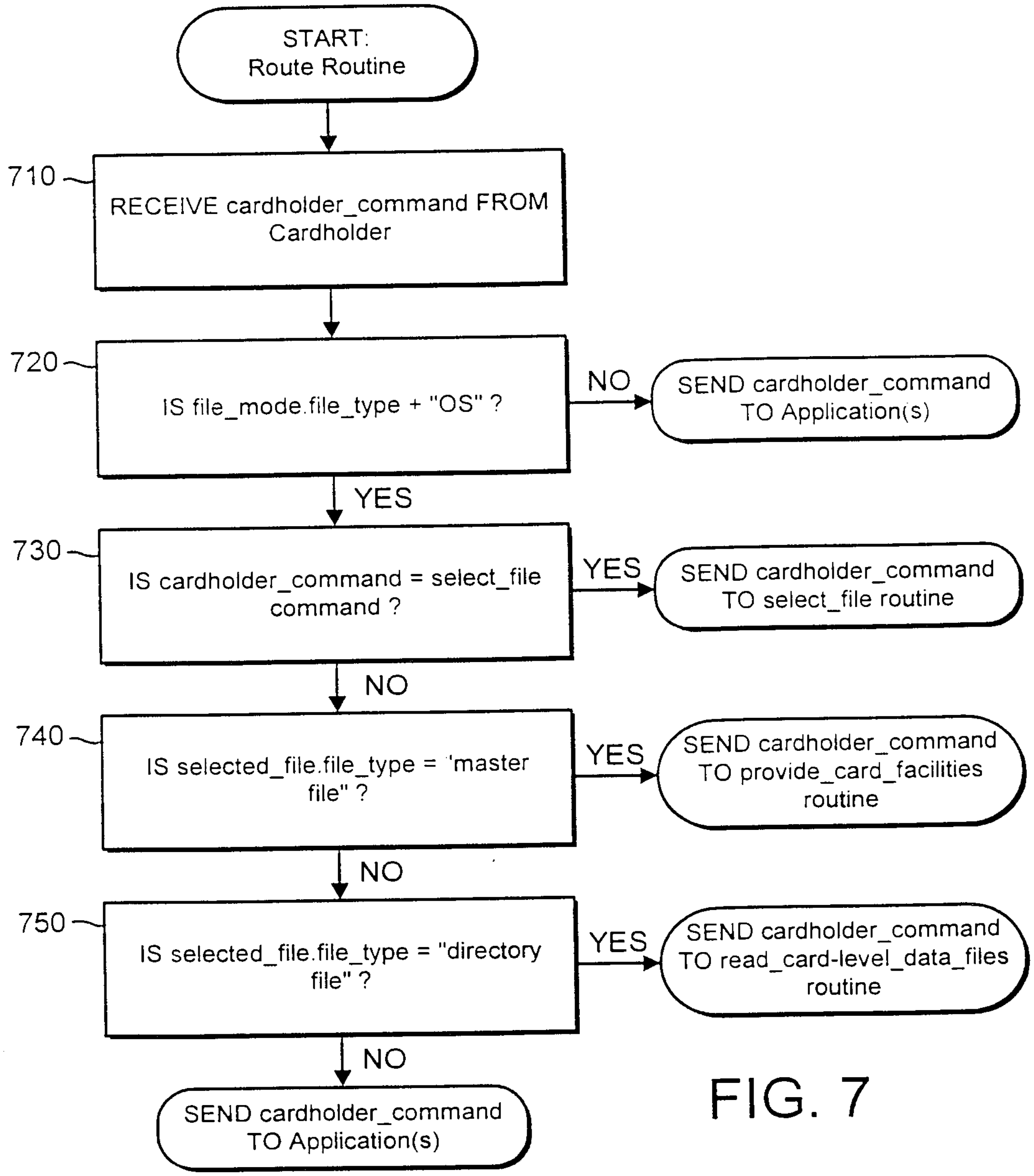

START:
Route Routine
710
RECEIVE cardholder_command FROM Cardholder
720
IS file_mode.file_type + "OS" ?
NO
SEND cardholder_command TO Application(s)
YES
730
IS cardholder_command = select_file command ?
YES
SEND cardholder_command TO select_file routine
NO
740
IS selected_file.file_type = 'master file" ?
YES
SEND cardholder_command TO provide_card_facilities routine
NO
750
IS selected_file.file_type = "directory file" ?
YES
SEND cardholder_command TO read_card-level_data_files routine
NO
SEND cardholder_command TO Application(s)

FIG. 7

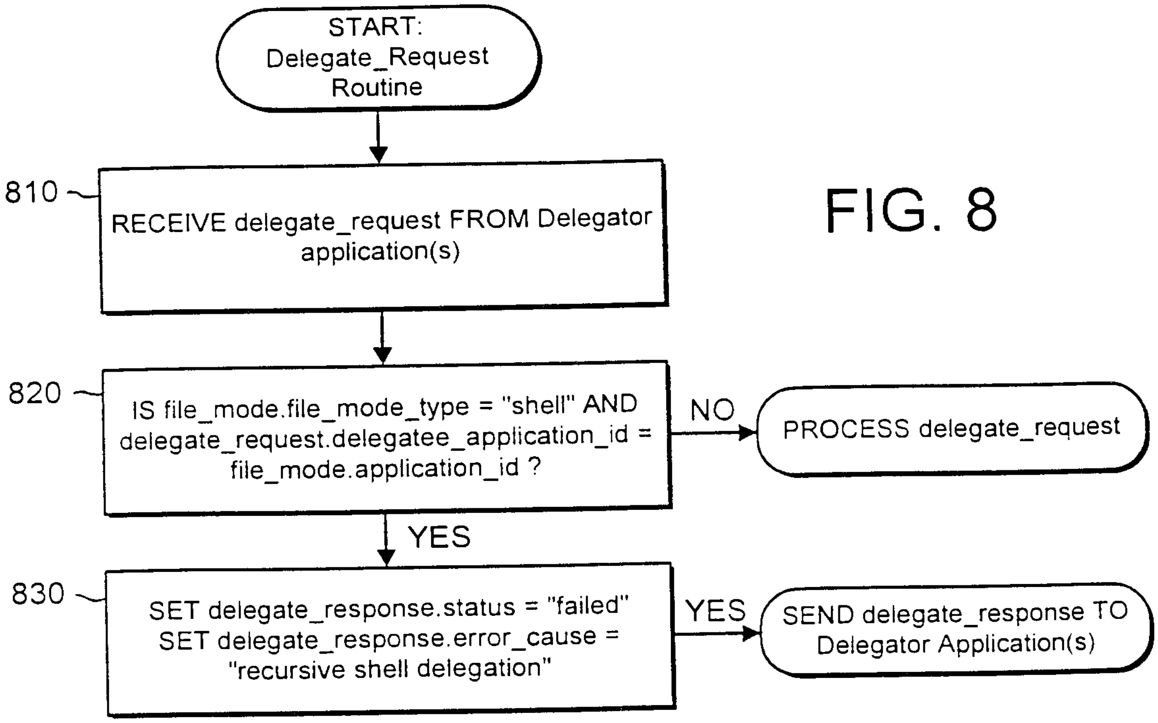

START:
Delegate_Request
Routine
810
RECEIVE delegate_request FROM Delegator application(s)
820
IS file_mode.file_mode_type = "shell" AND delegate_request.delegatee_application_id = file_mode.application_id ?
NO
PROCESS delegate_request
YES
830
SET delegate_response.status = "failed"
SET delegate_response.error_cause = "recursive shell delegation"
YES
SEND delegate_response TO Delegator Application(s)

FIG. 8

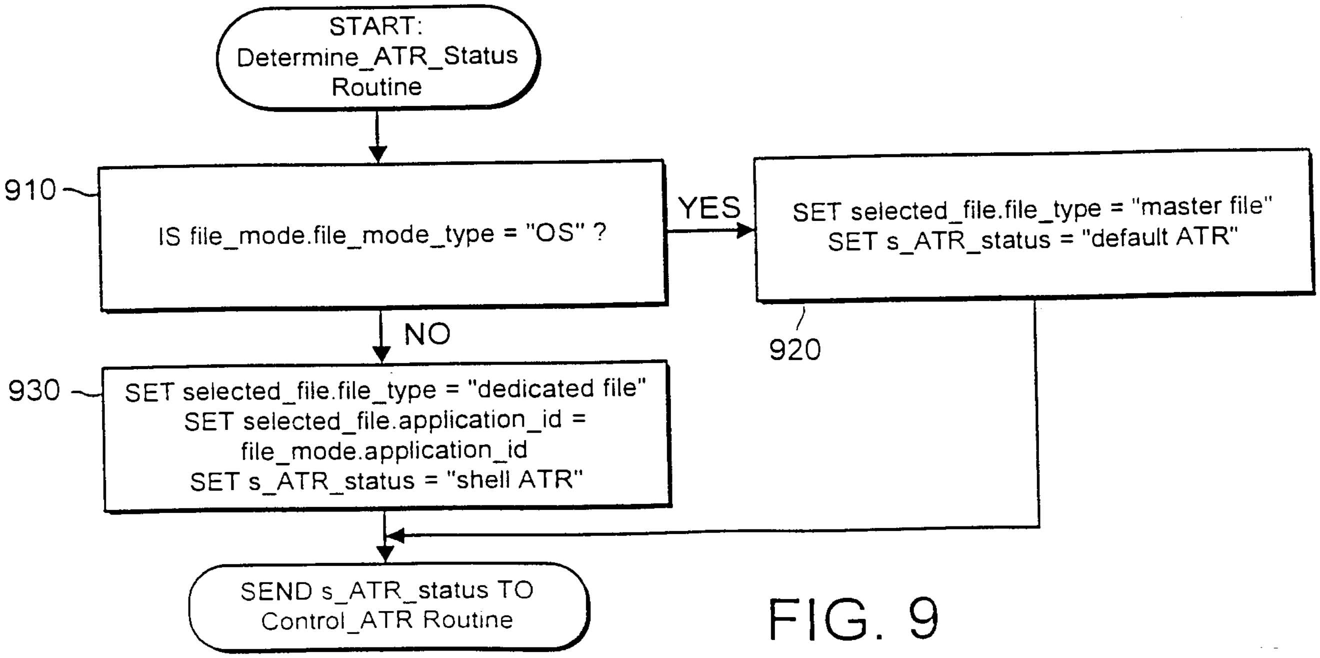

START:
Determine_ATR_Status
Routine
910
IS file_mode.file_mode_type = "OS" ?
YES
SET selected_file.file_type = "master file"
SET s_ATR_status = "default ATR"
920
NO
930
SET selected_file.file_type = "dedicated file"
SET selected_file.application_id = file_mode.application_id
SET s_ATR_status = "shell ATR"
SEND s_ATR_status TO Control_ATR Routine

FIG. 9

ANNEX A TO THE DRAWINGS

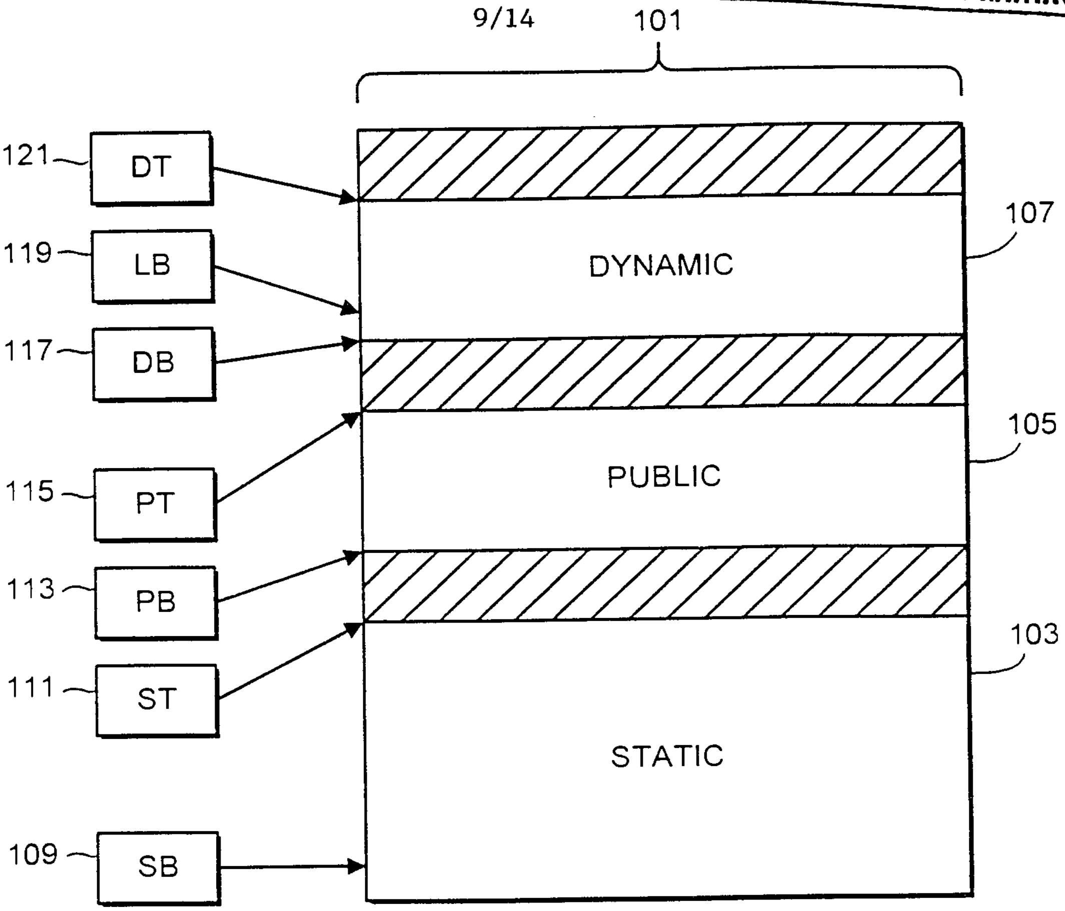


FIG. 1

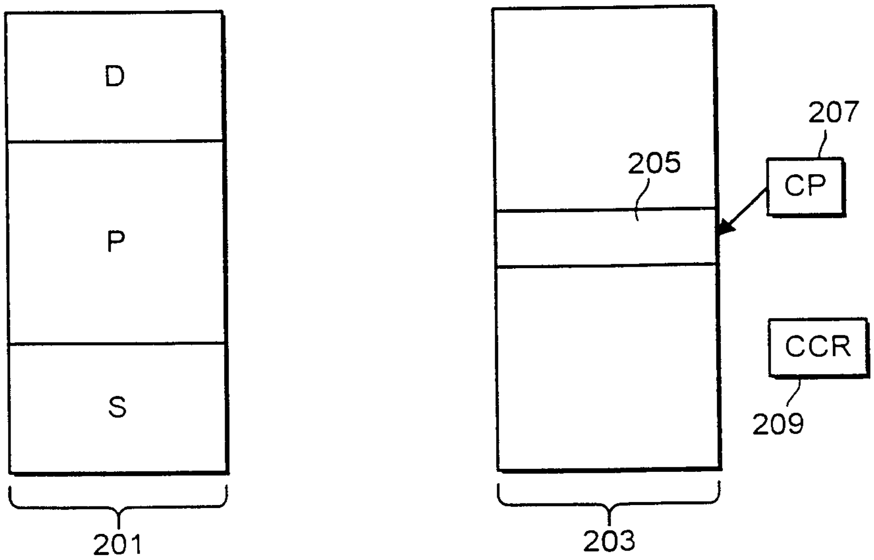


FIG. 2

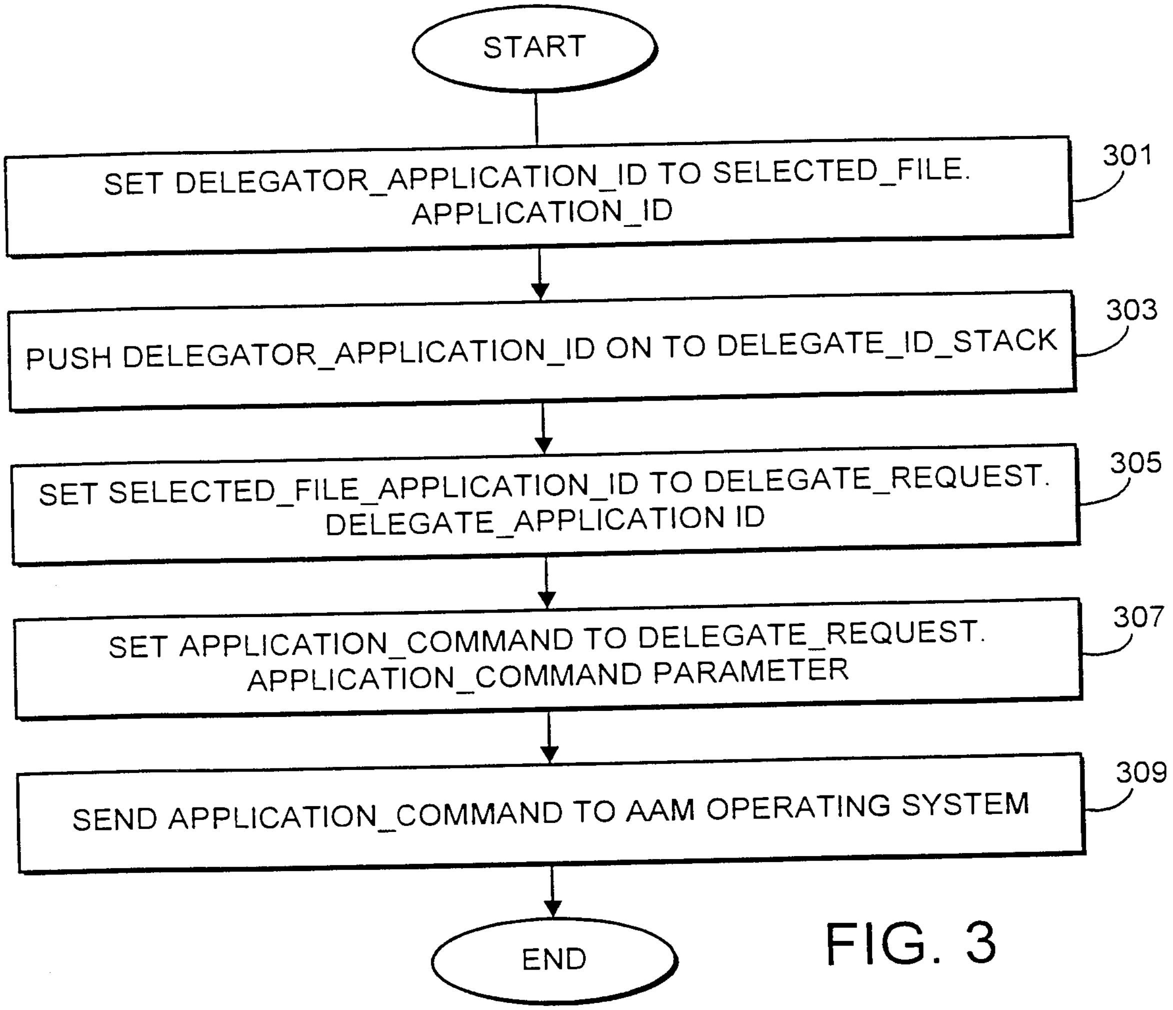

START
SET DELEGATOR_APPLICATION_ID TO SELECTED_FILE. APPLICATION_ID
301
PUSH DELEGATOR_APPLICATION_ID ON TO DELEGATE_ID_STACK
303
SET SELECTED_FILE_APPLICATION_ID TO DELEGATE_REQUEST. DELEGATE_APPLICATION ID
305
SET APPLICATION_COMMAND TO DELEGATE_REQUEST. APPLICATION_COMMAND PARAMETER
307
SEND APPLICATION_COMMAND TO AAM OPERATING SYSTEM
309
END

FIG. 3

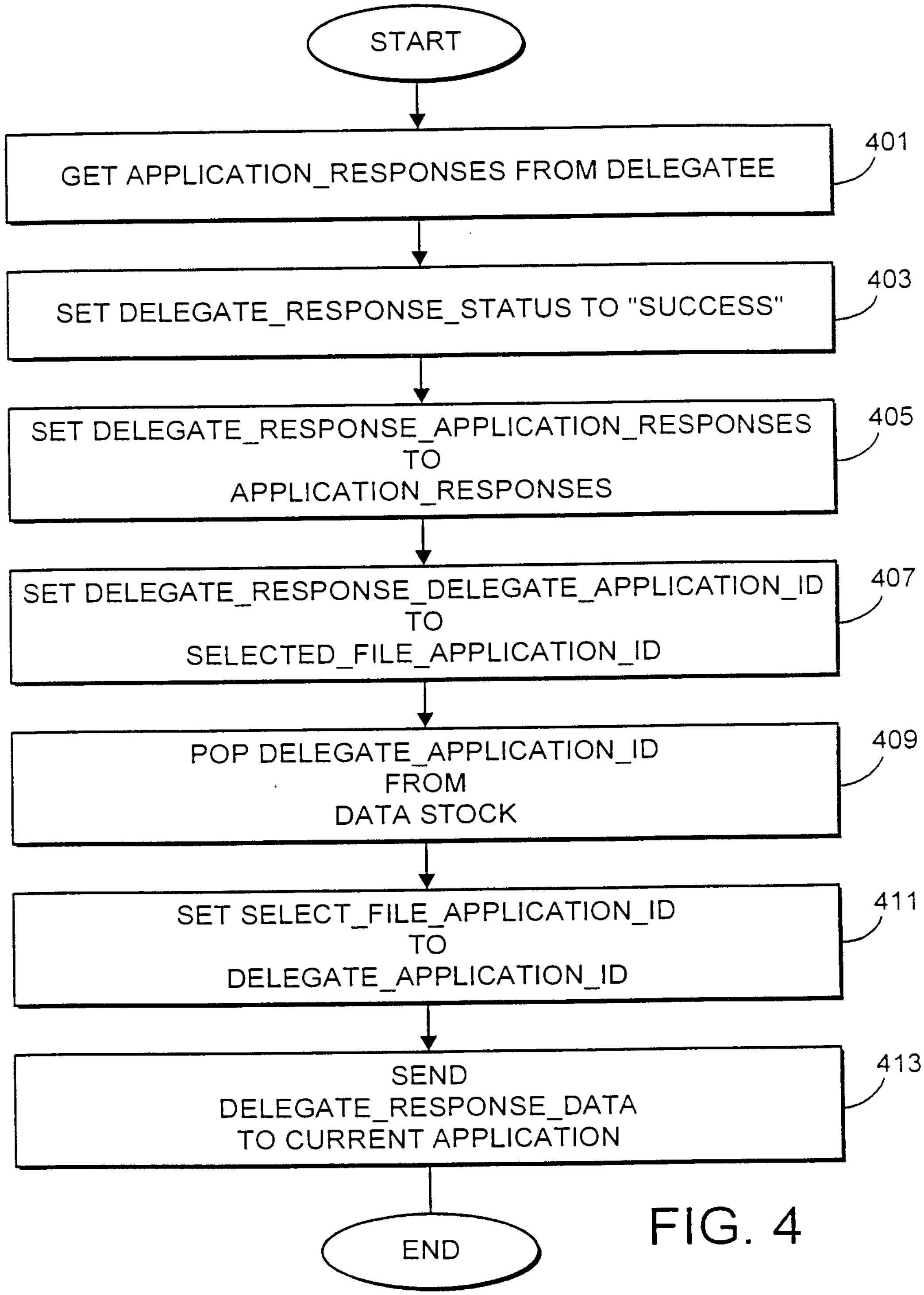

START
GET APPLICATION_RESPONSES FROM DELEGATEE
401
SET DELEGATE_RESPONSE_STATUS TO "SUCCESS"
403
SET DELEGATE_RESPONSE_APPLICATION_RESPONSES TO APPLICATION_RESPONSES
405
SET DELEGATE_RESPONSE_DELEGATE_APPLICATION_ID TO SELECTED_FILE_APPLICATION_ID
407
POP DELEGATE_APPLICATION_ID FROM DATA STOCK
409
SET SELECT_FILE_APPLICATION_ID TO DELEGATE_APPLICATION_ID
411
SEND DELEGATE_RESPONSE_DATA TO CURRENT APPLICATION
413
END

FIG. 4

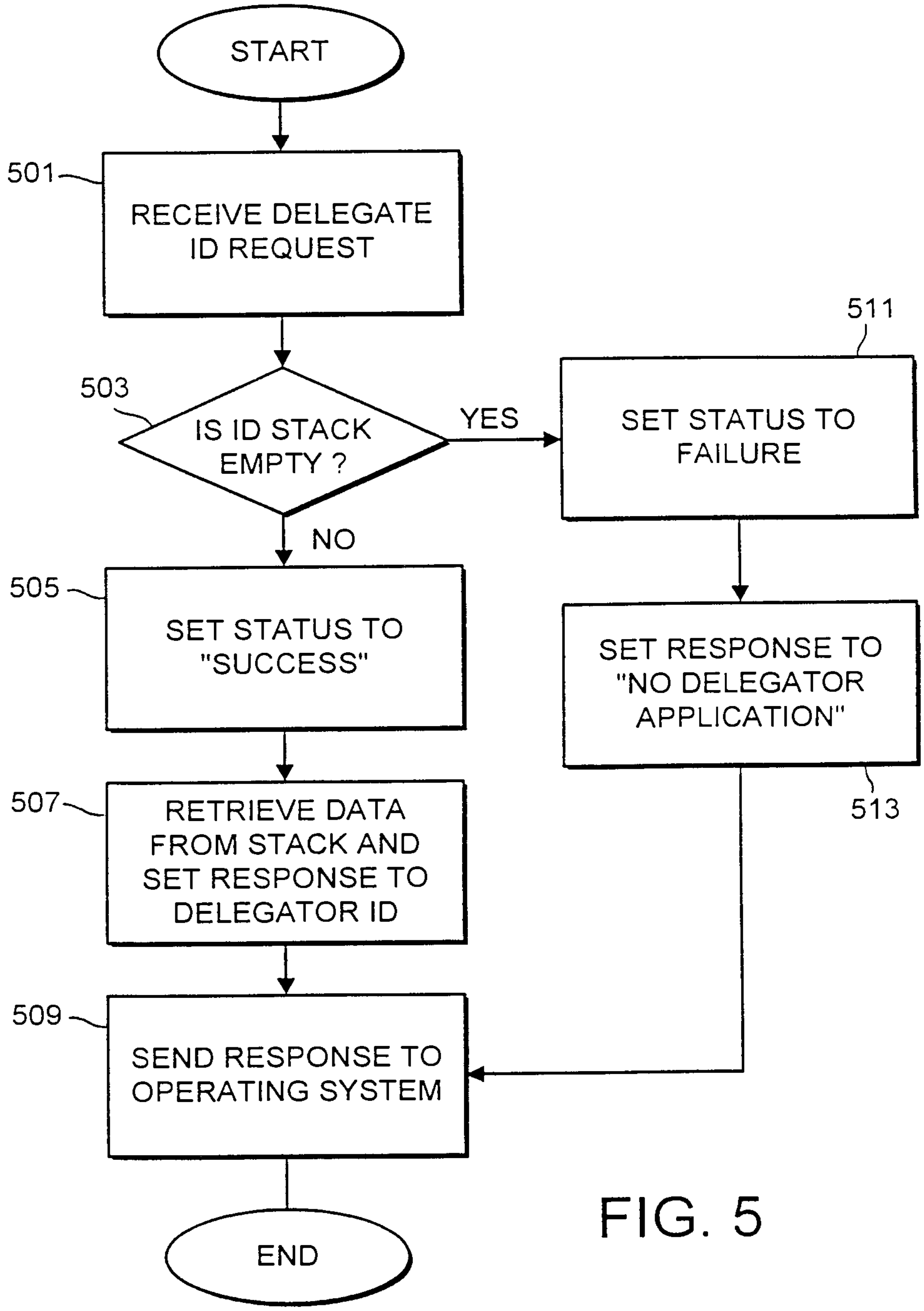

START
501
RECEIVE DELEGATE ID REQUEST
503
IS ID STACK EMPTY ?
YES
NO
511
SET STATUS TO FAILURE
505
SET STATUS TO "SUCCESS"
SET RESPONSE TO "NO DELEGATOR APPLICATION"
513
507
RETRIEVE DATA FROM STACK AND SET RESPONSE TO DELEGATOR ID
509
SEND RESPONSE TO OPERATING SYSTEM
END

FIG. 5

ANNEX A TO THE DRAWINGS

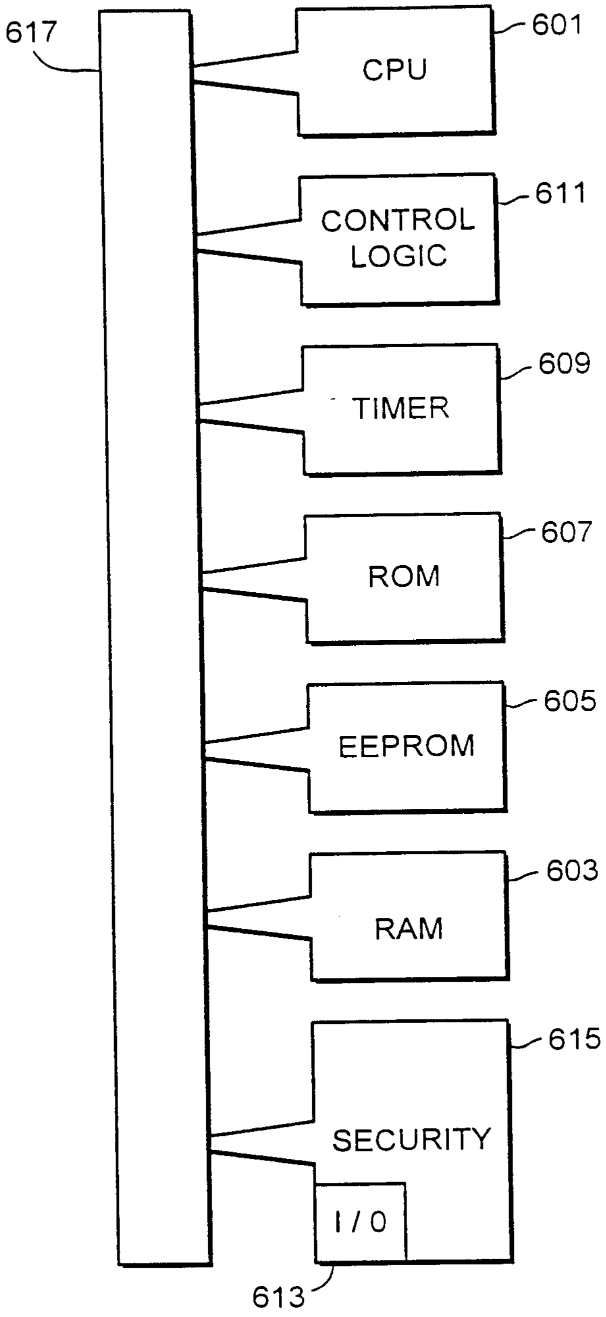


FIG. 6

14/14

ANNEX A TO THE DRAWINGS

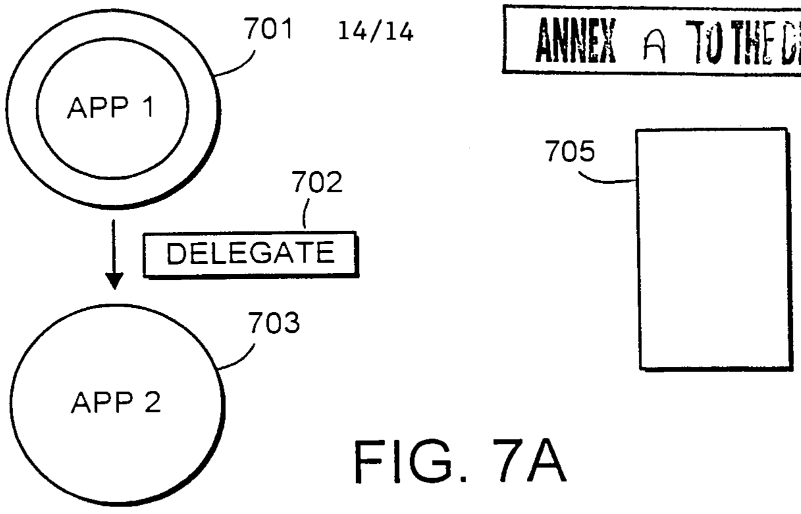


FIG. 7A

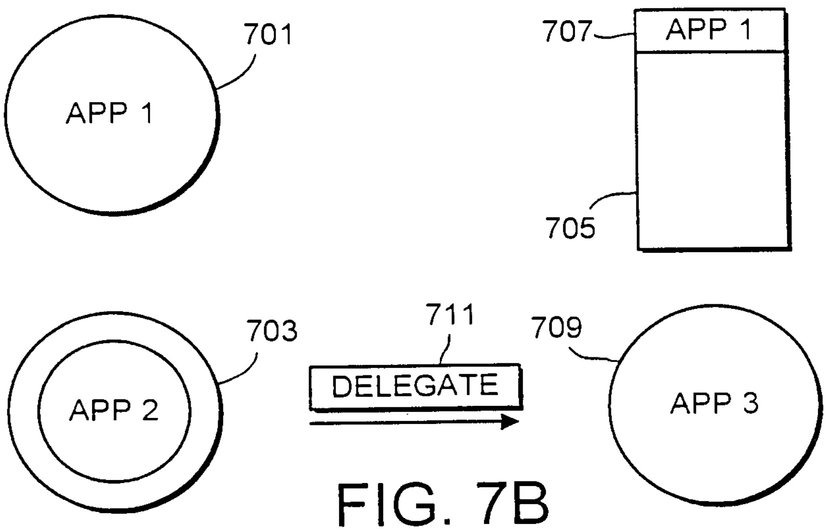


FIG. 7B

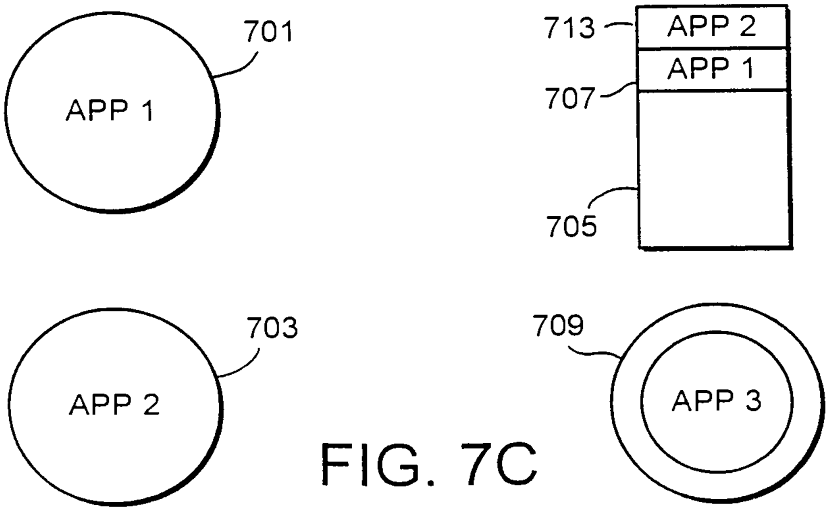


FIG. 7C

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G07F 7/10	A2	(11) International Publication Number: WO 98/52158 (43) International Publication Date: 19 November 1998 (19.11.98)

(21) International Application Number: PCT/GB98/01388

(22) International Filing Date: 14 May 1998 (14.05.98)

(30) Priority Data:

60/046,514	15 May 1997 (15.05.97)	US
60/046,543	15 May 1997 (15.05.97)	US
09/078,031	13 May 1998 (13.05.98)	US

(71) Applicant: MONDEX INTERNATIONAL LIMITED [GB/GB]; 47–53 Cannon Street, London EC4M 5SQ (GB).

(72) Inventors: EVERETT, David, Barrington; 31 Ashdown Avenue, Saltdean, Brighton, East Sussex BN2 8AH (GB). MILLER, Stuart, James; 9 Woodford Green, The Warren, Bracknell, Berks. RG12 9YQ (GB). PEACHAM, Anthony, David; 4 Lynwood, Groombridge, Tunbridge Wells, Kent TN3 9LX (GB). SIMMONS, Ian, Stephens; The Elms, School Road, Broughton, Cambs. PE17 3AT (GB). RICHARDS, Timothy, Philip; 32 Craig Mount, Radlett, Herts. WD7 7LW (GB). VINER, John, Charles; Hydes, Woodlands Lane, Windlesham GU20 6DL (GB).

(74) Agent: POTTER, Julian, Mark; D. Young & Co., 21 New Fetter Lane, London EC4A 1DA (GB).

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

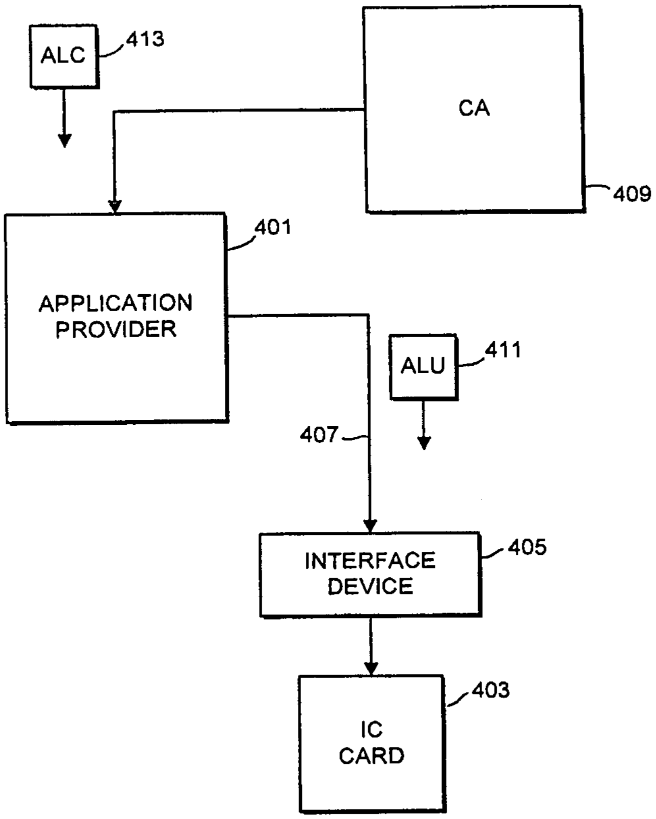
Published

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

(54) Title: INTEGRATED CIRCUIT CARD WITH APPLICATION HISTORY LIST

(57) Abstract

There is provided an integrated circuit card for loading an application copy thereon and a method of loading an application copy onto the integrated circuit card, wherein the application copy is one of a plurality of copies of an application. The application copy has an associated application identifier that uniquely identifies the application from other applications and an application copy number that is unique for each copy of the application. The integrated circuit card includes a microprocessor and a memory coupled to the microprocessor. The memory includes an application history list area for storing application identifiers and application copy numbers of applications that have been previously loaded onto the integrated circuit card. The method includes receiving by the integrated circuit card the application copy, the application identifier, and the application copy number; determining by the integrated circuit card whether the application identifier and the application copy number are contained in the application history list area; and failing to load the application copy by the integrated circuit card if the application identifier and the application copy number are contained in the application history list area.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

INTEGRATED CIRCUIT CARD WITH APPLICATION HISTORY LIST

BACKGROUND OF INVENTION

Integrated circuit (IC) cards are becoming increasingly used for many different purposes in the world today, principally because they are ideal tools for the delivery of distributed, secure information processing at a low cost. An IC card, also called a "smart card," is a card typically the size of a conventional credit card, but which contains a computer chip on the card. The computer chip on the IC card typically includes a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), a random access memory (RAM), an input/output (I/O) mechanism, and other circuitry to support the microprocessor in its operations. The computer chip can execute one or more applications stored on the card. Examples of applications that IC cards are being used to store and execute include credit/debit, electronic money/purse, telephone calling card, and loyalty reward applications.

When an application is initially loaded onto an IC card, the application may include data that is associated with the application. Such data may include, for example, data that identifies the cardholder, such as the cardholder's name and account number. Additionally, the associated data may also include a promotional or bonus value provided by the application provider to the cardholder for loading the application. For example, with a telephone calling card application, an application provider may provide a certain amount of free calling time. As another example, with an electronic purse application, an application provider may provide bonus electronic cash. As yet another example, with a frequent flyer loyalty application, an application provider may provide free miles.

The use of application data to provide promotional or bonus value creates a potential problem for the IC card manufacturer and the application provider regarding the integrity of loading applications. A solution is needed to prevent a cardholder from intentionally or unintentionally copying an application when it is first loaded, and reloading the application thereafter to reload the value in the data associated with the application. By repeated reloading of an application, a cardholder may potentially obtain an unlimited amount of promotional or bonus value to which he or she is not entitled. At the same time, however, cardholders may be required to reload an application for legitimate reasons, such as for updating an application.

Accordingly, a need exists for a method of loading an application onto an IC card such that a cardholder is prevented from illegitimately reloading an application once it has been loaded onto the IC card.

The foregoing technical challenges and needs are addressed by embodiments in accordance with the invention which provides technical solutions.

SUMMARY OF THE INVENTION

In accordance with a preferred embodiment of the present invention, there is provided a method of loading an application copy onto an integrated circuit card, wherein the application copy is one of a plurality of copies of an application. The application copy has an associated application identifier that uniquely identifies the application from other applications and an application copy number that is unique for each copy of the application. The integrated circuit card includes a microprocessor and a memory coupled to the microprocessor. The memory includes an application history list area for storing application identifiers and application copy numbers of applications that have been previously loaded onto the integrated circuit card. The method includes receiving by the integrated circuit card the application copy, the application identifier, and the application copy number; determining by the integrated circuit card whether the application identifier and the application copy number are contained in the application history list area; and failing to load the application copy by the integrated circuit card if the application identifier and the application copy number are contained in the application history list area.

As it is used in this specification and the appended claims, the term "unique" to refer to application copy numbers refers to two types of numbers: (1) non-random numbers that are actually determined to be unique, and (2) random numbers that are determined to be probabilistically unique for a given cardholder.

The method in accordance with the preferred embodiment of the present invention may further include the steps of allocating a predetermined portion of the memory for the application history list area; determining by the integrated circuit card whether the application history list area is full; and failing to load the application copy if the application history list is full.

The method in accordance with the preferred embodiment of the present invention may further include the step of adding the application identifier and the application copy number to the application history list area if the application identifier and the application copy number are not contained in the application history list area. Thus, once a copy of an application is loaded onto the integrated circuit card, the application identifier and the application copy number associated with the copy of the application are stored in the application history list area for future checking.

The method in accordance with the preferred embodiment of the present invention may also provide a mechanism by which application providers not concerned with repeated loading of applications may circumvent storage of the application identifier and the application copy number in the application history list area. For example, an application copy number of zero can be used to signify that an application may be reloaded as often as desired. Accordingly, the method of the preferred embodiment of the present invention may further include the step of adding the application identifier and the application copy number to the application history list area if the application identifier and the application copy number are not contained in the application history list area and the application copy number is not zero.

The application copy may include both application code and application data. The application identifier and the application copy number may be contained in the application data.

Preferably, the application copy, the application identifier, and the application copy number are transmitted to the integrated circuit card by an application provider. Preferably, before transmitting the application copy to the integrated circuit card, the application provider encrypts at least a portion of the application copy. It is also preferred that an application provider transmit a key transformation unit, which includes information relating to the encryption of the encrypted portion of the application copy. It is further preferred that the integrated circuit card has a first public key pair and that the application provider encrypts the key transformation unit with the public key of the first public key pair before transmitting the key transformation unit to the integrated circuit card.

When the application provider encrypts the key transformation unit with the public key of the first public key pair, the integrated circuit card may decrypt the encrypted key transformation unit with the secret key of the first public key pair. Once the key transformation unit is decrypted, the integrated circuit card may decrypt the application copy using the information contained in the decrypted key transformation unit.

It is also preferred that the application provider has a second public key pair and that the application provider form a signed application copy by encrypting the application copy with the secret key of the second public key pair. The application provider may then transmit both the application copy and the signed application copy to the integrated circuit card.

It is further preferred that the application provider register the public key of the second public key pair with a certification authority, which has a third public key pair. The certification authority may then provide a certificate to the application provider by encrypting the public key of the second public key pair with the secret key of the third public key pair. The application provider may transmit the certificate to the integrated circuit card.

When a certificate is transmitted to the integrated circuit card, the integrated circuit card may obtain the public key of the second key pair by decrypting the certificate using the public key of the third public key pair. The integrated circuit card may then verify the signed application copy using the public key of the second public key pair. The integrated circuit card may fail to load the application copy if the signed application copy is not verified.

In accordance with another preferred embodiment of the present invention, there is provided an integrated circuit card that includes a microprocessor and a memory coupled to the microprocessor. The memory includes an application history list area for storing application identifiers and application copy numbers, each application identifier and each application copy number being associated with an application copy. The application copy is one of a plurality of copies of an application. Each application identifier uniquely identifies an application from other applications, and each application copy number uniquely identifies an application copy from other application copies. The integrated circuit card of the invention further includes means for determining whether an application identifier and an application copy number associated with an application copy to be loaded into the memory area are contained in the application history list area and means for failing to load the application copy to be loaded if the associated application identifier and the associated application copy number are contained in the application history list area.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments in accordance with the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Fig. 1 is a schematic representation of an IC card in accordance with a preferred embodiment of the present invention;

Fig. 2 is a perspective view of an IC card and terminal in accordance with a preferred embodiment of the present invention;

Fig. 3 is a functional block diagram of an IC card in accordance with a preferred embodiment of the present invention;

Fig. 4 is a diagram of a system for remotely loading an application from an application provider onto an IC card in accordance with a preferred embodiment of the present invention;

Fig. 5 is a schematic representation of an application load unit in accordance with a preferred embodiment of the present invention;

Fig. 6 is a flowchart of exemplary steps for processing the application load unit of Fig. 5 in accordance with a preferred embodiment of the present invention; and

Fig. 7 is a flowchart illustrating exemplary steps of a file loading routine, which may be implemented by the operating system of an IC card in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 1 provides a schematic representation of a typical IC card 10 that can be used with the presently claimed invention. The IC card 10 includes an integrated circuit 12 having one or more electrical contacts 14 connected to the integrated circuit 12.

Fig. 2 shows an example of a device with which the IC card 10 communicates. As used in this specification and the appended claims, the terms "interface device" and "terminal" shall be used to generically describe devices with which an IC card may communicate. A typical terminal 20, as shown in Fig. 2, includes a card reader 22, a keypad 24, and a display 26. The keypad 24 and the display 26 allow a user of the IC card 10 to interact with the terminal. The keypad 24 allows the user to select a transaction, to enter a personal identification number ("PIN"), and to enter transactional information. The display 26 allows the user to receive informational messages and prompts for data entry. Other types of terminals may include IC card-compatible ATM machines and telephones.

Fig. 3 provides a functional block diagram of the integrated circuit 12. At a minimum, the integrated circuit 12 includes a processing unit 100 and a memory unit 110. Preferably, the integrated circuit 12 also includes control logic 150, a timer 160, security circuitry 170, input/output ports 180, and a co-processor 190. The control logic 150 provides, in conjunction with the processing unit 100, the control necessary to handle communications between the memory unit 110 and input/output ports 180. The timer 160 provides a timing reference signal for the processing unit 100 and the control logic 150. The security circuitry 170 preferably provides fusible links that connect the input/output ports 180 to internal circuitry for testing during manufacturing. The fusible links are burned after completion of testing to limit later access to sensitive circuit areas. The co-processor 190 provides the ability to perform complex computations in real time, such as those required by cryptographic algorithms.

The memory unit 110 may include different types of memory, such as volatile and non-volatile memory and read-only and programmable memory. For example, as shown in Fig. 3, the memory unit 110 may include read-only memory (ROM), electrically erasable programmable read-only memory (EEPROM), and random-access memory (RAM).

The memory unit 110 stores IC card data such as secret cryptographic keys and a user PIN. The secret cryptographic keys may be any type of well-known cryptographic keys, such as the private keys of public-key pairs. Preferably, the secret cryptographic keys are stored in a secure area of ROM or EEPROM that is either not accessible or has very limited accessibility from outside the IC card.

The memory unit 110 also stores the operating system of the IC card. The operating system loads and executes IC card applications and provides file management and other basic card services to the IC card applications. Preferably, the operating system is stored in ROM.

In addition to the basic services provided by the operating system, the memory unit 110 may also include one or more IC card applications. For example, if the IC card is to be used as an electronic cash card, an application called MONDEX™ PURSE (from Mondex International Limited) might be included on the IC card, which loads an electronic value of a certain currency from a user's account in a financial institution onto the IC card. Preferably, the operating system of the IC card 10 should support multiple applications, such as the MULTOS™ operating system from Mondex International Limited.

An IC card application may include both program and associated data files, which are typically stored in EEPROM. The application program may be written either in the native programming code of the processing unit 100 or it may be written in a higher level language that must be translated before it is executed on the processing unit 100. An example of such a higher level language for use on IC cards is the MULTOS™ Executable Language (MEL). Advantageously, by using a higher level language such as MEL, an application program is capable of running on multiple hardware platforms without any need for re-writing.

Because IC cards typically have limited memory capacity due to the size and cost restraints of placing memory on the IC cards, an IC card may also have primitives stored in ROM, which are subroutines that perform frequently used functions or procedures, such as mathematical functions. The primitives are usually written in the native language of the processing unit 100 so that they can be executed very quickly.

In Fig. 4, there is shown a diagram of a system for remotely loading an application from an application provider 401 onto an IC card 403. The application provider 401 may be a card issuer, a bank, or any other entity that provides application loading services. The IC card 403 communicates with the application provider 401 through an interface device 405, which may be a bank terminal, an ATM, or any other device that communicates with an IC card. The application provider 401 and the interface device 405 communicate by way of a data conduit 407, which can be a telephone line, a cable line, a satellite link, an Internet connection, an intra-net connection, or any other type of communications link.

When loading applications onto an IC card remotely, an application provider is required to address several security issues. First, an application provider must ensure that an application is sent only to the cardholder who is intended to receive the application. Second, the application provider must ensure the privacy of any confidential or trade secret information contained in the applications to be loaded. Third, because the data conduit 407 may be an open link and subject to third parties possibly intercepting or replacing applications being transmitted, an application provider must take security measures to enable the IC card to authenticate the application.

The solutions to these security issues typically involve encryption using symmetric and/or asymmetric cryptography techniques. Symmetric cryptography involves encoding and decoding data using the same mathematical number, called a "key," which must be kept secret. On the other hand, asymmetric cryptography, or "public key" cryptography as it is also called, involves encoding data with one key and decoding data with another key. The two keys are referred to as a key pair, and one of the key pair must be kept secret while the other of the key pair may be publicly distributed. Each key of a key pair may be used to encode data; however, once data is encoded by using one key, it can only be decoded by using the other key.

In the system of Fig. 4, it is assumed that the application provider 401 and the IC card 403 each have cryptographic key pairs. The generation of cryptographic keys is performed by any manner known by those skilled in the art. The system also utilizes a Certification Authority (CA) 409, which also has a cryptographic key pair. The CA 409 may be any entity that is trusted to keep the secret key of its public key pair private and to authenticate the identity of other entities —— as, for example, the identity of the application provider 401.

In the system of Fig. 4, the application provider 401 applies for registration of its public key with the CA 409. To do so, the application provider 401 must meet the identification requirements of the CA 409. If the application provider 401 meets these identification requirements, the CA 409 will issue an Application Load Certificate (ALC) 413, which includes the public key of the application provider 401 encoded or "signed" by the secret key of the CA 409. The ALC 413 may be decoded using the public key of the CA 409, which is publicly distributed. Since the CA 409 is trusted to keep its secret key private and to authenticate the identity of the application provider 401, any entity receiving the ALC 413 is assured that the public key contained within the certificate belongs to the application provider 401.

To load an application onto the IC card 403, the application provider 401 transmits an Application Load Unit (ALU) 411 to the interface device 405 via the data conduit 407. The contents of the ALU 411 are shown schematically in

Fig. 5. The ALU preferably includes an Application Unit (AU) 415, a signed Application Unit (AU_s) 417, a Key Transformation Unit (KTU) 419, and the ALC 413.

The AU 415 contains the application code and data that are to be stored on the IC card. Some or all of the application code and data may be encrypted to protect confidential or trade secret portions of the application code and data.

The AU_S 417 is the application code and data AU 415 signed with the secret key of the application provider 401. Using the public key of the application provider 401 provided in the ALC 413, the IC card 403 may decode the AU_S 417 and compare it to the AU 415 to ensure that the AU 415 has not been tampered with during transmission.

The KTU 419 contains information relating to the encrypted portions of the AU 415. This information allows the IC card 403 to decode those encrypted portions so that the application code and data can be accessed by the IC card 403. The KTU 419 is signed with the public key of the IC card 403, which ensures that only the intended IC card 403 can decode the KTU 419 (using the IC card's secret key). Once the KTU 419 is decoded, the IC card 403 may use the information contained in the KTU 419 to decode the encrypted portions of the application code and data of AU 415.

Fig. 6 shows a flow chart of the steps for processing the ALU 411 when it is received by the IC card 403. In step 601, the IC card 403 receives the ALU 411 from the application provider 401. The ALU 411 is placed in the

EEPROM of the IC card 403 along with header information indicating the location in memory of AU 415, AU_s 417, KTU 419 and ALC 413.

In step 603, the ALC 413 is decoded using the public key of the CA 409. The IC card 403 preferably stores in its memory a copy of the CA public key because it may be used in many transactions. Alternatively, the IC card could obtain the public key from a trusted storage location, such as the interface device 405. Once decoded, the ALC 413 provides the IC card 403 with a trusted copy of the public key of the application provider 401.

In step 605, the IC card 403 uses the application provider's public key to verify the AU 415 was not tampered with during transmission. Using the public key of the application provider 401, the IC card 403 decodes the AU_s 417, which was signed with the secret key of the application provider 401. Once the AU_s 417 is decoded, the decoded AU_s 417 is compared to the AU 415. If the two units match, then the AU 415 is verified.

In step 607, the KTU 419, which has been encrypted with the public key of the IC card 403, is decoded using the private key of the IC card 403. In step 609, the information in the decoded KTU 419 is used to decode the encrypted portions of the AU 415. The KTU 419 may contain, for example, either an algorithm or a key for use in decoding the AU 415.

In addition to the security and authentication measures discussed above, other security and authentication measures may also be employed. Additional methods of security and authentication have been addressed, for example, in the related International Patent Application No. PCT/GB98/00531 entitled "Multi-Application IC Card System" by Everett et al., filed February 19, 1998, and US Application entitled "Key Transformation Unit for an IC Card" by Richards et al., filed May 11, 1998. Both of these applications are hereby incorporated by reference to Annex A and Annex B respectively, and Annex C, all attached herewith.

In accordance with a preferred embodiment of the present invention, the data portion of the AU 415 includes an application identifier for the application to be loaded onto the IC card 403 and an application copy number, which is unique for each copy of an application to be loaded onto the IC card 403. As it is used in this specification and the amended claims, the use of the term “unique” in relation to application copy numbers refers both to non-random numbers that are actually determined to be unique and to random numbers that are determined to be probabilistically unique for a given IC card. Preferably, the data portion of the AU 415 containing the application identifier and the application copy number is encoded (and the KTU 419 contains the information necessary to decode this data portion).

Fig. 7 is a flowchart illustrating the steps of a file loading routine that may be implemented by the operating system of the IC card 403 to take advantage of the application identifier and the application copy number contained in the AU 415 to prevent a cardholder from repeatedly loading the same application onto the IC card 403. In the embodiment of Fig. 7, the application copy number is a random number, also called a “random seed.” In step 701, the file loading routine receives the file loading command *load_file_command* from the security manager of the operating system, *OS_Security_Manager*. The *OS_Security_Manager* of the operating system is responsible for verification and decoding of the ALU 411 as discussed with regard to Fig. 6.

In step 703, the application identifier and random seed associated with the application, referred to as *load_file_command.application_id* and *load_file_command.random_seed*, respectively, are checked against entries in an application history list stored on the IC card, referred to as *os_global_data.app_history_list*. The application history list contains entries for each set of application identifier and random seed associated with an application loaded onto the IC card 403. It is preferred that the application history list be stored in a secure area of EEPROM that is not accessible from outside the IC card.

If the application identifier and random seed associated with the application to be loaded are found in the application history list, in step 705, the response status *load_file_response.status* is set to "failed" and the error description *load_file_response.error_cause* is set to "application previously loaded." The error response *load_file_response* is returned to the *OS_Security_Manager*, indicating that the load file routine failed to load the application because the application had previously been loaded onto the IC card.

If the application identifier and random seed associated with the application to be loaded are not found in the application history list, in step 707, the random seed is checked to determine whether it is equal to zero and the application history list is checked to determine whether it is full. A random seed with a value of zero indicates that the application does not contain any economic value included in its data, and thus may be reloaded as often as desired. If the random seed associated with the application is not zero (indicating there is an economic value included with the application) and the application history list is full, the response status *load_file_response.status* is set to "failed" and the error description *load_file_response.error_cause* is set to "application history list full." In this case, the application cannot be loaded because the application history list is full and, therefore, the application identifier and random seed cannot be added to the application history list for future checking.

If an error condition has not been triggered in steps 703 or 707, in step 711, the directory file record associated with the application is added to the directory file of the IC card -- i.e., the application is loaded onto the IC card 403. In step 713, it is checked whether the random seed is equal to zero. If the random seed is not equal to zero (indicating that there is an economic value included with the application), the application identifier and the random seed are added to the application history list for checking against subsequent applications sought to be loaded onto the IC card. After updating the application history list, the response status *load_file_response.status* is set to "success" and sent to the *OS_Security_Manager.*

If the random seed is equal to zero (indicating that there is no economic value included with the application), the application identifier and random seed are not added to the application history list. Instead, step 717 is skipped, and the response status *load_file_response.status* is set to "success" and sent to the *OS_Security_Manager*.

Advantageously, the file loading routine of Fig. 7 prevents a cardholder from illegitimately reloading an application. If a cardholder intercepts and copies an application to be loaded onto an IC card, the cardholder cannot later reload the application because, once the application is loaded, the application identifier and random seed are stored permanently on the IC card. If a cardholder attempts to reload the application, the operating system of the IC card will fail to reload the application because the application identifier and random seed of the application will match an entry in the application history list of the IC card.

On the other hand, a cardholder is not prevented from legitimately reloading an application from an application provider. Since an application provider will generate a new random seed for each copy of an application it provides, it will be unlikely for a cardholder to receive a second copy of the application from the application provider with the same random seed. Of course, the application provider must use a random seed of sufficient length to ensure that the probability of any cardholder twice receiving the same random seed is sufficiently unlikely.

Alternatively, instead of using a random number, an application provider may use any unique number associated with copies of applications it provides to each cardholder. For example, an application provider may keep a counter that tracks the number of copies of an application that is has provided. The application provider may use the value of the counter to provide a unique number each time it provides a copy of the application to a cardholder. The random seed embodiment is preferred, however, because it is easier to manage (i.e., there is no information that is required to be stored or managed).

Although the present invention has been described with reference to certain preferred embodiments, various modifications, alterations, and substitutions will be known or obvious to those skilled in the art without departing from the spirit and scope of the invention, as defined by the appended claims.

The scope of the present disclosure includes any novel feature or combination of features disclosed therein either explicitly or implicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or mitigates any or all of the problems addressed by the present invention. The application hereby gives notice that new claims may be formulated to such features during the prosecution of this application or of any such further application derived therefrom. In particular, with reference to the appended claims, features from dependant claims may be combined with those of the independent claims in any appropriate manner and not merely in the specific combinations enumerated in the claims.

ANNEX A

MULTI-APPLICATION IC CARD SYSTEM

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. This allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application at its personalization stage. That application, however, cannot be modified or changed after the card is issued even if the modification is desired by the card user or card issuer. Moreover, if a card user wanted a variety of application functions to be performed by IC cards issued to him or her, such as

both an electronic purse and a credit/debit function, the card user would be required to carry multiple physical cards on his or her person, which would be quite cumbersome and inconvenient. If an application developer or card user desired two different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal, making the transaction difficult, lengthy and inconvenient.

The Applicant has recognised therefore, that it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card. Although multiple applications could be pre-selected and placed in the memory of the card during is production stage, it would also be beneficial to have the ability to load and delete applications for card post-production as needed.

The increased flexibility and power of storing multiple applications on a single card create new challenges to be overcome concerning the integrity and security of the information (including application code and associated data) exchanged between the individual card and the application provider as well as within the entire system when loading and deleting applications. The Applicant has further recognised that it would be beneficial to have the capability of the IC card system to exchange data among cards, card issuers, system operators and application

ANNEX A TO THE DESCRIPTION

providers securely and to load and delete applications securely at any time from either a terminal or remotely over a telephone line, internet or intranet connection or other data conduit. Because these data transmission lines are not typically secure lines, a number of security and entity-authentication techniques must be implemented to make sure that applications being sent over the transmission lines are only loaded on the intended cards.

As mentioned, it is important -- particularly where there is a continuing wide availability of new applications to the cardholder -- that the system has the capability of adding applications onto the IC card subsequent to issuance. This is highly advantageous since it protects the longevity of the IC cards; otherwise, once an application becomes outdated, the card would be useless. In this regard, to protect against the improper or undesired loading of applications onto IC cards, the Applicant has further recognised that it would be beneficial for the IC card system to have the capability of controlling the loading process and restricting, when necessary or desirable, the use of certain applications to a limited group or number of cards such that the applications are "selectively available" to the IC-cards in the system. This "selective capability" would allow the loading and deleting of applications at, for example, a desired point in time in the card's life cycle. It would also allow the loading of an application only to those cards chosen to receive the selected application.

Accordingly, it is an advantage of a preferred embodiment of the invention that it provides these important features and specifically a secure IC-card system that allows for selective availability of smart card applications which may be loaded onto IC cards.

ANNEX A TO THE DESCRIPTION

These and other advantages are achieved by an embodiment of the present invention which proves an IC card system comprising at least one IC card and an application to be loaded onto the card wherein the IC card contains card personalization date and the application is assigned application permissions data designating which IC card or group of IC cards upon which the application may be loaded. The system checks to determine whether the card's personalization data falls within the permissible set indicated by the application's permissions data. If it does, the application may be loaded onto the card.

In a preferred embodiment, the card personalization data is transferred onto the card by the personalization bureau after the card is manufactured. The data preferably includes data representing the card number, the issuer, product class (i.e., such as gold or platinum cards), and the date on which the card was personalized. The card further preferably contains enablement data indicating whether or not the card has been enabled with personalized data.

In a further preferred embodiment, the IC card secure system checks the enablement data prior to loading an application to determine whether or not the card has been enabled. Preferably, if the card has been enabled, the system checks if the card number, the issuer, the product class and/or the date on which the card was personalized are within the acceptable set indicated by the application's permissions data. If so, the application may be loaded onto the IC card.

ANNEX A TO THE DESCRIPTION

In yet another preferred embodiment, the application's permissions data may contain data representative of a blanket permission such that all cards would pass for application loading.

Further aspects, features and advantages of embodiments of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

Fig. 1 is block diagram illustrating the three stages in the life of a multi-application IC card in a secure system;

Fig. 2 is a block diagram illustrating the steps of the card manufacture process;

Fig. 3 is a flow diagram illustrating the steps involved in enabling each of the IC cards in the secure system;

Fig. 4 is a block diagram of an IC card chip which can be used in accordance with an embodiment of the invention;

Fig. 5 is a block diagram illustrating the data stored on the IC card as indicated in block 307 of Fig. 3;

Fig. 5A is a schematic of the data structures residing in an IC card and representing personalization data;

ANNEX A TO THE DESCRIPTION

Fig. 6 is a flowchart illustrating the steps of loading an application onto an IC card in the secure system;

Fig. 7 is a flow chart illustrating the checking steps as indicated in block 601 of Fig. 6;

Fig. 8 is a flowchart illustrating the steps undertaken in determining if loading of an application may proceed;

Fig. 9 is a block diagram showing the components of the system architecture for the enablement process of an IC card in a secure multi-application IC card system; and

Fig. 10 is a system diagram of entities involved with the use of the IC card once it has been personalized.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

ANNEX A TO THE DESCRIPTION

An embodiment of the present invention provides an IC card system and process which allow the flexibility to load and delete selected applications over the lifetime of a multi-application IC card in response to the needs or desires of the card user, card issuers and/or application developers. A card user who has such a card can selectively load and delete applications as desired if allowed by the card issuer in conjunction with the system operator or Certification Authority ("CA") which controls the loading and deleting process by certifying the transfer of information relating to the process.

By allowing applications to be selectively loaded and deleted from the card, a card issuer can extend additional functionality to an individual IC card without having to issue new cards. Moreover, application developers can replace old applications with new enhanced versions, and applications residing on the same card using a common multiple application operating system may interact and exchange data in a safe and secure manner. For example, a frequent flyer loyalty program may automatically credit one frequent flyer mile to a card user's internal account for every dollar spent with an electronic purse such as the Mondex purse or with a credit/debit application. By allowing the ability to selectively load and delete applications, the card user, subject to the requirements of the card issuer, also has the option of changing loyalty programs as desired.

A card issuer or application developer may intend that a particular application be loaded on only one card for a particular card user in a card system. A regional bank may desire to have a proprietary application reside only on the cards which

ANNEX A TO THE DESCRIPTION

the bank issues. Embodiments in accordance with the present invention would allow for this selective loading and specifically allow for the prevention of loading proprietary applications onto unauthorized cards issued by others.

To achieve these desired objectives, embodiments of the present invention give each card a specific indentity by storing "card personalization data" on the card. Morover, each application to be loaded or deleted on one or more cards in the system is assigned "application permissions data" which specify the cards upon which the applications may be loaded.

The type of personalized data can vary depending upon the needs and requirements of the card system. In the preferred embodiment, described in greater detail below, the personalization data include unique card identification designation data, the card issuer, the product class or type (which is defined by the card issuer) and the date of personalization. However, not all of these data elements are required to be used and additional elements could also be included.

The application permissions data associated with an application, also described in greater detail below, can be a single value in an identity field or could include multiple values in the identity field. For example, the application permissions data in the card issuer field could represent both product class A and product class B from a certain Bank X, indicating that the application could be loaded onto cards designated as product classes A and B issued by Bank X (as indicated in the card product ID field of the card's personalization data).

ANNEX A TO THE DESCRIPTION

In addition, a "global value" could be stored in the issuer field (or other field) of the application permissions data indicating that all IC cards in the system regardless of who issued the card would match this permissions field. In this case, for example, a data value of zero stored in the application permissions card-issuer field will match all of the cards' personalization card-issuer fields.

Figure 1 shows the three steps involved in providing an operational multi-application IC card in a secure system. The first step is the card manufacturing step 101. The second step is the personalization step 103 where card personalization data (also called entity authentication data) is loaded onto the card. The third step is the application loading step 105 which checks to see if a card is qualified to receive an application, i.e., when the personalization data is checked against the application permissions data associated with the application to be loaded. Each of these three steps is described in detail below.

Card Manufacture

Figure 2 shows the steps necessary in manufacturing an IC card in a secure system. Step 201 manufactures the physical IC card by creating the integrated circuit on silicon and placing it on the card. The integrated circuit chip will include RAM, ROM and EEPROM memories. When the card is first manufactured, a global public key of the system operator (in this case called the Certification Authority (CA)) is stored on each card in ROM in step 203. This will allow the card to authenticate that the source of any message to it is from the CA since the public key on the card will be matched to the CA's secret key.

ANNEX A TO THE DESCRIPTION

More specifically, this public key stored on the card will allow the individual card to verify data signed with the CA's private key. The public key of the CA, which is stored on the card, is used only for determining if the data sent to the card was signed with the proper CA private key. This allows the card to verify the source of any message coming from the CA.

Step 205 inserts a card enablement key in a secure portion of EEPROM in the card to facilitate card specific confidentiality during enablement, and step 207 inserts a card identifier in EEPROM of the card. The identifier, which can be accessed by any terminal, will allow the system to determine the identity of the card in later processes. The identifier is freely available and will not be used to authenticate messages.

Step 209 stores the operating system code in ROM on the card including any primitives which are called or supported by the operating system. The primitives are written in native language code (e.g., assembly language) and are stored in ROM. The primitives are subroutines which may be called by the operating system or by applications residing on the card such as mathematic functions (multiply or divide), data retrieval, data manipulation or cryptographic algorithms. The primitives can be executed very quickly because they are written in the native language of the processor.

After the IC cards are manufactured, they are sent to a personalization bureau ("PB") to enable and personalize the card by storing card personalization data in the memory of the card. The terms enablement and personalization are used interchangeably herein to indicate the preparatory steps taken to allow the card to be loaded securely with

ANNEX A TO THE DESCRIPTION

an application. The individual cards are preferably manufactured in batches and are sent to a personalization bureau in a group for processing.

Card Enablement/Personalization

Figure 3 shows the steps of the card enablement process when the card arrives at a personalization bureau. The personalization bureau may be the card issuer (e.g., a bank or other financial institution) or may be a third party that performs the service for the card issuer. The personalization bureau configures the card to a specific user or user class.

Figure 3 specifically shows the steps taken to enable and personalize each IC card which will work within the system. The cards can be placed in a terminal which communicates with IC cards and which reads the card identifier data (previously placed on the card during the manufacturing process -- see step 207). This card identification data is read from the card in step 301. The terminal will effectively send a "get identification data" command to the card and the card will return the identification data to the terminal.

The PB typically processes a group of cards at the same time, and will first compile a list of IC card identification data for the group of cards it is personalizing. The PB then sends electronically (or otherwise) this list of identification data to the Certification Authority ("CA") which creates a personalization (or enablement) data block for each card identifier. The data block includes the card personalization data organized in a number of identity fields and an individual key set for the card, discussed below. These data blocks are then encrypted and sent to the PB in step 302. By using the

ANNEX A TO THE DESCRIPTION

card identification data, the PB then matches the cards with the encrypted data blocks and separately loads each data block onto the matched card. To insure that the CA controls the identity of the card and the integrity of the system, the PB never obtains knowledge of the content of the data blocks transferred. Some aspects of the personalization are requested by the card issuer to the CA in order to affect their preferred management of the cards they issue. The following additional steps are performed.

Step 303 first checks to see if an enablement bit stored in EEPROM of the card has been already set. If it already has been set, the card has already been configured and personalized and the enablement process will end as shown in step 304. A card cannot be enabled and personalized twice. If the bit has not been set, then the process continues with step 305.

In step 305, the individualized card key set for the card being enabled (which key set is generated at the CA) is stored on the card. The keys can be used later in off-card verification (i.e., to verify that the card is an authentic card). This verification is necessary to further authenticate the card as the one for which the application was intended.

Step 307 generates four different MULTOS Security Manager (MSM) characteristic data elements (otherwise referred to herein as personalization data) for the card at the CA which are used for securely and correctly loading and deleting applications from a particular card. The MSM characteristics also allow for the loading of applications on specific classes of identified cards. (These MSM characteristics are further described in connection with Figure 5.)

ANNEX A TO THE DESCRIPTION

Other data can also be stored on the card at this time as needed by the system design such as an address table or further subroutines.

Step 311 sets the enablement bit in EEPROM of the card which indicates that the enablement process has been completed for the particular card. When this bit is set, another enablement process cannot occur on the card. This ensures that only one personalization and enablement process will occur to the card thus preventing illegal tampering of the card or altering the card by mistake. In the preferred embodiment, the enablement bit is initially not set when the card is manufactured and is set at the end of the enablement process.

Figure 4 shows an example of a block diagram of an IC card chip which has been manufactured and personalized. The IC card chip is located on an IC card for use. The IC card preferably includes a central processing unit 401, a RAM 403, a EEPROM 405, a ROM 407, a timer 409, control logic 411, an I/O ports 413 and security circuitry 415, which are connected together by a conventional data bus.

Control logic 411 in memory cards provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports. CPU 401 with its control logic can perform calculations, access memory locations, modify memory contents, and manage input/output ports. Some cards have a coprocessor for handling complex computations like cryptographic algorithms. Input/output ports 413 are used under the control of a CPU and control logic alone, for communications between the card and a card acceptance device. Timer 409 (which generates or provides a clock pulse) drives the control logic 411 and CPU 401 through the sequence of steps that

ANNEX A TO THE DESCRIPTION

accomplish memory access, memory reading or writing, processing, and data communication. A timer may be used to provide application features such as call duration. Security circuitry 415 includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed ("blown") upon completion of testing to prevent later access. The personalization data to qualify the card is stored in a secured location of EEPROM 405. The comparing of the personalization data to applications permissions data is performed by the CPU 401.

Figure 5 shows the steps of generating and loading the four elements of the card personalization data into the memory of the IC cards, and Fig. 5A shows a schematic of bit maps for each identity field residing in the memory of an IC card containing personalization data in accordance with the present invention. Each data structure for each identity field has its own descriptor code. Step 501 loads the data structure for the identity field "card ID" called "msm_mcd_permissions_mcd_no." This nomenclature stands for MULTOS system manager _ MULTOS card device _ permissions_ MULTOS card device number. Although this number is typically 8 bytes long as shown in Fig. 5A, the data could be any length that indicates a unique number for the card. In the preferred embodiment, 2 bytes are dedicated as a signal indicator, 2 bytes comprise a MULTOS Injection Security Module ID (MISM ID) indicating which security module injected the card with its injected keys when it was manufactured, and 4 bytes comprise an Integrated Circuit Card (ICC) serial number which identifies the individual card produced at the particular MISM.

ANNEX A TO THE DESCRIPTION

Step 503 loads the data structure for the identity field "issuer ID" called "msm_mcd_permissions_ mcd_issuer_id." This nomenclature stands for a MULTOS card device issuer identification number. Each card issuer (such as a particular bank, financial institution or other company involved with an application) will be assigned a unique number in the card system. Each IC card in the MULTOS system will contain information regarding the card issuer which personalized the card or is responsible for the card. A card issuer will order a certain number of cards from a manufacturer and perform or have performed the personalization process as described herein. For example, a regional bank may order 5,000 cards to be distributed to its customers. The "mcd_issuer_id" data structure on these cards will indicate which issuer issued the cards. In the preferred embodiment, the data structure is 4 bytes long (as shown in Fig. 5A at 503A) to allow for many different issuers in the system although the length of the data structure can vary with the needs of the card system.

Step 505 loads the data structure for the identity field "product ID" called "msm_mcd_permissions_mcd_ issuer_product_id." This nomenclature stands for MULTOS card device issuer product identification number. Each card issuer may have different classes of products or cards which it may want to differentiate. For example, a bank could issue a regular credit card with one product ID, a gold credit card with another product ID and a platinum card with still another product ID. The card issuer may wish to load certain applications onto only one class of credit cards. A gold credit card user who pays an annual fee may be entitled to a greater variety of applications than a regular credit card user who pays no annual fee. The product ID field identifies the card as a

particular class and will later allow the card issuer to check the product ID and only load applications onto cards which match the desired class.

Another way to differentiate products is by application type, such as by categorizing the application as financial, legal, medical and/or recreational, or by assigning particular applications to a group of cards. For example, one card issuer may have different loyalty programs available with different companies to different sets of card users. For example, a bank may have an American Airlines® loyalty program and a British Airways® loyalty program for different regions of the country dependent on where the airlines fly. The product type allows the issuer to fix the product classification of the card during the personalization process. When loading applications onto the card, the product type identification number on each card will be checked to make sure it matches the type of card onto which the issuer desires to load. The product type data structure is preferably an indexing mechanism (unlike the other personalization data structure) of 8 bits (as shown at 505A in Fig. 5A) but could be any length depending upon the needs of the card system. In the illustrated embodiment, the resulting instruction would be to locate the second bit (since the byte's indicated value is 2) in the array to be searched (see discussion of step 809 below).

Step 507 loads the data structure for the identity field data called "msm_mcd_permissions_mcd_ controls_data_ date." This nomenclature stands for the MULTOS card device controls data date or, in other words, the date on which the card was personalized so that, for example, the application loader can load cards dated only after a certain date, load cards before a certain date (e.g., for application updates) or load

ANNEX A TO THE DESCRIPTION

cards with a particular data date. The information can include the year, month and day of personalization or may include less information, if desired. The data_date data structure is preferably 1 byte in length (see 507A in Fig. 5A) although it could be any length depending upon the needs of the particular card system used.

Once all of the personalization data structures are loaded and stored in the card, the card has been identified by issuer, product class, date and identification number (and other data fields, if desired), and the card cannot change its identity; these fields cannot be changed in the memory of the card. If a card user wants to change the product_id stored in the card to gain access to different applications available to another product type, a new card will have to be issued to the user containing the correct personalization data. This system is consistent with a gold card member receiving a new card when the classification is changed to platinum.

After the card has been enabled and personalized by storing its individual card key set, MSM personalization characteristics and enablement bit as described in Fig. 3, the card is ready to have applications loaded into its memory.

Loading Applications

The application loading process contains a number of security and card configuration checks to ensure the secure and proper loading of an application onto the intended IC card. The application loading process is preferably performed at the personalization bureau so that the card will contain one or more applications when the card is issued. The card may contain certain common applications which will be present on every card the issuer sends out, such as an electronic purse application or a credit/debit

ANNEX A TO THE DESCRIPTION

application. Alternatively, the personalization bureau could send the enabled cards to a third party for the process of loading applications. The multiple application operating system stored in the ROM of each card and the card MSM personalization data is designed to allow future loading and deleting of applications after the card has been issued depending upon the desires of the particular card user and the responsible card issuer. Thus, an older version of an application stored on the IC card could be replaced with a new version of the application. An additional loyalty application could also be added to the card after it has been initially sent to the card user because the application is newly available or the user desires to use the new application. These loading and deleting functions for applications can be performed directly by a terminal or may be performed over telephone lines, data lines, a network such as the Internet or any other way of transmitting data between two entities. In the present IC card system, the process of transmitting the application program and data ensures that only IC cards containing the proper personalization data and which fit on application permissions profile will be qualified and receive the corresponding application program and data.

Figure 6 shows the preferred steps performed in loading an application onto an IC card in the MULTOS IC card system. For this example, the personalization bureau is loading an application from a terminal which enabled the same card. Step 601 performs an "open command" initiated by the terminal which previews the card to make sure the card is qualified to accept the loading of a specific application. The open command provides the card with the application's permissions data, the application's size, and instructs the card to determine (1) if the enablement bit is set indicating the card

ANNEX A TO THE DESCRIPTION

has been personalized; (2) whether the application code and associated data will fit in the existing memory space on the card; and (3) whether the personalization data assigned to the application to be loaded allows for the loading of the application onto the particular card at issue. The open command could also make additional checks as required by the card system. These checking steps during the open command execution will be described in detail in conjunction with Figure 7.

After the open command has been executed, the application loader via the terminal will be advised if the card contains the proper identification personalization data and if enough room exists in the memory of the card for the application code and related data. If there is insufficient memory, then a negative response is returned by the card and the process is abended (abnormally ended). If the identification personalization data does not match the applications permissions data, a warning response is given in step 603, but the process continues to the load and create steps. Alternatively, if there is no match, the process may automatically be abended. If a positive response is returned by the card to the terminal in step 605, the application loader preferably proceeds to next steps. The open command allows the application to preview the card before starting any transfer of the code and data.

Step 607 then loads the application code and data onto the IC card into EEPROM. The actual loading occurs in conjunction with create step 609 which completes the loading process and enables the application to execute on the IC card after it is loaded. The combination of the open, load and create commands are sent by the terminal, or another application provider source, to the IC card to perform the application

ANNEX A TO THE DESCRIPTION

loading process. The operating system in the IC cards is programmed to perform a specific set of instructions with respect to each of these commands so that the IC card will communicate with and properly carry out the instructions from the terminal.

Step 609 performs the create command which at least: (1) checks if an application load certificate is signed (encrypted) by the CA and therefore authenticates the application as a proper application for the system; and (2) checks the card personalization data stored on the card against the permissions profile for the application to be loaded to qualify the card for loading. It may do other checks as required. If one of the checks fails, then a failure response 610 is given and the process aborts. The application after it has passed these checks will be loaded into the memory of the card.

Figure 7 shows the various steps of the open step 601 of Fig. 6 in more detail. Step 701 determines if the enablement (i.e., control) bit is set. This bit is set when the card has completed its personalization process and has been assigned its personalization data. An application can be loaded on an IC card in the card system only if the card contains the personalization data. If the enablement bit is not set, the card has not been personalized and therefore the card returns a negative response 703 to the terminal. If the enablement bit is set, then the card has been enabled and the test conditions continue with step 711.

Step 711 checks if there is sufficient space in the memory on the card to store the application code and its associated data. Applications will typically have associated data related to their functions. This data will be used and manipulated when the application is run. Storage space in the memory of an IC card is a continuing concern

due to the relatively large physical space required for EEPROM and how it fits in the integrated circuit which is desired to be small enough to fit on a credit card sized card. An example of the size of a preset EEPROM on an IC card is 16K bytes although the actual size varies. Applications can range from 1K byte or less for a very simple application up to the size of available memory for a more sophisticated application. The data associated with an application can range from no data being stored in the card memory to a size constrained by the amount of available memory. These varied sizes of application code and data continually increase as applications become more advanced and diverse.

MULTOS as an operating system is not limited by the number of applications and associated data it can store on the card. Thus, if five applications can fit in the available memory of the card, the card user will have greatly increased functionality than if one or two applications were stored on the card. Once a card's memory is filled to its capacity, however, a new application cannot be loaded onto the card unless another application including its code and data of sufficient size can be deleted. Therefore, checking the amount of available space on the card is an important step. If there is not sufficient space, then an insufficient space response 713 will be returned to the terminal. The application loader can then decide if another existing application on the card should be deleted to make room for the new application. Deletion depends upon the card issuer having an application delete certificate from the CA. If there is sufficient space on the card, then the process continues with step 715.

ANNEX A TO THE DESCRIPTION

An example of the testing of memory spaces in step 711 is now described. The numbers used in this example in no way limit the scope of the invention but are used only to illustrate memory space requirements. An IC card may have 16K available EEPROM when it is first manufactured. The operating system data necessary for the operating system may take up 2K of memory space. Thus, 14K would remain. An electronic purse application's code is stored in EEPROM and may take up 8K of memory space. The purse application's required data may take up an additional 4K of memory space in EEPROM. The memory space which is free for other applications would thus be 2K (16K-2K-8K-4K=2K). If a card issuer wants to load a credit/debit application whose code is 6K bytes in size onto the card in this example, the application will not fit in the memory of the IC card. Therefore, the application cannot load the new application without first removing the purse application from the card. If a new credit/debit application was loaded into EEPROM of the IC card, then it would have to overwrite other application's code or data. The application loader is prevented from doing this.

Figure 8 shows the steps performed in determining whether the card's personalization data falls within the permissible set of cards onto which the application at issue may be loaded. These steps are preferably performed during the execution of the "create" command. However, these steps may be performed at any time during the loading or deleting of an application. As described previously, the card is personalized by storing data specific to the card (MSM personalization data) including: a card ID designation specific to an individual card, the card issuer number indicating the issuer of the card, the product type of the card, such as a gold or platinum card, and the date the

ANNEX A TO THE DESCRIPTION

card was personalized. This data uniquely identifies the card apart from all other IC cards in the system.

Accordingly, applications can be selectively stored on individual cards in the IC card system on virtually any basis, including the following. An application can be loaded selectively to cards containing one or more specific card numbers. An application can be selectively loaded on one or more cards containing a specified card issuer ID. Moreover, an application can be loaded only upon one type of product specified by the particular card issuer, and/or the application can be loaded only on cards which have a specified date or series of dates of personalization. Each of the personalization data allows an application to be selectively loaded onto certain cards or groups of cards and also ensures that cards without the proper permissions will not receive the application. Personalization data types in addition to the four described can also be used as needed.

The selection of IC cards upon which a particular application may be loaded is made possible by the use of "applications permissions data" which is assigned to the application and represents at least one set of cards upon which the application may be loaded. The set may be based on virtually any factor, including one or more of the following: card numbers, card issuers, product types or personalization dates. Although the individual card's personalization data typically identify one specific number, one card issuer, one product type and one date, the application's permissions data may indicate a card numbers or a blanket permission, a card issuer or a blanket permission, and a number of product types and dates.

ANNEX A TO THE DESCRIPTION

For example, a frequent loyalty program may be configured to allow its loading and use on cards in different product classes belonging to one card issuer. In addition, the application permissions data may indicate that the loyalty program can be used on gold and platinum product types if the card was issued after May, 1998. Thus, the MSM permissions check will determine if the card's individual personalization data is included in the allowed or permissible set of cards upon which the application may be loaded. If it is, the application will be loaded.

To expedite the comparison process, an alternative embodiment may include setting one or more permissions data at zero representing a blanket permission for that particular data. For instance, by placing a zero for the "card number" entry in the application permissions data or some other value indicating that all cards may be loaded regardless of their number, the system knows not to deny any cards based on their card number. Moreover, if a zero is placed in the application's permissions data "issuer ID," then all cards similarly will pass the "issuer" test comparison. This feature allows greater flexibility in selecting groups of cards. The zero indicator could also be used for other permissions data, as required.

Referring to Figure 8, each of the permissions data is checked in the order shown, but other orders could be followed because if any one of the permissions fails, the application will be prevented from being loaded on the IC card being checked. The permissions are preferably checked in the order shown. Step 801 checks if the application permissions product type set encompasses the card's product type number stored in the memory of the card. Each card product type is assigned a number by the

ANNEX A TO THE DESCRIPTION

system operator. The product types are specified for each card issuer because different card issuers will have different product types. The cards are selectively checked to ensure that applications are loaded only on cards of authorized product type. The application permissions product type set can be 32 bytes long which includes multiple acceptable product types or can be a different length depending upon the needs of the system. Using data structure 505A as an example, the operating system would check bit number 2 in the 256 bit array (32 bytes x 8 bits per byte) resulting from the 32 byte long application permissions data structure. If the permissions check fails, then the card returns a failure message to the terminal in step 803. If the product type check passes (for example, the value of bit no. 2 being 1), then the process continues with step 805.

Step 805 checks if the application permissions allowable card issuer number set encompasses the card's issuer number stored in the memory of the card or if the application permissions issuer data is zero (indicating all cards pass this individual permissions check). Each card issuer is assigned a number by the system operator and the cards are selectively checked to ensure that applications are loaded only on cards distributed by authorized card issuers. The application permissions card issuer number set can be 4 bytes long if one issuer is designated or can be longer depending upon the needs of the system. If the issuer check fails, then the card returns a failure message to the terminal in step 807. If the check passes, then the process continues with step 809.

Step 809 checks if the application permissions date set encompasses the card's data date stored in the memory of the card. The date that the IC card was personalized will be stored and will preferably include at least the month and year. The

ANNEX A TO THE DESCRIPTION

cards are selectively checked to ensure that applications are loaded only on cards with the authorized personalization date. The application permissions date set can be 32 bytes long which includes multiple dates or can be a different length depending upon the needs of the system. If the date permissions check fails, then the card returns a failure message to the terminal in step 811. If the date check passes, then the process continues with step 813.

Step 813 checks if the application permissions allowable card number set encompasses the card's ID number stored in the card memory or if the application permissions allowable card number data is zero (indicating all cards pass this individual permissions check). The testing of the permissions is performed on the card during the execution of the open, load and create commands. The application permissions card number data set can be 8 bytes long if one number is designated or can be longer depending upon the needs of the system. If the card number check fails, then the card returns a failure message to the terminal in step 815. If the check passes, then the process continues with step 817.

Summary of IC Card System's Process

Figure 9 shows the components of the system architecture for the card initialization process of an IC card in a secure multiple application IC card system. The system includes a card manufacturer 102, a personalization bureau 104, an application loader 106, the IC card 107 being initialized, the card user 109 and the certification authority 111 for the entire multiple application secure system. The card user 131 is the

ANNEX A TO THE DESCRIPTION

person or entity who will use the stored applications on the IC card. For example, a card user may prefer an IC card that contains both an electronic purse containing electronic cash (such as MONDEX™) and a credit/debit application (such as the MasterCard® EMV application) on the same IC card. The following is a description of one way in which the card user would obtain an IC card containing the desired applications in a secure manner.

The card user would contact a card issuer 113, such as a bank which distributes IC cards, and request an IC card with the two applications both residing in memory of a single IC card. The integrated circuit chip for the IC card would be manufactured by manufacturer 102 and sent to the card issuer 113 (or an entity acting on its behalf) in the form of an IC chip on a card. As discussed above (see steps 201-209), during the manufacturing process, data is transmitted 115 via a data conduit from the manufacturer 102 to card 107 and stored in IC card 107's memory. (Any of the data conduits described in this figure could be a telephone line, Internet connection or any other transmission medium.) The certification authority 111, which maintains encryption/decryption keys for the entire system, transmits 117 security data (i.e., global public key) to the manufacturer over a data conduit which is placed on the card by the manufacturer along with other data, such as the card enablement key and card identifier. The card's multiple application operating system is also stored in ROM and placed on the card by the manufacturer. After the cards have been initially processed, they are sent to the card issuer for personalization and application loading.

ANNEX A TO THE DESCRIPTION

The card issuer 113 performs, or has performed by another entity, two separate functions. First, the personalization bureau 104 personalizes the IC card 107 in the ways described above, and second, the application loader 106 loads the application provided the card is qualified, as described.

Regarding personalization, an individualized card key set is generated by the CA and stored on the card (see Fig. 3). The card is further given a specific identity using MSM personalization (see Fig. 3, step 307 and Fig. 5) including a card ID number, an issuer ID number identifying the card issuer which processed the card, a card product type number which is specified by the card issuer and the date upon which the personalization took place. After the card has been personalized, applications need to be loaded onto the card so that the card can perform desired functions.

The application loader 106, which could use the same terminal or data conduit as personalization bureau 104, first needs to have determined if the card is qualified to accept the application. This comparison process takes place on the card itself (as instructed by its operating system) using the permissions information. The card, if it is qualified, thus selectively loads the application onto itself based upon the card's identity and the card issuer's instructions. The application loader communicates 119 with the IC card via a terminal or by some other data conduit. After the applications have been loaded on the card, the card is delivered to the card user 109 for use.

The secure multiple application IC card system described herein allows for selective loading and deleting of applications at any point in the life cycle of the IC card after the card has been personalized. Thus, a card user could also receive a personalized

ANNEX A TO THE DESCRIPTION

card with no applications and then select a desired application over a common transmission line such as a telephone line or Internet connection.

Figure 10 is a system diagram of entities involved with the use of an IC card once it has been personalized. The system includes an IC card 151, a terminal 153, an application load/delete entity 155, the certification authority 157, a card issuer 171 and other IC cards 159 in the system. The arrows indicate communication between the respective entities. The CA 157 facilitates loading and deleting of applications. After providing the MSM permissions data and card specific keyset to the card during card enablements, the CA allows applications to be later loaded and deleted preferably by issuing an application certificate. Application specific keys are required to authenticate communication between a card and terminal. The IC card 151 also can communicate with other IC cards 159. Card issuer 171 is involved with all decisions of loading and deleting applications for a card which it issued. All communications are authenticated and transmitted securely in the system.

For instance, IC card 151 will use the following procedure to load a new application onto the card. IC card 101 is connected to terminal 153 and the terminal requests that an application be loaded. Terminal 153 contacts application load/delete entity 155 which, as a result and in conjunction with card issuer 171, sends the application code, data and application permissions data (along with any other necessary data) to terminal 153. Terminal 153 then queries card 151 to ensure it is the correct card onto which the application may be loaded. If IC card passes the checks discussed above, the application is loaded onto card 151. The CA 157 provides the application load or

ANNEX A TO THE DESCRIPTION

delete certificate that enables the application to be loaded or deleted from the card. This example shows one way to load the application, but other variations using the same principles could be performed, such as directly loading the application at the application load/delete entity 155.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

For example, it will be appreciated that the MSM personalization and permissions data may not only be used for loading applications onto IC cards but also for deleting applications from said cards. The same checks involving MSM permissions and loading applications are made for deleting applications. A delete certificate from the CA authorizing the deletion of an application will control from which cards the application may be deleted. This is accomplished through the personalization data stored on each IC card and the permissions check as described herein.

Moreover, the data may also be applicable to personal computers or other units onto which applications may be loaded which are not physically loaded on cards. In addition, the application's permissions data may actually include data representative of a set or sets of cards to be excluded, instead of included -- cards that cannot be loaded with the application.

ANNEX A TO THE DESCRIPTION

The scope of the present disclosure includes any novel feature or combination of features disclosed therein either explicitly or implicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or mitigates any or all of the problems addressed by the present invention. The applicant hereby gives notice that new claims may be formulated to such features during the prosecution of this application or of any such further application derived therefrom. In particular, with reference to the appended claims, features from dependent claims may be combined with those of the independent claims in any appropriate manner and not merely in the specific combinations enumerated in the claims.

CLAIMS:

1. An IC card system comprising at least one IC card, an application to be loaded onto said card and means for determining whether said card is qualified to accept the loading of said application onto said card.

2. The IC card system of claim 1, wherein said IC card contains card personalization data, and said application is assigned application permissions data representing at least one set of IC cards upon which said application may be loaded.

3. The IC card system of claim 2, wherein said determining means compares said card personalization data with said application permissions data.

4. The IC card system of claim 3, wherein whether said application is loaded onto said IC card depends on the result of said comparison, such that in the event the card personalization data matches said permissions data set the card is qualified and the application is loaded.

5. The IC card system of any of claims 2 to claim 4, wherein said personalization data comprises data representative of a unique card identification designation.

ANNEX A TO THE DESCRIPTION

6. The IC card system of any of claims 2 to claim 5, wherein said personalization data comprises data representative of a card issuer.

7. The IC card system of any of claims 2 to claim 6, wherein said personalization data comprises data representative of a product class.

8. The IC card system of any of claims 2 to claim 7, wherein said personalization data comprises data representative of a date.

9. An IC card system comprising at least one IC card and an application, wherein said IC card contains personalization data representative of that card and said application is assigned a permissions data set representing at least one IC card upon which said application may be loaded, said system further comprising means for determining whether said personalization data falls within said permissions data set.

10. The IC card system of claim 9 wherein said application is loaded onto said IC card in the event said determining means determines that said personalization data falls within said set.

11. The IC card system of claim 9 or claim 10 wherein said personalization data comprises data representing a card identification designation, and an issuer of said card.

ANNEX A TO THE DESCRIPTION

12. The IC card system of any of claims 9 to claim 11 wherein said personalization data comprises data representing a product class and a date.

13. The IC card system of any of claims 9 to 12 wherein said permissions data set includes a plurality of card identification designations.

14. The IC card system of any of claims 9 to 13 wherein said permissions data set includes one or more issuers of IC cards.

15. The IC card system of any of claims 9 to 14 wherein said permissions data set includes one or more product classes.

16. The IC card system of any of claims 9 to 15 wherein said permissions data set includes a plurality range of dates.

17. The IC card system of any of claims 9 to 16 wherein said permissions data set includes all IC cards which attempt to load the application.

18. An IC card system comprising at least one IC card, an application to be loaded onto said card and means for enabling said card to be loaded with said application.

19. The IC card system of claim 18 wherein said enabling means comprises means for storing personalization data onto said card.

20. The IC card system of claim 18 wherein said enabling means comprises means for setting an enablement bit.

21. The IC card system of claim 19 wherein said enabling means comprises means for setting an enablement bit.

22. The IC card system of claim 20 further comprising means for checking the enablement bit prior to enabling said IC card to determine whether or not said card has already been enabled.

23. The IC card system of claim 21 further comprising means for checking the enablement bit prior to enabling said IC card to determine whether or not said card has already been enabled.

24. A process for loading an application onto an IC card comprising the step of determining whether said IC card is qualified to accept the loading of said application onto said card.

25. The process of claim 24 wherein said determining step includes the steps of: providing said card with personalization data;

assigning to said application permissions data representing at least one set of IC cards upon which said application may be loaded;

comparing said personalization data with said permissions data; and

loading said application onto said IC card provided said personalization data falls within said set of cards upon which said application may be loaded.

26. The process of claim 25, wherein said personalization data comprises data representative of a card identification designation.

27. The process of claim 25 or claim 26, wherein said personalization data comprises data representative of a card issuer.

28. The process of any of claims 25 to claim 27, wherein said personalization data comprises data representative of a product class.

29. The process of any of claims 25 to claim 28. wherein said personalization data comprises data representative of a date.

ANNEX A TO THE DESCRIPTION

30. The process of any of claims 25 to claim 29 further comprising the first step of enabling said card to be loaded with said application.

31. The process of claim 30 wherein said enabling step includes the step of storing personalization data onto said card.

32. The process of claim 30 wherein said enabling step includes the step of setting an enablement bit indicating that the card has been enabled.

33. The process of claim 31 wherein said enabling step further includes the step of setting an enablement bit indicating that the card has been enabled.

34. The process of claim 32 wherein prior to said enabling step a checking step is performed to determine whether said card has been enabled.

35. The process of claim 33 wherein prior to said enabling step a checking step is performed to determine whether said card has been enabled.

36. A process for deleting an application from an IC card comprising the step of determining whether said IC card is qualified to delete said application based upon permissions data associated with said application.

ANNEX A TO THE DESCRIPTION

37. The process of claim 36 wherein said determining step includes the steps of:

providing said card with personalization data;

assigning to said application permissions data representing at least one set of IC cards from which said application may be deleted;

comparing said personalization data with said permissions data; and

deleting said application from said IC card provided said personalization data falls within said set of cards from which said application may be deleted.

38. The process of claim 37, wherein said personalization data comprises data representative of a card identification designation.

39. The process of claim 37 or claim 38, wherein said personalization data comprises data representative of a card issuer.

40. The process of any of claims 37 to claim 39, wherein said personalization data comprises data representative of a product class.

41. The process of any of claims 37 to claim 40, wherein said personalization data further comprises data representative of a date.

ANNEX A TO THE DESCRIPTION

42. An IC card system comprising at least one IC card, an application to be deleted from said card and means for determining whether said card is qualified to delete said application from said card.

43. The IC card system of claim 42, wherein said IC card contains card personalization data, and said application is assigned application permissions data set representing at least one set of IC cards from which said application may be deleted.

44. The IC card system of claim 43, wherein said determining means compares said card personalization data with said application permissions data.

45. The IC card system of claim 44, wherein whether said application is deleted from said IC card depends on the result of said comparison, such that in the event the card personalization data matches said permissions data set the card is qualified and the application is deleted.

ABSTRACT

Multi-Application IC Card System

A multi-application IC card system is disclosed having selective application loading and deleting capability. Prior to loading an application onto an IC card a test is conducted to determine if the card is qualified to receive the application using personalization data stored on the card and comparing it with permissions data associated with the application indicating one or more sets of cards upon which the application may be loaded. If the personalization data of the card falls within the allowable set of permissions for that application then the card may be loaded with the application. Preferably, the personalization data includes data representative of the card number, issuer, a product class and the date on which the card is personalized.

ANNEX B TO THE DESCRIPTION

ANNEX B

KEY TRANSFORMATION UNIT FOR AN IC CARD

ANNEX B TO THE DESCRIPTION

BACKGROUND OF INVENTION

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. This allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application when it is manufactured and before it is given to a card user. That application, however, cannot be modified or changed after the card is issued even if the modification is desired by the card user or card issuer. Moreover, if a card user wanted a variety of application functions to be performed by IC cards issued to him or her, such as both an electronic purse and a credit/debit function, the card user would be required to carry multiple physical cards on his or her person, which would be quite cumbersome and inconvenient. If an application developer or card user desired two

different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal, making the transaction difficult, lengthy and inconvenient.

Therefore, it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card. Although multiple applications could be preselected and placed in the memory of the card during its production stage, it would also be beneficial to have the ability to load and delete applications for the card post-production as needed.

The increased flexibility and power of storing multiple applications on a single card create new challenges to be overcome concerning the integrity and security of the information (including application code and associated data) exchanged between the individual card and the application provider as well as within the entire system when loading and deleting applications. It would be beneficial to have the capability in the IC card system to exchange data among cards, card issuers, system operators and application providers securely and to load and delete applications securely at any time from a local terminal or remotely over a telephone line, Internet or intranet connection or other data conduit. Because these data transmission lines are not typically secure lines, a number of security and

ANNEX 6 TO THE DESCRIPTION

entity authentication techniques must be implemented to make sure that applications being sent over the transmission lines are not tampered with and are only loaded on the intended cards.

As mentioned, it is important -- particularly where there is a continuing wide availability of new applications to the cardholder -- that the system has the capability of adding applications onto the IC card subsequent to issuance. This is necessary to protect the longevity of the IC cards; otherwise, once an application becomes outdated, the card would be useless. It would be beneficial to allow the addition of applications from a remote location as well as from a direct connection to an application provider's terminal. For example, it would be beneficial for a card user to be able to plug his IC card into his home computer and download an application over the Internet. This type of remote loading of applications raises a number of security risks when transmitting the application code and related data over an unsecured communications line such as the Internet. At least three issues need to be addressed in a system which provides such a capability.

The first issue is to make sure that the IC card receiving the application is the intended IC card and not another IC card. The second issue is determining how the IC card can authenticate that the application came from the proper application provider and not an unknown third party. The third issue concerns preventing third parties from reading the application and making an unauthorized copy. If a portion of the application is encrypted to address the latter issue, the intended IC card needs to have access to the correct key to decrypt the application. In a system with many IC cards and additionally many application

ANNEX B TO THE DESCRIPTION

providers, a secure key transfer technique is required so that the intended IC card can use the correct key for the application which is received. These concerns are raised by both remote application loading as well as local terminal application loading.

Accordingly, it is an object of this invention to provide a key transfer and authentication technique and specifically to provide a secure IC-card system that allows for the secure transfer of smart card applications which may be loaded onto IC cards.

SUMMARY OF THE INVENTION

These and other objectives are achieved by the present invention which provides an IC card system and method for securely loading an application onto an IC card including providing a secret and public key pair for the IC card, encrypting at least a portion of the application using a transfer key, encrypting the transfer key using the IC card's public key to form a key transformation unit, transmitting the encrypted application and the key transformation unit to the IC card, decrypting the key transformation unit using the IC card's secret key to provide the transfer key, decrypting the encrypted application using the provided transfer key and storing the decrypted application on the IC card.

In a preferred embodiment, the secure loading system and method allows the application provider to encrypt two or more portions of the application to be transmitted with two or more different keys, encrypt the two or more keys with the public key of the IC card to form a key transformation unit including the

ANNEX 6 TO THE DESCRIPTION

locations of the encrypted portions. Both the encrypted application and the key transformation unit are sent to the IC card. Because the decryption keys are encrypted with the IC card's public key, only the IC card's secret key can decrypt the key transformation unit. The transfer keys and the locations of the encrypted portions are recovered from the decrypted key transformation unit and the application is decrypted using the recovered transfer keys. This ensures that only the intended IC card can decrypt and use the application which was transmitted to that IC card.

In a preferred embodiment, an application load certificate is also sent to the IC card which is receiving the application. The application load certificate contains the public key of the application provider encrypted by the secret key of the certificate authority ("CA"), or the entity that manages the overall security of the IC card system. The IC card then uses a certificate authority public key to make sure that the certificate was valid by attempting to verify the application load certificate with the CA's public key. The IC card then uses the recovered application provider's public key to verify that the application provider was in fact the originator of the application by verifying the sent application signature generated with the application provider's corresponding secret key.

BRIEF DESCRIPTION OF THE DRAWINGS

Further objects, features and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

ANNEX 6 TO THE DESCRIPTION

Fig. 1 is block diagram of the application loading system which loads an application from an application provider to an IC card;

Fig. 2 is a graphic representation of the contents of an Application Loading Unit;

Fig. 3 is a graphic representation of an Application Unit;

Fig. 4 is a flow chart of the steps for providing an individual key set for an IC card;

Fig. 5 is a graphic representation of a Key Transformation Unit;

Fig. 6 is a graphic representation of a Key Transformation Unit plaintext;

Fig. 7 is a graphic representation of the Application Load Certificate;

Fig. 8 is a graphic representation of the Application Unit being decrypted;

Fig. 9 is a flowchart illustrating the steps undertaken in processing the Application Load Unit;

Fig. 10 is a flowchart illustrating the steps undertaken in processing the KTU; and

Fig. 11 is a block diagram showing the components of an IC card which can receive and process and Application Load Unit.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection

ANNEX 6 TO THE DESCRIPTION

with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

DETAILED DESCRIPTION OF THE INVENTION

It is beneficial to have the capability to load applications onto IC cards containing multiple application operating systems at any time during the lifetime of the IC card. This flexibility allows a user of a card to periodically add new applications to the IC card and also allows older applications to be updated with newer versions of the application when they are released. For example, a card user may start with an IC card that contains a purse, or electronic cash application (e.g., MONDEX™), being stored on his IC card. Some time after the user has the card, he or she may load an additional application onto the card such as a credit/debit application. Some time after loading the credit/debit application on the card, a new version of the credit/debit application may become available and the card user should be able to erase the old application on his IC card and replace it with the new version of the credit/debit application which may contain additional features.

The flexibility of loading applications at different times during the IC card's life cycle creates security issues with the process of loading applications onto the card. In a multiple application operating system environment, it is beneficial to be able to load applications both at terminals, such as a bank ATM machine, as well as over remote communication links, such as telephone lines, cable

ANNEX 6 TO THE DESCRIPTION

lines, the Internet, satellite or other communications means. When loading applications onto an IC card, the application provider and the card issuer (which could be the same entity) needs to provide security regarding the applications to be loaded. First, the application provider must make sure the application is only sent to the correct card user who is intended to receive the application. One solution to this problem is addressed in a related application entitled "Secure Multi-Application IC Card System Having Selective Loading and Deleting Capability" by Everett et al., filed February 12, 1998 and assigned to Mondex International, which is hereby incorporated by reference. Two additional security concerns also need to be addressed when loading an application from a remote source, or even from a local terminal, onto an IC card. First, the source of the application must be authenticated as the proper originator so that applications which may contain viruses or simply take up the limited storage memory in an IC card are not allowed to be loaded onto an IC card. Second, the application and associated data may contain private or trade secret information which needs to be encrypted so other people cannot view the contents of the encrypted application code and data. A portion of the application code and data may be secret while other portions are not. These concerns of authentication and protecting the contents of some or all of the application and associated data being loaded onto a card is addressed herein.

A number of encryption/decryption techniques are described herein. There are two basic types of encryption, symmetric encryption and asymmetric encryption. Symmetric encryption uses a secret key as part of a mathematical formula which encrypts data by transforming the data using the formula and key.

ANNEX B TO THE DESCRIPTION

After the data is encrypted, another party can decrypt the encrypted data using the same secret key with a related decryption algorithm. Thus the same key is used for encryption and decryption so the technique is symmetric. A conventional example of a symmetric algorithm is DES.

Asymmetric encryption techniques use two different keys of a pair for encrypting and decrypting information. The two keys are normally referred to as a private or secret key and a public key. When data is encrypted with one key of the pair, the other key is used to decrypt the data. If a sender of data signs the data with his secret key, anyone with the public key can verify the message. Since public keys are typically known to the public, the contents of a data signed with a secret key cannot be protected but the origination of the data can be verified by determining if a particular secret key signed the data. This authentication process is termed a digital signature. If person A wanted to authenticate a message he was sending to person B, the person A would sign the document with his secret key. When person B received the message, he would use person A's public key to decipher the message. If the message was readable after the public key was applied to it, person B would know that the document was signed with secret key of person A. Thus, the origin of the message has been authenticated.

The asymmetric key set can also be used to protect the contents of a message. If person A wanted to send an encrypted message to person B that no one else could read, he would encrypt the data or message with person B's public key and send it to person B. Now only the holder of B's secret key could decrypt the data. If a combination of keys is used, a person could both authenticate and

encrypt the message. The asymmetric pair of keys has some powerful applications with respect to card security and is more robust than symmetric encryption. However, asymmetric encryption is more processor costly that symmetric encryption. A example of an asymmetric encryption method is RSA.

A hybrid of symmetric encryption which makes the encryption method more powerful is to encrypt data using two symmetric keys. This technique is called triple DES which encodes data with symmetric key 1, decodes the data using symmetric key 2 (which in effect further encodes the data) and then further encodes the data using key 1 again. Once the data has arrived at its destination, key 1 is used to decode the data, key 2 is used to encode the data, and key 1 is used to decode the data. These extra steps of encoding and decoding make the technique more powerful and more difficult to properly decipher without both keys.

Figure 1 shows a block diagram of the entities used in a secure remote application loading process. The application provider 101 can be a card issuer, bank or other entity which provides application loading services. The application provider 101 initiates an application loading process onto IC card 103. Application Provider 101 is connected to data conduit 107 which is connected to interface device 105 (e.g., a terminal that communicates with an IC card). Data conduit 107 can be a telephone line, an intranet, the Internet, a satellite link or any other type of communications link. The application provider 101, which is remotely located from the IC card 103, desires to send and load an application to the IC card. However, because the data link is an open link and subject to third parties possibly intercepting or replacing applications being transmitted, security

ANNEX 6 TO THE DESCRIPTION

measures which authenticate the application itself, the application provider and the IC card must be used to ensure the integrity of the system. The Certificate Authority 109 may also be used to help authenticate that some data being transferred is part of an identified system.

In Figure 1, the application provider sends an application load unit 111 to the interface device 105 and finally to IC card 103. The ALU includes the application itself and security data required to authenticate and protect the application code and associated data. The ALU is discussed specifically in Figure 2 and in connection with the other figures herein. The ALU 111 also preferably contains Application Load Certificate (ALC) 113 data which is sent from the Certification Authority (CA) 109 to the application provider 101. The Certification Authority manages the overall security of the system by providing an Application Load Certificate for each application which is to be loaded onto an IC card. The application provider 101 and the IC card 103 both have individual public/secret keys sets provided to them. The authentication and security processes will now be described.

Figure 2 shows a diagram illustrating the components of an Application Load Unit which is sent from the application loader to the IC card during the application load process. The Application Load Unit (ALU) 201 contains an Application Unit (AU) 203, an Application Unit Signature (AU_S) 205, a Key Transformation Unit (KTU) 207 and an Application Load Certificate (ALC) 209. The ALU 201 is formatted in a conventional format used during data transmission. AU 203 contains the application code and data which are to be stored

ANNEX B TO THE DESCRIPTION

on the IC card, some or all of which is encrypted to protect a secret portion or portions of the code and/or data. AU 203 is described in further detail in connection with Figure 3.

AU_S 205 is the application code and data AU 203 digitally signed with the secret key of the application provider. The public key of the application provider is sent as part of the ALC 209 and is used to authenticate the application provider as the originator of the application. ALC 209 is made up of card identification information and the application provider's public key and is signed by the secret key of the certification authority. All these elements will be described in more detail below.

KTU 207 contains information relating to the encryption of the AU 203 (the code and data of the application) which allows the IC card to decrypt the designated portions so that the application and data can be accessed by the IC card but protects the data during transmission between the application provider and the IC card. KTU 207 is signed with a public key of the IC card for which the application is intended which ensures that only the intended IC card can decrypt the application code and data using the KTU information. This element will be described in connection with Figure 5.

Figure 3 shows a graphic representation of the Application Unit 203 which is part of the application load unit. The AU 203 contains both the program code and associated data which is to be loaded onto the IC card of the card user. The program code consists of a number of program instructions which will be executed by the microprocessor on the IC card. The program instructions can be

ANNEX B TO THE DESCRIPTION

written in any programming language which the operating system stored on the IC card can interpret.

For example, in the MULTOS system the program can be written in MEL™ (MULTOS Executable Language). Most applications have associated data which must be loaded onto the card. For instance, data which identifies the card user such as a person's name or account number may be loaded in a secure manner with the credit/debit application. An application provider may provide electronic cash represented by data as a promotion when installing an electronic purse application. Some or all of this data is desired to be kept secret from third parties. Additionally, the application code itself may be considered proprietary and portions may be desired to be kept secret from others. The use of a Key Transformation Unit (KTU) will allow an application provider to designate and encrypt selected portions of its application as confidential and protect it from third parties.

Application Unit portion 305 indicates the program code which is to be transferred from the application provider to the IC card. Application Unit portion 307 indicates the associated data which is to be transferred as part of the application to be loaded onto the IC card. In this example, three discrete areas of the application unit are shown to be encrypted using either single DES or triple DES. Any number of variations regarding the portions encrypted and the type of encryption can be employed using the techniques described herein.

In this example, encrypted location 309 shows the first portion of the Application Unit 203 which has been encrypted using a triple DES technique. The encryption process as described above involves using a symmetrical key and the

conventionally known DES algorithm to transform the data. The data can later be recovered by applying the key to the known DES algorithm. Encrypted location 311 shows a second portion of the application unit 203 which has been encrypted using triple DES. Encrypted location 313 shows a third portion which is encrypted using single DES. Single DES requires less computation to decrypt and takes up less space as part of the KTU as described below. If the application unit were intercepted by a third party while it was being transmitted from the application loader to the IC card, the encrypted portions could not be read unless the third party had the correct keys. That information, therefore, is protected in the KTU.

The KTU is used to allow the IC card for which the application and associated data is intended to decrypt the encrypted portions of the Application Unit by describing which portions of the application unit are encrypted, which encryption algorithm was used and the key or keys to be used to decipher the text. This information is highly confidential between the application provider and the intended IC card and therefore is protected in a manner unique to the intended card. In order to encrypt the KTU which is part of the overall ALU being transmitted, an individual key set for the particular intended IC card is used. The key set and its generation will now be described.

One of the security operations performed at the CA is to generate an individualized key set for each IC card which is stored on the card. The keys are used for off-card verification (i.e., to verify that the card is an authentic card) and for secure data transportation. The key generation process is shown generally in Figure 4. The key set is made up of three different key data items: the card's

ANNEX B TO THE DESCRIPTION

secret key which is known only to the card, the card's public key which is stored on the card and the card's public key certificate which is the card's public key signed by one of the CA's secret keys. The individual keys of the key set are described in more detail below.

Step 401 stores a card specific transport secret key for the individual IC card in the memory of the card. This secret key is generated by the CA and loaded onto the card via a card acceptance device. Once stored on the card, the CA deletes from its own memory any data relating to the secret key. Thus, only the card itself knows its secret key. The data element containing the secret key information in the card is called "mkd_sk" which stands for MULTOS key data secret key.

Step 403 stores a card specific transport public key for the individual IC card in the memory of the card. This public key is preferably generated by the CA from the asymmetric encryption technique used to produce the secret key in step 401. The data element containing the card's public key information is called "mkd_pk" which stands for MULTOS key data public key.

Step 405 stores a card specific transport public key certificate for the individual IC card in the memory of the card. The data element containing the card's public key certificate information is called "mkd_pk_c" which stands for MULTOS key data public key certificate. This public key certificate is preferably generated by encrypting the transport public key mkd_pk with the secret key of the CA, indicated as follows:

$$\text{mkd\_pk\_c} = [\text{mkd\_pk}]_{CA\_sk}$$

which means the individual card's public key certificate is formed by applying the CA's secret key to the individual card's public key. The process is carried out at the CA. The public key certificate is retained by the CA so that it can regenerate the public key as needed.

A terminal can read the public key certificate from the IC cards to verify that the CA had signed and therefore approved the individual IC card. This is accomplished by verifying the public key certificate with the public component of the CA key set used to sign the mkd_pk. The decrypted public key certificate can then be compared with the public key to verify that the key certificate was certified (signed) by the CA.

Figure 5 is a graphic depiction of the contents of KTU 207, which contains Header portion 501 and KTU Ciphertext portion 503. As shown in Figure 5, header information 501 includes, for example, identifier or permissions information 505 such as the application_id_no (application identification number), mcd_no (IC card no) and/or msm_control_data_date (the date the IC card was issued). Additional identifiers could also be included. These identifiers allow the system to verify that the IC card which receives the ALU is the intended IC card. The permissions data is discussed in detail in the above referenced related application.

KTU Ciphertext 503 corresponds to KTU Plaintext (not encrypted) encrypted with the public key mkd_pk of the intended IC card as shown in box 507. The KTU Plaintext in further described in Figure 6. The public key mkd_pk is obtained from the intended IC card by the application provider. The public key

ANNEX 6 TO THE DESCRIPTION

of an IC card is freely available to anyone and can be obtained directly from the card or from the CA. By signing the KTU Plaintext with the IC card public key, only the intended IC card can use its secret key of the public/secret key pair to decrypt the KTU Ciphertext. This means that only the intended IC card can determine the contents of the KTU plaint text, identify the encrypted portions of the application being loaded and use the keys provided to decrypt and recover the entire application and associate data. Because no other entity has the secret key of the IC card, the security and integrity of the program code and data being transmitted in ensured.

Figure 6 is a graphic representation of KTU Plaintext 601. KTU Plaintext 601 preferably includes identifier field 603, no_area_discriptors field 605, alg_id field 607, area_start field 609, area_length 611, key_length field 613, key_data field 615 and additional area and key fields depending upon the number of encrypted areas present in the Application Unit. Identifiers 603 contain identifying information of the Application Unit to which the KTU applies.

No_area_descriptors 605 indicates how many different portions of the AU have been encrypted. In the example of Figure 3, the number or area descriptors would be three. Field 607 contains the algorithm identifier for the first area which has been encrypted. The algorithm could be DES or triple DES, for example. Field 609 indicates the start of the first encrypted area. This indication could be an offset from the start of the AU. For example, the offset could be 100 which means that the first area starts at the 100th byte of the Application Unit. Field 611 indicates the area length for the first encrypted portions. This field allows the microprocessor on the IC card to know how large an area has been encrypted and when coupled with the start of the area, allows the IC card microprocessor to decrypt the correct portion of the Application Unit. Filed 613 indicates the key length for the particular encrypted portion of the application unit. The length of the key will differ for different encryption techniques. The key length field allows the IC card to know the length of the key data. Field 615 indicates the key data for the particular encrypted portion. The key data is used with the algorithm identity and the location of the encoded portion to decode the encrypted portion. If more than one encrypted area is indicated, then additional data referring of the algorithm, start location, length, key length and key data will be present in the KTU Plaintext. While a number of fields have been described, not all the fields are necessary for the invention. The most important field, however, is the key data itself.

Figure 7 is a graphic representation of the Application Load Certificate (ALC) 209. ALC 209 includes a header 701 and the Application Provider Public Key 703. Header 701 and Application Provider Public Key 703 are then signed (encrypted) with the CA secret key. Thus, the ALC 209 must be provided by the CA to the application provider for each application loaded because only the CA knows the CA private key. Header 701 contains information regarding the application provider and the IC card for which the application is intended. The ALC 209 is placed in the correct ALU by the application provider which can use the identification information. Application Provider Public Key 703 is provided to the CA along with the identification data. The CA then signs this information after verifying its authenticity and returns the signed ALC to the application provider.

The IC card, when it receives the ALC 209 as part of the ALU 201, will open the ALC 209 with the public key of the CA. This ensures that the CA signed the application load certificate and that it is genuine. After decrypting the information, the header identification information 701 is checked and the application provider public key is recovered. This public key will be used to verify that the application and code which is to be loaded onto the IC card originated with the proper application provider.

Figure 8 is a graphic representation of the use of the application provider's public key to decrypt the signed AU 205 in order to verify that AU 203 was signed by the application provider. AU signed 205 is verified with the Application Provider Public Key 801. The recovered AU 803 is then compared with AU 203. If the data blocks match, then the IC card has verified that the application provider signed (encrypted) the application unit and the application is genuine. This authentication is valid because only the application provider has its own secret key. The IC card can process this information because the application provider's public key is provided to it as part of the application load certificate 209 which is signed by the CA. Therefore, it does not need to retrieve the public key from an external location to authenticate the application.

Figure 9 shows a flow chart of the steps for processing the Application Load Unit when it is received by the IC card. Prior to receiving the ALU, identity checks as to the identity of the IC card can be performed if desired. The ALU processing techniques provide a number of further verifications including verifying that the application being loaded is: (1) from the correct application

ANNEX 6 TO THE DESCRIPTION

provider, (2) being loaded on the intended card and (3) certified by the CA. The ALU processing techniques also allow the transportation of transport decryption keys which enable the IC card to decrypt portions of the program code and associated data in a secure manner. In step 901, the IC card receives the ALU from the application provider. The ALU can be transmitted via a terminal connection, contactless connection, telephone, computer, intranet, Internet or any other communication means. The ALU is placed in the EEPROM of the IC card along with header information indicating the starting addresses of AU 203, AU signed 205, the KTU 207 and ALC 209. Alternatively, the IC card could determine the relative address locations of these four units.

Step 903 decrypts the ALC 209 with the CA public key. Each IC card preferably stores in its memory a copy of the CA public key because it is used in many transactions. Alternatively, the IC card could obtain the public key from a known storage location. If the CA public key successfully verifies the ALC 209, then the IC card has verified that the CA has signed the ALC 209 with its secret key and thus the Application Load Certificate is proper. If the IC card cannot verify the ALC successfully, then the ALC was not signed by the CA and the certificate is not proper. The application loading process would then end.

Step 905 then checks the identity of IC card against the identification information sent in the application load certificate to make sure the card is intended to receive the application. This permissions checking is described in the related patent application identified above. If there is no match of identification data, the application loading process ends. If the identification data does match, then the

ANNEX 6 TO THE DESCRIPTION

process continues.

Step 907 uses the application providers public key which was recovered from the verified ALC to verify the AU signature 205. When the ALU was generated by the application provider, the application unit 203 was signed with the application provider's secret key. The application provider then provides its public key to IC card through the ALC. The IC card then verifies the AU signed 205. If the ALU is successfully verified, then it is accepted as having been generated by the application provider. Because the application provider's public key is part of the ALC which is signed by the CA, the CA can make sure that the proper public key has been provided to the IC card. This unique key interaction between the application provider, CA and the intended IC card ensures that no counterfeit or unapproved applications or data are loaded onto an IC card which is part of the secure system.

Step 911 then processes a KTU authentication check which further verifies that only the intended card has received the application. The KTU authentication check makes sure that if a third party does somehow intercept the ALU, the third party cannot read the enciphered portions of the AU and cannot retrieve the keys to decrypt the AU. This step is further explained in Figure 10.

Figure 10 shows the steps of the KTU Authentication process. Step 1001, which is shown in dashed lines because it is preferably optional, checks the identification of the IC card a second time. The identification information can be sent as part of the KTU data. However, this check is optional as it has already been performed once in step 905.

ANNEX 6 TO THE DESCRIPTION

Step 1003 then decrypts KTU ciphertext 503 using the IC card's secret key (mkd_sk). The KTU Plaintext was previously encrypted using the intended card's public key (mkd_pk). This means that only the holder of the intended card's secret key could decrypt the encrypted message. The application provider obtains the intended IC card's public key either from the IC card itself (See Figure 4 and related text for a discussion of the mkd key set) or from a database holding the public keys. If the IC card cannot decrypt the KTU ciphertext properly then the KTU is not meant for that card and the application loading process halts. If the IC card does properly decipher the KTU ciphertext, then the process continues.

Step 1005 identifies an encrypted area of the application unit (AU). In the example of the KTU Plaintext described in connection with Figure 6, the IC card uses a relative starting address and area length field to determine the encrypted portion. Step 1005 also identifies which encryption technique was used to encrypt the identified portion so that the proper decryption technique can be used. For example, the technique could by single or triple DES. Alternatively, the technique could be a default technique used in the system and need not be identified.

Step 1007 then retrieves the key from KTU Plaintext and decrypts the identified portion with the identified decryption technique. This allows the IC card to have the decrypted portion of the AU which it will store in its static memory once all the encrypted portions have been decrypted.

Step 1009 checks if there are any other additional encrypted areas. In the example described in Figure 3, there are three encrypted areas. The number

ANNEX B TO THE DESCRIPTION

of encrypted areas was a field in the example of Figure 6. However, the number of portions can be determined using other conventional means. If there are additional encrypted portions, the process jumps to step 1005. If there are no additional encrypted portions, then the process continues with step 1011.

Step 1011 then loads the decrypted AU into the memory of the IC card. The ALU has passed all of the authentication and decryption checks and the application can now properly reside on the IC card and be executed and used by the card user. While the different checks have been presented in a particular order in Figures 9 and 10, the checks can be performed in any order. While all of the described techniques used in conjunction with the ALU provide the best security, one or more of the individual techniques could be used for their individual purposes or combined with other conventional security techniques.

Figure 11 shows an example of a block diagram of an IC card chip upon which an ALU can be loaded and processed. An integrated circuit is located on an IC card for use. The IC card preferably includes a central processing unit 1101, a RAM 1103, an EEPROM 1105, a ROM 1107, a timer 1109, control logic unit 1111, an I/O port 1113 and security circuitry 1115, which are connected together by a conventional data bus.

Control logic 1111 in memory cards provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports. CPU 1101 with its control logic can perform calculations, access memory locations, modify memory contents, and manage input/output ports. Some cards have a coprocessor for handling complex computations like performing

ANNEX B TO THE DESCRIPTION

cryptographic operations. Input/output ports 1113 are used under the control of a CPU and control logic, for communications between the card and a card interface device. Timer 1109 (which generates or provides a clock pulse) drives the control logic 1111 and CPU 1101 through the sequence of steps that accomplish memory access, memory reading or writing, processing, and data communication. A timer may be used to provide application features such as call duration. Security circuitry 1115 includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed ("blown") upon completion of testing to prevent later access. The AU data after the ALU has been authenticated and verified is stored in EEPROM 1105. The authentication process as described herein is performed by the CPU 1101.

Figure 11 also shows a possible configuration for the integrated circuit chip for the application provider and for the certification authority. CPU 1101 present in the IC chip for the application provider encrypts the necessary information using encryption techniques described herein and performs the necessary data operations. CPU 1101 at the certification authority is used to sign the Application Load Certificate as described herein.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

For example, while loading an application is discussed herein, the

same secure loading process can apply to transmitting other types of data such as data blocks, database files, word processing documents or any other type of data need to be transmitted in a secure manner.

ANNEX 6 TO THE DESCRIPTION

I CLAIM:

1. A method for securely loading an application onto an IC card comprising the steps of:

providing a secret key and public key pair for said IC card;

encrypting at least a portion of said application using a transfer key;

encrypting said transfer key using said IC card's public key to form a key transformation unit;

transmitting said encrypted application and said key transformation unit to said IC card;

decrypting said key transformation unit using said IC card's secret key to recover said transfer key; and

decrypting said encrypted application using said recovered transfer key.

2. The method of claim 1, further including the step of storing said decrypted application on said IC card.

3. The method of claim 1, wherein said encryption technique using said transfer key transfer key is symmetric.

4. The method of claim 3, wherein said symmetric technique is DES.

ANNEX 6 TO THE DESCRIPTION

5. The method of claim 1, wherein said IC card's public and private keys are provided using an asymmetric technique.

6. The method of claim 5, wherein said asymmetric technique is RSA.

7. The method of claim 1, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

8. The method of claim 1, further including the steps of enciphering a second portion of said application exclusive of said at least a portion of said application.

9. The method of claim 8, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

10. The method of claim 8, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

11. The method of claim 8, wherein said key transformation unit indicates the location of said second portion of said application.

12. The method of claim 1, wherein said key transformation unit indicates the location of said at least a portion of said application.

13. The method of claim 1, wherein said key transformation unit indicates the number of encrypted portions of said application.

14. The method of claim 1, further including the steps of providing a public key and secret key set for an application provider; providing a public and secret key set for a certification authority; encrypting said application provider's public key using said certificate authorities' secret key to produce an application load certificate; further signing said encrypted application using said application provider's secret key to produce a signed application and transmitting said signed application and said application load certificate to said IC card.

15. The method of claim 14, further including the step of the IC card verifying said application load certificate with said certification authority's public key.

16. The method of claim 15, further including the steps of verifying the signed encrypted application using the application provider's public key from said decrypted application load certificate.

ANNEX 6 TO THE DESCRIPTION

17. The method of claim 16, wherein said verified application signature is compared to sent encrypted application to determine if they are equivalent.

18. An IC card system comprising:

at least one IC card;

an application provider for providing an application to said at least one IC card;

a communications link coupled to said at least one IC card and said application provider;

a public key and secret key set generated for said IC card;

a transport key generated for use by said applications provider; and

an application, wherein at least a portion of said application is encrypted by said application provider using said transport key; said transport key is encrypted using said IC card's public key to form a key transformation unit; wherein said encrypted application and said key transformation unit are then transmitted to said IC card over said communications link; said transmitted key transformation unit is decrypted using said IC card's private key to recover said transport key; and said transmitted application is decrypted using said recovered transport key to recover said application.

19. The system of claim 18, wherein said recovered application is stored on said card.

ANNEX 6 TO THE DESCRIPTION

20. The system of claim 18, wherein said encryption technique using said transfer key transfer key is symmetric.

21. The system of claim 20, wherein said symmetric technique is DES.

22. The system of claim 18, wherein said IC card's public and private keys are provided using an asymmetric technique.

23. The system of claim 22, wherein said asymmetric technique is RSA.

24. The system of claim 18, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

25. The system of claim 18, further including the steps of enciphering a second portion of said application independently of said at least a portion of said application.

26. The system of claim 25, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

27. The system of claim 25, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

ANNEX B TO THE DESCRIPTION

28. The system of claim 25, wherein said key transformation unit indicates the location of said second portion of said application.

29. The system of claim 18, wherein said key transformation unit indicates the location of at least a portion of said application.

30. The system of claim 18, wherein said key transformation unit indicates the number of encrypted portions of said application.

31. The system of claim 18, further including a certification authority, wherein a public key and secret key set is provided for an application provider; a public and secret key set is provided for said certification authority; said certificate authority's secret key is used to sign said application provider's public key to produce an application load certificate; said application provider's secret key is used to further sign said encrypted application to produce a signed encrypted application and said signed encrypted application and said application load certificate is transmitted to said IC card.

32. The system of claim 31, wherein the IC card verifies said application load certificate with said certification authority's public key.

ANNEX B TO THE DESCRIPTION

33. The system of claim 32, wherein said IC card verifies the signed encrypted application using the application provider's public key from said verified application load certificate.

34. The system of claim 33, wherein said verified application signature is compared to said encrypted application to determine if they are equivalent.

35. A method for transmitting data in a secure manner from a first microprocessor based device to a second microprocessor based device, comprising the steps of:

encrypting at least a portion of said data at said first device using a transfer key;

encrypting said transfer key with a second key at said first device to form a key transformation unit;

transmitting said encrypted data and said key transformation unit to said second device;

decrypting said key transformation unit at said second device to recover said transfer key; and

decrypting said encrypted data using said recovered transfer key.

36. The method of claim 35, further including the step of storing said decrypted data in said second device.

37. The method of claim 35, wherein said second key is from a public key and private key set used in asymmetric encryption.

38. The method of claim 35, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

39. The method of claim 35, further including the steps of enciphering a second portion of said application independently of said at least a portion of said application.

40. The method of claim 39, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

41. The method of claim 39, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

42. The method of claim 39, wherein said key transformation unit indicates the location of said second portion of said application.

43. The method of claim 35, wherein said key transformation unit indicates the location of said at least a portion of said application.

44. The method of claim 35, further including the steps of providing a public key and secret key set for an application provider; providing a public and secret key set for a certification authority; signing said application provider's public key using said certificate authority's secret key to produce an application load certificate; further signing said encrypted application using said application provider's secret key to produce a signed encrypted application and transmitting said signed application and said application load certificate to said IC card.

45. A method for processing a data transmission comprising the steps of:

receiving said data transmission comprising an application encrypted with a first key and a key transformation unit encrypted with a second key, wherein said key transformation unit comprises said first key;

decrypting said key transformation unit to recover said first key;

decrypting said encrypted application using said first key; and

storing said decrypted application.

46. The method of claim 45, wherein said second key is from a public key and private key set used in asymmetric encryption.

47. The method of claim 45, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

48. The method of claim 45, further including the steps of enciphering a second portion of said application independently of said at least a portion of said application.

49. The method of claim 48, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

50. The method of claim 48, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

51. The method of claim 48, wherein said key transformation unit indicates the location of said second portion of said application.

52. The method of claim 45, wherein said key transformation unit indicates the location of said at least a portion of said application.

53. The method of claim 45, further including the steps of providing a public key and secret key set for an application provider; providing a public and secret key set for a certification authority; signing said application provider's public key using said certificate authorities' secret key to produce an application load certificate; further encrypting said encrypted application using said application provider's secret key to produce a signed encrypted application and transmitting

said signed application and said application load certificate to said IC card.

54. The method of claim 53, further including the step of the IC card verifying said application load certificate with said certification authority's public key.

55. The method of claim 54, further including the steps of verifying the signed encrypted application using the application provider's public key from said verified application load certificate.

56. The method of claim 55, wherein said verified application signature is compared to said encrypted application to determine if they are equivalent.

57. An apparatus for processing a data transmission comprising the steps of:

means for receiving said data transmission comprising an application encrypted with a first key and a key transformation unit encrypted with a second key, wherein said key transformation unit comprises said first key;

means for decrypting said key transformation unit to recover said first key;

means for decrypting said encrypted application using said first key; and

means for storing said decrypted application.

58. The apparatus of claim 57, wherein said second key is from a public key and private key set used in asymmetric encryption.

59. The apparatus of claim 57, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

60. The apparatus of claim 57, further including means for enciphering a second portion of said application exclusive of said at least a portion of said application.

61. The apparatus of claim 60, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

62. The apparatus of claim 60, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

63. The apparatus of claim 60, wherein said key transformation unit indicates the location of said second portion of said application.

64. The apparatus of claim 57, wherein said key transformation unit indicates the location of said at least a portion of said application.

ANNEX B TO THE DESCRIPTION

65. The apparatus of claim 60, further including means for verifying an application load certificate with said certification authority's public key.

66. The apparatus of claim 65, further including means for verifying the signed encrypted application using an application provider's public key located in said verified application load certificate.

67. The apparatus of claim 66, wherein said verified application signature is compared to the said encrypted application to determine if they are equivalent.

ANNEX B TO THE DESCRIPTION

ABSTRACT OF THE DISCLOSURE

A multi-application IC card system and method is disclosed providing a secure data transmission technique. The method is used, for example, to load an application from an application provider, which could be remote, to an IC card. At least a portion of the application is encrypted using a transfer key. The transfer key is then encrypted using the public key of a public/secret key pair of the intended IC card to form a key transformation unit. The encrypted application and key transformation unit are then sent to the IC card and the IC card decrypts the key transformation unit using its secret key. The transfer key is then recovered and used to decrypt the encrypted application. The application can then by stored on the IC card and accessed by the card user.

ANNEX C

IC CARD TRANSPORTATION KEY SET

ANNEX C TO THE DESCRIPTION

BACKGROUND OF INVENTION

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. This allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application when it is manufactured and before it is given to a card user. That application, however, cannot be modified or changed after the card is issued even if the modification is desired by the card user or card issuer. Moreover, if a card user wanted a variety of application functions to be performed by IC cards issued to him or her, such as both an electronic purse and a credit/debit function, the card user would be required to carry multiple physical cards on his or her person, which would be quite

ANNEX C TO THE DESCRIPTION

cumbersome and inconvenient. If an application developer or card user desired two different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal, making the transaction difficult, lengthy and inconvenient.

Therefore, it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card. Although multiple applications could be preselected and placed in the memory of the card during its production stage, it would also be beneficial to have the ability to load and delete applications for the card post-production as needed.

The increased flexibility and power of storing multiple applications on a single card create new challenges to be overcome concerning the integrity and security of the information (including application code and associated data) exchanged between the individual card and the application provider as well as within the entire system when loading and deleting applications. It would be beneficial to have the capability in the IC card system to exchange data among cards, card issuers, system operators and application providers securely and to load and delete applications securely at any time from a local terminal or remotely over a telephone line, Internet or intranet connection or other data conduit. Because these data transmission lines are not typically secure lines, a number of security and entity authentication techniques must be implemented to make sure that applications being sent over the transmission lines are not tampered with and are only loaded on the intended cards.

As mentioned, it is important -- particularly where there is a continuing wide availability of new applications to the cardholder -- that the system has the capability of adding applications onto the IC card subsequent to issuance. This is necessary to protect the longevity of the IC cards; otherwise, once an application becomes outdated, the card would be useless. It would be beneficial to allow the addition of applications from a remote location as well as from a direct connection to an application provider's terminal. For example, it would be beneficial for a card user to be able to plug his or her IC card into a home computer and download an application over the Internet. This type of remote loading of applications raises a number of security risks when transmitting the application code and related data over an unsecured communications line such as the Internet.

An entity which transmits an application or data to an IC card requires that only the intended IC card should receive the transmitted data. Third parties should not be able to intercept and view the data. Additionally, a transmitting entity will require verification that the IC card which has requested information is actually part of the overall IC card system and not simply posing as being part of the system. These concerns are raised by both remote application loading as well as local terminal application loading.

ANNEX C TO THE DESCRIPTION

Accordingly, it is an object of this invention to provide a secure transfer technique and specifically to provide a secure IC-card system that allows for the secure transfer of data including smart card applications which may be loaded onto IC cards.

SUMMARY OF THE INVENTION

These and other objectives are achieved by the present invention which provides an IC card method and apparatus for securely transporting data including an application onto an IC card including storing a secret and public key pair on the IC card, retrieving the stored public key from the IC card, encrypting at least a portion of the data to be transported using the public key, transmitting the encrypted data to the IC card and decrypting the encrypted data using the IC card's secret key.

In a preferred embodiment, a certification authority ("CA") or the entity that manages the overall security of the IC card system, encrypts (or digitally signs) a copy of the IC card's public key and the signed copy is also stored on the IC card. The entity transmitting the data to the IC card can verify that the CA has approved the card by retrieving using the IC card's signed public key and verifying the signed public key using the public key of the CA. If verification is successful, the entity has verified that the CA approved the IC card.

ANNEX C TO THE DESCRIPTION

BRIEF DESCRIPTION OF THE DRAWINGS

Further objects, features and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

Fig. 1A is a block diagram of the secure data transfer system which securely transfers data from a transferring entity to an IC card.

Fig. 1B is block diagram of the application loading system which loads an application from an application provider to an IC card;

Fig. 2 is a graphic representation of the contents of an Application Loading Unit;

Fig. 3 is a graphic representation of an Application Unit;

Fig. 4 is a flow chart of the steps for providing an individual key set for an IC card;

Fig. 5 is a graphic representation of a Key Transformation Unit;

Fig. 6 is a graphic representation of a Key Transformation Unit plaintext;

Fig. 7 is a graphic representation of the Application Load Certificate;

Fig. 8 is a graphic representation of the Application Unit being decrypted;

Fig. 9 is a flowchart illustrating the steps undertaken in processing the Application Load Unit;

Fig. 10 is a flowchart illustrating the steps undertaken in processing

ANNEX C TO THE DESCRIPTION

the KTU; and

Fig. 11 is a block diagram showing the components of an IC card which can receive and process and Application Load Unit.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

DETAILED DESCRIPTION OF THE INVENTION

It is beneficial to have the capability to load applications onto IC cards containing multiple application operating systems at any time during the lifetime of the IC card. This flexibility allows a user of a card to periodically add new applications to the IC card and also allows older applications to be updated with newer versions of the application when they are released. For example, a card user may start with an IC card that contains a purse, or electronic cash application (e.g., MONDEX™), being stored on his IC card. Some time after the user has the card, he or she may load an additional application onto the card such as a credit/debit application. Some time after loading the credit/debit application on the card, a new version of the credit/debit application may become available and the

ANNEX C TO THE DESCRIPTION

card user should be able to erase the old application on his IC card and replace it with the new version of the credit/debit application which may contain additional features. Additionally, an IC card needs to receive data regarding personal information such as new credit card account numbers or updated information.

The flexibility of loading applications and transmitting data at different times during the IC card's life cycle creates security issues with the process of loading applications onto the card. In a multiple application operating system environment, it is beneficial to be able to load applications and data both at terminals, such as a bank ATM machine, as well as over remote communication links, such as telephone lines, cable lines, the Internet, satellite or other communications means. When loading applications and data onto an IC card, the application provider needs to provide security regarding the applications to be loaded. First, the application provider must make sure the application is only sent to the correct card user who is intended to receive the application. Second, the application and associated data may contain private or trade secret information which needs to be encrypted so entities other than the IC card cannot view the contents of the encrypted application code and data. A portion of the application code and data may be secret while other portions are not. These concerns of authentication and protecting the contents of some or all of the application and associated data being loaded onto a card is addressed herein.

A number of encryption/decryption techniques are described herein. There are two basic types of encryption, symmetric encryption and asymmetric encryption. Symmetric encryption uses a secret key as part of a mathematical formula which encrypts data by transforming the data using the formula and key. After the data is encrypted, another party can decrypt the encrypted data using the same secret key with a decryption algorithm. Thus the same key is used for encryption and decryption so the technique is symmetric. A conventional example of a symmetric algorithm is DES.

Asymmetric encryption techniques use two different keys of a pair for encrypting and decrypting information. The two keys are normally referred to as a private or secret key and a public key. When data is encrypted with one key of the pair, the other key is used to decrypt the data. If a sender of data signs the data with his secret key, anyone with the public key can verify the message. Since public keys are typically known to the public, the contents of a data signed with a secret key cannot be protected but the origination of the data can be verified by determining if a particular secret key signed the data. This authentication process is termed a digital signature. If person A wanted to authenticate a message he was sending to person B, the person A would sign the document with his secret key. When person B received the message, he would use person A's public key to verify the message. If the message was verified with the public key, person B would know that the document was signed with secret key of person A. Thus, the origin of the message has been authenticated.

The asymmetric key set can also be used to protect the contents of a message. If person A wanted to send an encrypted message to person B that no one else could read, he would encrypt the data or message with person B's public key and send it to person B. Now only the holder of B's secret key could decrypt the data. If a combination of keys is used, a person could both authenticate and encrypt the message. The asymmetric pair of keys has some powerful applications with respect to card security. However, asymmetric encryption is relatively processor costly (processor cost is associated with computation time) compared with symmetric encryption. An example of asymmetric encryption method is RSA®.

A hybrid of symmetric encryption which makes the encryption method more powerful is to encrypt data using two symmetric keys. This technique is called triple DES which encodes data with key 1, decodes the data using key 2 (which in effect further encodes the data) and then further encodes the data using key 1 again. Once the data has arrived at its destination, key 1 is used to decode the data, key 2 is used to encode the data, and key 1 is used to decode the data. These extra steps of encoding and decoding make the technique more powerful and more difficult to properly decipher without both keys.

Figure 1A shows a block diagram of the entities used in transporting data in a secure manner in an IC card system. The transmitting entity 1 can be a card issuer, bank, IC card or other entity which desires to transport data to an IC card 3. The transmitting entity 1 preferably initiates the data transfer process. Alternatively, the IC card 3 can initiate the data transfer process if the card requires data from the transmitting entity 1.

The transmitting entity 1 is connected to interface device 5 (e.g., a terminal that communicates with an IC card). Data conduit 7 can be a telephone line, an intranet, the Internet, a satellite link or any other type of communications link. In this example, the transmitting entity 1, which is remotely located from IC

ANNEX C TO THE DESCRIPTION

card 3, desires to send data in a secure manner to the IC card. However, because the data link is an "open" link (i.e. not a private link) and subject to third parties possibly intercepting or replacing data being transmitted, security measures are needed to guarantee that only the intended IC card will receive the transmitted data. The Certificate Authority 9 can also be used to authenticate that the IC card has been validated as part of the IC card system.

In Figure 1A, a private (or secret) key 19 and corresponding public key 15 is generated for IC card 3. The keys are preferably generated using an asymmetric encryption algorithm such as RSA®. The keys can be generated at the CA 9 or any other location because they are specific only to the IC card 3 and no other copies need to be kept. A third data item, the public key certificate 17, is also generated and stored on the IC card 3.

The public key certificate 17 is generated by signing the public key 15 with the private key of the CA 9. This allows a person with the public key of the CA 9 to verify that the CA digitally signed the IC card's public key in order to certify the IC card's individual key set. The public key certificate can be generated by the CA at the time the IC card private/public key set is generated or at a subsequent time.

When a data transfer is initiated by the transmitting entity 1, the IC card 3 is contacted through the interface device 5 and the IC card 3 sends its public key 15 and its public key certificate 17 to the transmitting entity 1. The transmitting entity then verifies the public key certificate with public key of the CA 13 (which is publicly available from the CA 9 and may be stored in the transmitting

ANNEX C TO THE DESCRIPTION

entity 1) thus determining if the CA 9 digitally signed the public key and verifying that the IC card is a valid card.

The transmitting entity 1 then encrypts the data to be transmitted with the IC card's public key. The transmitting entity 1 then transmits the encrypted data 11 to the interface device 5 and to the IC card 3. The IC card 3 decrypts the encrypted data with its corresponding private (also called secret) key 19. The data can then be processed by the IC card 3. Only the IC card 3 has a copy of its private key so only the intended IC card can access the encrypted data. This ensures that third parties cannot access the encrypted data and correspondingly that only the intended IC card will be able to read and process the data.

Figure 1B shows a secure method for loading applications onto an IC card. Figure 1B shows a block diagram of the entities used in a secure remote application loading process. The application provider 101 can be a card issuer, bank or other entity which provides application loading services. The application provider 101 initiates an application loading process onto IC card 103. IC card 103 is connected to data conduit 107 which is connected to interface device 105 (e.g., a terminal that communicates with an IC card). Data conduit 107 can be a telephone line, an intranet, the Internet, a satellite link or any other type of communications link. The application provider 101, which is remotely located from the IC card 103, desires to send and load an application to the IC card. However, because the data link is an open link and subject to third parties possibly intercepting or replacing applications being transmitted, security measures which authenticate the application itself, the application provider and the IC card must be used to ensure the integrity of the system. The CA 109 may also be used to help authenticate that some data being transferred is part of an identified system.

In Figure 1B, the application provider sends an application load unit 111 to the interface device 105 and finally to IC card 103. The ALU includes the application itself and security data required to authenticate and protect the application code and associated data. The ALU is discussed specifically in Figure 2 and in connection with the other figures herein. The ALU 111 also preferably contains Application Load Certificate (ALC) 113 data which is sent from the Certification Authority (CA) 109 to the application provider 101. The Certification Authority manages the overall security of the system by providing an Application Load Certificate for each application which is to be loaded onto an IC card. The application provider 101 and the IC card 103 both have individual public/secret keys sets. The authentication and security processes will now be described.

Figure 2 shows a diagram illustrating the components of an Application Load Unit which is sent from the application loader to the IC card during the application load process. The Application Load Unit (ALU) 201 contains an Application Unit (AU) 203, an Application Unit Signature (AU_S) 205, a Key Transformation Unit (KTU) 207 and an Application Load Certificate (ALC) 209. The ALU 201 is formatted in a conventional format used during data transmission. AU 203 contains the application code and data which are to be stored on the IC card, some or all of which is encrypted to protect a secret portion or portions of the code and/or data. AU 203 is described in further detail in connection with Figure 3.

ANNEX C TO THE DESCRIPTION

AU_S 205 is the application code and data AU 203 digitally signed with the secret key of the application provider. The public key of the application provider is sent as part of the ALC 209 and is used to authenticate the application provider as the originator of the application. ALC 209 is made up of card identification information and the application provider's public key and is signed by the secret key of the certification authority. All these elements will be described in more detail below.

KTU 207 contains information relating to the encryption of the AU 203 (the code and data of the application) which allows the IC card to decrypt the designated portions so that the application and data can be accessed by the IC card but protects the data during transmission between the application provider and the IC card. KTU 207 is encrypted with the public key of the IC card for which the application is intended which ensures that only the intended IC card can decrypt the application code and data using the KTU information. This element will be described in connection with Figure 5.

Figure 3 shows a graphic representation of the Application Unit 203 which is part of the application load unit. The AU 203 contains both the program code and associated data which is to be loaded onto the IC card of the card user. The program code consists of a number of program instructions which will be executed by the microprocessor on the IC card. The program instructions can be written in any programming language which the operating system stored on the IC card can interpret.

For example, in the MULTOS system the program can be written in

ANNEX C TO THE DESCRIPTION

MEL™ (MULTOS Executable Language). Most applications have associated data which must be loaded onto the card. For instance, data which identifies the card user such as a person's name or account number may be loaded in a secure manner with the credit/debit application. An application provider may provide electronic cash represented by data as a promotion when installing an electronic purse application. Some or all of this data is desired to be kept secret from third parties. Additionally, the application code itself may be considered proprietary and portions may be desired to be kept secret from others. The use of a Key Transformation Unit (KTU) will allow an application provider to designate and encrypt selected portions of its application as confidential and protect it from third parties.

Application Unit portion 305 indicates the program code which is to be transferred from the application provider to the IC card. Application Unit portion 307 indicates the associated data which is to be transferred as part of the application to be loaded onto the IC card. In this example, three discrete areas of the application unit are shown to be encrypted using either single DES or triple DES. Any number of variations regarding the portions encrypted and the type of encryption can be employed using the techniques described herein.

In this example, encrypted location 309 shows the first portion of the Application Unit 203 which has been encrypted using a triple DES technique. The encryption process as described above involves using a symmetric key and the conventionally known DES-based algorithm to transform the data. The data can later be recovered by applying the key to a conventionally known DES-based decryption algorithm. Encrypted location 311 shows a second portion of the

ANNEX C TO THE DESCRIPTION

application unit 203 which has been encrypted using triple DES. Encrypted location 313 shows a third portion which is encrypted using single DES. Single DES requires less computation to decrypt and takes up less space as part of the KTU as described below. If the application unit were intercepted by a third party while it was being transmitted from the application loader to the IC card, the encrypted portions could not be read unless the third party had the correct keys and decryption algorithm. That information, therefore, is protected in the KTU.

The KTU is used to allow the IC card for which the application and associated data is intended to decrypt the encrypted portions of the Application Unit by describing which portions of the application unit are encrypted, which encryption algorithm was used and the key or keys to be used to decipher the text. This information is highly confidential between the application provider and the intended IC card and therefore is protected in a manner unique to the intended card. In order to encrypt the KTU which is part of the overall ALU being transmitted, an individual key set for the particular intended IC card is used. The key set and its generation will now be described.

In accordance with the present invention, one of the security operations performed at the CA is to generate an individualized key set for each IC card which is stored on the card. The keys are used for off-card verification (i.e., to verify that the card is an authentic card) and for secure data transportation. The key generation process is shown generally in Figure 4. The key set is made up of three different key data items: the card's secret key which is known only to the card, the card's public key which is stored on the card and the card's public key

ANNEX C TO THE DESCRIPTION

certificate which is the card's public key signed by the CA's secret key. The individual keys of the key set are described in more detail below.

Step 401 stores a card specific transport secret key for the individual IC card in the memory of the card. This secret key is generated by the CA from a standard asymmetric encryption technique such as RSA® and loaded onto the card via a card acceptance device. Once stored on the card, the CA deletes from its own memory any data relating to the secret key. Thus, only the card itself knows its secret key. The data element containing the secret key information in the card is called "mkd_sk" which stands for MULTOS key data secret key.

Step 403 stores a card specific transport public key for the individual IC card in the memory of the card. This public key is preferably generated by the CA from the asymmetric encryption technique used to produce the secret key in step 401. As with the secret key, once the public key is stored on the card, the CA (or other key provider) deletes from its systems the public key data so that the only copy of the public key is kept in the card. The data element containing the card's public key information is called "mkd_pk" which stands for MULTOS key data public key.

Step 405 stores a card specific transport public key certificate for the individual IC card in the memory of the card. The data element containing the card's public key certificate information is called "mkd_pk_c" which stands for MULTOS key data public key certificate. This public key certificate is preferably generated by signing the transport public key mkd_pk with the secret key of the CA, indicated as follows:

$$\text{mkd\_pk\_c} = [\text{mkd\_pk}]_{\text{CA\_sk}}$$

which means the individual card's public key certificate is formed by applying the CA's secret key to the individual card's public key. The process is carried out at the CA. The public key certificate is retained by the CA so that it can regenerate the public key as needed.

A terminal can read the public key certificate from the IC cards to verify that the CA had signed and therefore approved the individual IC card. This is accomplished by verifying the public key certificate with the public component of the CA key set used to sign the mkd_pk.

Figure 5 is a graphic depiction of the contents of KTU 207, which contains Header portion 501 and KTU Ciphertext portion 503. As shown in Figure 5, header information 501 includes, for example, identifier or permissions information 505 such as the application_id_no (application identification number), mcd_no (IC card no) and/or msm_control_data_date (the date the IC card was issued). Additional identifiers could also be included. These identifiers allow the system to verify that the IC card which receives the ALU is the intended IC card. The permissions data is discussed in detail in the above referenced related application.

KTU Ciphertext 503 corresponds to KTU Plaintext (not encrypted) encrypted with the public key mkd_pk of the intended IC card as shown in box 507. The KTU Plaintext in further described in Figure 6. The public key mkd_pk is obtained from the intended IC card by the application provider. The public key of an IC card is freely available to anyone and can be obtained directly from the card or from the CA. By encrypting the KTU Plaintext with the IC card public key, only the intended IC card can use its secret key of the public/secret key pair to decrypt the KTU Ciphertext. This means that only the intended IC card can determine the contents of the KTU plaint text, identify the encrypted portions of the application being loaded and use the keys to decrypt and recover the entire application and associate data. Because no other entity has the secret key of the IC card, the security and integrity of the program code and data being transmitted in ensured.

Figure 6 is a graphic representation of KTU Plaintext 601. KTU Plaintext 601 preferably includes identifier field 603, no_area_discriptors field 605, alg_id field 607, area_start field 609, area_length 611, key_length field 613, key_data field 615 and additional area and key fields depending upon the number of encrypted areas present in the Application Unit. Identifiers 603 contain identifying information of the Application Unit to which the KTU applies. No_area_descriptors 605 indicates how many different portions of the AU have been encrypted. In the example of Figure 3, the number or area descriptors would be three. Field 607 contains the algorithm identifier for the first area which has been encrypted. The algorithm could be DES or triple DES, for example. Field 609 indicates the start of the first encrypted area. This indication could be an offset from the start of the AU. For example, the offset could by 100 which means that the first area starts at the 100th byte of the Application Unit. Field 611 indicates the area length for the first encrypted portions. This field allows the microprocessor on the IC card to know how large an area has been encrypted and when coupled with

ANNEX C TO THE DESCRIPTION

the start of the area, allows the IC card microprocessor to decrypt the correct portion of the Application Unit. Filed 613 indicates the key length for the particular encrypted portion of the application unit. The length of the key will differ for different encryption techniques. The key length field allows the IC card to know the length of the key data. Field 615 indicates the key data for the particular encrypted portion. The key data is used with the algorithm identity and the location of the encoded portion to decode the encrypted portion. If more than one encrypted area is indicated, then additional data referring to the algorithm, start location, length, key length and key data will be present in the KTU Plaintext. While a number of fields have been described, not all the fields are necessary for the invention. The most important field, however, is the key data itself.

Figure 7 is a graphic representation of the Application Load Certificate (ALC) 209. ALC 209 includes a header 701 and the Application Provider Public Key 703. Header 701 and Application Provider Public Key 703 are then signed (encrypted) with the CA secret key. Thus, the ALC 209 must be provided by the CA to the application provider for each application loaded because only the CA knows the CA private key. Header 701 contains information regarding the application provider and the IC card for which the application is intended. The ALC 209 is placed in the correct ALU by the application provider which can use the identification information. Application Provider Public Key 703 is provided to the CA along with the identification data. The CA then signs this information after verifying its authenticity and returns the signed ALC to the application provider. The IC card, when it receives the ALC 209 as part of the ALU 201, will verify the

ANNEX C TO THE DESCRIPTION

ALC 209 with the public key of the CA. This ensures that the CA signed the Application Load Certificate and that it is genuine. After verifying the information, the header identification information 701 is checked and the application provider public key is recovered. This public key will be used to verify that the application and code which is to be loaded onto the IC card originated with the proper application provider.

Figure 8 is a graphic representation of the use of the application provider's public key to verify the signature of the AU 205 in order to verify that AU 203 was signed by the application provider. AU signature 205 is verified with the Application Provider Public Key 801 and compared with AU 203. If the data blocks match, then the IC card has verified that the application provider signed (encrypted) the application unit and the application is genuine. This authentication is valid because only the application provider has its own secret key. The IC card can process this information efficiently because the application provider's public key is provided to it as part of the Application Load Certificate 209 which is signed by the CA. Therefore, it does not need to retrieve the public key from an external location to authenticate the application.

Figure 9 shows a flow chart of the steps for processing the Application Load Unit when it is received by the IC card. Prior to receiving the ALU, identity checks as to the identity of the IC card can be performed if desired. The ALU processing techniques provide a number of further verifications including verifying that the application being loaded is: (1) from the correct application provider, (2) being loaded on the intended card and (3) certified by the CA. The

ALU processing techniques also allow the transportation of transport decryption keys which enable the IC card to decrypt portions of the program code and associated data in a secure manner. In step 901, the IC card receives the ALU from the application provider. The ALU can be transmitted via a terminal connection, contactless connection, telephone, computer, intranet, Internet or any other communication means. The ALU is placed in an I/O buffer of the IC card along with header information indicating the starting addresses of AU 203, AU signed 205, the KTU 207 and ALC 209. Alternatively, the IC card could determine the relative address locations of these four units.

Step 903 verifies the ALC 209 with the CA public key. Each IC card preferably stores in its memory a copy of the CA public key because it is used in many transactions. Alternatively, the IC card could obtain the public key from a known storage location. If the CA public key verifies the ALC 209 properly, then the IC card has verified that the CA has signed the ALC 209 with its secret key and thus the Application Load Certificate is proper. If the IC card cannot verify the ALC properly, then the ALC was not signed by the CA and the certificate is not proper. The application loading process would then end.

Step 905 then checks the identity of IC card against the identification information sent in the Application Load Certificate to make sure the card is intended to receive the application. This permissions checking is described in the related patent application identified above. If there is no match of identification data, the application loading process ends. If the identification data does match, then the process continues.

ANNEX C TO THE DESCRIPTION

Step 907 uses the application providers public key which was recovered from the verified ALC to verify AU signature 205. When the ALU was generated by the application provider, the application unit 203 was signed with the application provider's secret key to authenticate that the application was provided by the correct application provider. The application provider then provides its public key to IC card through the ALC. The IC card then verifies the AU signature 205. If the two data blocks match, then the ALU is verified as being generated by the application provider. Because the application provider's public key is part of the ALC which is signed by the CA, the CA can make sure that the proper public key has been provided to the IC card. This unique key interaction between the application provider, CA and the intended IC card ensures that no counterfeit or unapproved applications or data are loaded onto an IC card which is part of the secure system.

Step 911 then processes a KTU authentication check which further verifies that only the intended card has received the application. The KTU authentication check makes sure that if a third party does somehow intercept the ALU, the third party cannot read the enciphered portions of the AU and cannot retrieve the keys to decrypt the AU. This step is further explained in Figure 10.

Figure 10 shows the steps of the KTU Authentication process. Step 1001, which is shown in dashed lines because it is preferably optional, checks the identification of the IC card a second time. The identification information can be sent as part of the KTU data. However, this check is optional as it has already been performed once in step 905.

Step 1003 then decrypts KTU ciphertext 503 using the IC card's secret key (mkd_sk). The KTU Plaintext was previously encrypted using the intended card's public key (mkd_pk). This means that only the holder of the intended card's secret key could decrypt the encrypted message. The application provider obtains the intended IC card's public key either from the IC card itself (See Figure 4 and related text for a discussion of the mkd key set) or from a database holding the public keys. If the IC card cannot decrypt the KTU ciphertext properly then the KTU is not meant for that card and the application loading process halts. If the IC card does properly decipher the KTU ciphertext, then the process continues.

Step 1005 identifies an encrypted area of the application unit (AU). In the example of the KTU Plaintext described in connection with Figure 6, the IC card uses a relative starting address and area length field to determine the encrypted portion. Step 1005 also identifies which encryption technique was used to encrypt the identified portion so that the proper decryption technique can be used. For example, the technique could by single or triple DES. Alternatively, the technique could be a default technique used in the system and need not be identified.

Step 1007 then retrieves the key from KTU Plaintext and decrypts the identified portion with the identified decryption technique. This allows the IC card to have the decrypted portion of the AU which it will store in its EEPROM once all the encrypted portions have been decrypted.

Step 1009 checks if there are any other additional encrypted areas. In the example described in Figure 3, there are three encrypted areas. The number

ANNEX C TO THE DESCRIPTION

of encrypted areas was a field in the example of Figure 6. However, the number of portions can be determined using other conventional means. If there are additional encrypted portions, the process jumps to step 1005. If there are no additional encrypted portions, then the process continues with step 1011.

Step 1011 then loads the decrypted AU into the memory of the IC card. The ALU has passed all of the authentication and decryption checks and the application can now properly reside on the IC card and be executed and used by the card user. While the different checks have been presented in a particular order in Figures 9 and 10, the checks can be performed in any order. While all of the described techniques used in conjunction with the ALU provide the best security, one or more of the individual techniques could be used for their individual purposes or combined with other conventional security techniques.

Figure 11 shows an example of a block diagram of an IC card chip upon which an ALU can be loaded and processed. An integrated circuit is located on an IC card for use. The IC card preferably includes a central processing unit 1101, a RAM 1103, an EEPROM 1105, a ROM 1107, a timer 1109, control logic 1111, an I/O port 1113 and security circuitry 1115, which are connected together by a conventional data bus.

Control logic 1111 in memory cards provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports. CPU 1101 with its control logic can perform calculations, access memory locations, modify memory contents, and manage input/output ports. Some cards have a coprocessor for handling complex computations like cryptographic operations. Input/output ports 1113 are used under the control of a CPU and control logic, for communications between the card and a card interface device. Timer 1109 (which generates or provides a clock pulse) drives the control logic 1111 and CPU 1101 through the sequence of steps that accomplish memory access, memory reading or writing, processing, and data communication. A timer may be used to provide application features such as call duration. Security circuitry 1115 includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed ("blown") upon completion of testing to prevent later access. The AU data after the ALU has been authenticated and verified is stored in EEPROM 1105. The IC card private key will be stored in a secure memory location. The IC card public key and public key certificate is preferably stored in EEPROM 1105. The authentication process as described herein is performed by the CPU 1101.

Figure 11 also shows a possible configuration for the application provider, transmitting entity and for the CA. CPU 1101 present in the application provider encrypts the necessary information using encryption techniques described herein and performs the necessary data operations. CPU 1101 present in the certification authority is used to sign the Application Load Certificate and the public key certificate as described herein.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

For example, while loading an application is discussed herein, the same secure loading processes can apply to transmitting other types of data such as data blocks, database files, word processing documents or any other type of data need to be transmitted in a secure manner.

ANNEX C TO THE DESCRIPTION

WE CLAIM:

1. A method for securely transporting data onto an integrated circuit card by using an individualized key set for said card, comprising the steps of:

storing a private key and public key pair unique to said integrated circuit card in said memory located on said integrated circuit card;

retrieving said stored public key from said integrated circuit card;

encrypting at least a portion of said data to be transported onto said card, using said retrieved public key;

transmitting said encrypted data to said integrated circuit card; and

decrypting said encrypted data using said integrated circuit card's private key to recover said transported data.

2. The method of claim 1, further including the step of storing said decrypted data on said integrated circuit card.

3. The method of claim 1, wherein a certification authority digitally signs said integrated circuit card's public key to produce a public key certificate unique to said card and stored thereon, and wherein said public key certificate is verified prior to said transmitting step.

4. The method of claim 3, wherein said public key certificate is verified with said certification authority's stored public key prior to said transmitting steps.

5. The method of claim 4, wherein said retrieved public key certificate is recovered and compared with said stored public key.

6. The method of claim 5, wherein said integrated circuit card's public and private keys are provided using an asymmetric technique.

7. The method of claim 6, wherein said asymmetric technique is RSA.

8. A method performed by an integrated circuit card for processing incoming data transmission to said integrated circuit card by using an individualized key set for the card, comprising the steps of:

receiving said data transmission comprising data encrypted with a public key stored on said integrated circuit card, said public key forming part of said individualized key set;

retrieving a unique private key for said integrated circuit card which is part of said individualized key set; and

decrypting said encrypted data with said unique private key to recover said data.

ANNEX C TO THE DESCRIPTION

9. The method of claim 8, further including the step of storing said decrypted data on said integrated circuit card.

10. The method of claim 8, wherein said individualized key set is generated by asymmetric encryption.

11. The method of claim 8, wherein a certification authority digitally signs said integrated circuit card's public key to produce a public key certificate unique to said card and stored thereon, and wherein said public key certificate is verified prior to said transmitting step.

12. The method of claim 11, wherein said public key certificate is retrieved prior to said transmitting steps.

13. The method of claim 12, wherein said retrieved public key certificate is verified using said certification authority's stored public key.

14. An apparatus located on an integrated circuit card by using an individualized key set for said card for processing an incoming secure data transmission comprising:

means for receiving said data transmission comprising data encrypted with a public key stored on said integrated circuit card, said public key forming part of said individualized key set;

ANNEX C TO THE DESCRIPTION

means for retrieving a unique public key for said integrated circuit card which is part of said individualized key set; and

means for decrypting said encrypted data with said unique private key to recover said data.

15. The apparatus of claim 14, further comprising means for storing said data on said integrated circuit card.

16. The apparatus of claim 14, further including means for retrieving a public key certificate which is generated by a certificate authority digitally signing said unique public key.

17. The apparatus of claim 16, further including means for transmitting said public key certificate prior to said receiving means receiving.

18. The apparatus of claim 17, wherein said transmitted public key certificate is verified using said certification authority's stored public key.

19. A method of securely transporting data onto an integrated circuit card by using an individualized key set for the card, comprising the steps of:

providing a first unique private and public key pair for a certification authority;

storing a second unique private and public key pair which

form said individualized key set for said integrated circuit card in a memory located on said integrated circuit card;

encrypting said second public key with said first certification authority's private key to form a public key certificate;

storing said public key certificate on said integrated circuit card;

retrieving said stored public key certificate from said integrated circuit card;

verifying said public key certificate with said first public key to ensure that said public key certificate is valid;

encrypting at least a portion of said data using said retrieved second public key;

transporting said encrypted data to said integrated circuit card; and

decrypting said encrypted data using said second private key to retrieve said data.

20. The method of claim 19, wherein said data comprises an application.

ANNEX C TO THE DESCRIPTION

ABSTRACT OF THE DISCLOSURE

Method and apparatus for securely transporting data onto an IC card. The method is used, for example, to transport data, including application programs, in a secure manner from a source located outside the IC card. At least a portion of the data is encrypted using the public key of a public/secret key pair of the intended IC card unit. The encrypted data is then sent to the IC card and the IC card verifies the key transformation unit using its unique secret key. The data can then be stored on the IC card. A copy of the public key signed by a certification authority can be used to verify that the card is authorized to be part of the overall authorized system.

CLAIMS

I CLAIM:

1. A method of loading an application copy onto an integrated circuit card, wherein said application copy is one of a plurality of copies of an application, said application copy having an associated application identifier that uniquely identifies said application from other applications and an application copy number that is unique for each copy of said application, said integrated circuit card comprising a microprocessor and memory coupled to said microprocessor, said memory comprising an application history list area for storing application identifiers and application copy numbers of applications that have been previously loaded onto said integrated circuit card, said method comprising:

receiving by said integrated circuit card said application copy, said application identifier, and said application copy number;

determining by said integrated circuit card whether said application identifier and said application copy number are contained in said application history list area; and

failing to load said application copy by said integrated circuit card if said application identifier and said application copy number are contained in said application history list area.

2. The method of claim 1, further comprising the steps of:

allocating a predetermined portion of said memory for said application history list area;

determining by said integrated circuit card whether said application history list area is full; and

failing to load said application copy if said application history list is full.

3. The method of claim 1 or claim 2, further comprising the step of:

adding said application identifier and said application copy number to said application history list area if said application identifier and said application copy number are not contained in said application history list area.

4. The method of claim 1 or claim 2, further including the step of:

adding said application identifier and said application copy number to said application history list area if said application identifier and said application copy number are not contained in said application history list area and said application copy number is not zero.

5. The method of any preceding claim, wherein said application copy comprises application code and application data and a portion of said application data comprises units of value that may be exchanged for goods or services.

6. The method of any preceding claim, wherein said application copy comprises application code and application data and wherein said application identifier and said application copy number are contained in said application data.

7. The method of any preceding claim, further comprising the step of:

transmitting said application copy, said application identifier, and said application copy number to said integrated circuit card by an application provider.

8. The method of claim 7, further comprising the step of:

encrypting by said application provider at least a portion of said application copy before transmitting said application copy to said integrated circuit card.

9. The method of claim 8, further comprising the step of:

transmitting by said application provider a key transformation unit comprising information relating to the encryption of said portion of said application copy.

10. The method of claim 9, wherein said integrated circuit card has a first public key pair, and further comprising the step of:

encrypting said key transformation unit by said application provider with the public key of said first public key pair before transmitting said key transformation unit to said integrated circuit card.

11. The method of claim 10, further comprising the steps of:

decrypting by said integrated circuit card said encrypted key transformation unit with the secret key of said first public key pair; and

decrypting said application copy using the information contained in said decrypted key transformation unit.

12. The method of claim 7 or any claim dependent thereon, wherein said application provider has a second public key pair, and further comprising the steps of:

forming a signed application copy by said application provider by encrypting said application copy with the secret key of said second public key pair; and

transmitting by said application provider said signed application copy to said integrated circuit card.

13. The method of claim 12, further comprising the steps of:

registering the public key of said second public key pair with a certification authority, which has a third public key pair.

providing a certificate by said certification authority to said application provider by encrypting the public key of said second public key pair with the secret key of said third public key pair; and

transmitting said certificate by said application provider to said integrated circuit card.

14. The method of claim 13, further comprising the steps of:

obtaining the public key of said second key pair by said integrated circuit card by decrypting said certificate using the public key of said third public key pair;

verifying by said integrated circuit card said signed application copy using the public key of said second public key pair;

failing to load said application copy by said integrated circuit card if said signed application copy is not verified.

15. An integrated circuit card, comprising:

a microprocessor;

a memory coupled to said microprocessor, said memory including an application history list area for storing application identifiers and application copy numbers, each application identifier and each application copy number being associated with an application copy, said application copy being one of a plurality of copies of an application, each application identifier uniquely identifying an application from other applications, and each application copy number uniquely identifying an application copy from other application copies;

means for determining whether an application identifier and an application copy number associated with an application copy to be loaded into said memory area are contained in said application history list area; and

means for failing to load said application copy to be loaded if said associated application identifier and said associated application copy number are contained in said application history list area.

16. The integrated circuit card of claim 15, wherein said application history list area is an allocated, predetermined portion of said memory, and further comprising:

means for determining whether said application history list area is full; and

means for failing to load said application copy to be loaded if said application history list area is full.

17. The integrated circuit card of claim 15 or claim 16, further comprising means for adding said associated application identifier and said associated application copy number of said application copy to be loaded into said application history list area if said application identifier and said application copy number are not contained in said application history list area.

18. The integrated circuit card of any of claims 15 to 17, further comprising means for adding said associated application identifier and said associated application copy number of said application copy to be loaded into said application history list area if said application identifier and said application copy number are not contained in said application history list area and said application copy number is not zero.

1/29

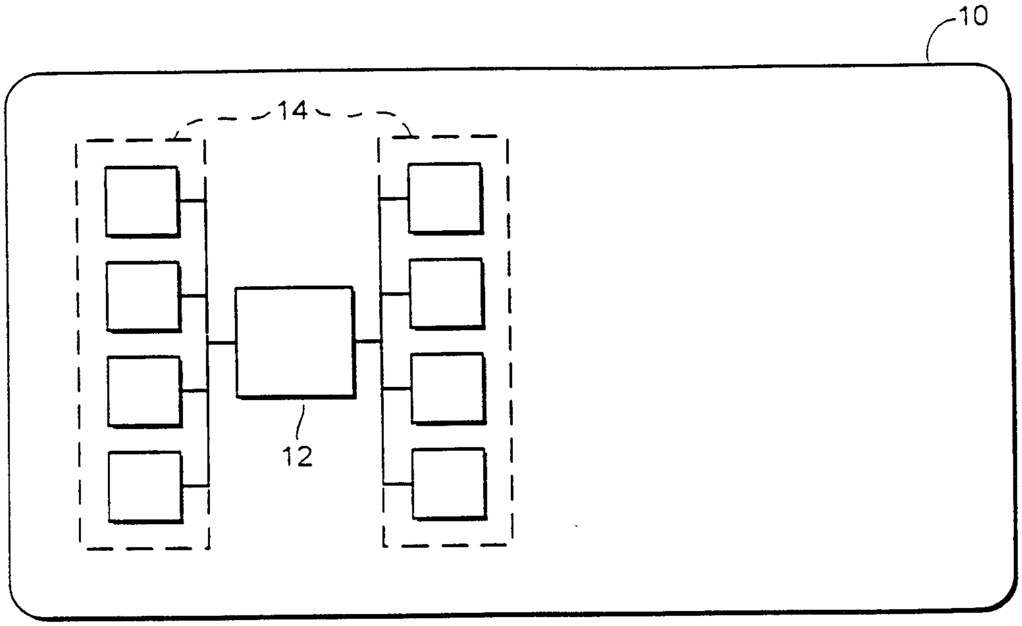


FIG. 1

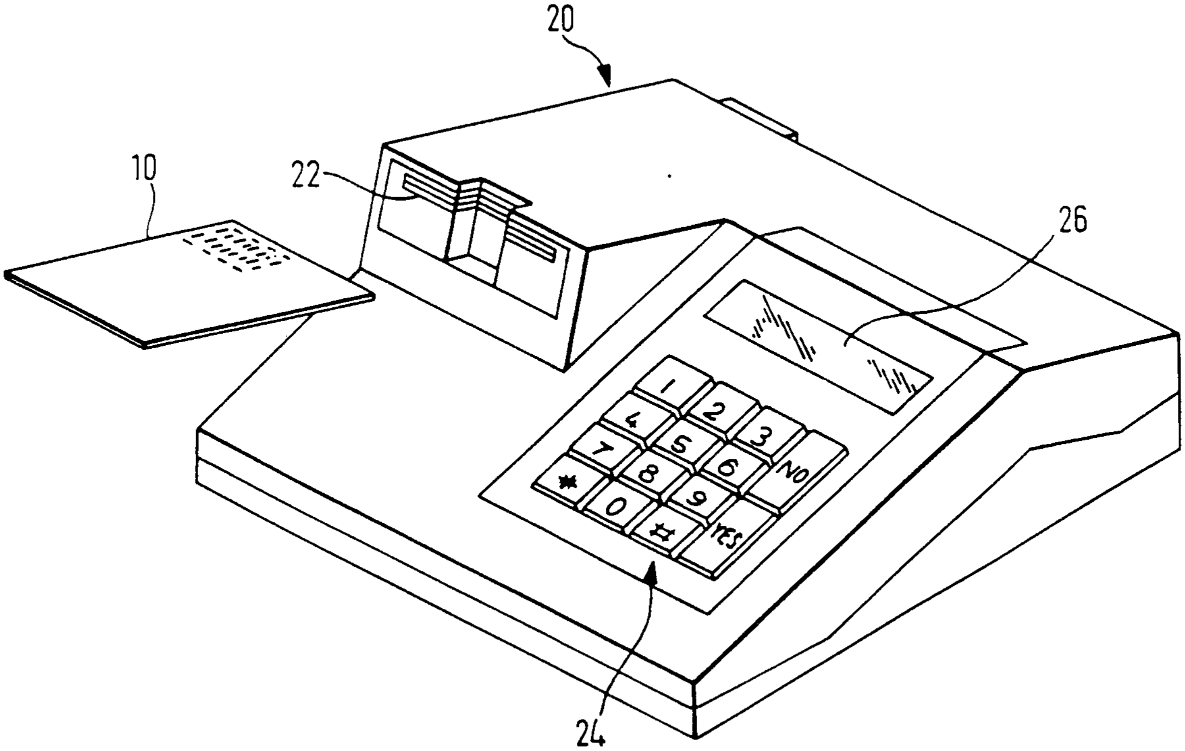


FIG. 2

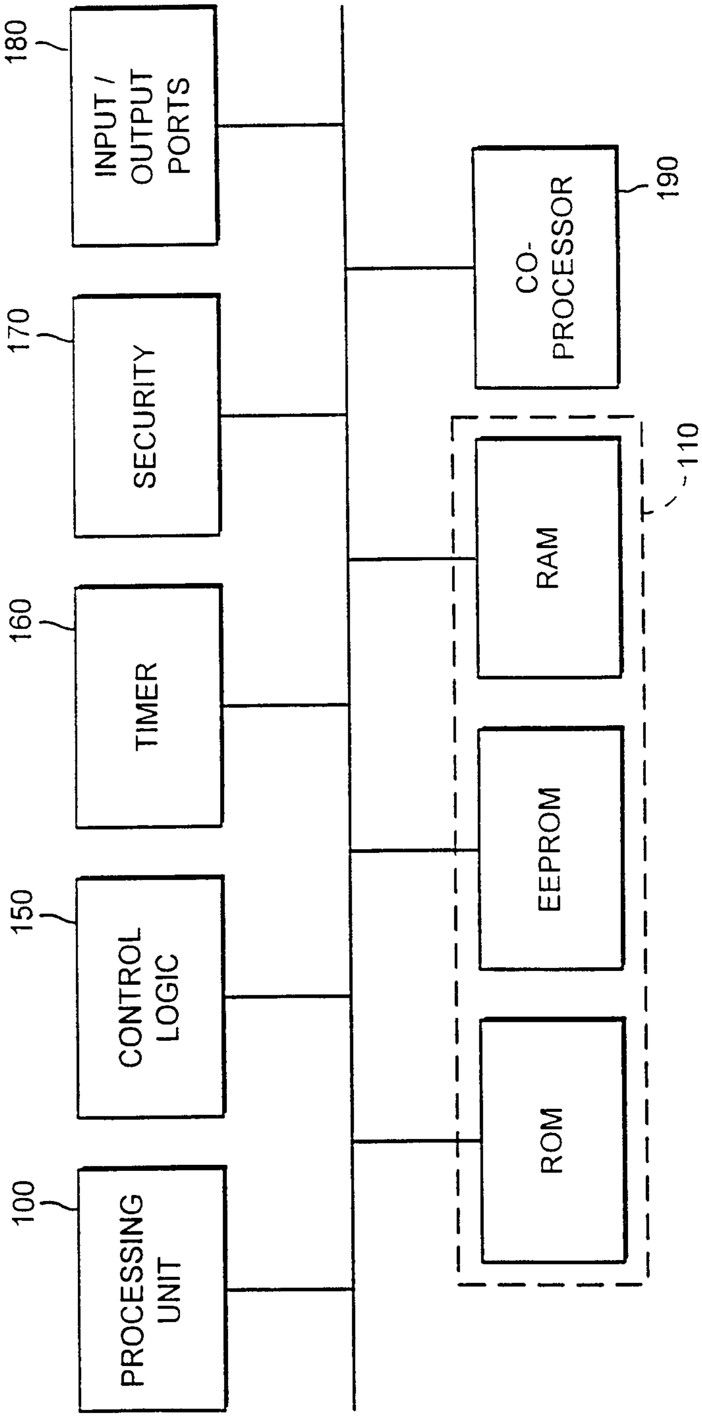

PROCESSING UNIT
100
CONTROL LOGIC
150
TIMER
160
SECURITY
170
INPUT / OUTPUT PORTS
180
ROM
EEPROM
RAM
110
CO-PROCESSOR
190

FIG. 3

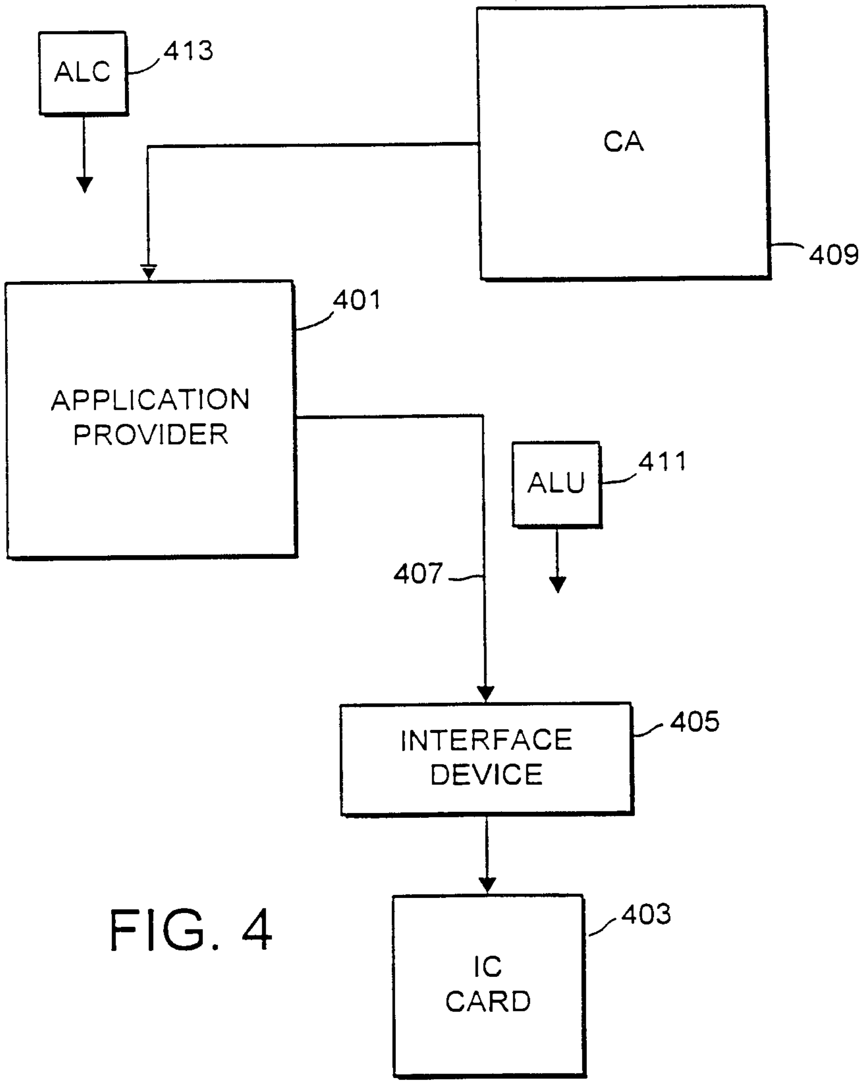

ALC
413
CA
409
APPLICATION PROVIDER
401
ALU
411
407
INTERFACE DEVICE
405
IC CARD
403

FIG. 4

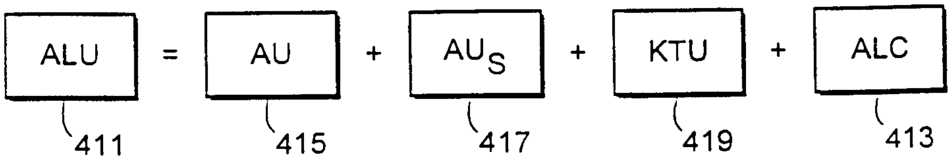

ALU
=
AU
+
AU$_S$
+
KTU
+
ALC
411
415
417
419
413

FIG. 5

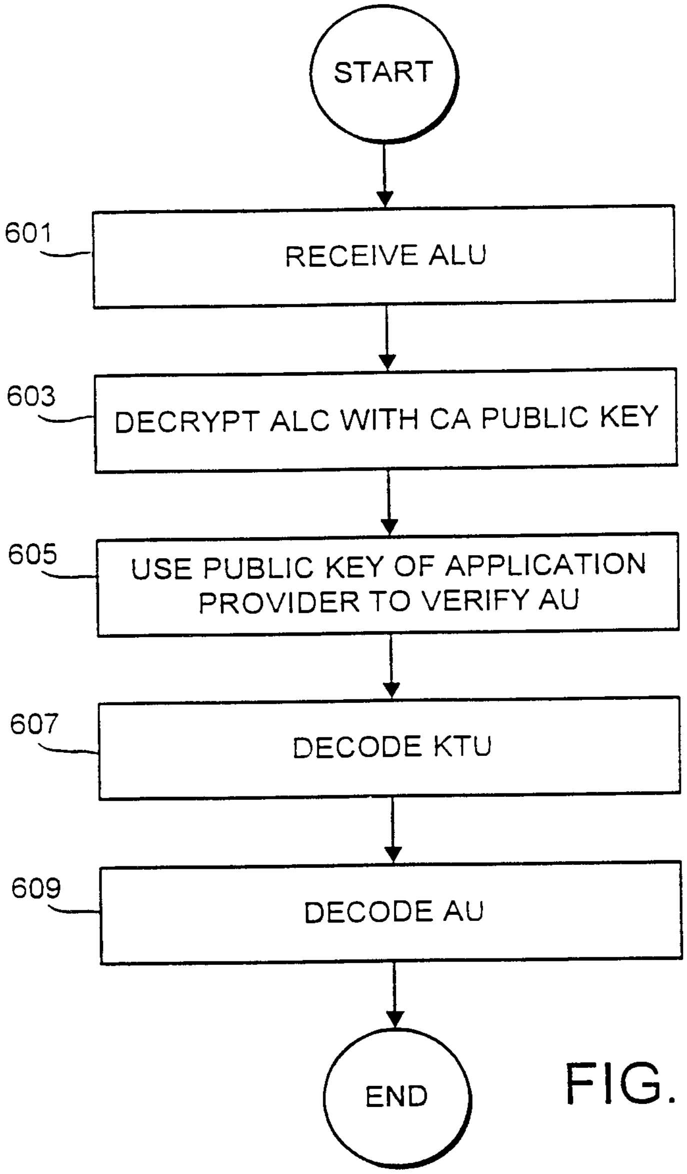

START
601
RECEIVE ALU
603
DECRYPT ALC WITH CA PUBLIC KEY
605
USE PUBLIC KEY OF APPLICATION PROVIDER TO VERIFY AU
607
DECODE KTU
609
DECODE AU
END

FIG. 6

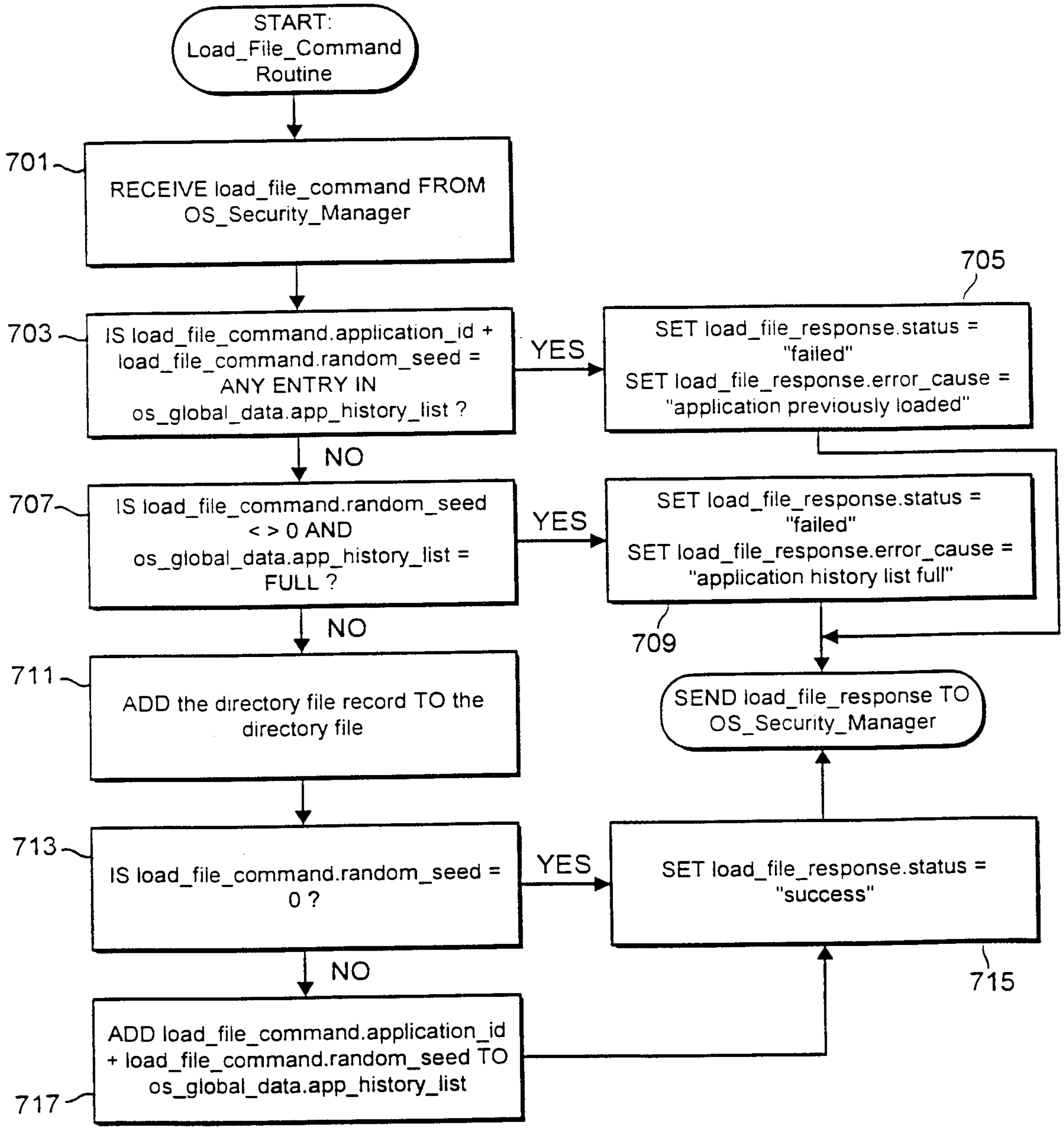

START:
Load_File_Command
Routine
701
RECEIVE load_file_command FROM
OS_Security_Manager
703
IS load_file_command.application_id +
load_file_command.random_seed =
ANY ENTRY IN
os_global_data.app_history_list ?
YES
705
SET load_file_response.status =
"failed"
SET load_file_response.error_cause =
"application previously loaded"
NO
707
IS load_file_command.random_seed
< > 0 AND
os_global_data.app_history_list =
FULL ?
YES
SET load_file_response.status =
"failed"
SET load_file_response.error_cause =
"application history list full"
709
NO
711
ADD the directory file record TO the
directory file
SEND load_file_response TO
OS_Security_Manager
713
IS load_file_command.random_seed =
0 ?
YES
SET load_file_response.status =
"success"
715
NO
ADD load_file_command.application_id
+ load_file_command.random_seed TO
os_global_data.app_history_list
717

FIG. 7

ANNEX A TO THE DRAWINGS

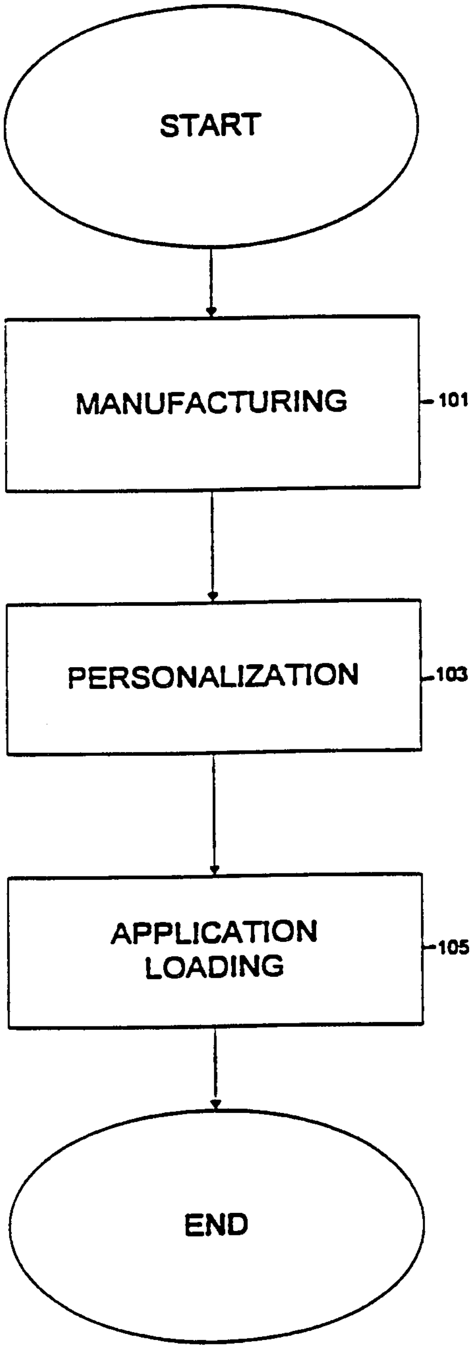


FIG. 1

ANNEX A TO THE DRAWINGS

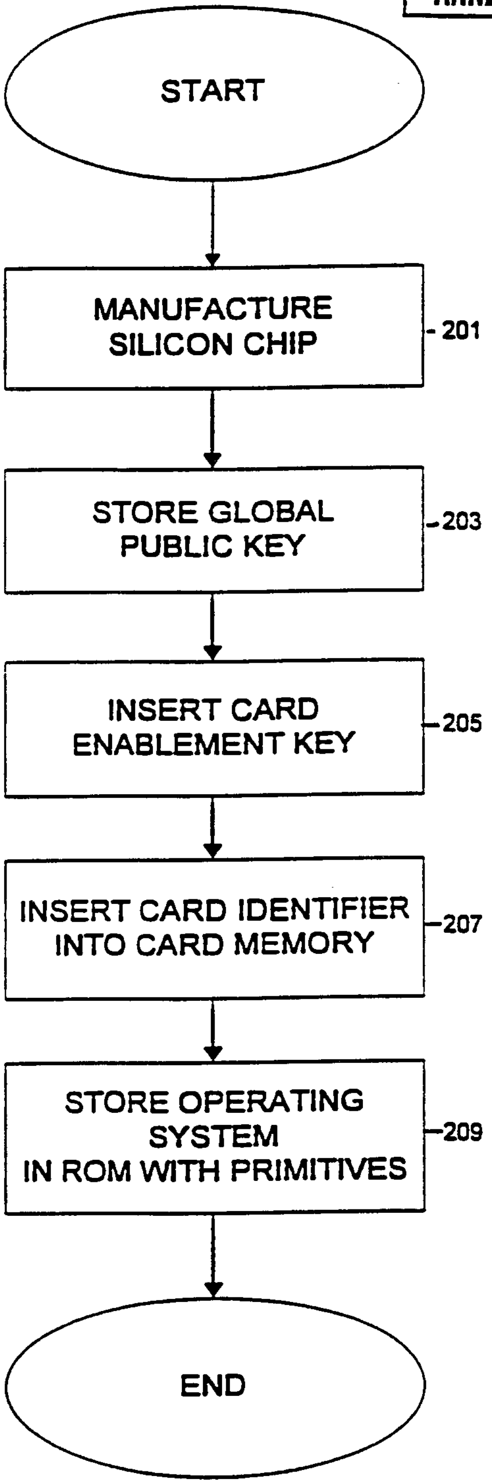

START
MANUFACTURE SILICON CHIP
201
STORE GLOBAL PUBLIC KEY
203
INSERT CARD ENABLEMENT KEY
205
INSERT CARD IDENTIFIER INTO CARD MEMORY
207
STORE OPERATING SYSTEM IN ROM WITH PRIMITIVES
209
END

FIG. 2

ANNEX A TO THE DRAWINGS

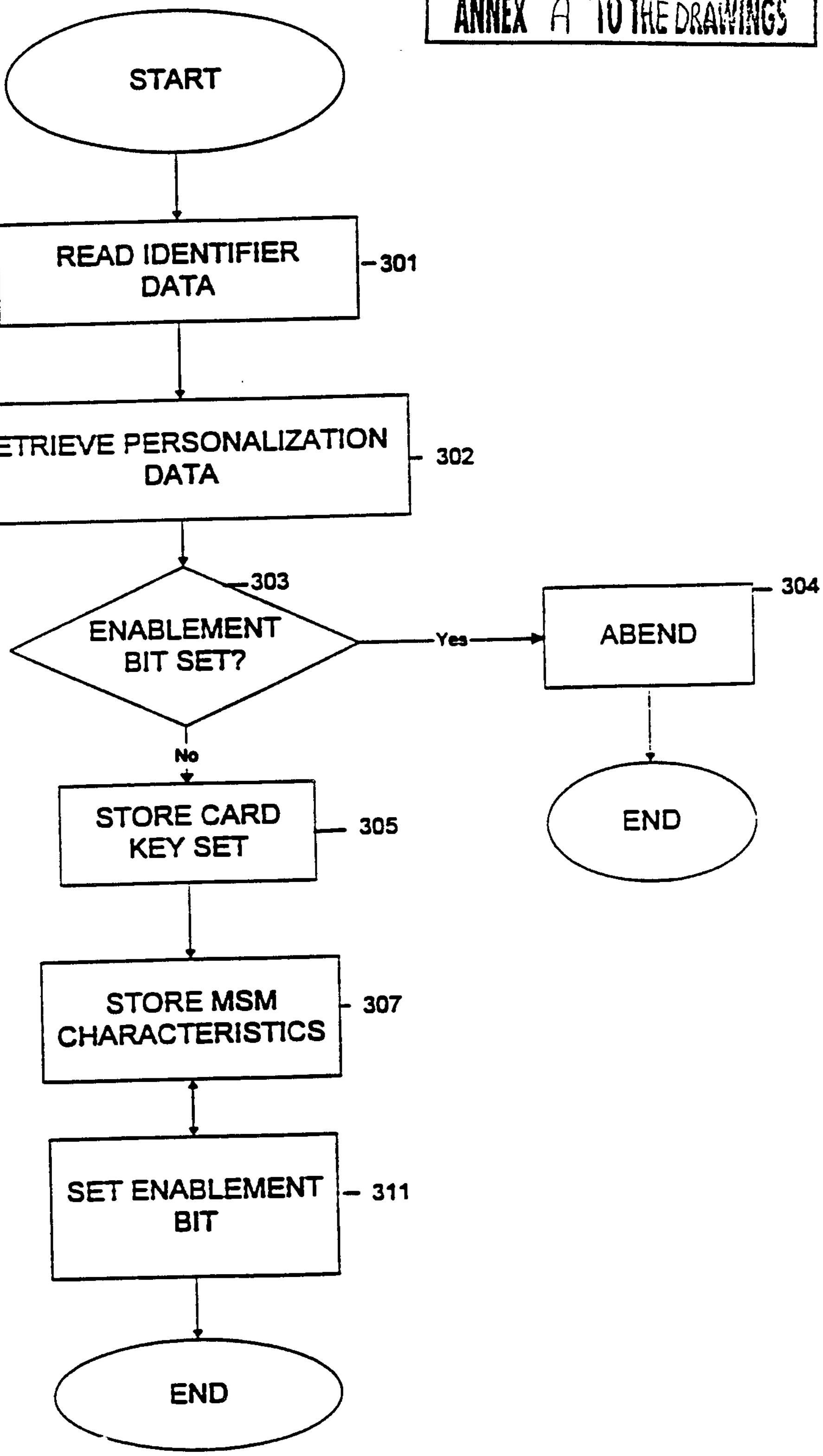


FIG. 3

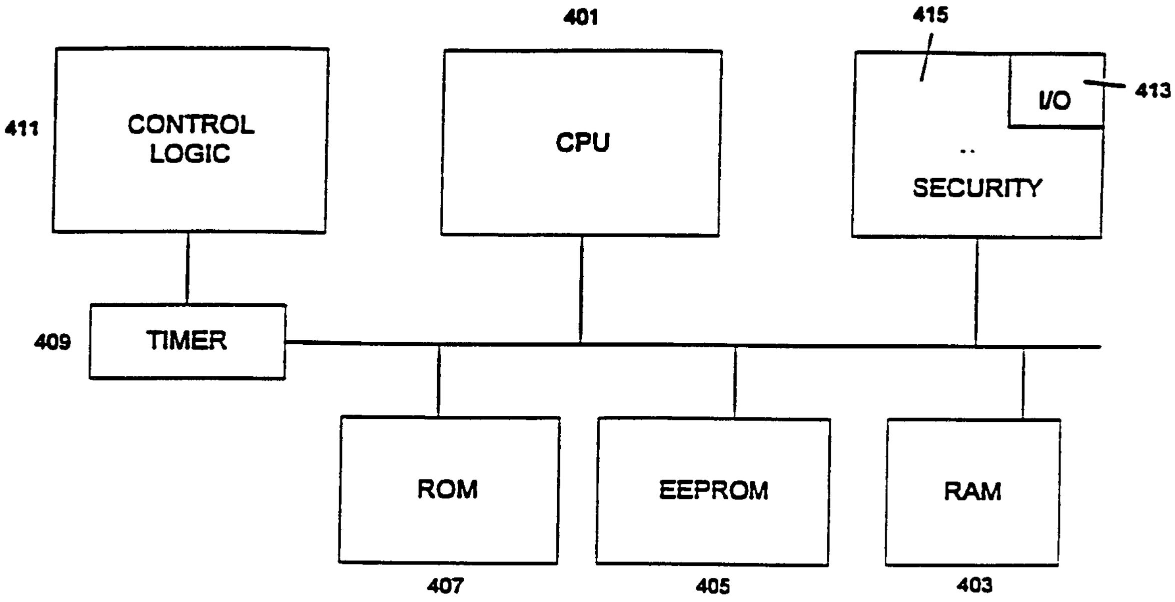

401
415
413
411
CONTROL LOGIC
CPU
I/O
SECURITY
409
TIMER
ROM
EEPROM
RAM
407
405
403

FIG. 4

ANNEX A TO THE DRAWINGS

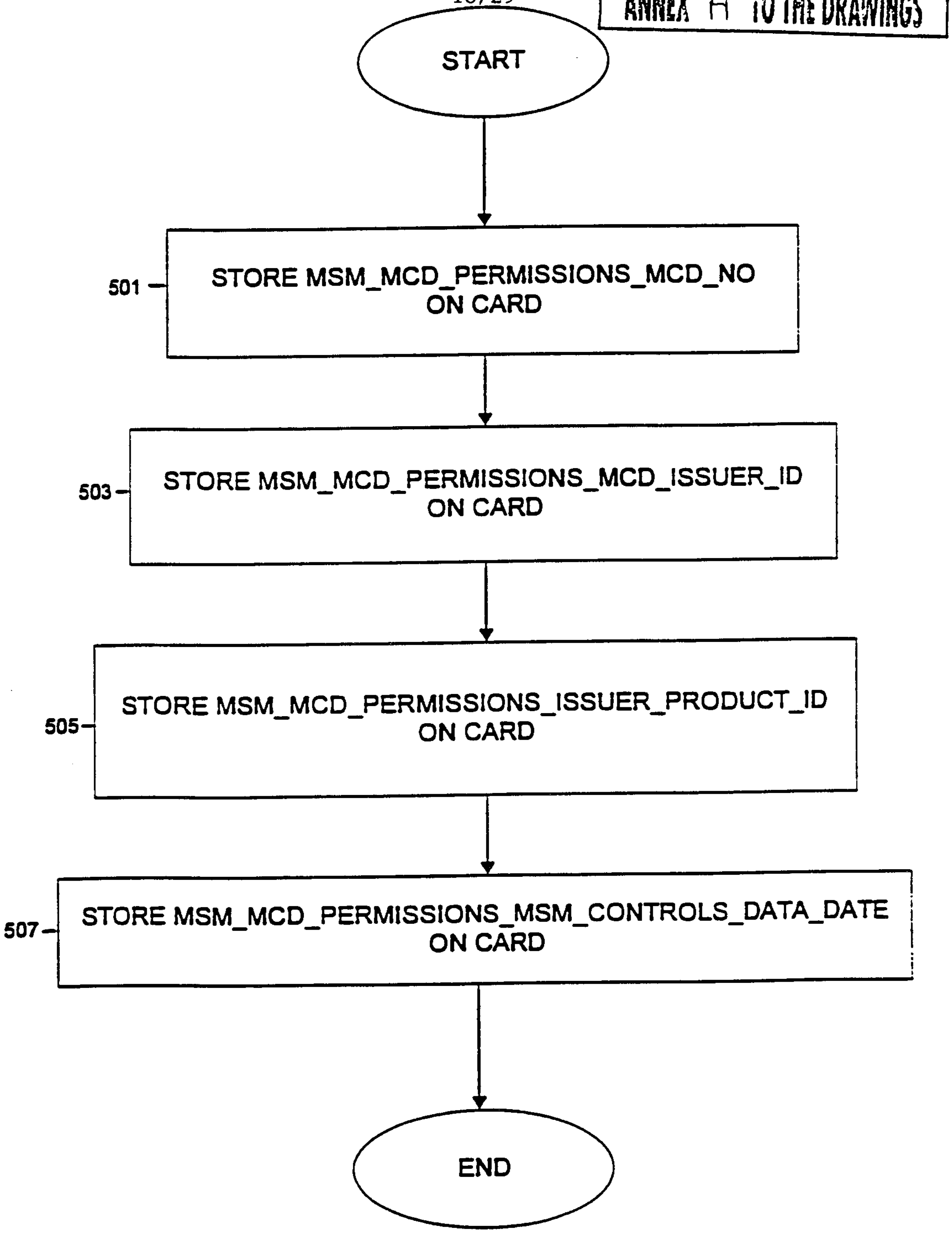

START
501
STORE MSM_MCD_PERMISSIONS_MCD_NO ON CARD
503
STORE MSM_MCD_PERMISSIONS_MCD_ISSUER_ID ON CARD
505
STORE MSM_MCD_PERMISSIONS_ISSUER_PRODUCT_ID ON CARD
507
STORE MSM_MCD_PERMISSIONS_MSM_CONTROLS_DATA_DATE ON CARD
END

FIG..5

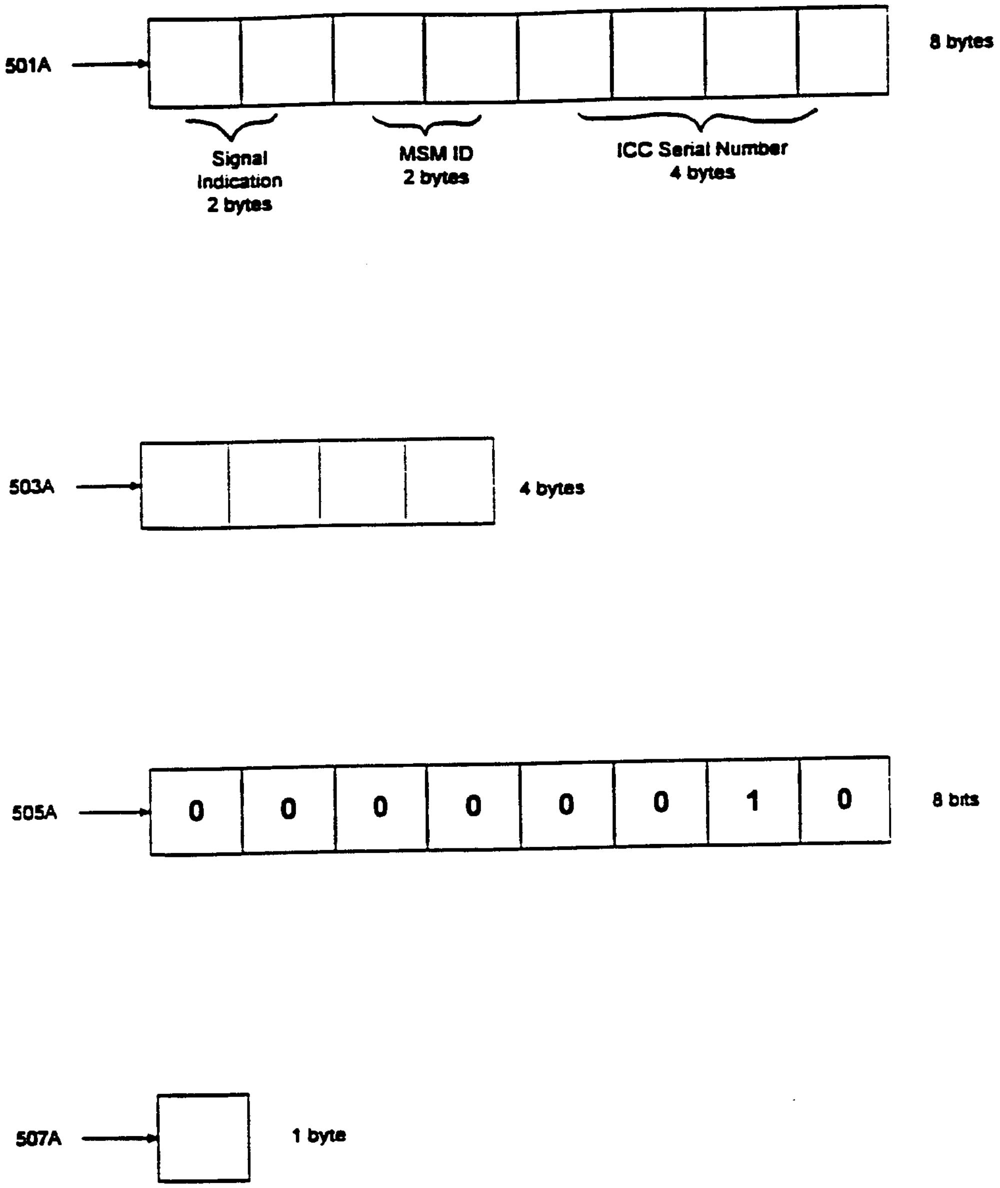

501A
8 bytes
Signal Indication 2 bytes
MSM ID 2 bytes
ICC Serial Number 4 bytes
503A
4 bytes
505A
0 0 0 0 0 0 1 0
8 bits
507A
1 byte

FIG. 5A

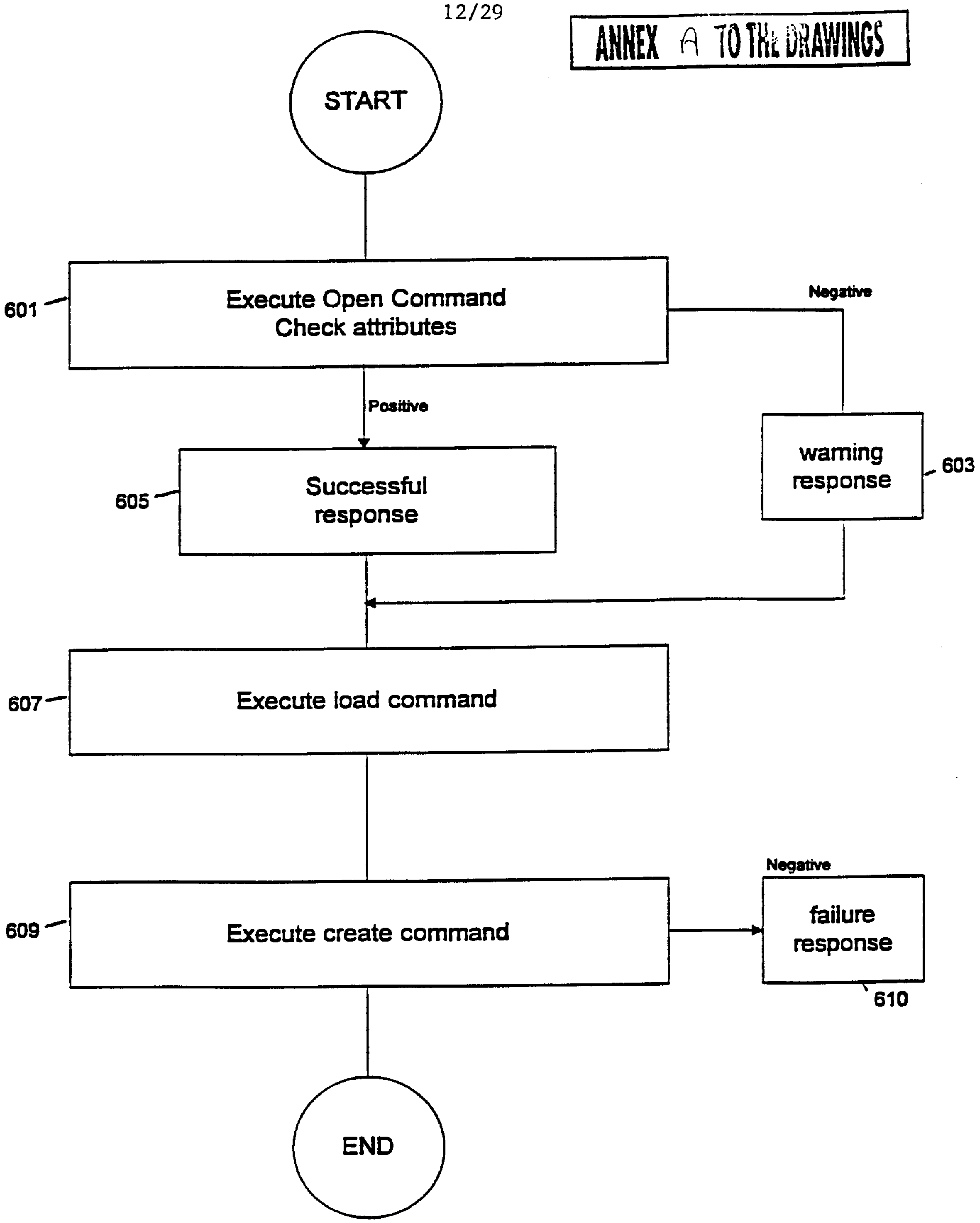

START
601
Execute Open Command
Check attributes
Negative
Positive
605
Successful
response
warning
response
603
607
Execute load command
609
Execute create command
Negative
failure
response
610
END

FIG. 6

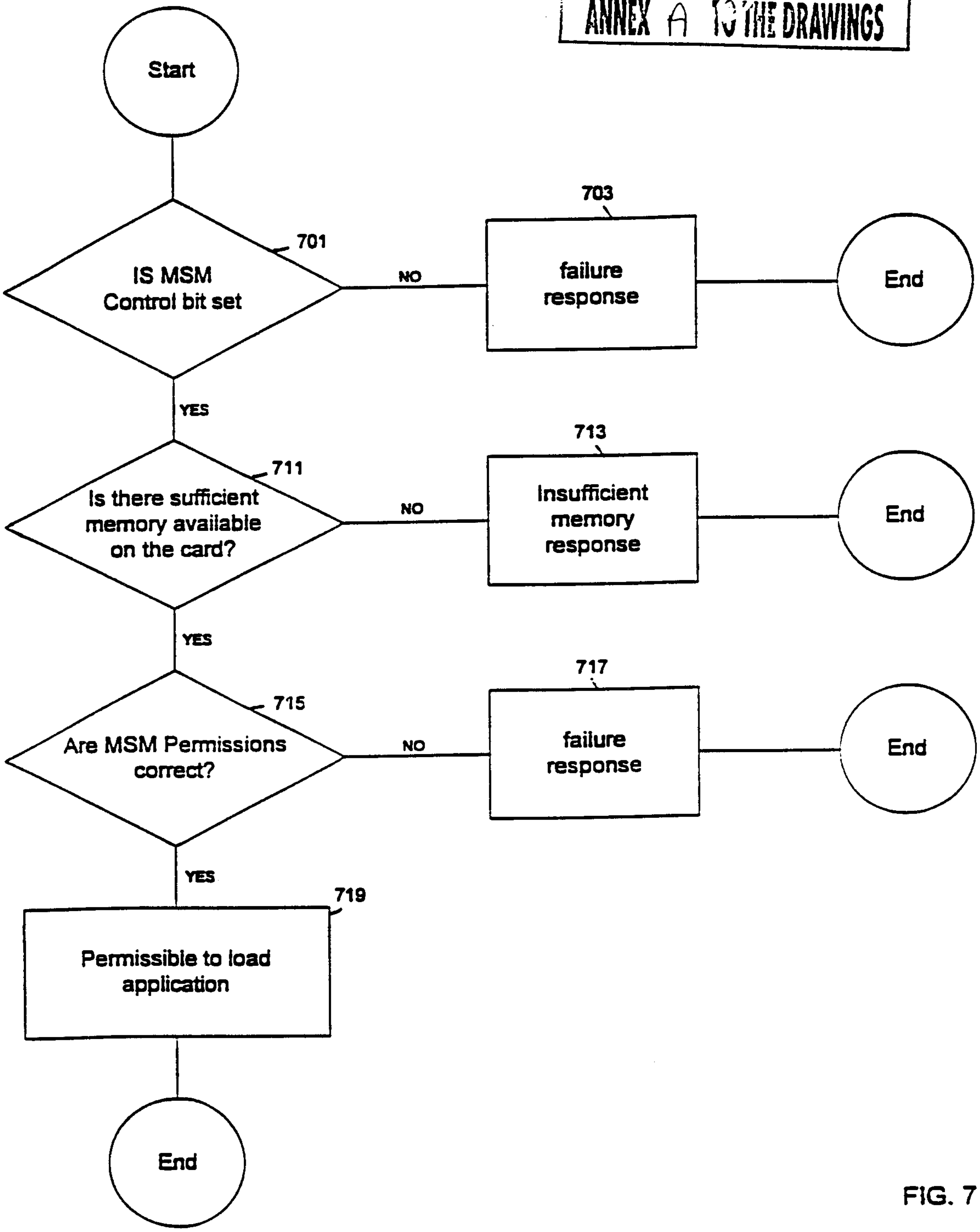

Start
IS MSM Control bit set
701
NO
703
failure response
End
YES
Is there sufficient memory available on the card?
711
NO
713
Insufficient memory response
End
YES
Are MSM Permissions correct?
715
NO
717
failure response
End
YES
719
Permissible to load application
End

FIG. 7

ANNEX A TO THE DRAWINGS

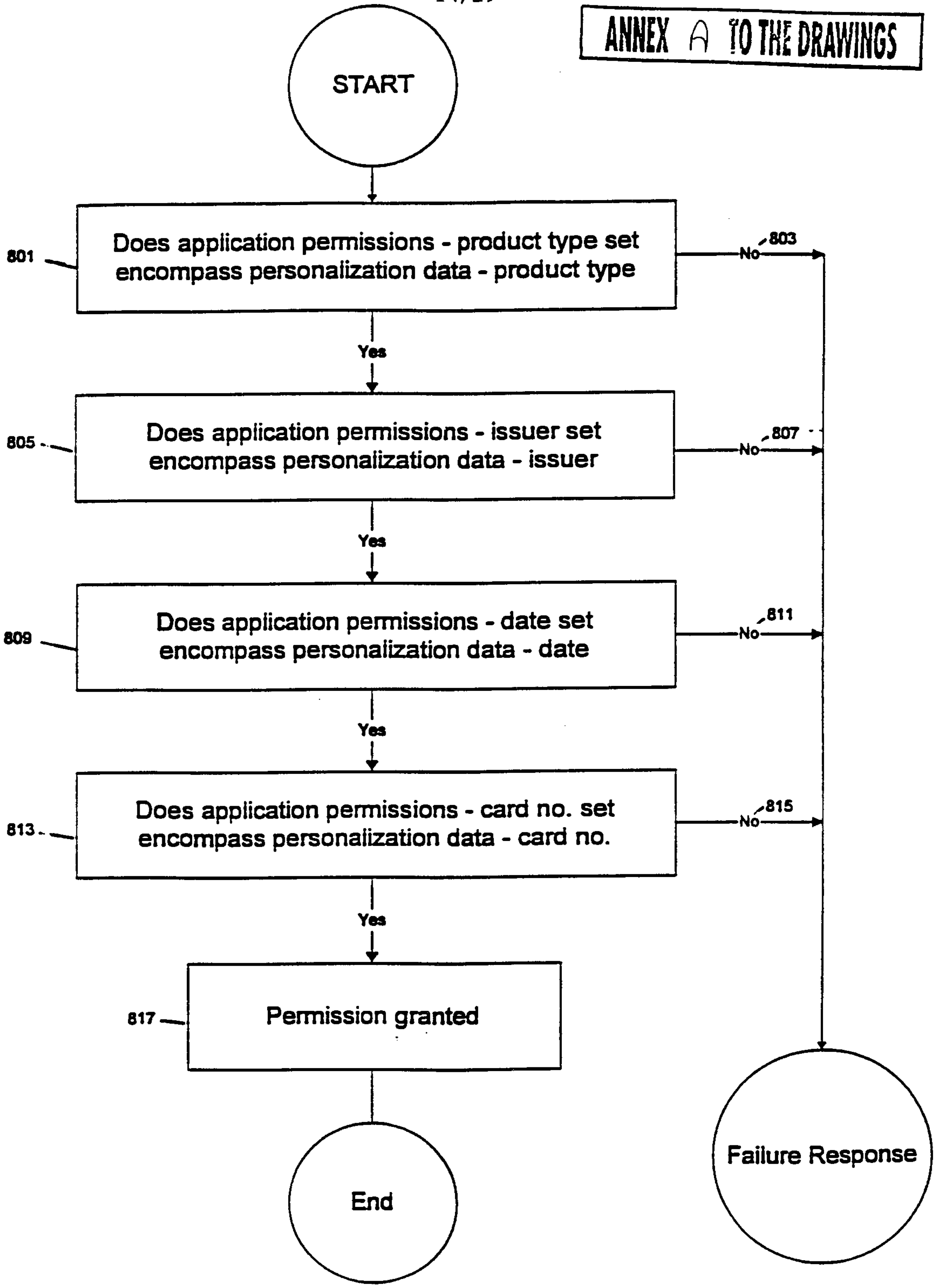

START
801
Does application permissions - product type set encompass personalization data - product type
No
803
Yes
805
Does application permissions - issuer set encompass personalization data - issuer
No
807
Yes
809
Does application permissions - date set encompass personalization data - date
No
811
Yes
813
Does application permissions - card no. set encompass personalization data - card no.
No
815
Yes
817
Permission granted
End
Failure Response

FIG. 8

ANNEX A TO THE DRAWINGS

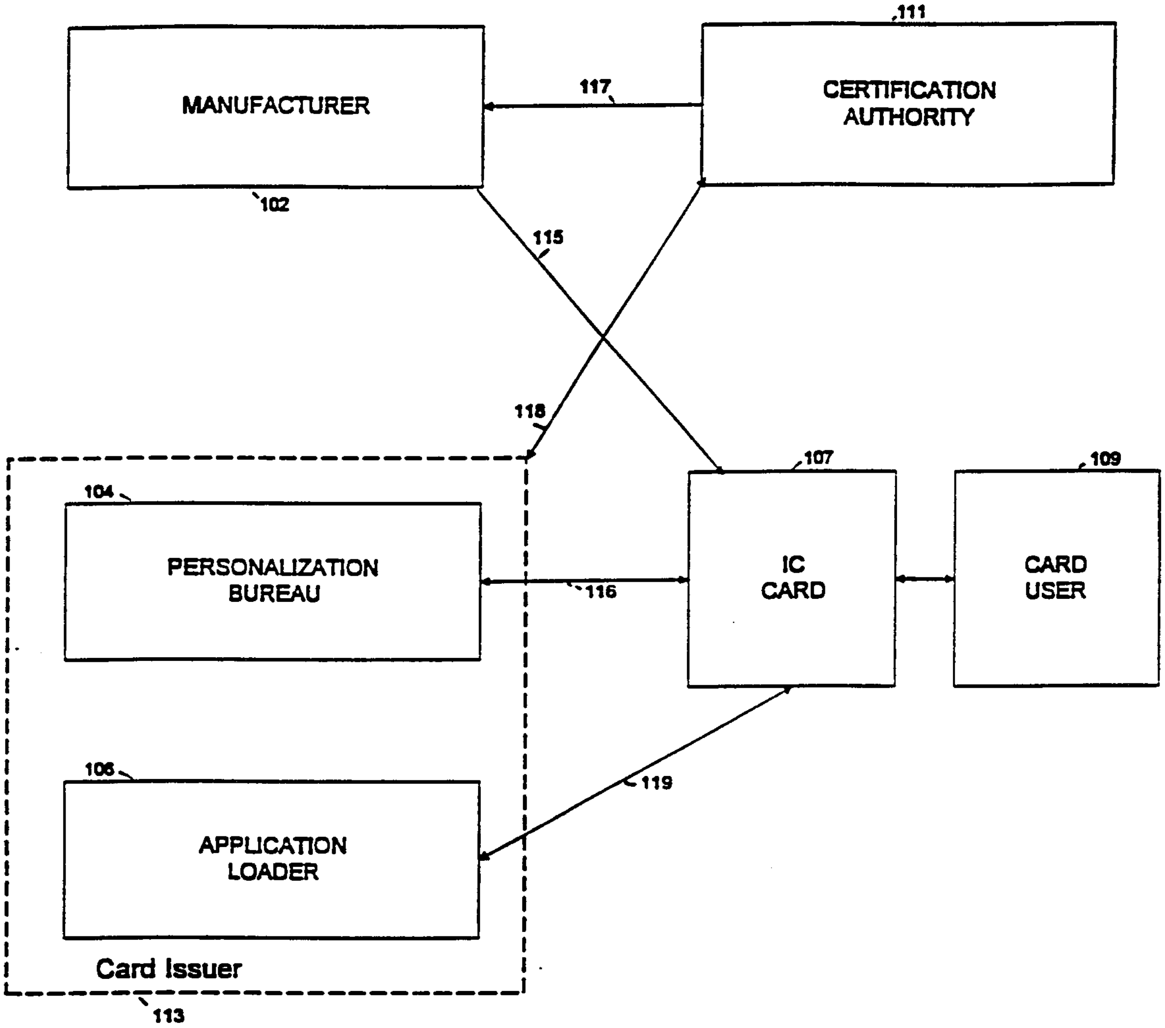


FIG. 9

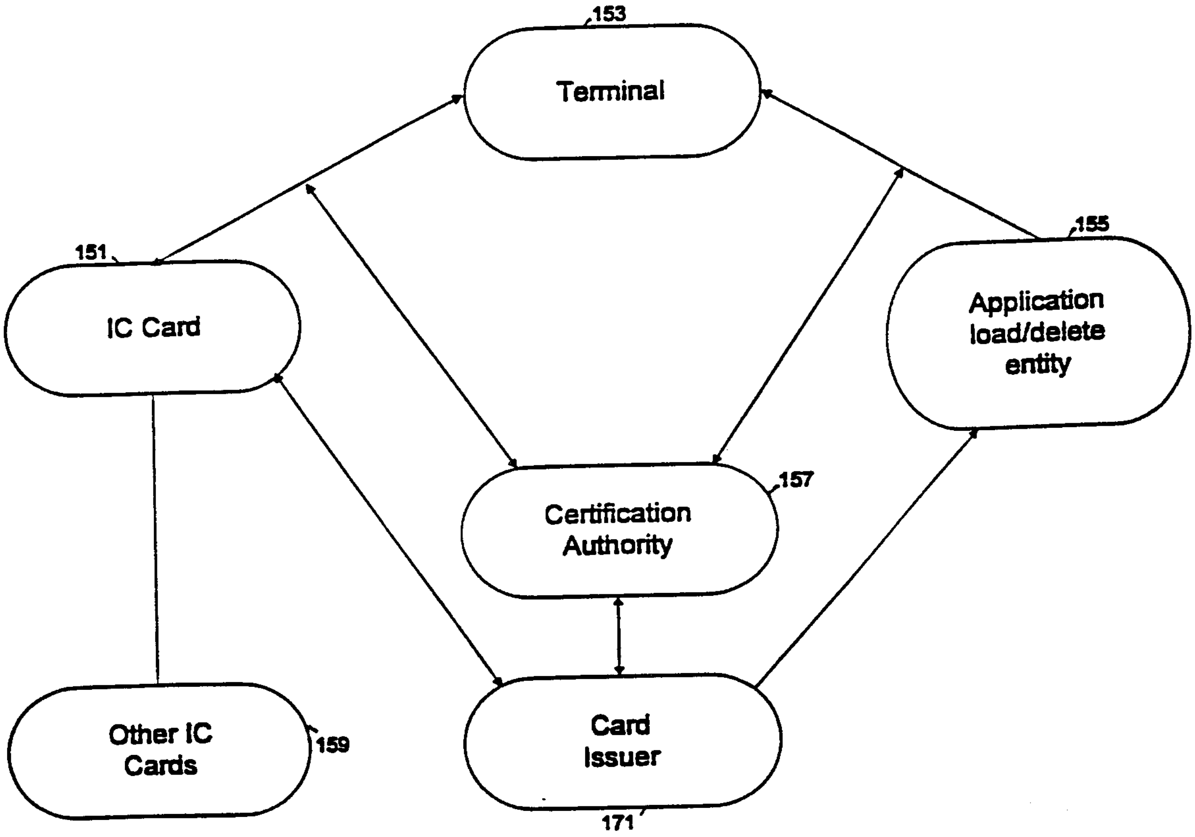

153
Terminal
151
IC Card
155
Application load/delete entity
157
Certification Authority
Other IC Cards
159
Card Issuer
171

FIG. 10

ANNEX B TO THE DRAWINGS

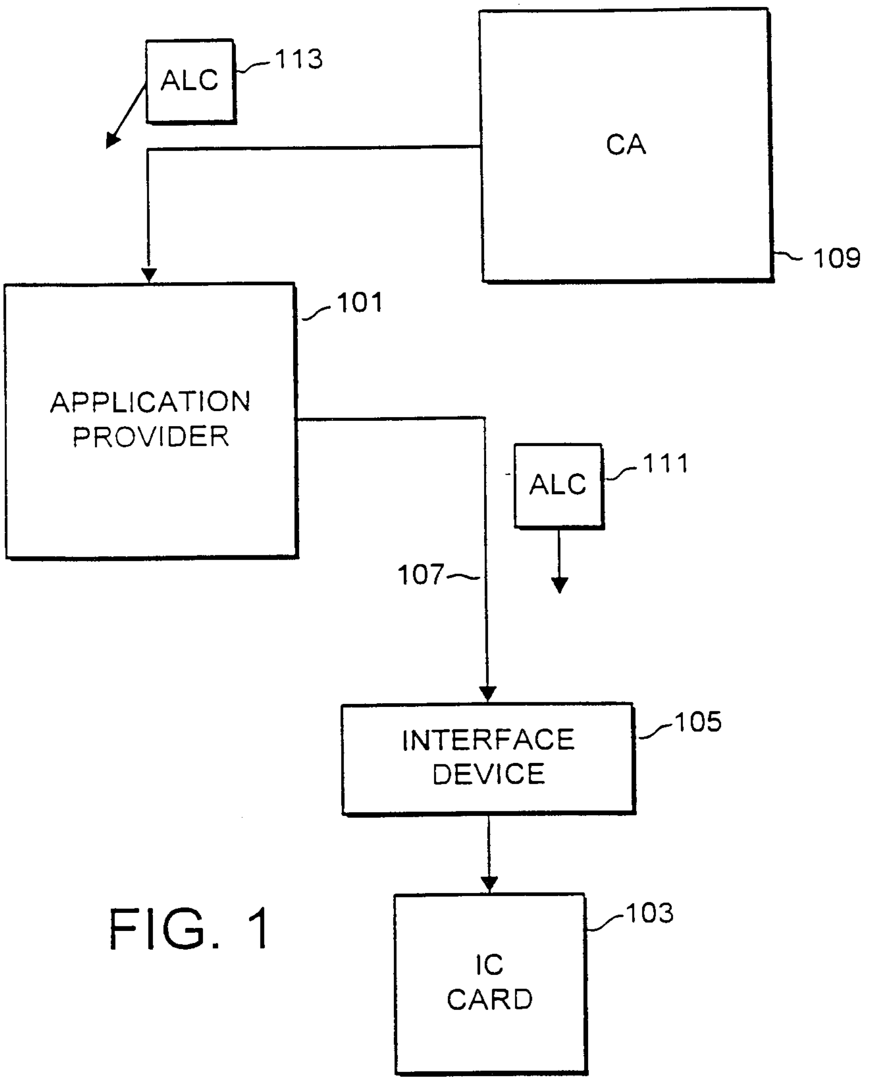


FIG. 1

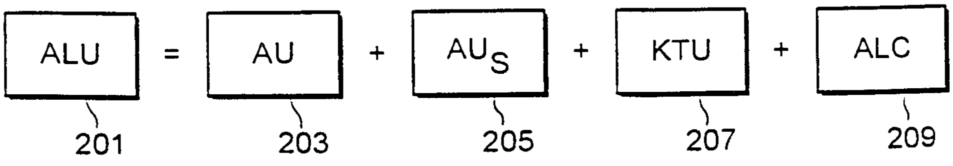


FIG. 2

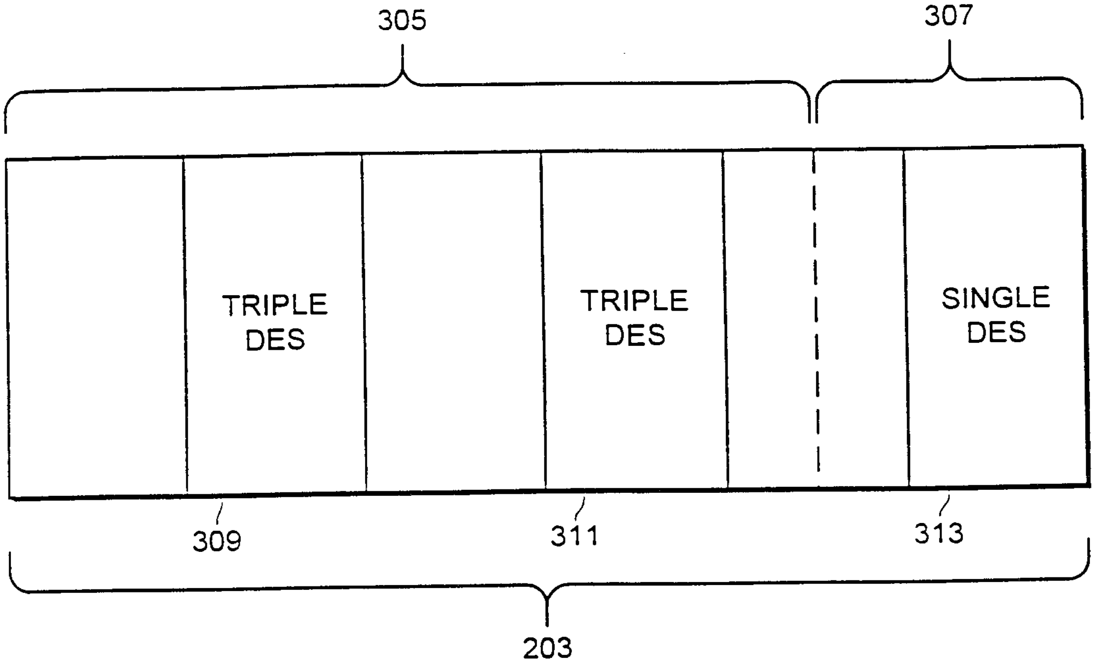

305
307
TRIPLE DES
TRIPLE DES
SINGLE DES
309
311
313
203

FIG. 3

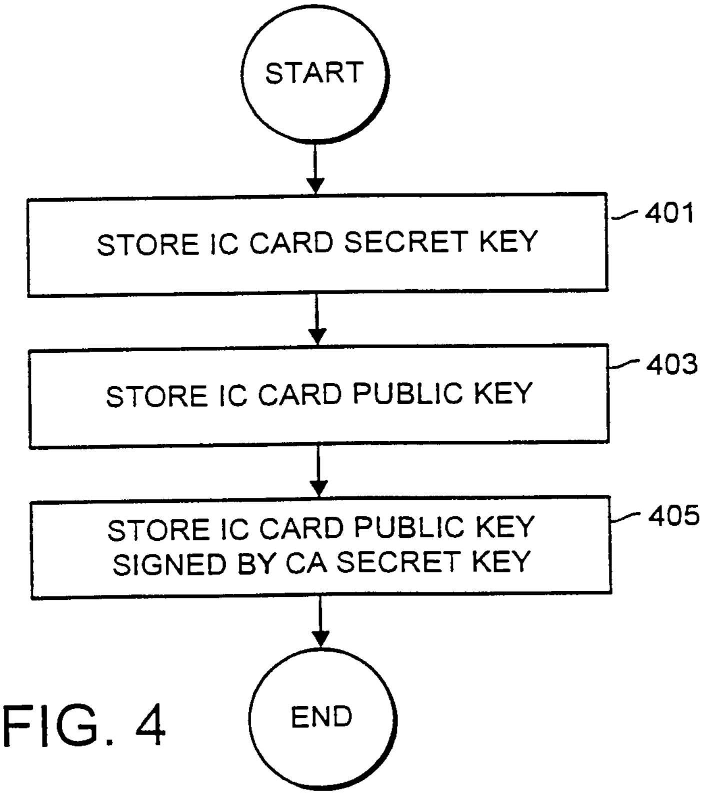

START
STORE IC CARD SECRET KEY
401
STORE IC CARD PUBLIC KEY
403
STORE IC CARD PUBLIC KEY SIGNED BY CA SECRET KEY
405
END

FIG. 4

ANNEX B TO THE DRAWINGS

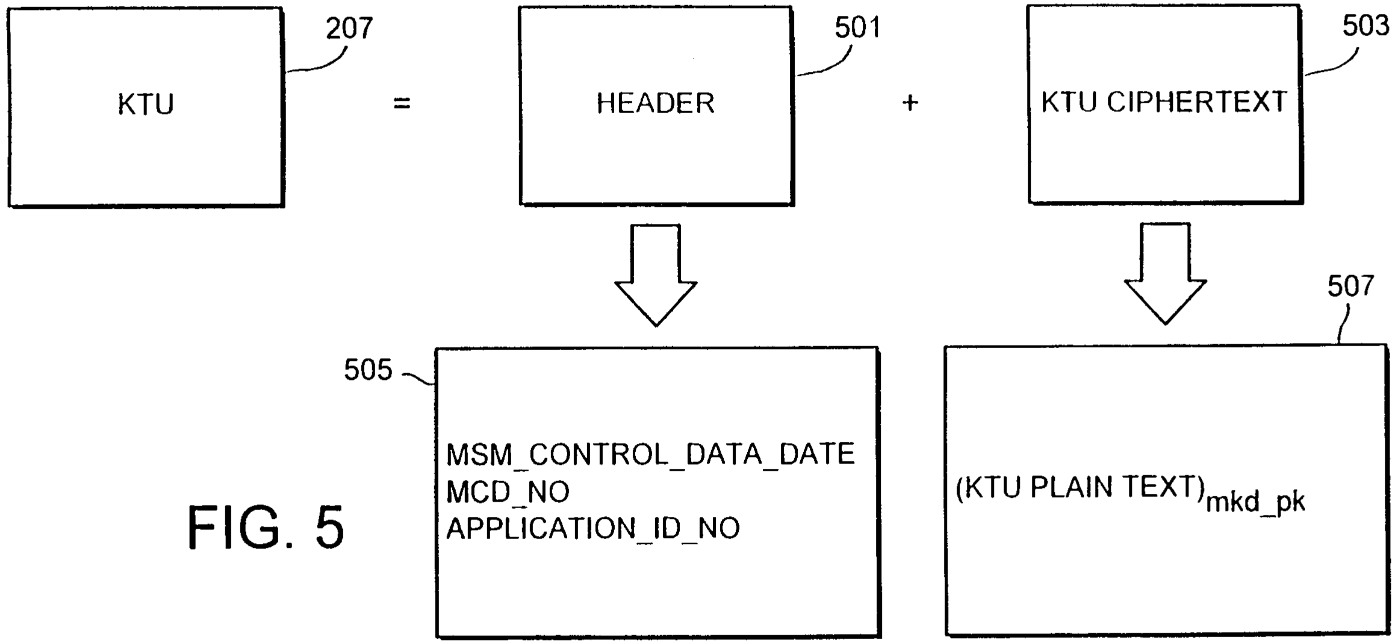


FIG. 5

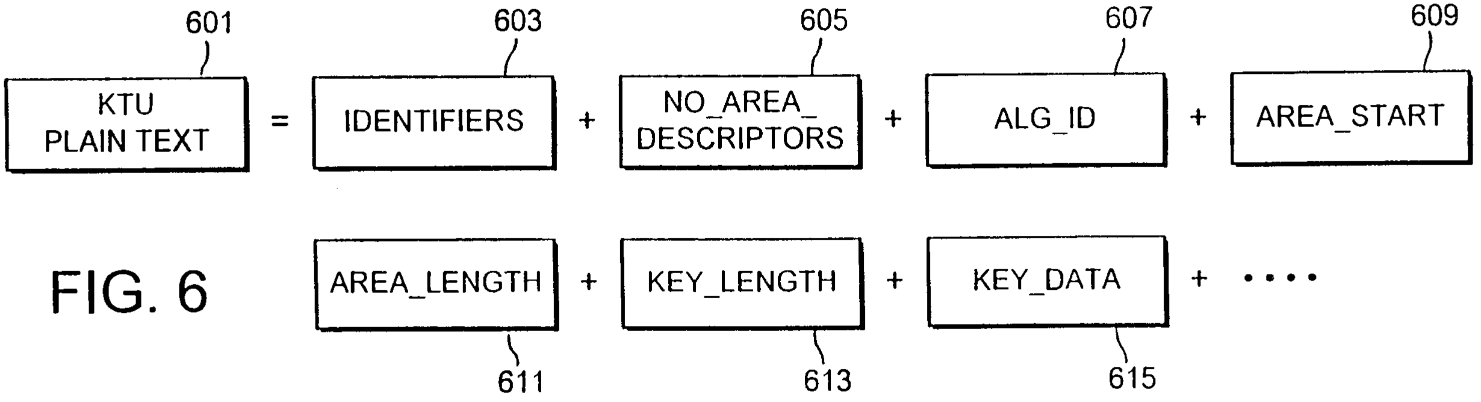


FIG. 6

ANNEX 6 TO THE DRAWINGS

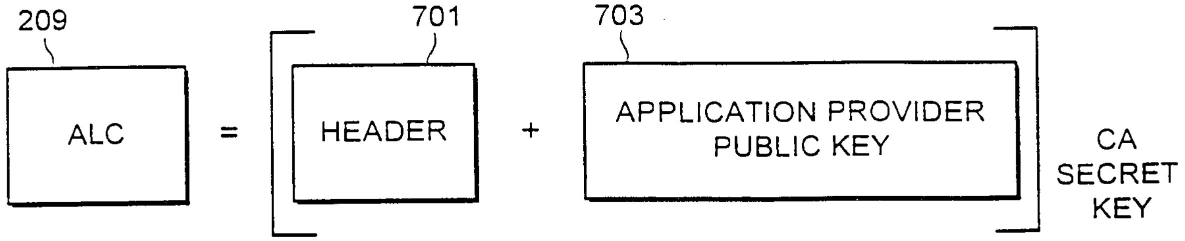


FIG. 7

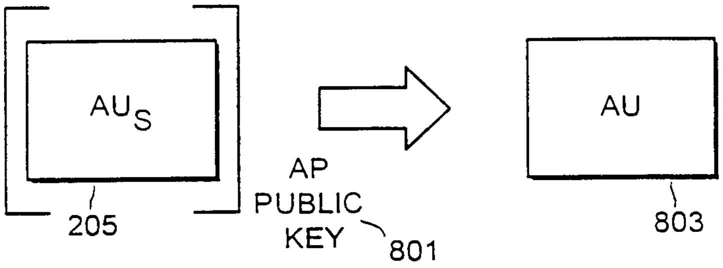


FIG. 8

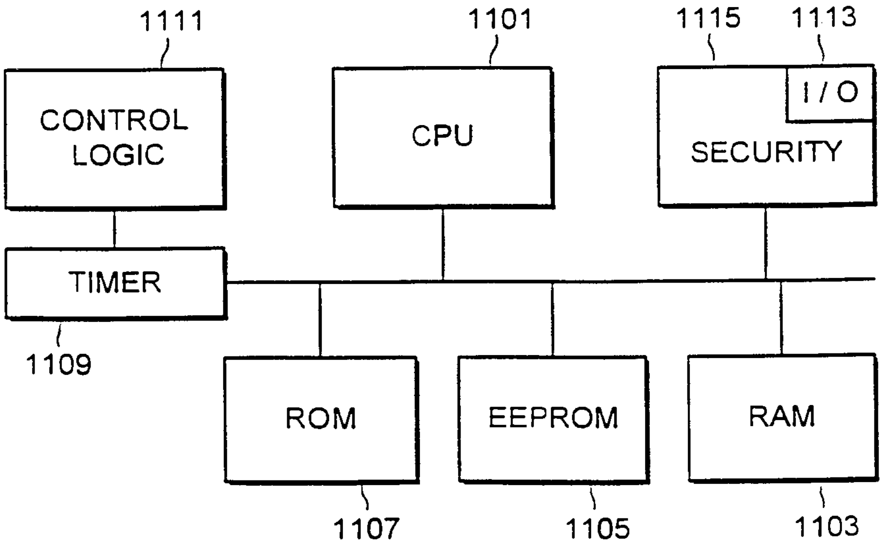


FIG. 11

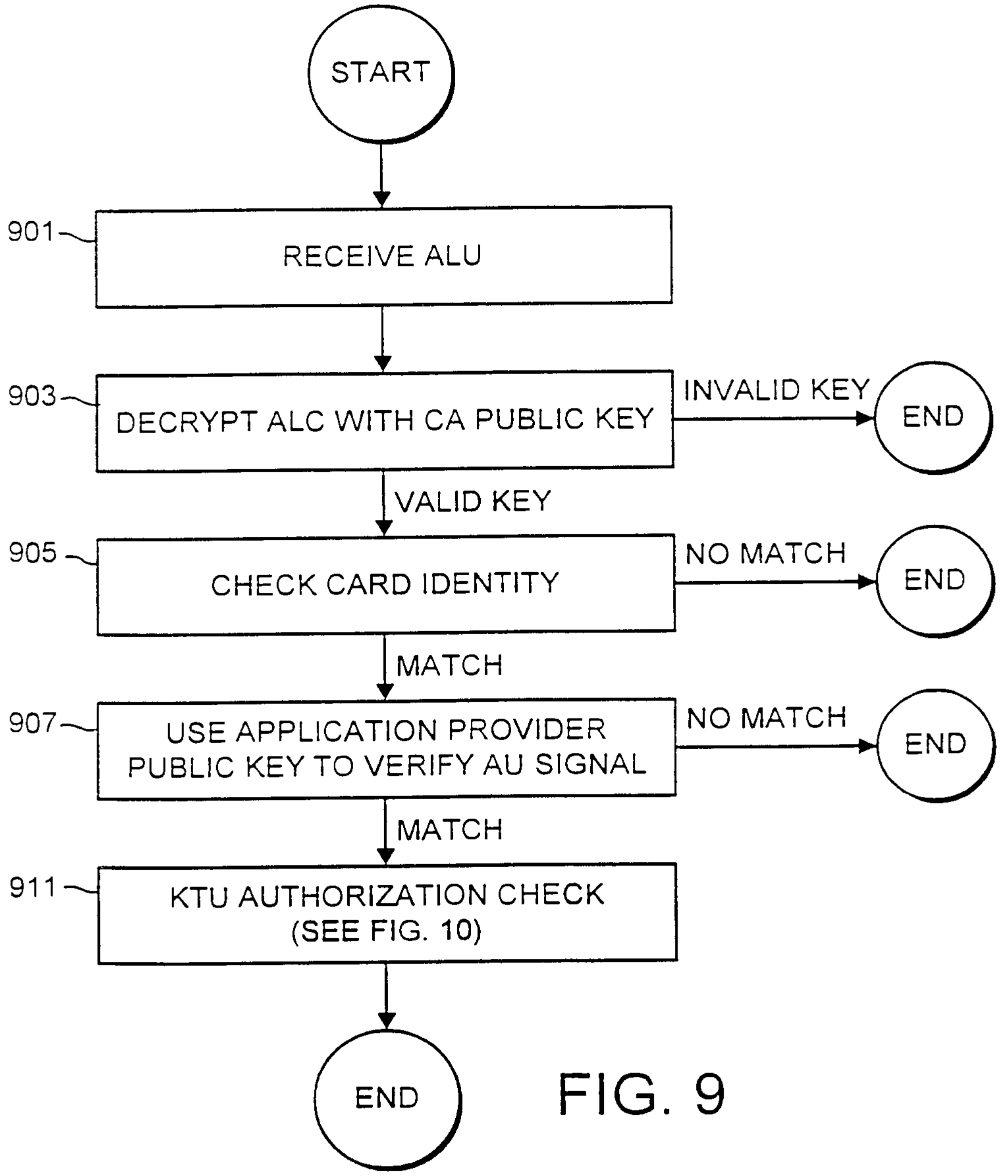

START
901
RECEIVE ALU
903
DECRYPT ALC WITH CA PUBLIC KEY
INVALID KEY
END
VALID KEY
905
CHECK CARD IDENTITY
NO MATCH
END
MATCH
907
USE APPLICATION PROVIDER PUBLIC KEY TO VERIFY AU SIGNAL
NO MATCH
END
MATCH
911
KTU AUTHORIZATION CHECK (SEE FIG. 10)
END

FIG. 9

ANNEX 6 TO THE DRAWINGS

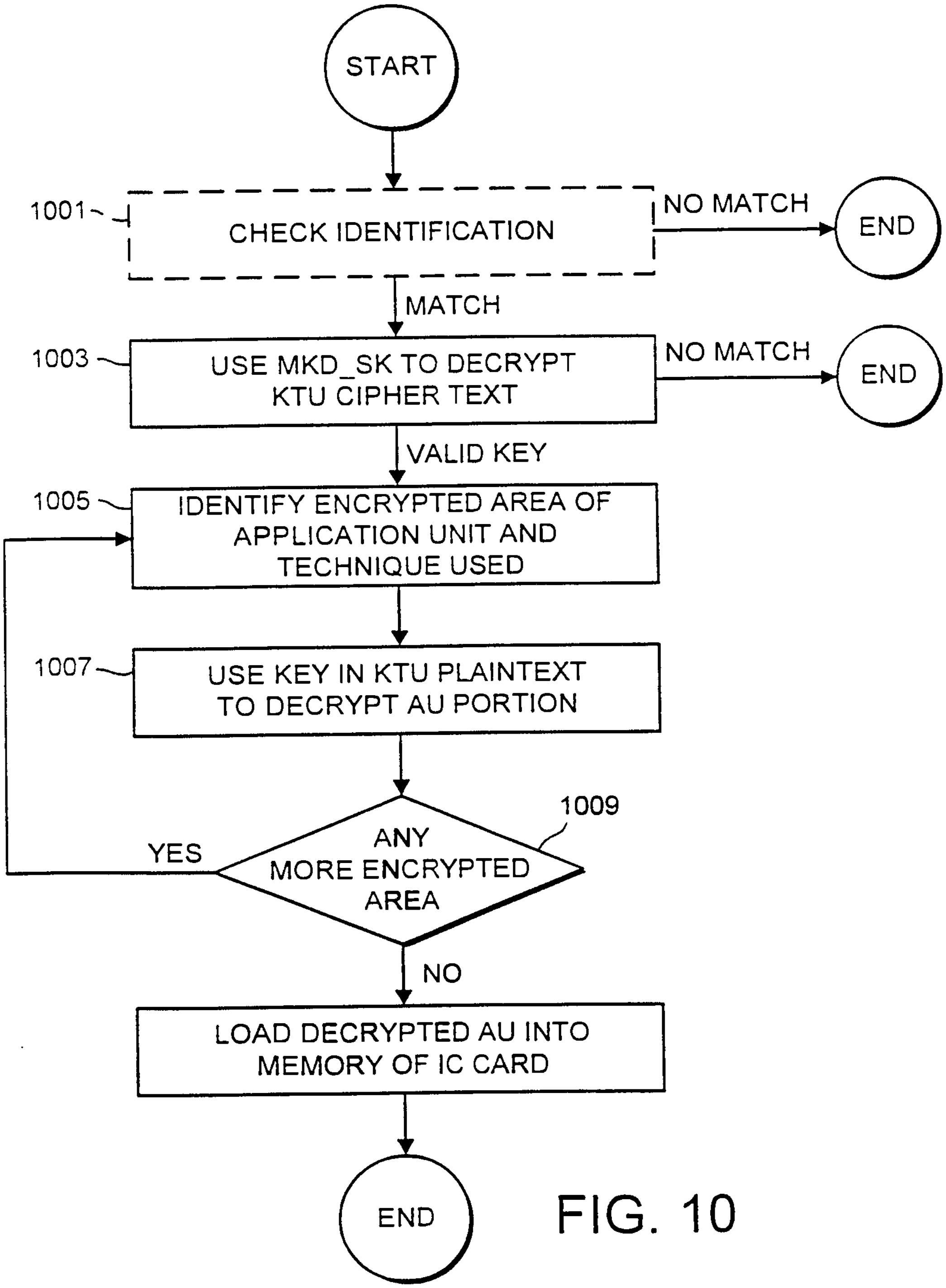

START
1001
CHECK IDENTIFICATION
NO MATCH
END
MATCH
1003
USE MKD_SK TO DECRYPT KTU CIPHER TEXT
NO MATCH
END
VALID KEY
1005
IDENTIFY ENCRYPTED AREA OF APPLICATION UNIT AND TECHNIQUE USED
1007
USE KEY IN KTU PLAINTEXT TO DECRYPT AU PORTION
1009
ANY MORE ENCRYPTED AREA
YES
NO
LOAD DECRYPTED AU INTO MEMORY OF IC CARD
END

FIG. 10

ANNEX C TO THE DRAWINGS

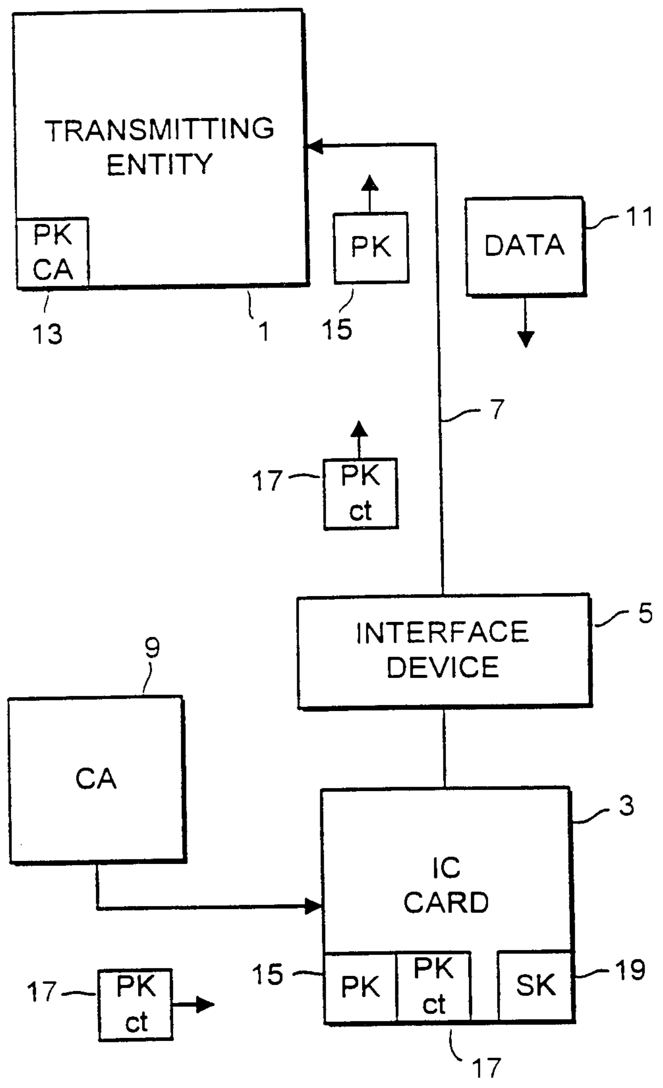


FIG. 1A

ANNEX C TO THE DRAWINGS

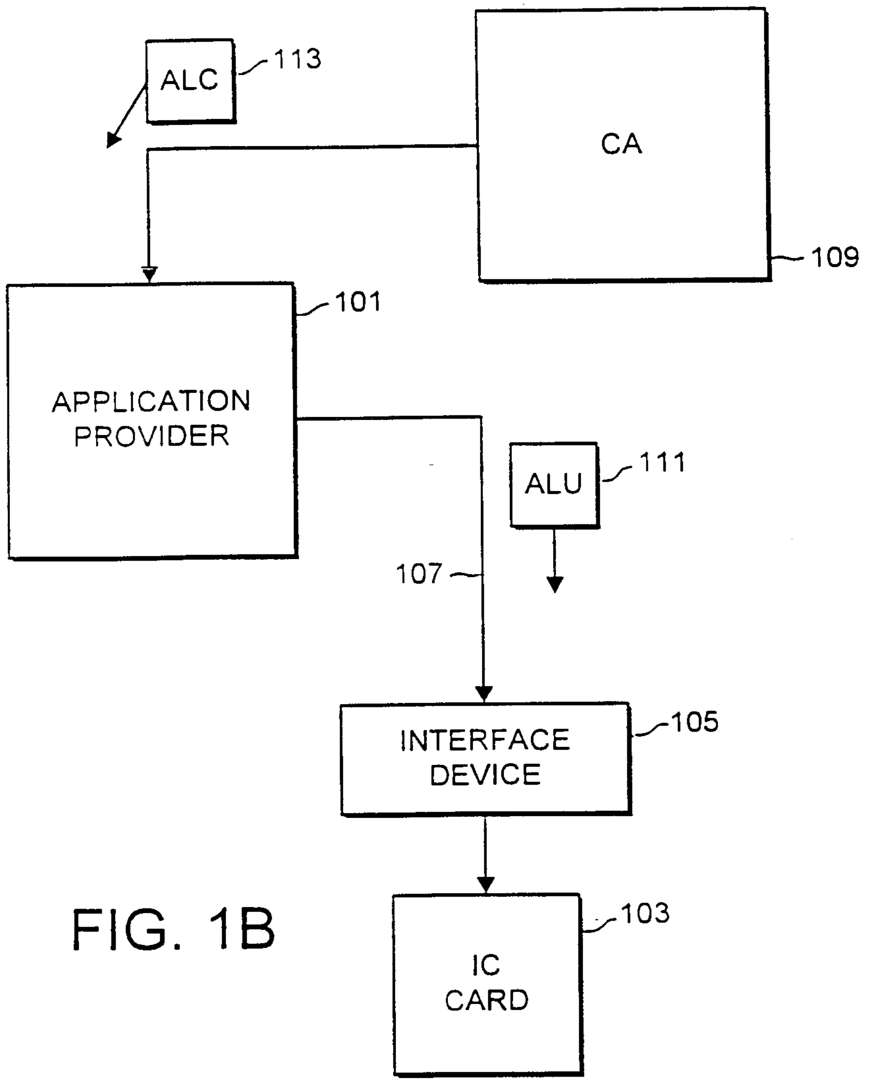


FIG. 1B

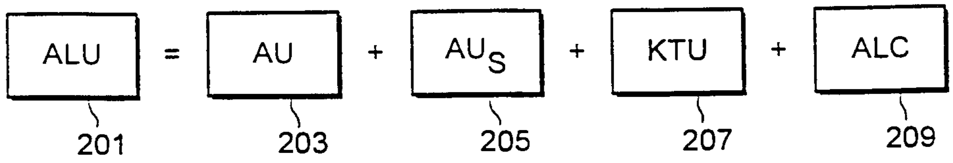


FIG. 2

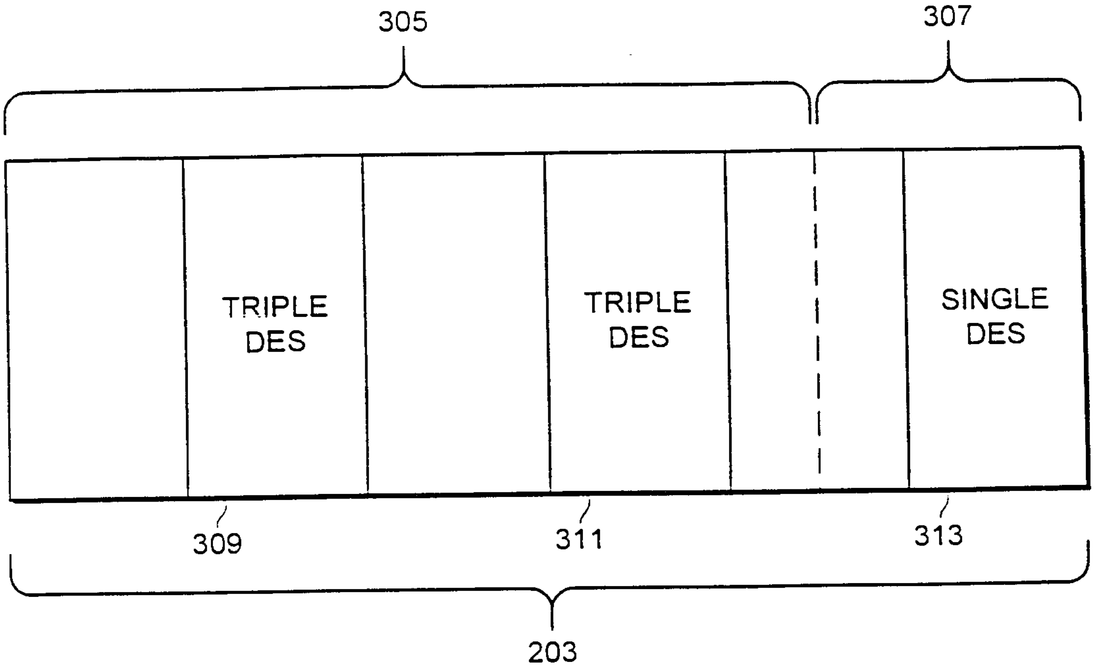

305
307
TRIPLE DES
TRIPLE DES
SINGLE DES
309
311
313
203

FIG. 3

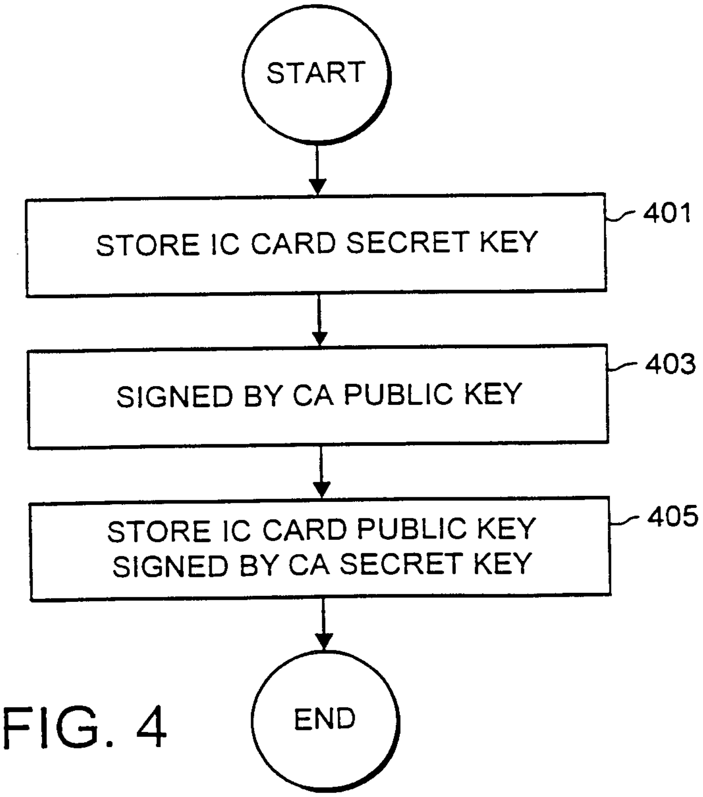

START
STORE IC CARD SECRET KEY
401
SIGNED BY CA PUBLIC KEY
403
STORE IC CARD PUBLIC KEY SIGNED BY CA SECRET KEY
405
END

FIG. 4

26/29

ANNEX C TO THE DRAWINGS

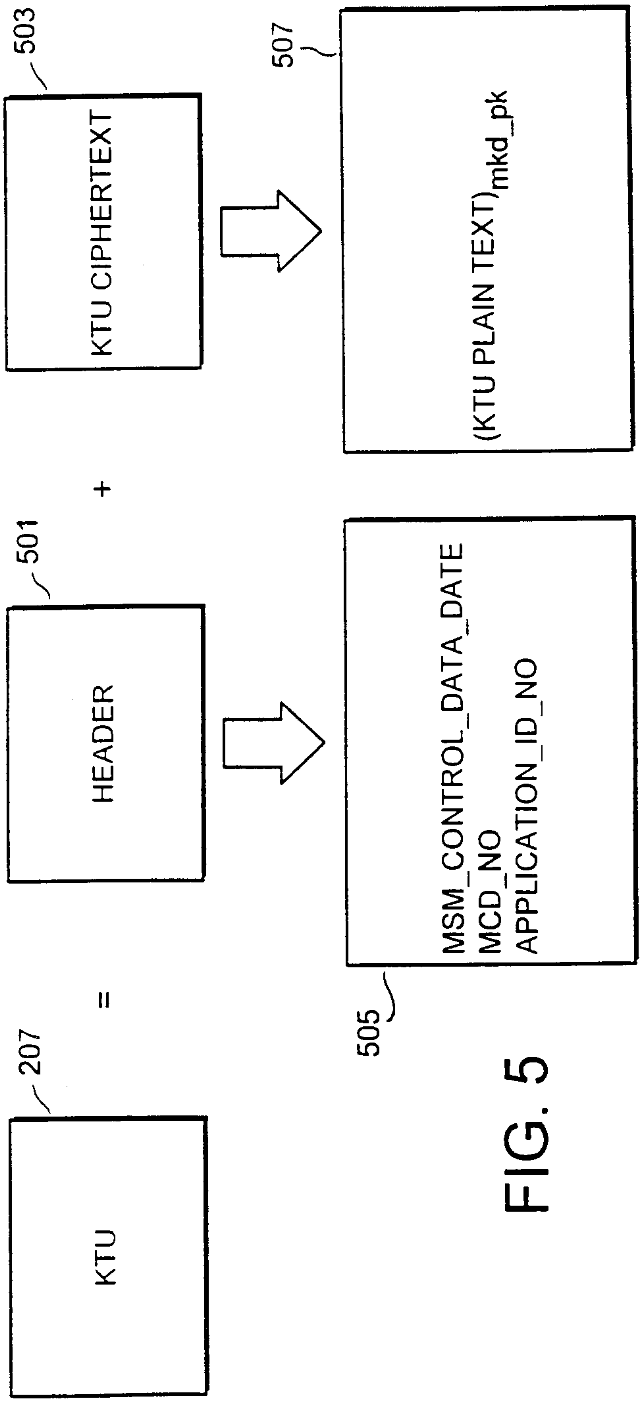


FIG. 5

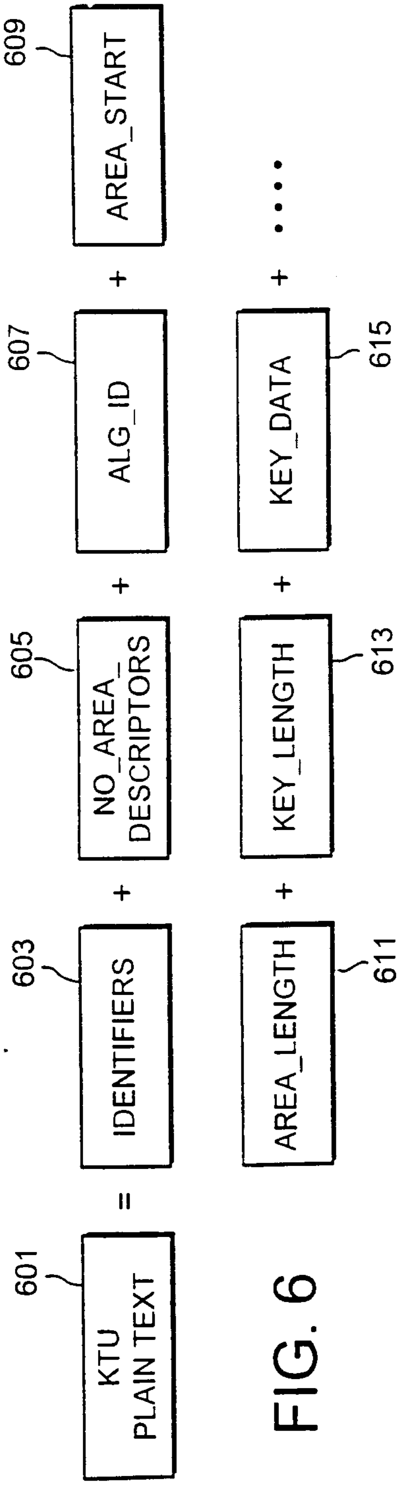


FIG. 6

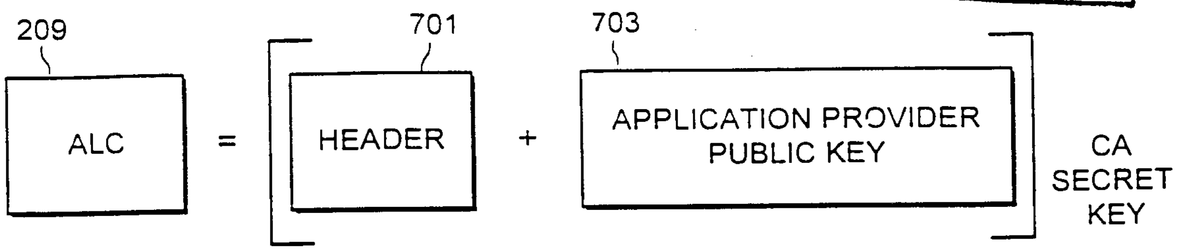


FIG. 7

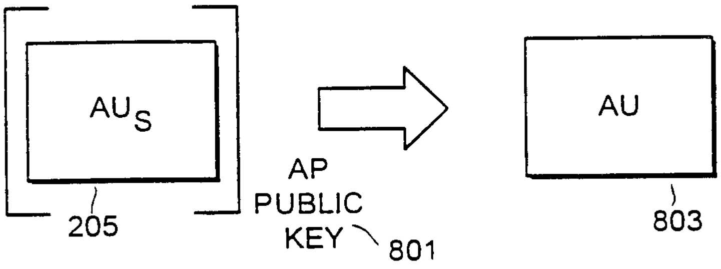


FIG. 8

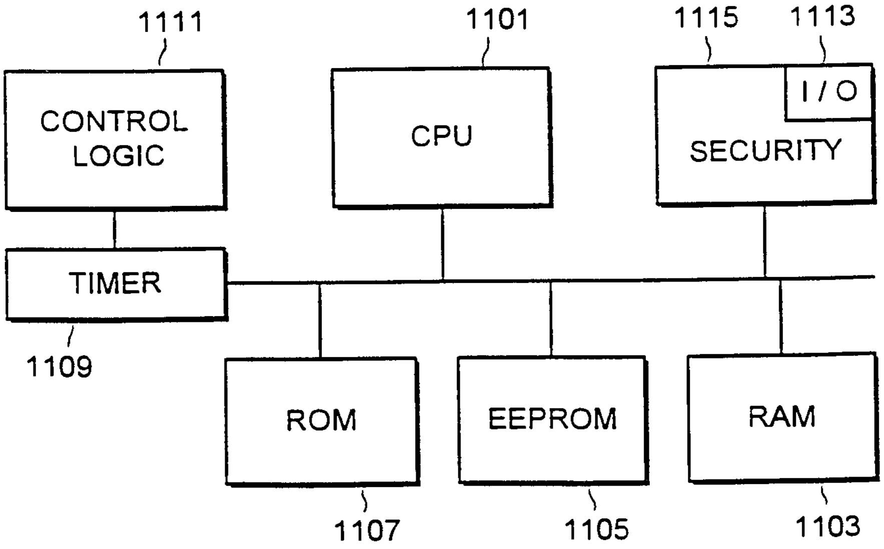


FIG. 11

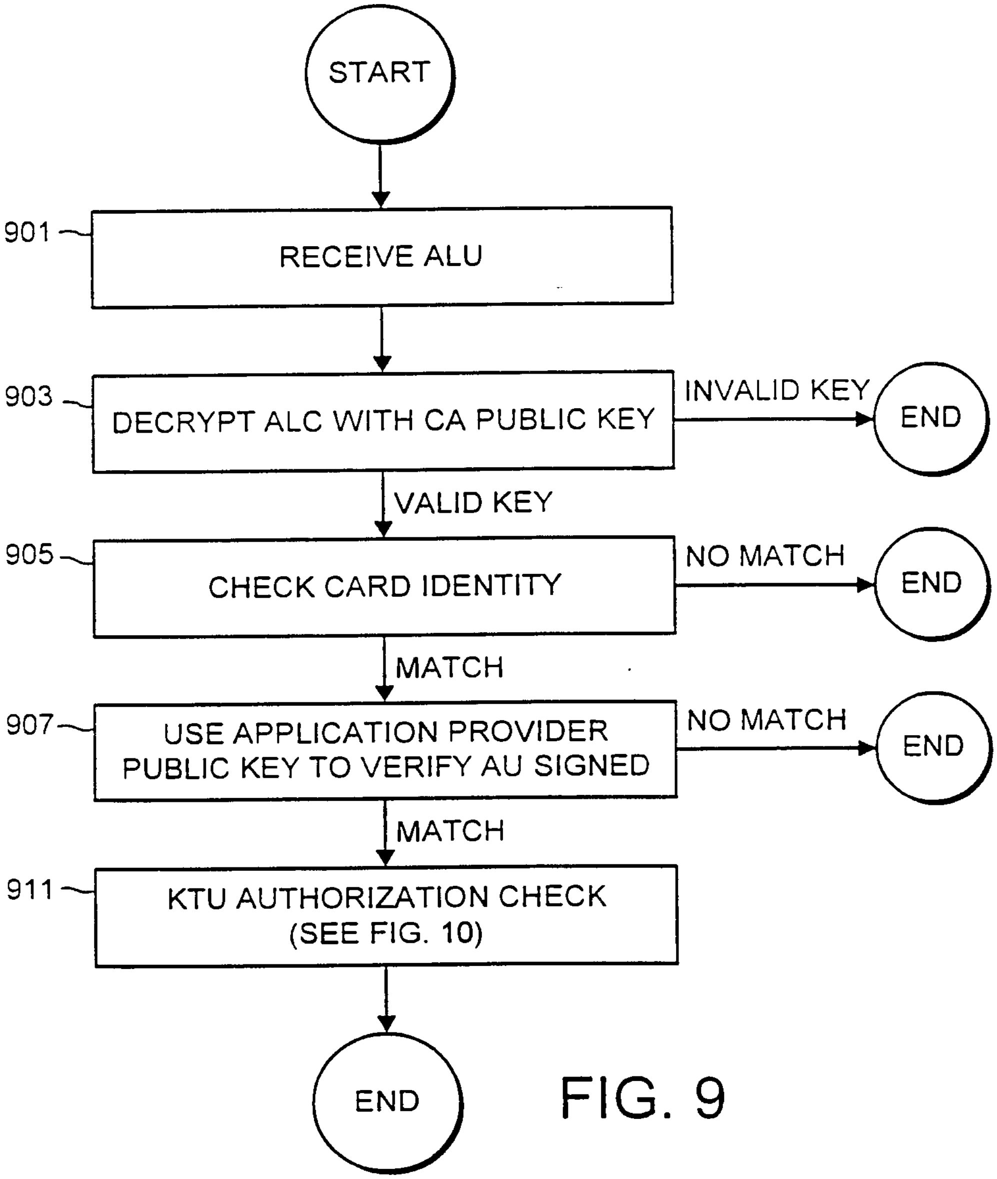

START
901
RECEIVE ALU
903
DECRYPT ALC WITH CA PUBLIC KEY
INVALID KEY
END
VALID KEY
905
CHECK CARD IDENTITY
NO MATCH
END
MATCH
907
USE APPLICATION PROVIDER PUBLIC KEY TO VERIFY AU SIGNED
NO MATCH
END
MATCH
911
KTU AUTHORIZATION CHECK (SEE FIG. 10)
END

FIG. 9

ANNEX C TO THE DRAWINGS

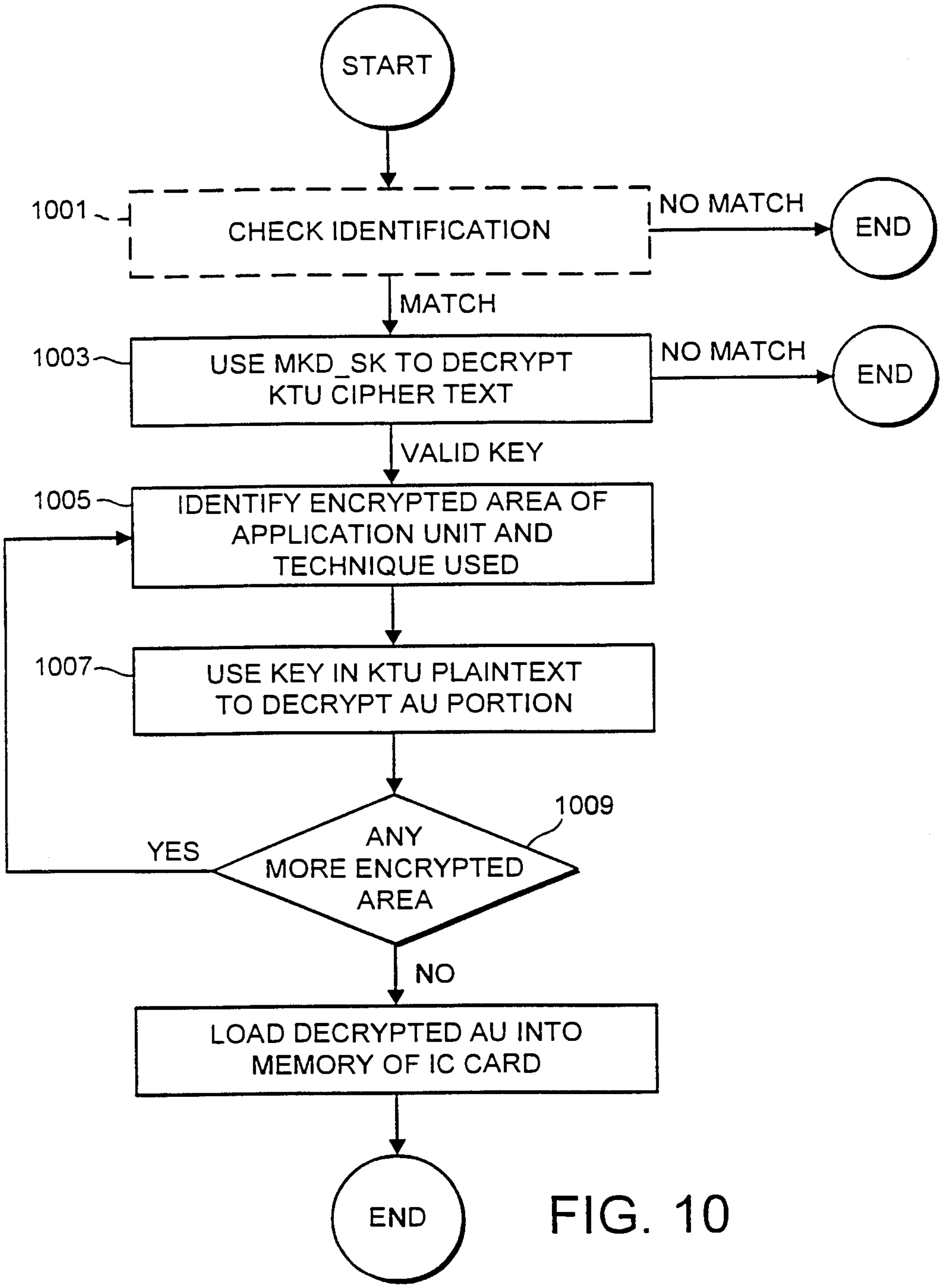

START
1001
CHECK IDENTIFICATION
NO MATCH
END
MATCH
1003
USE MKD_SK TO DECRYPT KTU CIPHER TEXT
NO MATCH
END
VALID KEY
1005
IDENTIFY ENCRYPTED AREA OF APPLICATION UNIT AND TECHNIQUE USED
1007
USE KEY IN KTU PLAINTEXT TO DECRYPT AU PORTION
1009
ANY MORE ENCRYPTED AREA
YES
NO
LOAD DECRYPTED AU INTO MEMORY OF IC CARD
END

FIG. 10

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
G07F 7/10

A2

(11) International Publication Number: WO 98/52159

(43) International Publication Date: 19 November 1998 (19.11.98)

(21) International Application Number: PCT/GB98/01390

(22) International Filing Date: 14 May 1998 (14.05.98)

(30) Priority Data:

60/046,514	15 May 1997 (15.05.97)	US
60/046,543	15 May 1997 (15.05.97)	US
09/064,915	23 April 1998 (23.04.98)	US

(71) Applicant: MONDEX INTERNATIONAL LIMITED [GB/GB]; 47–53 Cannon Street, London EC4M 5SQ (GB).

(72) Inventors: EVERETT, David, Barrington; 31 Ashdown Avenue, Saltdean, Brighton, East Sussex BN2 8AH (GB). MILLER, Stuart, James; 9 Woodford Green, The Warren, Bracknell, Berks RG12 9YQ (GB). PEACHAM, Anthony, David; 4 Lynwood, Groombridge, Tunbridge Wells, Kent TN3 9LX (GB). SIMMONS, Ian, Stephens; The Elms, School Road, Broughton, Cambs PE17 3AT (GB). RICHARDS, Timothy, Philip; 32 Craig Mount, Radlett, Herts WD7 7LW (GB). VINER, John, Charles; Hydes, Woodlands Lane, Windlesham (GB).

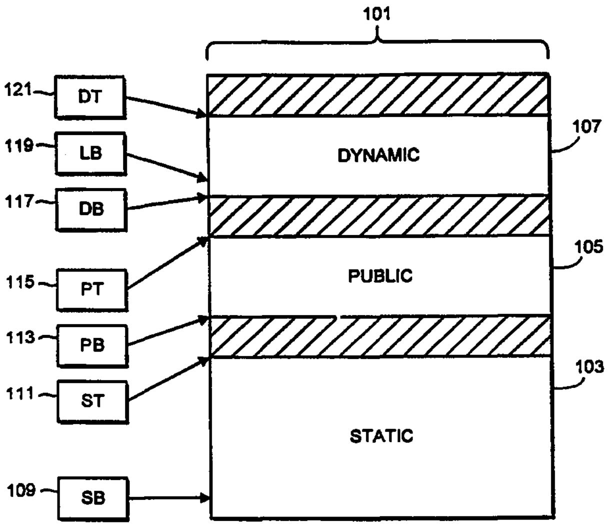
(74) Agent: POTTER, Julian, Mark; D. Young & Co., 21 New Fetter Lane, London EC4A 1DA (GB).

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

Published

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

(54) Title: MULTI–APPLICATION IC CARD WITH DELEGATION FEATURE



(57) Abstract

A multi–application IC card which processes two or more applications using an Application Abstract Machine architecture. The AAM architecture only allows one application to be executed at a time and allows for shared processing by performing a delegation function to a second application. A data space for each application is allocated when the application is selected to be executed. The data space includes a volatile and non–volatile region. The delegation, function temporarily interrupts the execution of the first application, saves the temporary data of the first application, shares any data needed with the second application and the second application is executed until the delegated task is completed. The first application then retrieves the saved data and completes its execution. A delegator stack is used to keep track of the delegator's identity when multiple delegations occur. The AAM model allows for a high level of security while transferring data between applications.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

MULTI-APPLICATION IC CARD WITH DELEGATION FEATURE

BACKGROUND OF INVENTION

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), a random access memory (RAM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. The multiple application operating system present on the IC card allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application card and only executes that one application when inserted into a terminal. For example, a telephone card could only be used to charge a telephone call and could not be used as a credit/debit card. If a card user desires a variety of application functions to be performed by single application IC cards issued to him or her, such as both an electronic purse and a credit/debit function, the card user would be required to carry multiple physical cards on his or her person, which would be quite cumbersome and inconvenient. If an application developer or card user desired two different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal during the transaction, making the transaction difficult, lengthy and inconvenient.

Therefore, it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card.

The increased flexibility and power of storing multiple applications on a single card create new technical challenges to be overcome concerning the integrity and security of the information (including application code and associated data) exchanged between the individual card and the application provider as well as within the entire system when communicating information between applications.

For instance, the existence of multiple applications on the same card allows for the exchange of data between two applications, while one of the applications is being executed. As stated above, a frequent flyer loyalty program may need to be accessed during the execution of an electronic purse application. If data is passed between applications in an insecure manner, it may be possible for a third party monitoring the transaction to determine the contents of the transferred data or even other private data associated with one or both of the applications. Thus, it would be beneficial to provide an application architecture and memory organization which protects an application's data from being discovered by a third party when it is exchanged with other applications present on the IC card.

Accordingly, it is an object of embodiments in accordance with the invention to provide an application architecture and memory organization which provides for data interaction between applications having increased security and allows multiple applications to be accessed while performing a desired task or function.

SUMMARY OF THE INVENTION

The present invention provides for a multiple application architecture for an IC card called an application abstract machine (AAM) and a method for implementing that architecture. The processing of multiple applications is accomplished by generating for at least one application (the "first application") a data memory space including at least two segments, a volatile memory segment and a non-volatile memory segment, commencing the execution of the first application's instructions; delegating or switching execution from the first application to the delegated application and in so doing, saving any data generated by the first application in the logical data memory space associated with the first application; executing the second application's instructions; retrieving the saved data and completing with this data the execution of the first application's instructions.

Additional delegation commands can be issued by the second application or other subsequent applications. The command delegated is interpreted by a delegated application in the same manner as a selection command being issued directly by a terminal and therefore each application performs the security functions at the same level as if a terminal is issuing the command.

The volatile memory segment can further be separated into public ("Public") and dynamic ("Dynamic") portions. Data can be exchanged between a plurality of applications and/or a terminal when stored in the Public region of the data memory. The Dynamic memory region can be used solely as temporary work space for the specific application being executed.

BRIEF DESCRIPTION OF THE DRAWINGS

Further objects, features and advantages of embodiments of the invention will become apparent from the following detailed description taken by way of example only and in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

Fig. 1 is block diagram illustrating the data memory space segment and associated registers for an IC card application using the AAM organization;

Fig. 2 is a block diagram illustrating the code memory and the data memory spaces for an IC card application using the AAM architecture;

Fig. 3 is a flow diagram illustrating the steps of performing a request for a delegation function by one application to another;

Fig. 4 is a flow diagram illustrating the steps of performing a return delegation control function for a delegate application to a delegator application;

Fig. 5 is a flow diagram illustrating the steps of performing an inquire delegator ID request of a delegation function;

Fig. 6 is a block diagram of an IC card chip which can be used as a platform in accordance with the invention; and

Figures 7A, 7B and 7C illustrate multiple delegation calls made between three applications.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

DETAILED DESCRIPTION OF THE INVENTION

An embodiment of the present invention provides for a method and apparatus for processing multiple application programs with associated data stored on an IC card which can be accessed and executed. An application stored on the card can be selected by a terminal, or other interface device, or another application. Each application program which is stored on the IC card when executed is allocated a memory space organized by the program's software code (instructions which are executed by a processor located on the IC card) and the associated data which the application stores and uses during execution of the program.

For example, a multi-application card may store a purse application, or an electronic money application, and a specific loyalty application such as a frequent flyer awards application. Each application has software code and associated data to support the execution of that software code. Each application is allocated a memory space when executed. In this example, there is interaction between the two applications stored on the card. For each dollar electronically spent to make a purchase, the user may be entitled to one frequent flyer mile which is stored and processed by the frequent flyer program. The purse application need not be aware of the specific loyalty program stored on the card, but instead may contain an instruction to communicate with any loyalty program stored on the card. The loyalty program will require input data representative of the amount of a particular electronic value so that it can update its own stored data of current frequent flyer miles for the user of the card.

When two applications need to communicate during the same transaction, a system architecture is required to process both applications in an efficient and secure manner. One approach could be a windows type model where both applications could be running at the same time. Presently, however, IC card platforms are not powerful enough to simultaneously operate multiple programs efficiently. Also, transferred data may be exposed to unwanted third party access. To address this problem, embodiments in accordance with the current invention, which is described in greater detail below, selectively interrupt the execution of applications in a secure manner. This allows the integrity of the applications' data to be maintained and allows the best utilization of the available memory space in the IC card.

An efficient architecture for processing multi applications in an IC card is termed an Application Abstract Machine (AAM) architecture and is described herein. The AAM Architecture applies to any platform independent of its hardware and enables developers to write applications to store on the IC cards which are portable across many different types of platforms (e.g., IC cards built by different manufacturers with different processor configurations) without the need for knowledge about the specific hardware of the platform.

An application abstract machine (AAM), a term for the memory allocation and organization for the data stored and used by each application, is created for each application stored on the IC card which is executed by the processor on the card. In order to ensure data integrity and security when data is transferred between applications which are executed on the IC card, only one application on the IC card is allowed to be executed at a time. Each application has a data memory space which is virtually allocated and mapped onto the physical memory addresses available in the IC card memories. Data is then passed between two or more applications within a specified memory location and in a manner consistent with transferring data to an external terminal or device with which the IC card is securely interacting. At a general level, each AAM space created for each application being executed includes two separate address spaces, one for the program code itself and one for the program data which is stored and/or used by the application. The program data address space is effectively divided into three segments: a Static segment, a Dynamic segment and a Public segment which are described in more detail in conjunction with Figure 1. As stated above, the Static, Dynamic and Public segments are logically mapped to the physical memory; they are virtual memory segments as opposed to physical memory segments. The AAM data address space is preferably addressed and processed using seven different address registers and two control registers.

Figure 1 shows an illustrative diagram of a logical data space allocation 101 created for an application used in conjunction with the present invention. The AAM data portion 101 includes a Static data space 103, a Public data space 105 and a Dynamic data space 107. Also shown are a series of address registers: the Static base address register 109, the Static top address register 111, the Public base address register 113, the Public top address register 115, the Dynamic base address register 117, the Dynamic top address register 121 and local base address register 119 which serves as a local stack frame pointer in the Dynamic data space when the application is being executed. The address registers can contain physical memory addresses but preferably contain offset addresses for the various data address spaces in order to be hardware independent. An example of the overall address space is 64K bytes, although the size varies with the applicable platform and the available memory size. The registers can also be considered pointers or can be any other conventional addressing mechanism.

Within the allocated AAM data space 101, the Static portion of the memory is non-volatile which is not erased after power is removed from the IC card (such as EEPROM), the Dynamic space is volatile (such as RAM) which may be erased after power is removed from the card and the Public space is also volatile (such as RAM). An IC card can receive power from a terminal after it is interfaced into the terminal. Although an IC card may contain a battery to maintain some power for memory and circuitry, volatile memory will typically be erased after the IC card is removed from its power source.

The defined AAM data space has bytes in each segment which are contiguous, so that applications can perform pointer and offset arithmetic. For example, if the segment addresses "1515" and "1516," or any other pair of sequential numbers, are both valid and are present within the same segment, then they address adjacent bytes. This allows offset values stored in registers to determine the location of a desired memory address. The segment address of the first byte of the Static segment is zero, so that the segment address of a given location within the Static region is equal to its offset.

Pointers to other specific regions of the Static data area can be stored in the Static data because the Static region is non-volatile. For example, if the card user's name is stored in the Static memory of a credit/debit application, the application will know the card user's name will always be stored in the 5th memory location above the starting point for the Static portion of memory. The location can be noted as SB[5] or the 5th byte above the Static Bottom. Since the Static memory is non-volatile, it will not be erased after each transaction and the application will always know of its location relative to the Static segments' address registers.

On the other hand, the segment address of any location in the Dynamic or Public segments is not always equal to a particular offset from the beginning of the respective segment because the contents of those segments change for each operation. The fourth location in the Dynamic segment will be different for each operation performed by the application. The address of a memory location of Dynamic or Public segment is fixed preferably only for the duration of one command-response pair operation. Because segment addresses in Dynamic or Public are not fixed, MULTOS Executable Language (MEL)™ instructions (or any other program instructions) cannot refer to data using only segment addresses. Instead, a tagged address preferably is used to identify data which is to be retrieved, manipulated, transferred and/or stored with the IC card system.

A tagged address is a nineteen bit value consisting of a three bit tag (address register number) and a sixteen bit offset. Each of the seven address registers for the AAM data space contain a segment physical address. For instance, the address registers SB 109 and ST 111 point to the boundaries of the Static, the address registers PB 113 and PT 115 point to the boundaries of the Public and the address registers DB 117 and DT 121 point to the boundaries of the Dynamic. For each segment, the top register points to the byte immediately after the last valid byte. For example, the last valid byte of the Static is ST[-1]. Register LB functions as a stack frame pointer. It points to a location in the Dynamic segment to indicate a specific byte of local data for the currently executing application.

Referring to Figure 1, the allocated Static segment 103 contains the application's non-volatile data. Static data includes data which is associated with each application for every transaction such as the card user's name, account number, PIN value and address. Static data also includes variable data which is stored for use in future transactions using the application. For example, in a purse transaction, the electronic value data would be read from the Static segment and later saved in the Static segment at the end of the transaction. Additionally, transaction information data or available credit limits in the case of a credit/debit application would be stored in Static data.

The Static data is addressed using register SB (Static Base) and the register ST (Static Top) as offset registers. These registers contain the offset value from a physical address in a memory on the IC card. The individual memory location is then further offset from these starting points such as SB[3] or ST[-5]. SB is defined as zero and ST is equal to the size of the application's Static data which is set when the application is loaded onto the IC card. The multiple application operating system ensures that no other application can read or write the data stored in the Static segment of a particular application. Using current technology, the Static segment is preferably mapped onto an EEPROM (Electrically Erasable Programmable Read-Only Memory) which is non-volatile.

The Dynamic segment 107 contains the application's volatile or temporary data. Dynamic data includes data which is temporarily used during the execution of an application such as intermediate values used in calculations or working variables. For example, a purse application may temporarily store the value of a transaction in order to reduce the amount of the value in the purse. The temporary data is used much like conventional computer programs use RAM to perform their assigned operations. The Dynamic segment preferably is divided into two parts, the session data portion and the stack data portion. The size of the session data is a constant for each application and is determined when the application is loaded. The stack holds variable data which is unique to the particular transaction being executed. The stack data portion stores data in a last-in-first-out manner. The stack is initially empty, but expands and contracts during execution of the application.

The Dynamic data is addressed from the register DB 117 to register DT 121. Register LB 119 serves as a local stack frame pointer to particular memory locations in the Dynamic segment for delegate commands or function calls. Register LB 119 is used to address the topmost frame, that of the currently executing function's session data. Register DT 121 serves as an address offset for the stack pointer. A one byte data item at the top of the stack is addressed as DT[-1], the next byte below is addressed by DT[-2], and so on. A push operation increments the relative value of DT for each item on the stack and a pop operation decrements the relative value of DT for each item on the stack. For example, a data element located at DT[-5] will be located at DT[-6] after an additional data item is placed on the stack.

When an application is being executed, the Dynamic segment created for that application also contains the application's session data which is used in performing the assigned task(s) or operation(s). The multiple application operating system ensures that no other application can read or write the data stored in the Dynamic segment of a particular application. The session data is set to zero upon the start of the execution of the application. Stack data will be saved in the stack if the application delegates a task or operation to another application.

A delegation function occurs when one application selects another application to process a command instead of processing the command itself. An example of a delegation function occurs when a delegator application receives a command that it does not recognize or is not programmed to process. The selected application should not reject the command and provide an error response to the interface device (IFD), but instead should pass the command to the appropriate receiver, or delegated application. In order to perform a delegation, the delegator calls the Delegate primitive. The Delegate primitive is a subroutine recognized by the multiple application operating system which is executed when the operating system interprets the Delegate instruction. Primitives can be stored as part of the operating system itself, loaded as a separate routine when the operating system is installed. Primitives are preferably written in machine executable language so that they can be executed quickly although they could be written in a higher level language. When a Delegate command is executed, execution of the delegating application is suspended, and the delegated application is executed instead. The delegated application then generates its own data memory space according to the AAM architecture. The data stored in the Public memory space of the first application (stored in RAM) is sent to the Public memory space of the second application (which could be physically the same memory but is allocated separately for each application) so that data can be passed between the applications. The Dynamic memory space is also shared although data is saved in a stack for the delegator and the other portions initialized before the delegated application is executed because the Dynamic data is secret.

In most cases, the delegated application processes the command exactly as though the command has arrived directly from an interface device. When the delegated application has finished processing the command, and has written a response into the allocated Public memory segment, it exits as normal. The delegator then resumes execution at the instruction address following the executed instruction which called the Delegate primitive. The response generated by the delegated application is retrieved or accessed from the allocated Public memory space. The delegator application may simply exit in turn, thus sending the response to the IFD, or may carry out further processing before exiting.

Another example of a delegation operation occurs when two applications need to share data. If an application A always returns a data item N when processing a command B, then another application which also returns data item N in response to a command can delegate the function B to application A in order to reduce the need for duplicate codes stored on the IC card. For example, if a PIN needs to be checked before an application is executed, an application stored on the card can delegate the "retrieve PIN function" to a PIN application which returns a stored universal PIN for the card.

Preferably, a new session begins whenever the IFD, e.g. a terminal, successfully selects an application, even if the application has been previously selected during the transaction. For example, if a card user goes to a terminal and transfers twenty dollars of electronic cash using a purse application, charges thirty dollars using a credit/debit application and then transfers ten dollars using the purse application again, three separate sessions will have occurred even though only two applications were used during the entire transaction. Each time an application delegates a task or function to another application, the delegated application treats the delegate function as if the IFD devices had selected the application to perform the task or function. However, performing a delegation function as described below has a different effect on session data.

The following examples will help explain when the session data is initialized (i.e., erased) versus when it is saved to be used in further operations. If application A is selected by an IFD device, and receives commands X, Y and Z from the terminal, application A may delegate all three commands to application B. For example, delegations may occur in response to delegation commands in the program code. Both applications A and B will have their session and stack data in their respective Dynamic segments initialized (set to zero) when they receive command X, but the stack will not be initialized when they receive the subsequent commands Y and Z.

In a second example, application A is selected, and receives commands X, Y and Z from the terminal. Application A processes X itself, but delegates Y and Z to application B. Application A will have its session and stack data initialized when it receives X, but not when it receives the subsequent commands Y and Z. Application B will have its session and stack data initialized when it receives Y, but not Z.

One example of a use of session data is to support the use of a session Personal Identification Number (PIN). The application could reserve one byte of session data to support the PIN-receiving flag. On receiving the PIN check command, the selected delegated application could update the flag as follows: if the PIN command is received and the inputted PIN is equal to the stored pin, then it will set the session data DB[0] to 1. If not, the application will check if the PIN flag is already set by checking the value in DB[0]. In either of the above cases, the application will process the rest of the commands in the session because the PIN has been verified. If neither of the cases is true, then the application will not process the command because the PIN is not proper. The PIN checking function could be a delegated function from the selected application to a PIN checking application.

The Public segment 105 is used for command and response data being passed between an IFD and an application. During a delegate command, the Public segment contains the data passed between two applications, the delegator (the application initiating the delegation) and the delegated application (the application which performs the delegated function). An application may also use the Public segment as a further temporary working storage space if required. The Public data is addressed using offsets stored in register PB 113 as a starting address, to register PT 115 as an ending address. Register PB 113 and Register PT 115 are fixed for the duration of a command-response pair being initiated by the IFD or delegator. Public data can include data inputted into or supplied by a terminal such as a transaction amount, vendor identification data, terminal information, transmission format or other data required or used by an application resident on the IC card. Public data can also include data which is to be transmitted to an IFD device or other application such as an electronic dollar value, card user information transmission format or other data required or used by the terminal or other delegated application.

The multiple application operating system ensures that the data stored in the Public segment remains private to the application until the application exits or delegates. Preferably, the data in the Public segment is then made available to other entities as follows: (1) if the application delegates, the whole of the Public segment becomes available to the delegated application; (2) if the application exits, and is itself delegated by another, the whole of the Public segment becomes available to the delegator; or (3) if the application exits, and is not itself delegated, then a portion of the Public segment containing the I/O response parameters and data are made available to the IFD.

An application may write secret data into the Public memory segment during execution of the application, but the application must make sure it overwrites the secret portion of the Public segment before delegating or exiting. If the application abnormally ends (abends), then the operating system on the IC card preferably overwrites all of the data in the Public segment automatically so that no unwanted entities can have access to the secret data. If the MULTOS carrier device (MCD) is reset, the operating system overwrites data in the Public segment automatically, so that no secret data is revealed. A portion of the Public memory segment is also used as a communications buffer. The I/O protocol data and parameters are preferably stored at the top of the Public memory space. In another preferred embodiment, the top seventeen bytes are reserved for the communications protocol between the IFD device and the IC card application. However, additional or less bytes can also be used depending upon the particular application and operating system being utilized.

The spaces shown between the memory segments in Figure 1 will vary depending upon the specific application and commands being processed. There could be no memory space between the memory segments so that the memory segments are contiguous.

Figure 2 shows an extended illustration of the AAM implemented architecture. Data memory space 201 includes the three segments Static, Public and Dynamic as previously described. Code memory space 203 contains the program instructions for an application stored on the IC card. The application instructions are preferably stored in an executable form which can be interpreted by the resident operating system but can also be stored in machine executable form. Instruction 205 is stored at one location in the code memory space 203. Additional instructions are stored in other locations of memory space 203. Two additional registers 207 and 209 are used in the AAM architecture. A code pointer (CP) register 207 indicates the particular code instruction to be next executed. In the figure, the register indicates, e.g., through an offset or pointer means, that instruction 205 is the next to be executed. Condition Control Register 209 contains eight bits, four of which are for use by the individual application and four of which are set or cleared depending upon the results of the execution of an instruction. These condition codes can be used by conditional instructions such as Branch, Call or Jump. The condition codes can include a carry bit, an overflow bit, a negative bit and a zero bit.

All address and control registers are set to defined values prior to executing the selected or delegated application. The values are set either when the application is first loaded onto the card and the size of the code and non-volatile data can be ascertained or at the moment when the application passes control to the application. When the application is loaded, SB is set to zero and ST is equal to the number of bytes in the application's Static database. The other address registers are initialized when the application is given control. CP 207 is set to zero and all eight bits in CCR 209 are cleared at the start of executing the application.

A communications interface mechanism is present between the IFD and an application which includes the use of the Public data segment as a communications buffer for command-response parameters. A command-response parameter means an application is given a command to perform and returns a response to the entity issuing the command. Applications interact with an IFD by receiving commands, processing them and returning responses across the IFD-Application Interface. When an application has completed executing a command, the application will place the response into the Public segment starting at PB[0] which can be read by the IFD device and will set the proper interface parameters in the reserved Public space relative to PT[0].

While an application can be called directly from an IFD and return a response directly to an IFD, it can also delegate a request to another application where appropriate. The subsequently-called application will then process the request on behalf of the first application. The delegation can be directly in response to a received command in which the delegator acts as a controller for delegating commands or subcommands to other appropriate applications. Alternatively, the delegated command can be embedded in an application's code which delegates control of the processor when the first application needs to interact with another application during its execution, such as updating frequent flyer miles or verifying a PIN.

Figure 3 shows a flow chart of the steps which are performed when a delegate request is executed. Step 301 sets the parameter named delegator_application_id (delegator ID) to be equal to the selected_file.application_id (selected ID). The selected ID indicates the current application which is selected and which is currently being executed. The delegator ID indicates the application which delegates a function to another delegated application stored on the IC card. Step 303 then pushes (stores) the delegator ID onto the top of the delegate_id_stack (delegate stack). The data referenced in the Dynamic portion of allocated memory is saved so that the current application can complete its execution after the delegated function is complete. Data which is to be shared with the delegated application is referenced in the Public portion of allocated memory. The delegate stack is preferably stored outside of an application's AAM memory space and keeps track of which applications have delegated functions. Each application is suspended when it delegates a function so the delegate stack can act in a Last-In-First-Out (LIFO) manner so that if a number of applications are suspended due to delegation requests, the proper application is started in the right order. The delegate stack thus keeps track of which application was the last delegator when multiple layered delegation functions are performed. The delegate stack preferably operates in a LIFO manner although different stack schemes could be used as appropriate.

Step 305 then sets the selected ID to the delegate_request.delegate_application_id (delegate ID) value. This step selects the application which will be called to perform the delegated function or functions. The identities of the delegated application can be specifically called by the delegator application or a particular function can be matched up with an application in a look up table. For example, a PIN match operation may be delegated to different applications depending upon which applications are present on the card. Step 307 then sets the application_command parameter to the value stored in the delegate_request.application_command parameter. This step specifies the command to be delegated to the delegate application. Applications typically have the ability to process many different commands. Alternatively, the entire application could be executed to perform one or more functions. The delegator application can choose which command it is delegating to another application. Step 309 then sends the application_command to the AAM operating system for execution by the delegatee application. The delegator application is then suspended (or interrupted). Any data that is required to pass between the applications is transferred via the Public memory space.

Figure 4 is a flow chart of the steps for performing a "return delegation control" command by the delegatee application. This command is executed by the operating system when a delegated application has completed its delegated function. Step 401 gets application_responses from the Public memory space of the delegated AAM. The response data is passed in the Public memory segment of the delegatee AAM. Step 403 then sets the delegate_response.status variable to a success condition. This means that a delegation operation has been successfully completed. Step 405 sets the delegate_ response.application_responses parameter to the application_responses values which were stored in the Public segment of the delegatee application.

Step 407 sets the delegate_response.delegate_application_id parameter to selected_file.application_id (the delegatee application ID). Step 409 pops the top (i.e., reads the last data stored in the stack) delegate_application_id from the delegate_id_stack. This information indicates the identity of the delegator application for the command which was just delegated and completed by the delegated application. Step 411 sets the select_file.application_id value to the delegator_application_id value. This selects the delegator application which was identified from the delegate ID stack as the current application which will resume running. The Dynamic data for the delegator application will be retrieved for the delegator application from its stored location so that the application will continue to execute where it left off with all data intact but will also have the response information from the delegated function. In step 413, the delegate_response data is sent to the current application for further processing. The response data is passed through the Public data space which could be the same physical RAM memory location because all applications share the physical volatile memory space.

Figure 5 shows a flow chart of the steps involved for inquiring about a delegator ID when a delegate command is received by a delegated application. The delegated application may need to know the identity of the delegator because it may perform operations differently for different delegator applications. For example, an airline loyalty program may need to know if awarded frequent flyers will be based on actual dollars processed or a lump sum award for some other activity such as performing a bill payment operation. This information could be passed to the delegated application as a variable or could be ascertained using an inquiry. The delegator inquiry operation could be implemented as a primitive as previously described.

Step 501 receives the delegator_id_enq_request from the AAM operating system. The request is used to identify the identity of the delegator. Step 503 checks if the delegate_id_stack is empty. If the stack is empty, then no delegation operations have occurred and no applications have been suspended. Thus step 511 sets the delegator_id_enq_response.status parameter to a failure indicator. Step 513 then sets the value of delegator_is_enq_request.error_cause to a value indicating "no delegator application." There is no delegator application. The process then continues with step 509.

If the delegate_id_stack is not empty, than one or more delegations have occurred. In that case, step 505 sets the delegator_id_enq_response.status parameter to a value indicating "success". Step 507 then sets the delegator_id_enq_response.delegator_ application_id parameter to the value stored in delegate_id_stack.delegator_ application_id. This sets the inquiry response to indicate the delegator application ID at the top of the stack. As explained above, the stored data at the top of the stack indicates the last delegator application to call a delegate function. Step 509 then sends the delegator_id_enq_ response back to the AAM operator system which delivers the information to the application or IFD entity requesting the information.

Figure 6 shows an example of a block diagram of an integrated circuit located on an IC card chip which can be used in conjunction with the invention. The integrated circuit chip is located on a chip on the card. The IC chip preferably includes a central processing unit 601, a RAM 603, a EEPROM 605, a ROM 607, a timer 609, control logic 611, I/O ports 613 and security circuitry 615, which are connected together by a conventional data bus 617 or other conventional means.

Control logic 611 in the smart card provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports 612. CPU 601 in conjunction with control logic 611 can perform many different functions including performing calculations, accessing memory locations, modifying memory contents, and managing input/output ports. Some IC cards also include a coprocessor for handling complex computations like cryptographic algorithms. Input/output ports 613 are used for communication between the card and an IFD which transfers information to and from the card. Timer 609 (which generates and/or provides a clock pulse) drives the control logic 611, CPU 601 and other components requiring a clock signal through the sequence of steps that accomplish functions including memory access, memory reading and/or writing, processing, and data communication. Security circuitry 615 (which is optional) preferably includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed upon completion of testing to prevent later access. The Static memory space is preferably mapped to memory locations in EEPROM 605 which is non-volatile. The Dynamic memory space is preferably mapped to RAM 603 which is volatile memory which has quick access. The Public memory space is also preferably mapped to RAM 603 which is volatile memory. The Dynamic data and Public data will be stored in different portions of RAM 603, while RAM is identified as a preferred non-volatile memory and EEPROM is identified as a preferred volatile memory. Other types of memory could also be used with the same characteristics.

Figures 7A, 7B and 7C illustrate an example of a delegation function being performed in order to process multiple applications on an IC card. Figure 7A shows a first application being executed as denoted with a double ringed circle 701. At some point during the execution of the first application, a delegation function 702 is called to delegate an operation to the second application which is indicated by circle 703. Also shown in Figure 7A is an empty delegator ID stack 705. Since the stack is empty, there is no data associated with it and it is shown only for illustrative purposes.

The multiple application operating system receives the delegate command and interrupts the execution of the first application 701 and gives control of the integrated circuit to application 703 as shown in Figure 7B. The execution of the second application 703 is illustrated with a double ringed circle. The term “gives control” means that the microprocessor and other circuitry on the card will process the instructions and allocate memory space for the application which is delegated. When the delegate command is processed, the delegator ID 707 is placed on top of the stack 705. The delegator ID stack is operated in a LIFO manner. Also shown in Figure 7B is a third application 709 resident on the card. At some point during the execution of the second application, a delegate function 711 is called to delegate the operation to the third application.

The multiple application operating system receives the delegate command 711 shown in Figure 7B interrupts the execution of the second application 703 and gives control of the integrated circuit to the third application 709 as shown in Figure 7C. When the delegate command is processed, the delegator ID 713 of the second application is pushed onto the delegator ID stack 705. The delegator ID 707 of the first application whose execution is still interrupted is pushed down in the stack consistent with a LIFO stack management. Thus when the third application has finished its execution, the delegator ID at the top of the stack is popped to indicate that execution of the second application should be resumed first. The delegator ID 707 from the first application will then be at the top of the stack so that when the second application is finished executing, the first application will resume its execution.

Additional applications can be managed by the delegator ID stack in a similar manner. By interrupting the execution of the applications when a delegate command is processed and keeping track of the order of delegations, the security and integrity of the data for each individual application can be maintained which is important because IC cards will store data for applications which is private to the card user such as account numbers, social security number, address and other personal information.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous apparatus, systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

The scope of the present disclosure includes any novel feature or combination of features disclosed therein either explicitly or implicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or mitigates any or all of the problems addressed by the present invention. The application hereby gives notice that new claims may be formulated to such features during the prosecution of this application or of any such further application derived therefrom. In particular with reference to the appended claims, features from dependant claims may be combined with those of the independent claims in any appropriate manner and not merely in the specific combinations enumerated in the claims.

WE CLAIM:

1. An integrated circuit card comprising:

a microprocessor; a volatile memory coupled to said microprocessor; a non-volatile memory coupled to said microprocessor; and a plurality of applications stored in said non-volatile memory, wherein upon execution of each said application, said microprocessor allocates for each said executing application an associated data memory space comprising at least a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data; and further comprising means for delegating the performance of a function from a first executing application to a second executing application.

2. The integrated circuit card of claim 1, wherein said non-volatile memory segment is divided into at least two regions, including a public region and a dynamic region.

3. The integrated circuit card of claim 2, wherein said public region is used to share data between said first and second applications.

4. The integrated circuit card of claim 2 or claim 3, wherein said dynamic region is used to reference temporary data utilized during an application's execution.

5. The integrated circuit card of any preceding claim, further comprising at least one register coupled to said microprocessor which is used to determine the starting locations of each of said segments.

6. The integrated circuit card of any preceding claim, further comprising at least one register coupled to said microprocessor which is used to determine the top locations of each of said segments.

7. The integrated circuit card of any preceding claim, further comprising at least one register coupled to said microprocessor which is used as a local dynamic pointer.

8. The integrated circuit card system of any preceding claim, wherein each said application comprise a plurality of program instructions and wherein at least one of said program instructions when executed causes said memory referenced by said volatile memory segment to be accessed.

9. The integrated circuit card of any preceding claim, wherein said volatile memory segment references RAM and said non-volatile memory segment references EEPROM.

10. A method for processing a plurality of applications stored in a memory of an integrated circuit:

selecting a first application for execution;

allocating a data space for said first application including at least two memory segments comprising a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data;

executing said first application, interrupting execution of said first application and saving data referenced by said volatile memory segment;

executing a second application;

utilizing said saved data from said volatile memory segment for execution of said first application; and

completing said execution of said first application.

11. The method of claim 10, wherein said first application's identity is stored in a data stack during said delegation step.

12. The method of claim 11, wherein said data stack is accessed following said completion of said second application.

13. The method of claim 11 or 12, further including the step of inquiring said first application's identity by accessing said delegator stack.

14. The method of any of claims 10 to 13, wherein said non-volatile memory segment is divided into at least two regions, including a public region and a dynamic region.

15. The method of claim 14, wherein said public region is used to share data between said first application and said second application.

16. The method of claim 14 or 15, wherein data referenced by said dynamic region is utilized during the execution of said first application.

17. The method of any of claims 10 to 15, further including the step of allocating a second data space including at least two memory segments for said second application.

18. The method of claim 17, wherein said second data space's segments comprise a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data.

19. The method of claim 18, wherein said second application's non-volatile segment is divided into at least two regions, including a public region and a dynamic region.

20. The method of claim 19, wherein said second application's public region is used to share data between said first and second applications.

21. The method of claim 19 or 20, wherein said data referenced by second application's dynamic region is utilized during said execution of said second application.

22. The method of any of claims 10 to 21, further including the step of delegating use of said microprocessor from said second application to a third application stored on said IC card.

23. The method of claim 22, wherein a third data space for said third application is allocated which includes a volatile memory segment for referencing temporary data and non-volatile memory segment for referencing static data, wherein said third application's volatile segment includes a public and dynamic portion.

24. An apparatus for processing a plurality of applications stored in a memory of a single integrated circuit card comprising:

means for allocating a data space comprising at least a non-volatile memory segment for referencing static data and a volatile memory segment for referencing temporary data; means for executing a first application; means for interrupting execution of said first application, means for saving data from at least a portion of said volatile memory segment; and means for executing a second application; means for retrieving said saved data; and means for completing said execution of said first application.

25. The apparatus of claim 24, further including means for storing said first application's identity on a data stack.

26. The apparatus of claim 25, further including means for inquiring of said first application's identity.

27. The apparatus of any of claims 24 to 26, wherein said first application's non-volatile memory segment is divided into at least two regions, including a public region and a dynamic region.

28. The apparatus of claim 27, wherein said public region references random access memory.

29. The apparatus of claim 27 or 28, wherein said dynamic region references random access memory.

30. The apparatus of any of claims 24 to 29, further including means for allocating a second data space including at least two segments for said second application.

31. The apparatus of claim 30, wherein said second data space includes a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data.

32. The apparatus of claim 31, wherein said second data space's non-volatile segment is divided into at least two regions, including a public region and a dynamic region.

33. The apparatus of claim 32, wherein said public region references random access memory.

34. The apparatus of claim 32 or 33, wherein said dynamic region references random access memory.

35. The apparatus of any of claims 24 to 34, further including means for delegating operation of said IC card from said second application to a third application stored on said IC card.

36. The apparatus of claim 35, wherein a third data space for said third application is allocated which includes a volatile memory segment for referencing temporary data and non-volatile memory segment for referencing temporary data, wherein said third application's volatile memory segment includes a public and dynamic portion.

37. A system for processing a plurality of applications stored on an IC card comprising:

a non-volatile memory coupled to a databus;

a volatile memory coupled to said databus;

a first and second application program stored in said non-volatile memory, wherein each application has an associated identifier;

a data stack accessible by said databus for storing said applications' identifier if said application is interrupted during its execution;

processor means for executing instructions from said application programs wherein said processor means allocates a data memory space for said application which is being executed and said data memory space is mapped to at least one address in said non-volatile memory and at least one address in said volatile memory; and

wherein said processor means interrupts said first application at least once during its execution to execute said second application.

38. The system of claim 37, wherein data memory space comprises at least a volatile memory segment for referencing temporary data stored in said volatile memory and a non-volatile memory segment for referencing static data stored in said non-volatile memory.

39. The system of claim 37 or 38, further including means for storing said first application's identity on a data stack.

40. The system of any of claims 37 to 39, further including means for inquiring of said first application's identity.

41. The system of any of claims 38 to 40, wherein said first application's non-volatile memory segment is divided into at least two regions, including a public region and a dynamic region.

42. The system of claim 41, wherein said public region references random access memory.

43. The system of claim 41 or 42, wherein said dynamic region references random access memory.

44. The system of any of claims 37 to 43, further including means for allocating a second data space including at least two segments for said second application.

45. The system of claim 44, wherein said second data space comprises at least a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data.

46. The system of claim 45, wherein said second data space's non-volatile segment is divided into at least two regions, including a public region and a dynamic region.

47. The system of claim 46, wherein said public region references random access memory.

48. The system of claim 46, wherein said dynamic region references random access memory.

49. The system of any of claims 37 to 48, further including means for delegating use of said processor means from said second application to a third application stored on said IC card.

50. The system of claim 49, wherein a third data space for said third application is allocated which includes a volatile memory segment for referencing temporary data and non-volatile memory segment for referencing temporary data, wherein said third application's volatile memory segment includes a public and dynamic portion.

51. An integrated circuit card comprising:

a plurality of applications and a microprocessor for controlling execution of said applications wherein execution of at least one first application is interrupted and execution is transferred to another second application, further comprising means for sharing data by said first and second applications and means for resuming execution of said first application at the appropriate location at least after completion of execution of said second application.

52. The integrated circuit card of claim 51, further comprising means for allocating a data memory space comprises at least a volatile memory segment for referencing temporary data stored in said volatile memory and a non-volatile memory segment for referencing static data stored in said non-volatile memory.

53. The integrated circuit card of claim 51 or 52, further including means for storing said first application's identity on a data stack.

54. The integrated circuit card of any of claims 51 to 53, further including means for inquiring of said first application's identity.

55. The integrated circuit card of claim 52, wherein said first application's non-volatile memory segment is divided into at least two regions, including a public region and a dynamic region.

56. The integrated circuit card of claim 55, wherein said public region references random access memory.

57. The integrated circuit card of claim 55 or 56, wherein said dynamic region references random access memory.

58. The integrated circuit card of any of claims 52 to 57, further including means for allocating a second data space including at least two segments for said second application.

59. The integrated circuit card of claim 58, wherein said second data space comprises at least a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data.

60. The integrated circuit card of claim 59, wherein said second data space's non-volatile segment is divided into at least two regions, including a public region and a dynamic region.

61. The integrated circuit card of claim 60, wherein said public region references random access memory.

62. The integrated circuit card of claim 60 or 61, wherein said dynamic region references random access memory.

63. The integrated circuit card of any of claims 51 to 62, further including means for delegating use of said processor means from said second application to a third application stored on said IC card.

64. An integrated circuit card comprising:

a microprocessor; a volatile memory coupled to said microprocessor; a non-volatile memory coupled to said microprocessor for storing a plurality of applications, and adapted for each application to be executed to allocate an associated data memory space comprising at least a volatile memory segment for referencing temporary data and a non-volatile memory segment for referencing static data prior to execution of each said application and further comprising means for delegating the performance of a function from a first executing application to a second executing application.

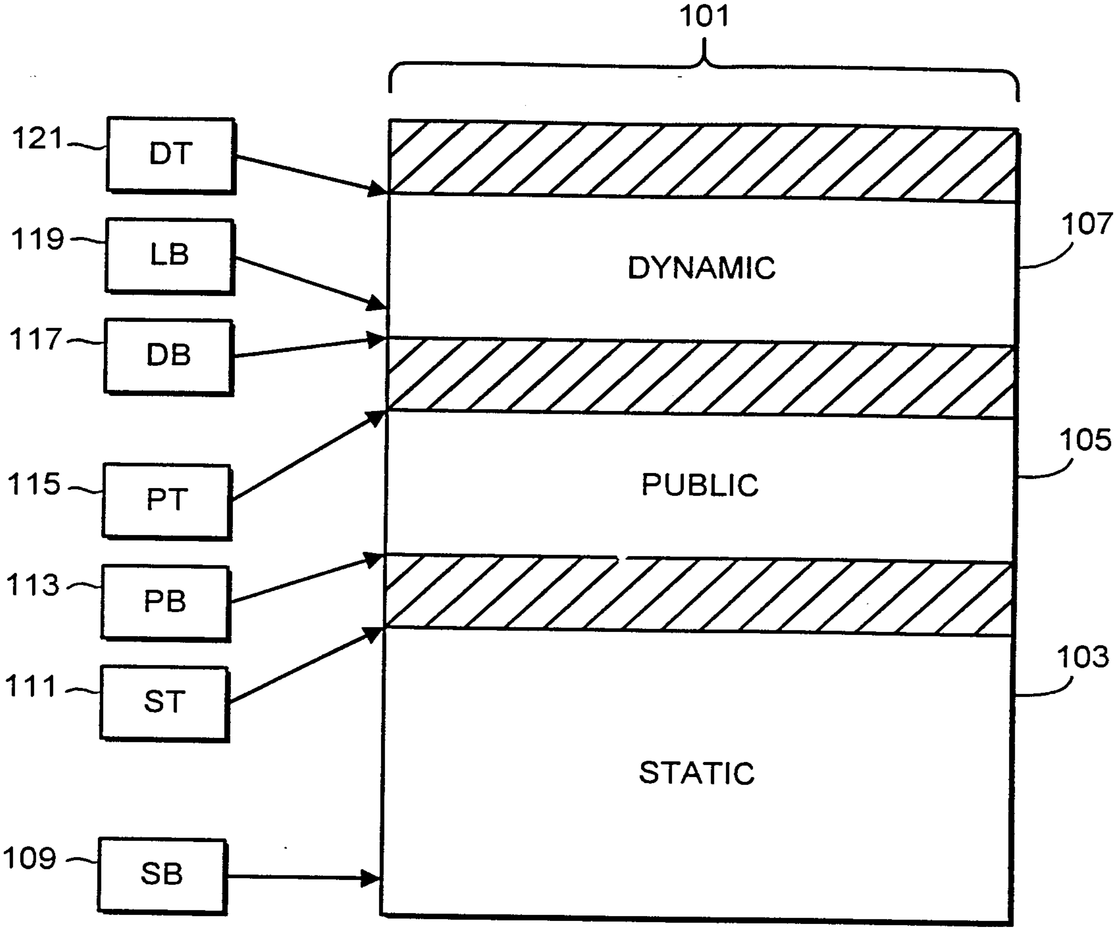

101
121
DT
119
LB
117
DB
115
PT
113
PB
111
ST
109
SB
DYNAMIC
107
PUBLIC
105
STATIC
103

FIG. 1

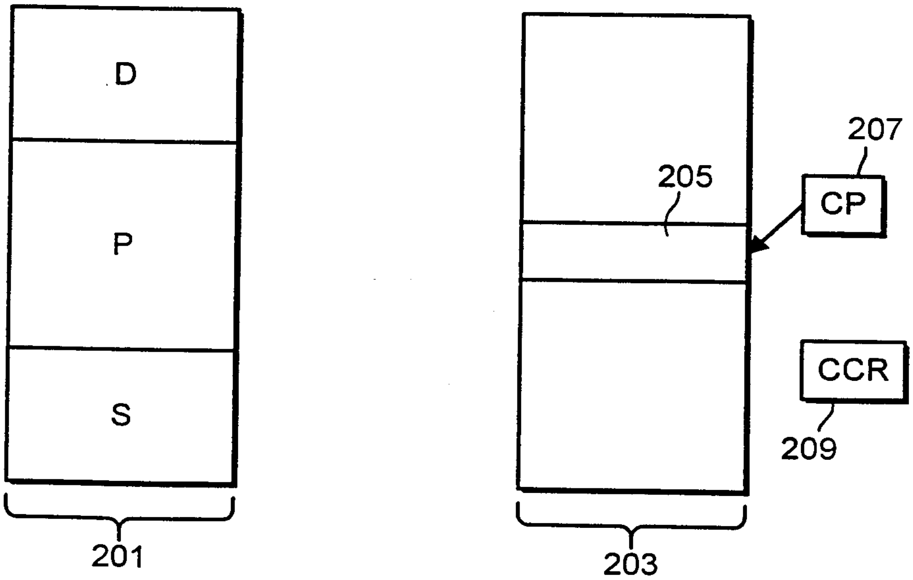

D
P
S
201
205
207
CP
CCR
209
203

FIG. 2

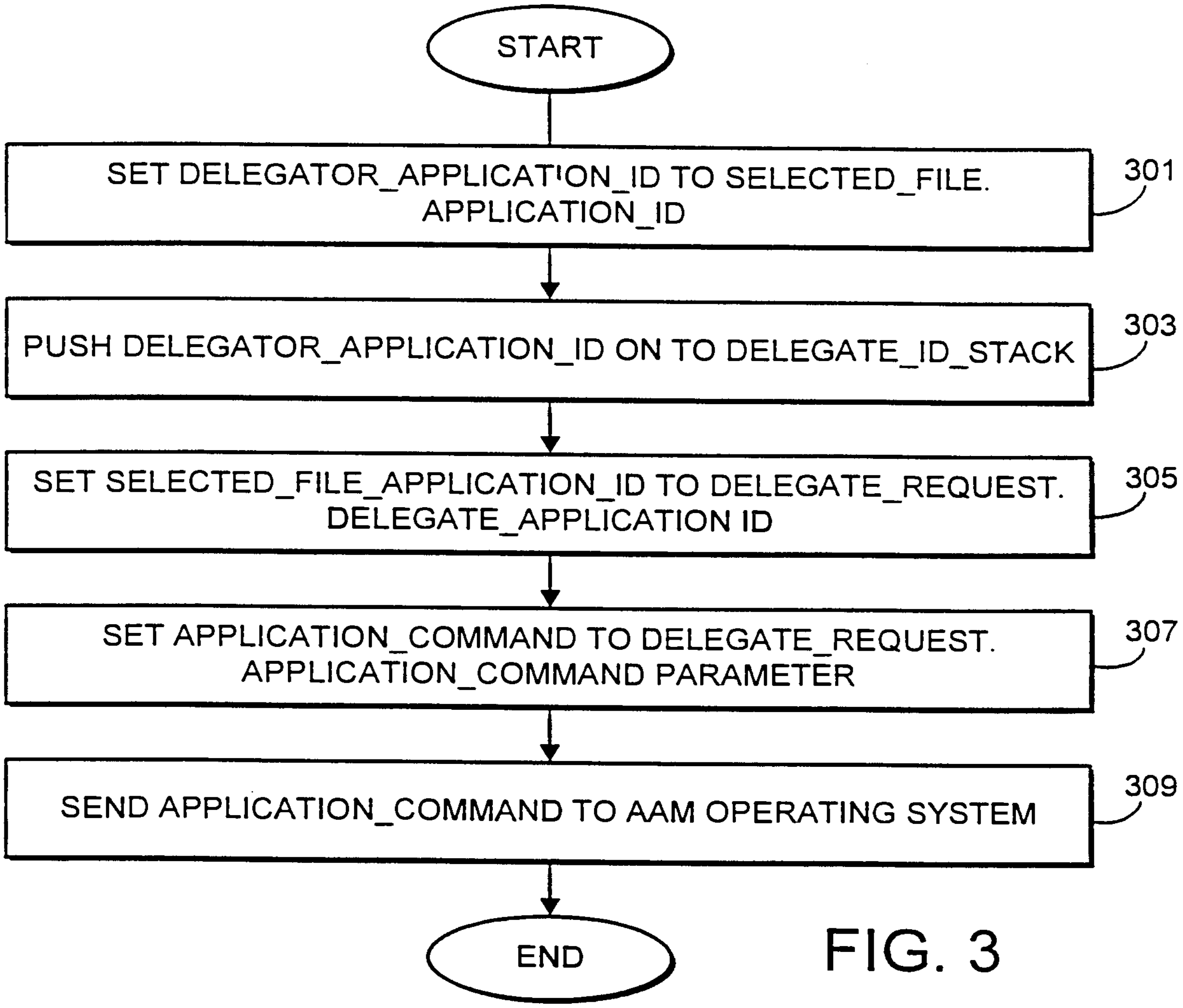

START
SET DELEGATOR_APPLICATION_ID TO SELECTED_FILE. APPLICATION_ID
301
PUSH DELEGATOR_APPLICATION_ID ON TO DELEGATE_ID_STACK
303
SET SELECTED_FILE_APPLICATION_ID TO DELEGATE_REQUEST. DELEGATE_APPLICATION ID
305
SET APPLICATION_COMMAND TO DELEGATE_REQUEST. APPLICATION_COMMAND PARAMETER
307
SEND APPLICATION_COMMAND TO AAM OPERATING SYSTEM
309
END

FIG. 3

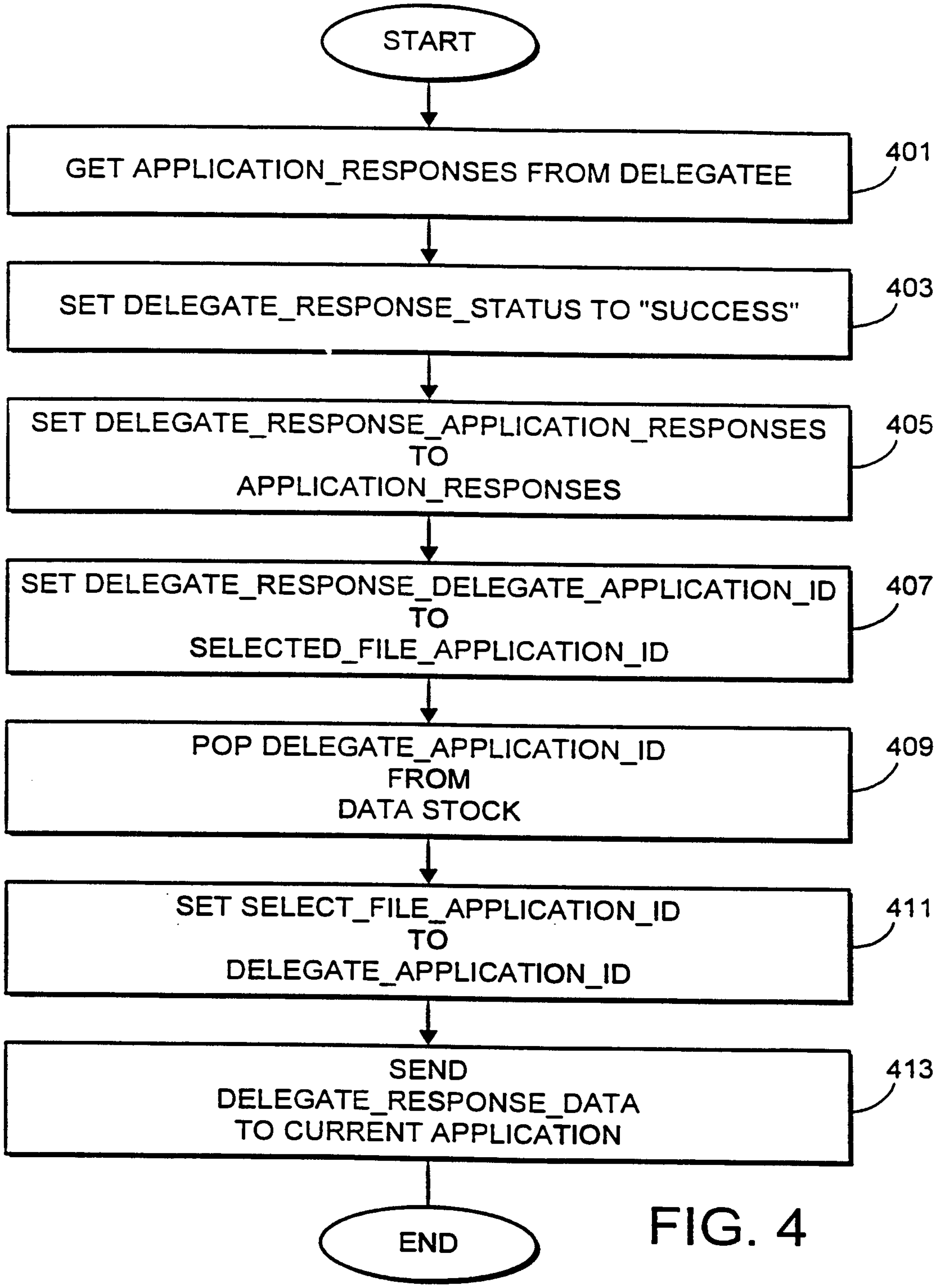

START
GET APPLICATION_RESPONSES FROM DELEGATEE
401
SET DELEGATE_RESPONSE_STATUS TO "SUCCESS"
403
SET DELEGATE_RESPONSE_APPLICATION_RESPONSES TO APPLICATION_RESPONSES
405
SET DELEGATE_RESPONSE_DELEGATE_APPLICATION_ID TO SELECTED_FILE_APPLICATION_ID
407
POP DELEGATE_APPLICATION_ID FROM DATA STOCK
409
SET SELECT_FILE_APPLICATION_ID TO DELEGATE_APPLICATION_ID
411
SEND DELEGATE_RESPONSE_DATA TO CURRENT APPLICATION
413
END

FIG. 4

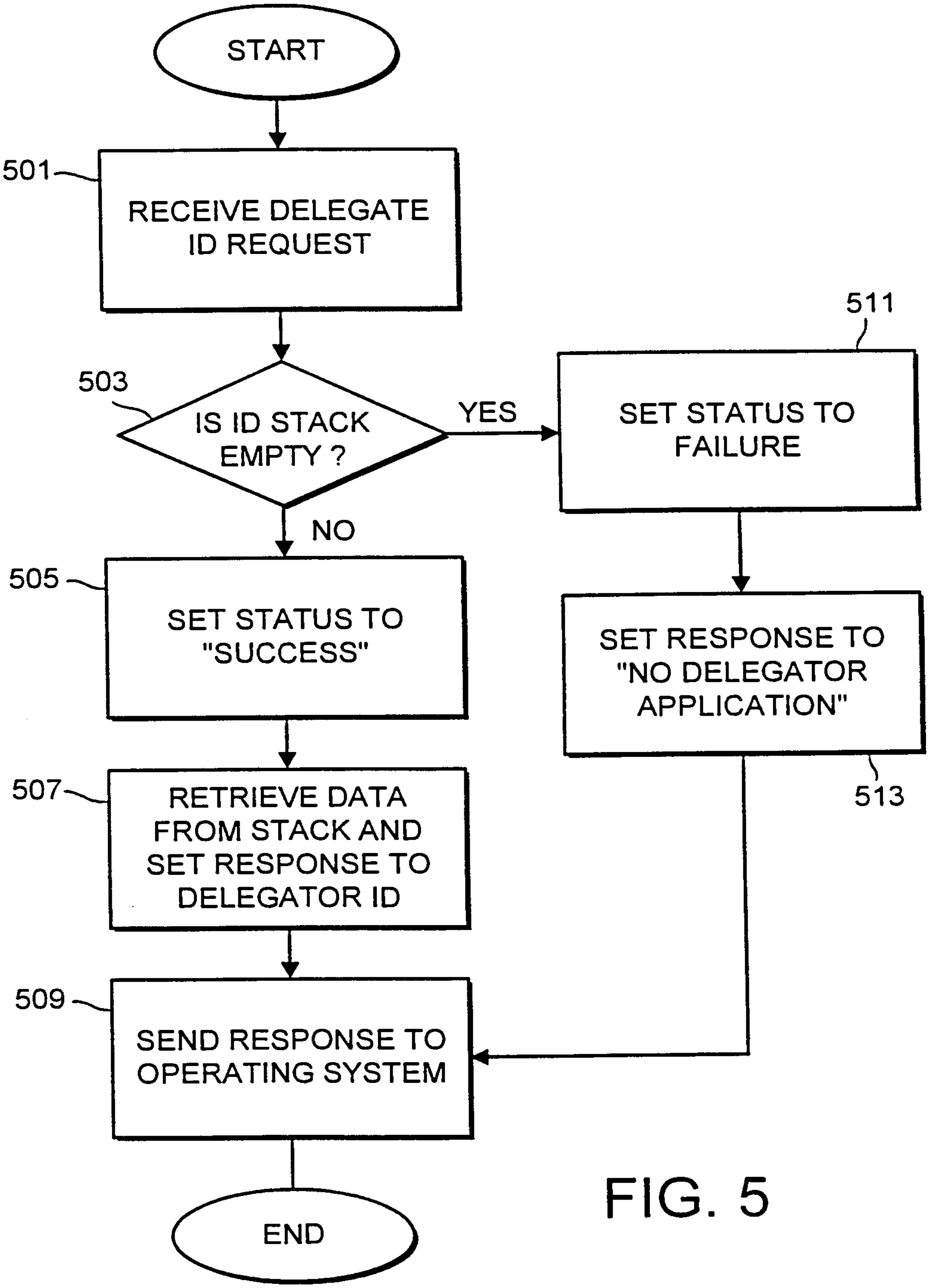

START
501
RECEIVE DELEGATE ID REQUEST
503
IS ID STACK EMPTY ?
YES
NO
511
SET STATUS TO FAILURE
505
SET STATUS TO "SUCCESS"
SET RESPONSE TO "NO DELEGATOR APPLICATION"
513
507
RETRIEVE DATA FROM STACK AND SET RESPONSE TO DELEGATOR ID
509
SEND RESPONSE TO OPERATING SYSTEM
END

FIG. 5

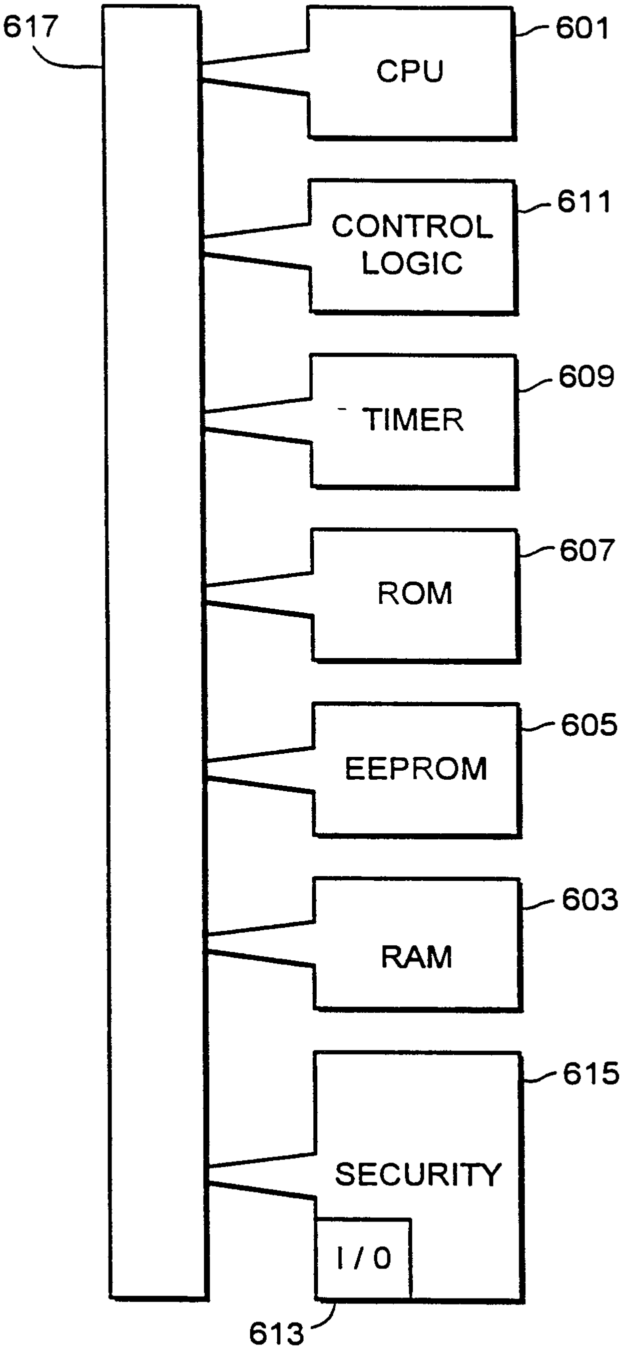

617
CPU
601
CONTROL LOGIC
611
TIMER
609
ROM
607
EEPROM
605
RAM
603
SECURITY
615
I / 0
613

FIG. 6

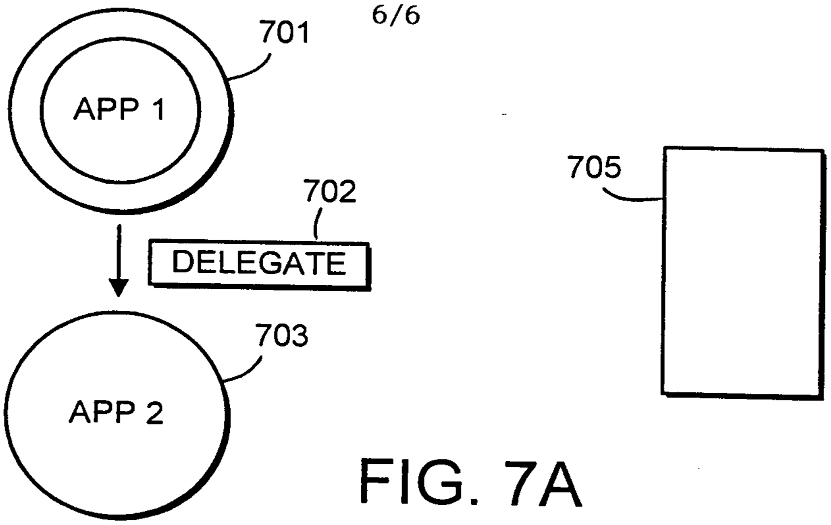


FIG. 7A

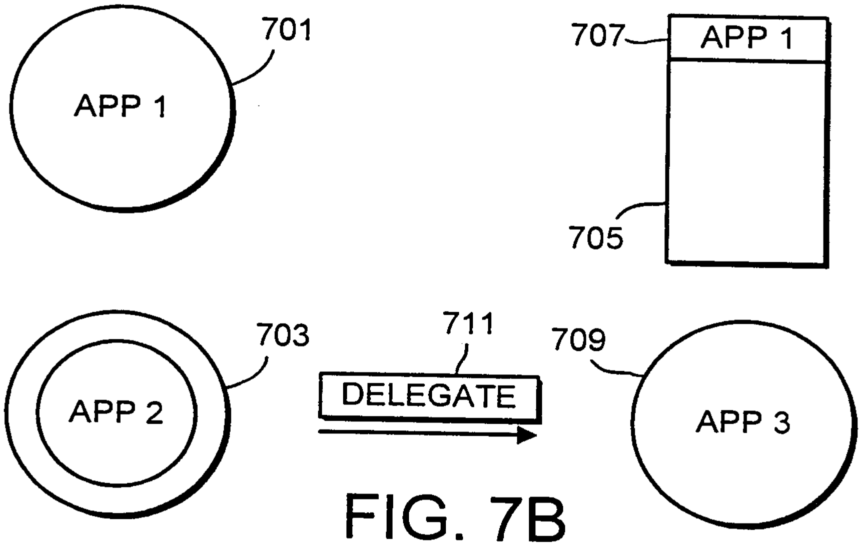


FIG. 7B

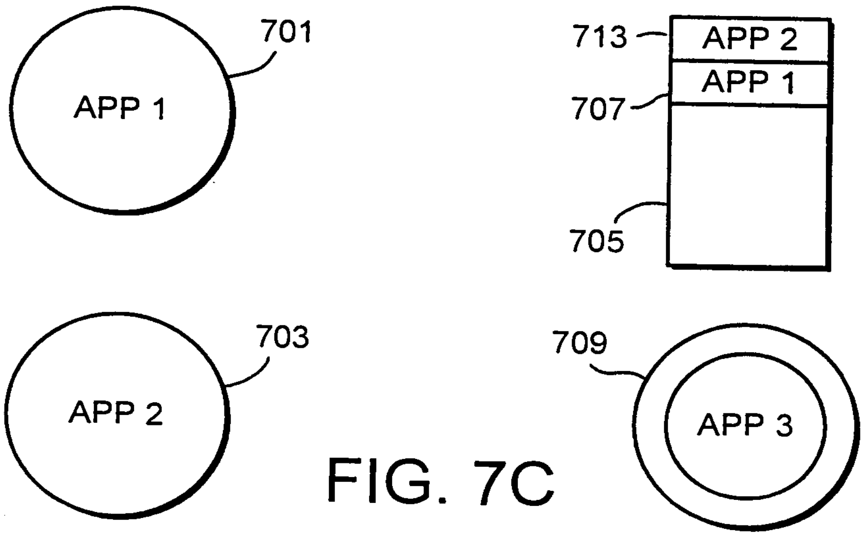


FIG. 7C

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G07F 7/10	A2	(11) International Publication Number: WO 98/52160 (43) International Publication Date: 19 November 1998 (19.11.98)

(21) International Application Number: PCT/GB98/01393

(22) International Filing Date: 14 May 1998 (14.05.98)

(30) Priority Data:

60/046,514	15 May 1997 (15.05.97)	US
09/076,550	12 May 1998 (12.05.98)	US

(71) Applicant: MONDEX INTERNATIONAL LIMITED [GB/GB]; 47–53 Cannon Street, London EC4M 5SQ (GB).

(72) Inventors: EVERETT, David, Barrington; 31 Ashdown Avenue, Saltdean, Brighton, East Sussex BN2 8AH (GB). MILLER, Stuart, James; 9 Woodford Green, The Warren, Bracknell, Berkshire RG12 9YQ (GB). PEACHAM, Anthony, David; 4 Lynwood, Groombridge, Tunbridge Wells, Kent TN3 9LX (GB). SIMMONS, Ian, Stephen; The Elms, School Road, Broughton, Cambridgeshire PE17 3AT (GB). RICHARDS, Timothy, Philip; 32 Graig Mount, Radlett, Hertfordshire WD7 7LW (GB). VINER, John, Charles; Hydes, Woodlands Lane, Windlesham (GB).

(74) Agent: POTTER, Julian, Mark; D. Young & Co., 21 New Fetter Lane, London EC4A 1DA (GB).

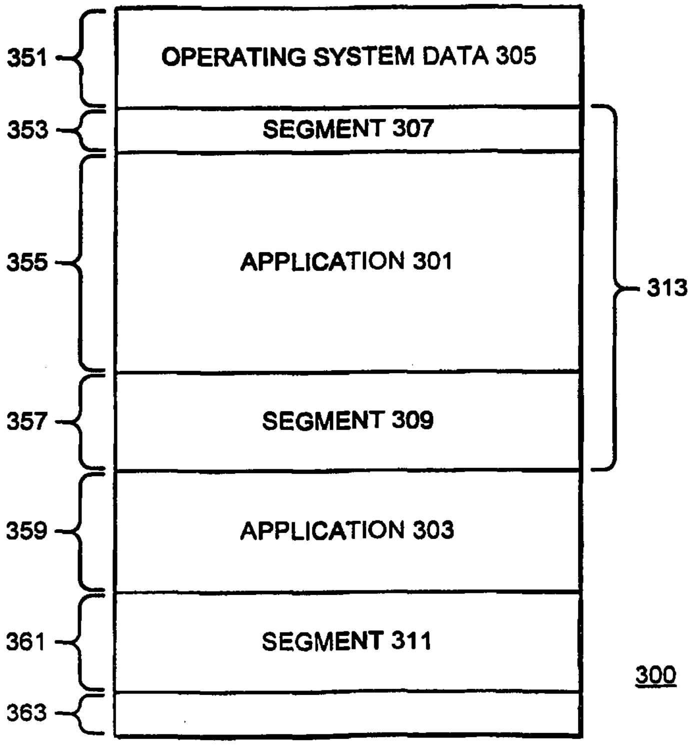
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW,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

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

(54) Title: SYSTEM AND METHOD FOR FLEXIBLY LOADING AN IC CARD

(57) Abstract

A system and method of flexibly loading an application and its associated data from an application provider onto an IC card. The application and its associated data is divided into segments which can each fit into the input buffer of an Integrated circuit card. Each segment is transmitted separately and the Integrated circuit card then stores the segment in an available space in the IC card's memory. The segments can be placed in non–contiguous memory in order to reduce memory fragmentation.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

SYSTEM AND METHOD FOR FLEXIBLY LOADING AN IC CARD

BACKGROUND OF INVENTION

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. This allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application when it is manufactured and before it is given to a card user. That application, however, cannot be modified or changed after the card is issued even if the modification is desired by the card user or card issuer. Moreover, if a card user wanted a variety of application functions to be performed by IC cards issued to him or her, such as both an electronic purse and a credit/debit function, the card user would be required to carry multiple physical cards on his or her person, which would be quite cumbersome and inconvenient. If an application developer or card user desired two different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal, making the transaction difficult, lengthy and inconvenient.

Therefore, it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card. Although multiple applications could be preselected and placed in the memory of the card during its production stage, it would also be beneficial to have the ability to load and delete applications for the card post-production as needed.

It is important, particularly where there is a continuing wide availability of new applications to the cardholder, that the system has the capability of adding applications onto the IC card subsequent to issuance. This is necessary to protect the longevity of the IC cards; otherwise, once an application becomes outdated, the card would be useless. It would be beneficial to allow the addition of applications from a remote location as well as from a direct connection to an application provider's terminal.

For example, it would be beneficial for a card user to be able to plug his IC card into his home computer and download an application over the Internet. Alternatively, it would be beneficial for an application provided by Bank A to be loaded from a terminal (such as an ATM) located at Bank B which is connected to Bank A by a network or series of interconnected networks.

The increased flexibility and power of storing multiple applications on a single card create new technical challenges to be overcome concerning the application loading process in which information (including application code and associated data) is exchanged between the application provider and the individual card. The IC card only has a finite amount of memory on the card for storing applications. Applications and their associated data can vary drastically in size depending upon the application. When multiple applications are stored on a card, and a series of application additions and deletions have occurred, memory fragmentation where memory which is free cannot be used because of size limitations.

Additionally, an IC card has limited space in its input buffer, which can be separate or combined with an output buffer, i.e., an Input/Output (I/O) buffer. It may not be possible to fit the entire application and its associated data into an I/O buffer of an IC card at one time. In order to achieve the flexibility of selectively loading and deleting applications on an IC card, the problems of limited I/O buffer space and fragmentation must be addressed.

Accordingly, it is an object of preferred embodiments of this invention to provide a system and method that allows for flexible loading of an application and its associated data onto an IC card by segmenting the application and associated data into selected segments in order to limit the size of the data packets being transmitted at one time and reduce fragmentation in the memory of the IC card.

SUMMARY OF THE INVENTION

These and other objectives are achieved by an embodiment of the present invention which provides an IC card system and method for flexibly loading an application and its associated data from an application onto an IC card. The application provider divides the application and its associated data into segments which will fit into the I/O buffer of the intended IC card. Each segment is transmitted separately and the IC card stores the segment in an available space in the IC card's memory. The segments can be placed in non-contiguous memory in order to reduce memory fragmentation. The IC card's microprocessor can additionally determine the smallest memory space which will store the segment in order to minimize fragmentation.

In a preferred embodiment, the application provider determines the size of the IC card's I/O buffer so that it can correctly select the size of each segment.

BRIEF DESCRIPTION OF THE DRAWINGS

Further objects, features and advantages of embodiments of the invention will become apparent from the following detailed description taken by way of example only and in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

Fig. 1 is block diagram of the flexible loading system of the present invention;

Fig. 2 is a block diagram of an IC card chip upon which an application and its associated data can be flexibly loaded and stored;

Fig. 3 is a graphic example of a memory map of EEPROM on an IC card;

Fig. 4 is a flow chart of an example of multiple segments being loaded onto the IC card;

Fig. 5 is a flow chart of the steps of segmenting the application and its associated data by the application provider; and

Fig. 6 is a flow chart of the steps of receiving and processing the segmented information by the IC card.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

DETAILED DESCRIPTION OF THE INVENTION

It is beneficial to have the capability to load applications onto IC cards containing multiple application operating systems at any time during the lifetime of the IC card. This flexibility allows a user of a card to periodically add new applications to the IC card and also allows older applications to be updated with newer versions of the application when they are released. For example, a card user may start with an IC card that contains a purse, or electronic cash application (e.g., MONDEX™), being stored on her IC card. Some time after the user has the card, she may load an additional application onto the card such as a credit/debit application. Some time after loading the credit/debit application on the card, a new version of the credit/debit application may become available and the card user should be able to erase the old application on her IC card and replace it with the new version of the credit/debit application which may contain additional features.

In a multiple application operating system environment, it is beneficial to be able to load applications both at terminals, such as a bank ATM machine, as well as over remote communication links, such as telephone lines, cable lines, the Internet, satellite or other communications means. When loading applications onto an IC card, the application provider and the card issuer (which could be the same entity) need to provide security regarding the applications to be loaded. The IC card has a limited amount of available I/O space and a limited amount of memory to store applications and their associated data. In order to address these limitations, the application and its associated data is segmented such that each segment will fit in the IC card's I/O buffer at one time. The segment is then stored in the IC card's storage memory, e.g., EEPROM, in a manner that can be non-contiguous to reduce memory fragmentation. This system and technique will now be described in detail.

Figure 1 shows a block diagram of the entities used in a remote application loading process of an application and its associated data. While Figure 1 shows a remote loading system, the flexible loading technique also applies to local loading such as a terminal located at the application provider. System 100 includes an application provider for transmitting an IC card application and its associated data to an intended IC card 103, an IC card interface device 105 and a data conduit 107. The application provider 101 can be a card issuer, bank or other entity which provides application loading services. The application provider 101 preferably initiates an application loading process onto IC card 103. Alternatively, the IC card 103 can request the loading process. Application Provider 101 is connected to data conduit 107 which is connected to interface device 105 (e.g., a terminal that communicates with an IC card). Data conduit 107 can be a telephone line, an intranet, the Internet, a satellite link or any other type of communications link. The application provider 101, which is remotely located from the IC card 103 in this example, desires to send and load an application to the IC card. Application provider 101 has an I/O buffer 113 and IC card 103 has an I/O buffer 115. In addition, interface device 105 also contains an I/O buffer 117. Each of the I/O buffers has a maximum storage capacity. The I/O buffers could be a combined input or output buffer or the input buffer and output buffer could be separate. However, the IC card 103 will typically have the smallest I/O buffer due to physical size limitations. The IC card 103 also has a memory 119 in which it stores the loaded application and its associated data.

In the illustrative embodiment of Figure 1, the application provider 101 sends two application segments S1, 109 and S2, 111 to the interface device 105 which is coupled to IC card 103. The application segments are discussed in more detail in connection with Figure 4. The application and its associated data are broken into two or more segment units in order for each of the data segments to fit in the I/O buffer of the I/O card. Additionally, the segmentation of the application and associated data helps to reduce fragmentation of the memory of the IC card which stores the application and associated data being loaded.

Figure 1 shows two segments 109 and 111 which are transferred at discrete times from the application provider to the IC card. However, any number of segments could be used depending upon such factors including the size of the application being loaded, the size of the associated data being loaded, the size of the respective I/O buffers, the availability of memory space on the IC card and the amount of memory fragmentation already on the IC card.

The application could be loaded directly at a terminal and not remotely. In that case, a separate interface device 105 would not be required because the application provider would have its own terminal capable of communicating with the IC card. For example, a bank could load an application onto an IC card by requiring the customer to insert his or her card into the bank's ATM machine. In that case, the application provider communicates with the IC card locally and transmissions are not sent over telephone lines or the Internet. Embodiments of the present invention are applicable to both the remote loading and local loading.

Figure 2 shows an example of a block diagram of an IC card chip upon which an application can be flexibly loaded and stored. An integrated circuit is located on an IC card for use. The IC card preferably includes a central processing unit 201, a RAM 203, an EEPROM 205, a ROM 207, a timer 209, control logic unit 211, an I/O port 213 and security circuitry 215, which are connected together by a conventional data bus.

Control logic 211 in memory cards provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports. CPU 201 with its control logic can perform calculations, access memory locations, modify memory contents, and manage input/output ports. Some cards have a coprocessor for handling complex computations like performing cryptographic operations. Input/output ports 213 are used under the control of a CPU and control logic, for communications between the card and a card interface device. Input/Output ports 213 include an I/O buffer. Timer 209 (which generates or provides a clock pulse) drives the control logic 211 and CPU 201 through the sequence of steps that accomplish memory access, memory reading or writing, processing, and data communication. A timer may be used to provide application features such as call duration. Security circuitry 215 preferably includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed ("blown") upon completion of testing to prevent later access. The application segments are stored in EEPROM 205. The storage and memory management process as described herein is performed by the CPU 201.

Figure 2 also shows a possible configuration for the integrated circuit for the application provider. CPU 201 present in the integrated circuit for the application provider determines the size of the IC card's I/O buffer, controls the segmentation of the application and associated data described herein and performs any other necessary operation.

Figure 3 shows a graphic representation of a memory map of EEPROM 300 on IC card 103. In this illustrative example, three applications are stored in EEPROM of an IC card. The first application 301 is stored in a contiguous memory space 355. Contiguous memory space means that the application occupies sequential memory addresses with no skipped memory addresses. A second application 303 is stored in contiguous memory space 359. Operating system data required for the execution of the operating system is stored in memory space 351. One example of a cause of fragmentation existing in the IC card is a previous application being deleted which was previously located at memory space 313. The next application loaded onto the IC card after the initial application was deleted can be a different size than the initial application and thus not all the freed up available memory space can be used in such a manner where two or more programs and data are stored contiguously without leaving small portions of unused memory space. In the example of Figure 1, the last application and its associated data which was loaded was segmented into three segments 307, 309 and 311. These segments are smaller portions of the entire application and its associated data set which could be placed in smaller areas of available memory. Thus fragmentation in the IC card's memory was alleviated by segmenting the application and its associated data.

The operating system stored on the card maintains a record of the physical location of the different segments and can access the physical locations when a logical address is called out when a program or operating system is being executed. The physical address look-up data can be stored in a table, a stack, a pointer or any conventional means for indicating the physical locations. Memory space 363 in Figure 3 is shown as not storing any data in the example and that memory space could be later used for storing new segmented applications and their associated data.

Figure 4 shows a flow chart of an illustrative example of loading multiple segments into a memory of an IC card from an application provider. In this example, six initial segments are created to be loaded onto the IC card. Two of the segments are further divided into components which results in a total of nine segments individually being sent to the IC card.

Step 401 loads a segment corresponding to the program code of the application to be provided to the IC card. The program code includes the program instructions which will be executed by the microprocessor located on the IC card. If the code segment is too large to fit into the I/O buffer of either the IC card or the application provider, then the segment can be further split into two or more components which can be separately transmitted to the IC card. In Figure 4, three components are illustrated for the program code, components 413, 415 and 417. The components are preferably stored in contiguous memory locations in the memory of the IC card. However, the components can be stored in non-contiguous locations if component pointers or tables are supported by the operating system on the IC card.

Step 403 loads the application data segment onto the IC card. The application data segment includes necessary and optional data needed for the execution of the application code. For example, if the application is a credit/debit application, the card user's account number, identification data and credit limit may be needed for the application to run. Another example is a health related application where a customer's medical history may be stored on the card for quick access at remote locations. The medical history data may be quite large and require further segmentation into two or more components. In Figure 4, components 419 and 421 are shown as subsets of the data segment being loaded in step 403.

Step 405 loads a Key Transformation Unit (KTU) segment for the application being loaded. If the application is being loaded from a remote location, there is a need to make sure the transmission is secure from third party access. The KTU information preferably contains information regarding the encryption key used to encipher the application program and associated data. The key information is sent with the application because applications can be transmitted from any application provider to any IC card with an IC card system. Since different encryption techniques can be used by different application providers, the KTU information in necessary. However, the flexible loading technique also applies when no encryption scheme is used and this information could also be included in another segment depending upon its size.

Step 407 loads a file control segment onto the IC card. File control information preferably includes an application identifier, security information and application and data size requirements. The file control information will be used by the operating system on the IC card to process the application. While in this example the file control information is a separate segment, it could be included in another segment depending upon its size.

Step 409 loads a directory information segment onto the IC card. The directory information preferably includes the name of the application which can be used by the operating system to identify the application. For example, if a select file command is initiated by a terminal, the name of the file to be selected which accompanies the command will be recognized by the operating system on the IC card. If the MONDEX™ Purse is selected by a customer as a terminal, the terminal will send a command to the IC card in the form of a "Select File Mondex" and the IC card will correlate MONDEX with a previously loaded application with the directory name Mondex. While in this example the directory information is a separate segment, the information could be included in another segment depending upon its size.

Step 411 loads an application signature segment onto the IC card. The application signature segment preferably includes data signed with the digital signature of the application provider. This allows the IC card to verify that the application provider is the genuine application provider and not an imposter. The IC card verifies the signature with the public key of an asymmetric encryption key pair of the application provider. While in this example the application signature is a separate segment, the information could be included in another segment if its size permitted it.

The segments could be organized in any manner and sent in any order. The IC card will need to have identified the subject matter of the incoming segment or component so that it can later locate a specific segment or component when needed. This information can be part of the load control information or can be obtained prior to the loading of the application. While Figure 4 describes a number of different segments, the subject matter of the segments transmitted will vary and depend upon the particular application and associated data.

Figure 5 shows a flowchart of the steps the application provider performs when segmenting the application and associated data to be loaded upon the IC card. Step 501 determines the I/O buffer size of the IC card. Alternatively, the input buffer size is determined if the input and output buffers are separate on the IC card. In most cases, the IC card I/O buffer will be smaller than the application provider I/O buffer because of the limited memory on the IC card. However, if the application provider I/O buffer or the Interface I/O buffer is smaller than the IC card I/O buffer, the smallest I/O buffer will control the size of the segments. The application provider can determine the IC card memory buffer size by some preliminary information exchange which identifies the IC card as the correct card upon which to load the application. Alternatively, some agreement or standard can be followed so that the application provider can create segments which will fit in an IC card which follows the agreement or standard.

Step 503 then segments the application and associated data in two or more segments. In the example of Figure 4, six initial segments were created and some of the segments were further divided to form two or more components. The segmented information is preferably divided in a predetermined organization to aid the IC card processing of the segments.

Step 505 then sends the segments to the IC card one at a time. When the IC card receives a segment in its I/O buffer, it will store that segment in a location of its memory thus freeing up its I/O buffer for the next incoming segment. After all the segments have been transmitted, the application provider can send a transmission indicating no more segments are being transmitted or the number of segments can be sent at the beginning of the transmission. Alternatively, a known segment protocol can be followed.

Figure 6 is a flow chart of the steps of processing the segmented information performed by the IC card. Step 601 receives a transmitted segment in the I/O buffer of the IC card. The entire segment will fit within the I/O buffer because of the processing performed at the application provider. Step 603 then stores the segment in available memory space after the microprocessor on the IC card identifies the proper memory space. The processor can check for the first available free memory space that is sufficient to store the segment. Once the segment is stored at a physical location, that location is recorded either in a segment address table, by a pointer or by any other conventional means. Different memory architectures can be used for storing the segments. For example, all the similar types of segments (e.g., program code) for the stored applications can be stored contiguously if desired. Alternatively, the processor can determine the space that is closest in size to the segment to be stored by scanning the memory. This will reduce any problems of fragmentation in the limited size IC card memory.

Step 605 determines if there are any additional segments to be stored. This step can be accomplished by checking earlier information regarding the number of segments which were being sent. It can also be accomplished by receiving a transmission indicating no more segments. Alternatively, the IC card can simply remain in a wait status until additional data or instructions is sent to the card. If the IC card determines that additional segments are being transmitted, the technique jumps back to step 602. If no more segments, the process ends.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

For example, while loading an application and its associated data is discussed herein, the same flexible loading process can apply to transmitting other types of data such as data blocks, database files, word processing documents or any other type of data requiring to be transmitted in a segmented manner.

The scope of the present disclosure includes any novel feature or combination of features disclosed therein either explicitly or implicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or mitigates any or all of the problems addressed by the present invention. The application hereby gives notice that new claims may be formulated to such features during the prosecution of this application or of any such further application derived therefrom. In particular, with reference to the appended claims, features from dependant claims may be combined with those of the independent claims in any appropriate manner and not merely in the specific combinations enumerated in the claims.

WE CLAIM:

1. A method for loading an application and its associated data from an application provider onto an integrated circuit card, wherein said integrated circuit card comprises a memory, comprising the steps of:

dividing said application and its associated data into a plurality of segments;

separately transmitting each said segment to said integrated circuit card; and

storing each said separately transmitted segment in an available area of said integrated circuit card's memory.

2. The method of claim 1, wherein at least two of said plurality of segments are not stored contiguously.

3. The method of claim 1 or claim 2, further including the step of determining an available area in said integrated circuit card's memory to store each said segment.

4. The method of claim 3, wherein said determining step identifies the smallest available area in said integrated circuit card's memory in which said segment can be stored.

5. The method of any preceding claim, wherein at least a first portion of said application is not stored contiguously with said application's remaining portion in said integrated circuit card's memory.

6. The method of any preceding claim, wherein said application is not stored contiguously with said associated data in said integrated circuit card's memory.

7. The method of any preceding claim, wherein said application is divided into a plurality of segments.

8. The method of any preceding claim, wherein said associated data is divided into a plurality of segments.

9. The method of any preceding claim, further including the step of determining said integrated circuit card's input buffer size.

10. A system for loading an application and its associated data onto an integrated circuit card comprising:

an application provider comprising means for dividing said application and its associated data into a plurality of segments and means for separately transmitting each said segment to said integrated circuit card; and

an integrated circuit card comprising a memory, means for receiving said transmitted segments and means for storing each said transmitted segment in an available area of said integrated circuit card's memory.

11. The system of claim 10, wherein at least two of said plurality of segments are not stored contiguously in said integrated circuit card.

12. The system of claim 10 or claim 11, wherein said card further includes means for determining an available area in said memory to store each said segment.

13. The system of claim 12, wherein said determining means identifies the smallest available area in which said segment can be stored.

14. The system of any of claims 10 to 13, wherein at least a first portion of said application is not stored contiguously with said application's remaining portion in said memory.

15. The system of any of claims 10 to 14, wherein said application is not stored contiguously with said associated data in said memory.

16. The system of any of claims 10 to 15, wherein said application is divided into a plurality of segments.

17. The system of any of claims 10 to 16, wherein said associated data is divided into a plurality of segments.

18. The system ofany of claims 10 to 17, wherein said application provider further includes means for determining said integrated circuit card's input buffer size.

19. The system of any of claims 10 to 18, wherein said means for receiving said transmitted segments has a size capacity smaller than said application and associated data's size.

20. The system of any of claims 10 to 19, wherein said integrated circuit card is remotely located from said application provider.

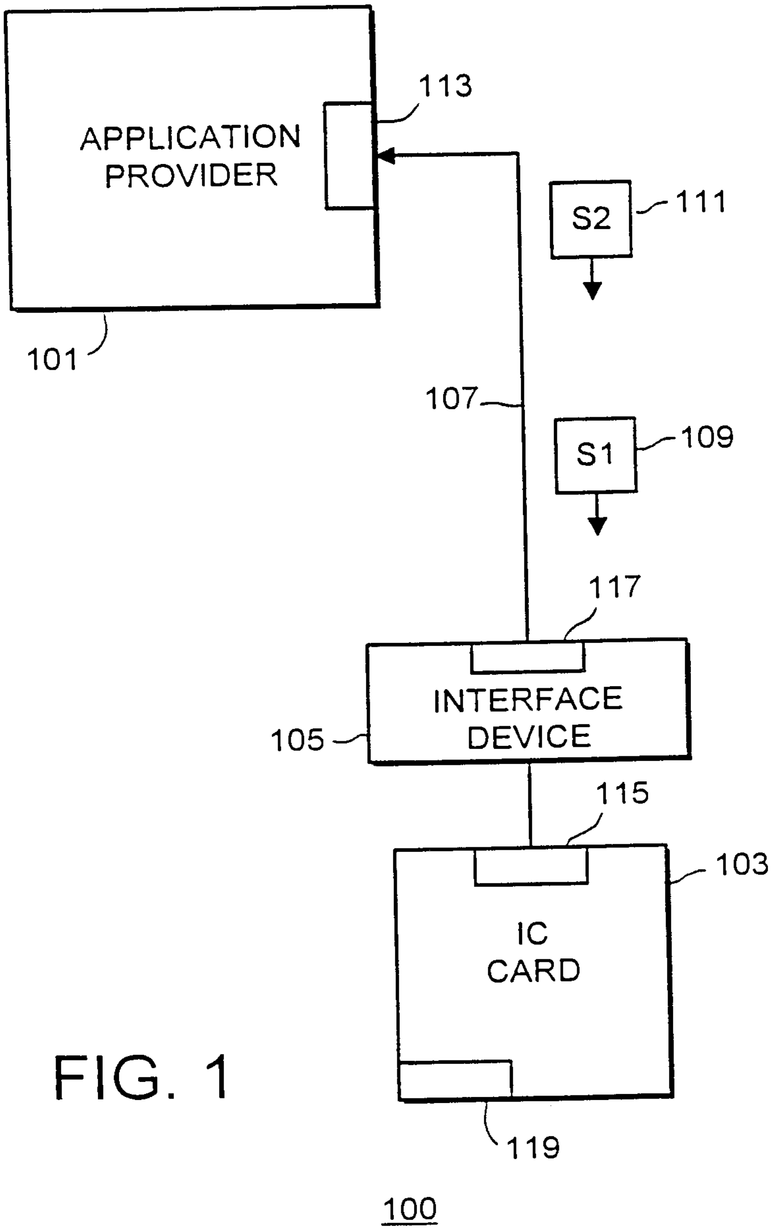

APPLICATION
PROVIDER
113
101
S2
111
107
S1
109
117
INTERFACE
DEVICE
105
115
103
IC
CARD
119
100

FIG. 1

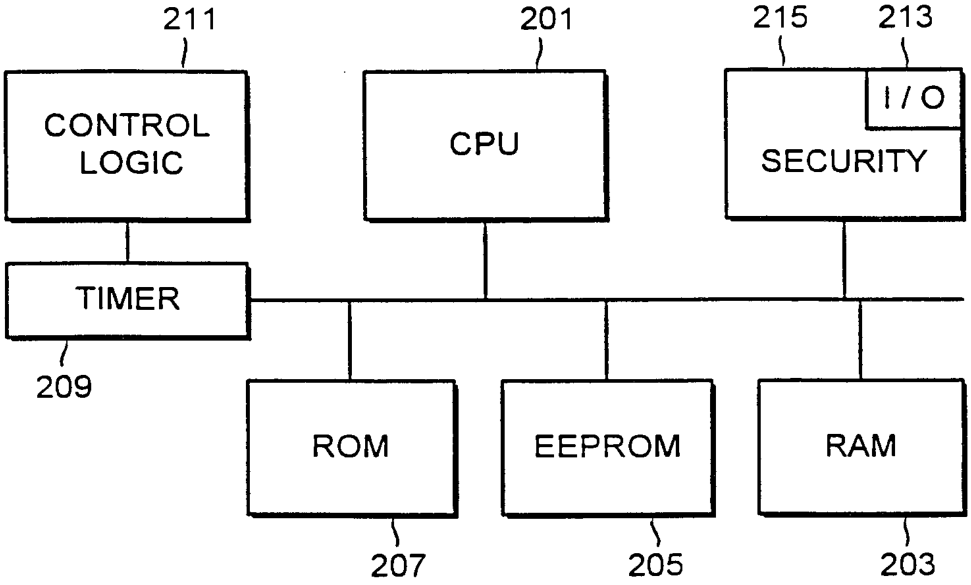

211
CONTROL LOGIC
201
CPU
215
213
I / O
SECURITY
TIMER
209
ROM
207
EEPROM
205
RAM
203

FIG. 2

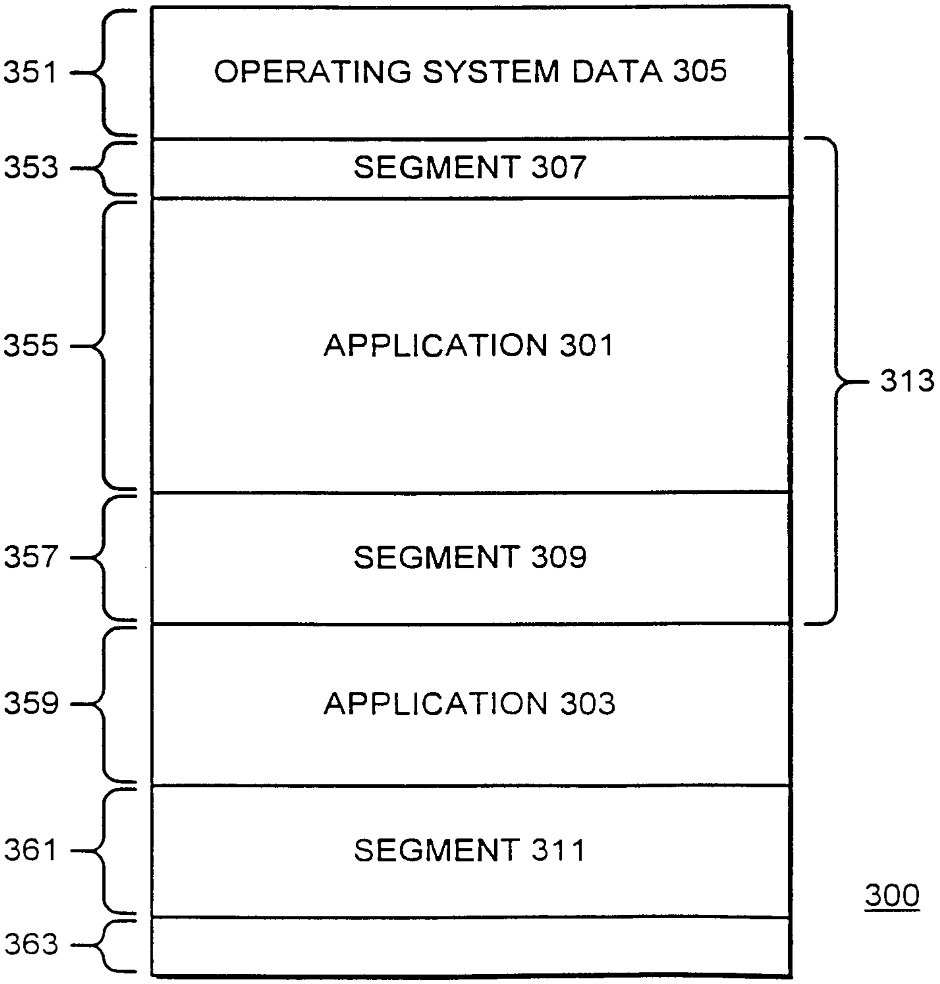

351
OPERATING SYSTEM DATA 305
353
SEGMENT 307
355
APPLICATION 301
313
357
SEGMENT 309
359
APPLICATION 303
361
SEGMENT 311
300
363

FIG. 3

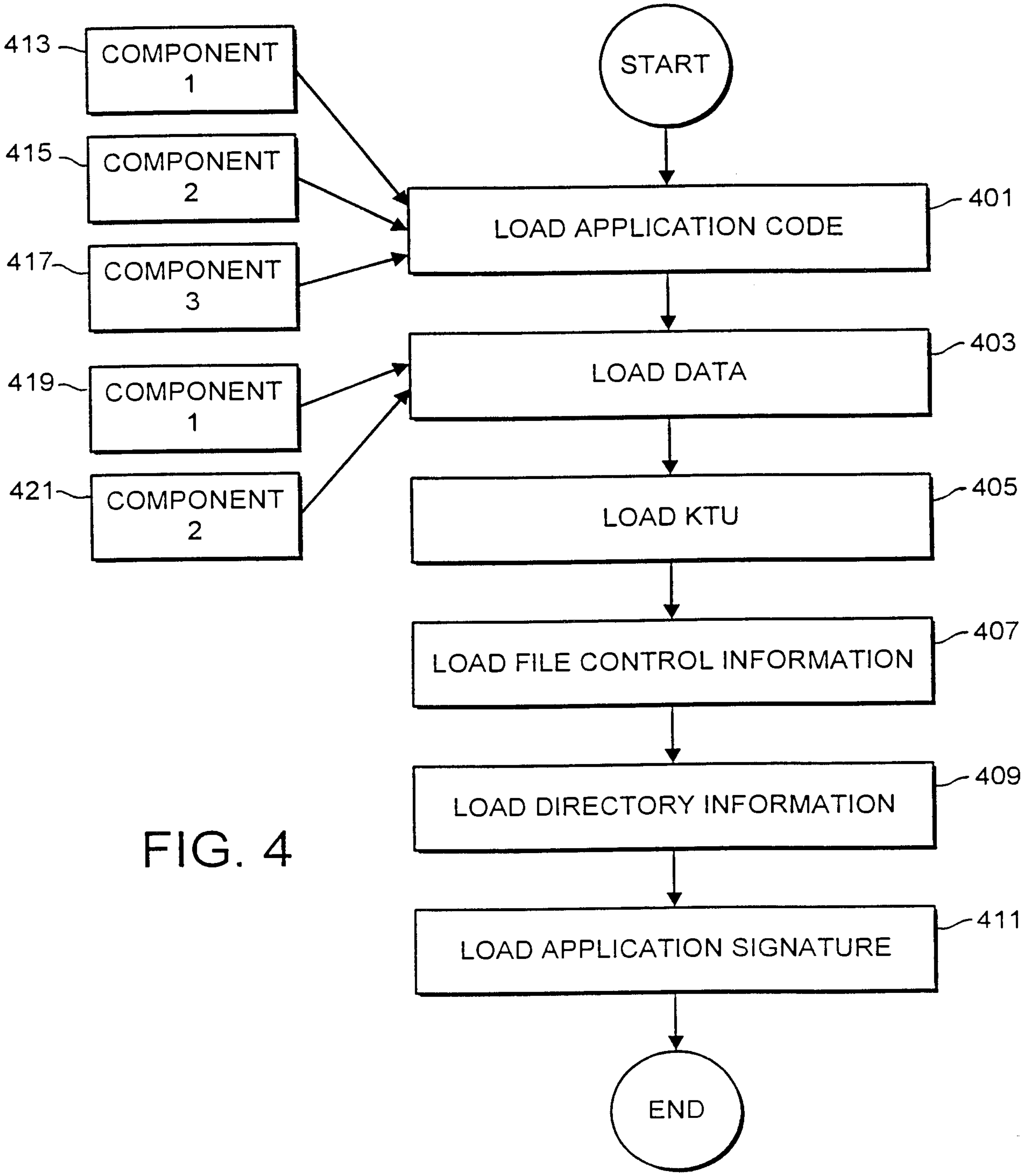

413
COMPONENT 1
415
COMPONENT 2
417
COMPONENT 3
419
COMPONENT 1
421
COMPONENT 2
START
LOAD APPLICATION CODE
401
LOAD DATA
403
LOAD KTU
405
LOAD FILE CONTROL INFORMATION
407
LOAD DIRECTORY INFORMATION
409
LOAD APPLICATION SIGNATURE
411
END

FIG. 4

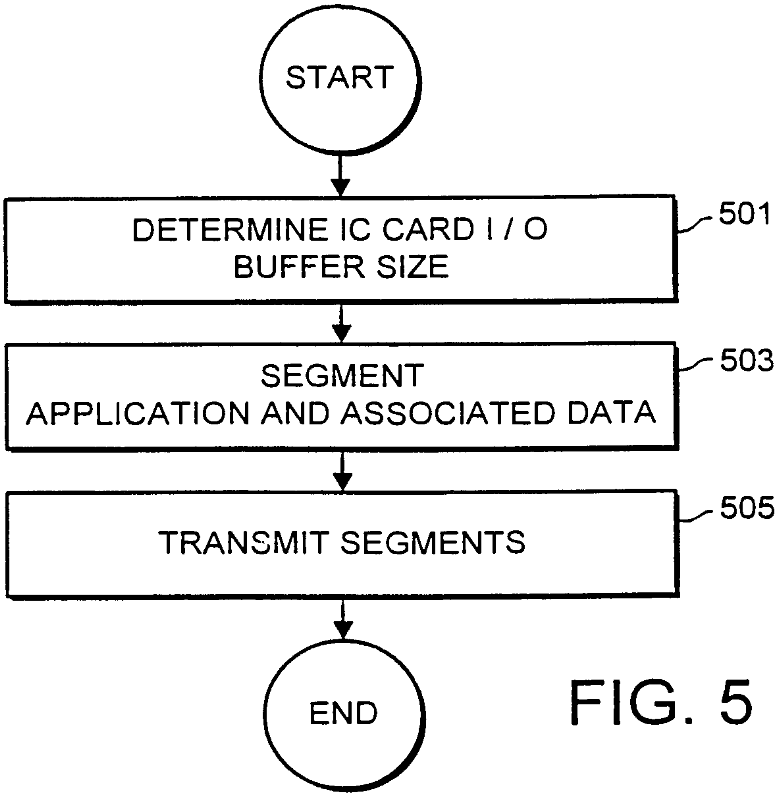

START
DETERMINE IC CARD I / O BUFFER SIZE
501
SEGMENT APPLICATION AND ASSOCIATED DATA
503
TRANSMIT SEGMENTS
505
END

FIG. 5

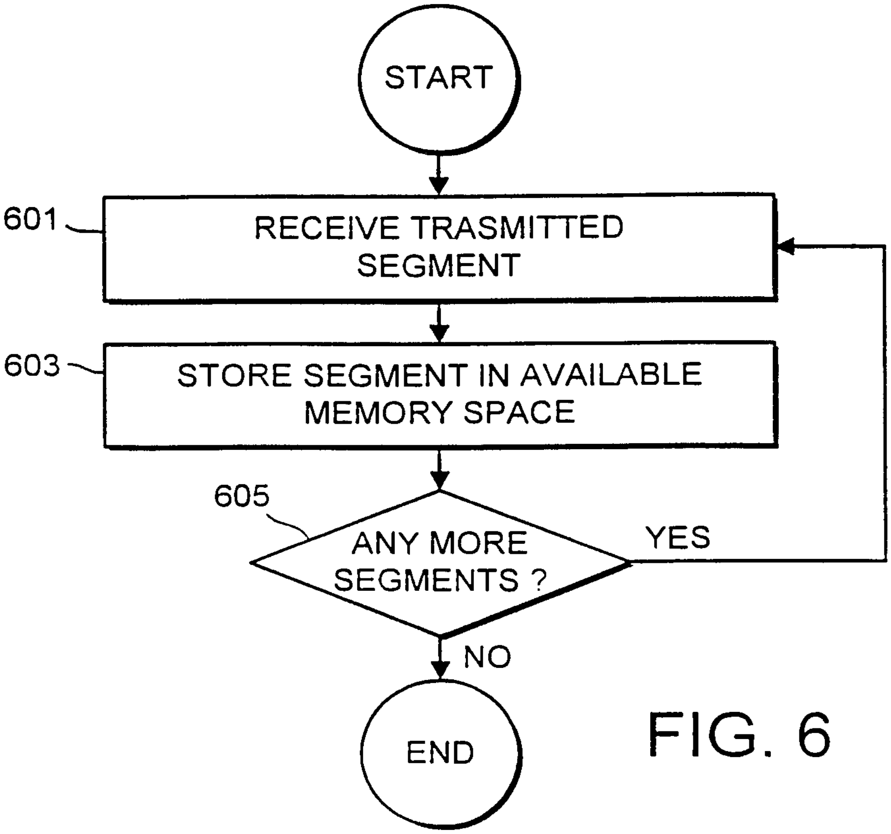

START
601
RECEIVE TRASMITTED SEGMENT
603
STORE SEGMENT IN AVAILABLE MEMORY SPACE
605
ANY MORE SEGMENTS ?
YES
NO
END

FIG. 6

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) **International Patent Classification ^{6} :**
G07F 7/10

A2

(11) **International Publication Number:** **WO 98/52161**

(43) **International Publication Date:** 19 November 1998 (19.11.98)

(21) **International Application Number:** PCT/GB98/01394

(22) **International Filing Date:** 14 May 1998 (14.05.98)

(30) **Priority Data:**

60/046,514	15 May 1997 (15.05.97)	US
09/075,974	11 May 1998 (11.05.98)	US

(71) **Applicant:** MONDEX INTERNATIONAL LIMITED [GB/GB]; 47–53 Cannon Street, London EC4M 5SQ (GB).

(72) **Inventor:** RICHARDS, Timothy, Philip; 32 Craig Mount, Radlett, Herts. WD7 7LW (GB).

(74) **Agent:** POTTER, Julian, Mark; D. Young & Co., 21 New Fetter Lane, London EC4A 1DA (GB).

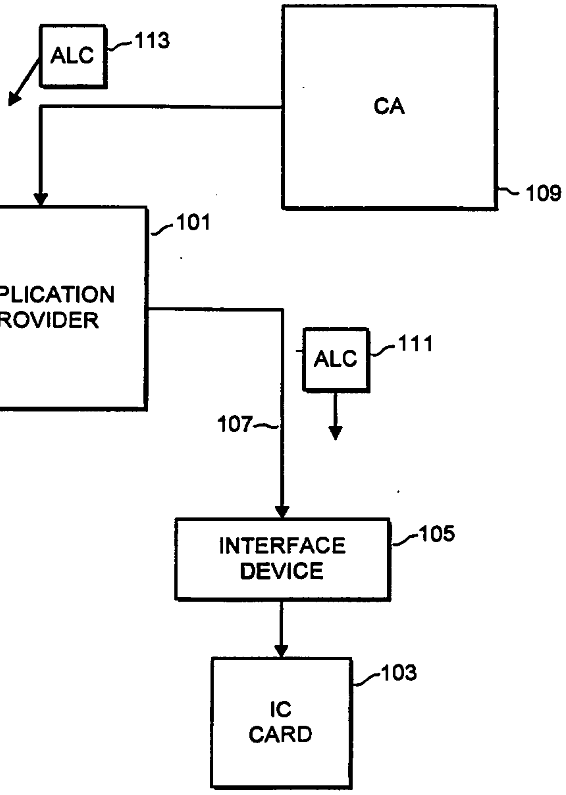
(81) **Designated States:** AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU,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

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

(54) **Title:** KEY TRANSFORMATION UNIT FOR AN IC CARD

(57) **Abstract**

A multi–application IC card system is disclosed having selective application laoding and deleting capability. Prior to loading an application onto an IC card a test is conducted to determine if the card is qualified to receive the application using personalization data stored on the card and comparing it with permissions data associated with the application indicating one ore more sets of cards upon which the application may be loaded. If the personalization data of the card falls within the allowable set of permissions for that application then the card may be loaded with the application. Preferably, the personalization data includes data representative of the card number, issuer, a product class and the date on which the card is personalized.



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		

KEY TRANSFORMATION UNIT FOR AN IC CARD

BACKGROUND OF INVENTION

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. This allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application when it is manufactured and before it is given to a card user. That application, however, cannot be modified or changed after the card is issued even if the modification is desired by the card user or card issuer. Moreover, if a card user wanted a variety of application functions to be performed by IC cards issued to him or her, such as both an electronic purse and a credit/debit function, the card user would be required to carry multiple physical cards on his or her person, which would be quite cumbersome and inconvenient. If an application developer or card user desired two

different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal, making the transaction difficult, lengthy and inconvenient.

Therefore, it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card. Although multiple applications could be preselected and placed in the memory of the card during its production stage, it would also be beneficial to have the ability to load and delete applications for the card post-production as needed.

The increased flexibility and power of storing multiple applications on a single card create new technical challenges to be overcome concerning the integrity and security of the information (including application code and associated data) exchanged between the individual card and the application provider as well as within the entire system when loading and deleting applications. It would be beneficial to have the capability in the IC card system to exchange data among cards, card issuers, system operators and application providers securely and to load and delete applications securely at any time from a local terminal or remotely over a telephone line, Internet or intranet connection or other data conduit. Because these data transmission lines are not typically secure lines, a number of security and entity authentication techniques must be implemented to make sure that applications being sent over the transmission lines are not tampered with and are only loaded on the intended cards.

As mentioned, it is important -- particularly where there is a continuing wide availability of new applications to the cardholder -- that the system has the capability of adding applications onto the IC card subsequent to issuance. This is necessary to protect the longevity of the IC cards; otherwise, once an application becomes outdated, the card would be useless. It would be beneficial to allow the addition of applications from a remote location as well as from a direct connection to an application provider's terminal. For example, it would be beneficial for a card user to be able to plug his IC card into his home computer and download an application over the Internet. This type of remote loading of applications raises a number of security risks when transmitting the application code and related data over an unsecured communications line such as the Internet. At least three issues need to be addressed in a system which provides such a capability.

The first issue is to make sure that the IC card receiving the application is the intended IC card and not another IC card. The second issue is determining how the IC card can authenticate that the application came from the proper application provider and not an unknown third party. The third issue concerns preventing third parties from reading the application and making an unauthorized copy. If a portion of the application is encrypted to address the latter issue, the intended IC card needs to have access to the correct key to decrypt the application. In a system with many IC cards and additionally many application

providers, a secure key transfer technique is required so that the intended IC card can use the correct key for the application which is received. These concerns are raised by both remote application loading as well as local terminal application loading.

Accordingly, it is an object of embodiments of this invention to provide a key transfer and authentication technique and specifically to provide an IC-card system having improved security that allows for improved security for transfer of smart card applications which may be loaded onto IC cards.

SUMMARY OF THE INVENTION

These and other objectives are achieved by the present invention which provides an IC card system and method for securely loading an application onto an IC card including providing a secret and public key pair for the IC card, encrypting at least a portion of the application using a transfer key, encrypting the transfer key using the IC card's public key to form a key transformation unit, transmitting the encrypted application and the key transformation unit to the IC card, decrypting the key transformation unit using the IC card's secret key to provide the transfer key, decrypting the encrypted application using the provided transfer key and storing the decrypted application on the IC card.

In a preferred embodiment, the loading system and method allows the application provider to encrypt two or more portions of the application to be transmitted with two or more different keys, encrypt the two or more keys with the public key of the IC card to form a key transformation unit including the locations of the encrypted portions. Both the encrypted application and the key transformation unit are sent to the IC card. Because the decryption keys are encrypted with the IC card's public key, only the IC card's secret key can decrypt the key transformation unit. The transfer keys and the locations of the encrypted portions are recovered from the decrypted key transformation unit and the application is decrypted using the recovered transfer keys. This ensures that only the intended IC card can decrypt and use the application which was transmitted to that IC card.

In a preferred embodiment, an application load certificate is also sent to the IC card which is receiving the application. The application load certificate contains the public key of the application provider encrypted by the secret key of the certificate authority ("CA"), or the entity that manages the overall security of the IC card system. The IC card then uses a certificate authority public key to make sure that the certificate was valid by attempting to verify the application load certificate with the CA's public key. The IC card then uses the recovered application provider's public key to verify that the application provider was in fact the originator of the application by verifying the sent application signature generated with the application provider's corresponding secret key.

BRIEF DESCRIPTION OF THE DRAWINGS

Further objects, features and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

Fig. 1 is block diagram of the application loading system which loads an application from an application provider to an IC card;

Fig. 2 is a graphic representation of the contents of an Application Loading Unit;

Fig. 3 is a graphic representation of an Application Unit;

Fig. 4 is a flow chart of the steps for providing an individual key set for an IC card;

Fig. 5 is a graphic representation of a Key Transformation Unit;

Fig. 6 is a graphic representation of a Key Transformation Unit plaintext;

Fig. 7 is a graphic representation of the Application Load Certificate;

Fig. 8 is a graphic representation of the Application Unit being decrypted;

Fig. 9 is a flowchart illustrating the steps undertaken in processing the Application Load Unit;

Fig. 10 is a flowchart illustrating the steps undertaken in processing the KTU; and

Fig. 11 is a block diagram showing the components of an IC card which can receive and process and Application Load Unit.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection

with and by way of example only of the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

DETAILED DESCRIPTION OF THE INVENTION

It is beneficial to have the capability to load applications onto IC cards containing multiple application operating systems at any time during the lifetime of the IC card. This flexibility allows a user of a card to periodically add new applications to the IC card and also allows older applications to be updated with newer versions of the application when they are released. For example, a card user may start with an IC card that contains a purse, or electronic cash application (e.g., MONDEX™), being stored on his IC card. Some time after the user has the card, he or she may load an additional application onto the card such as a credit/debit application. Some time after loading the credit/debit application on the card, a new version of the credit/debit application may become available and the card user should be able to erase the old application on his IC card and replace it with the new version of the credit/debit application which may contain additional features.

The flexibility of loading applications at different times during the IC card's life cycle creates security issues with the process of loading applications onto the card. In a multiple application operating system environment, it is beneficial to be able to load applications both at terminals, such as a bank ATM machine, as well as over remote communication links, such as telephone lines, cable

lines, the Internet, satellite or other communications means. When loading applications onto an IC card, the application provider and the card issuer (which could be the same entity) needs to provide security regarding the applications to be loaded. First, the application provider must make sure the application is only sent to the correct card user who is intended to receive the application. One solution to this problem is addressed in a related PCT application entitled "Multi-Application IC Card System Having Selective Loading and Deleting Capability" by Everett et al., filed February 19, 1998 and assigned to Mondex International, which is hereby incorporated by reference to Annex B attached hereto. Two additional security concerns also need to be addressed when loading an application from a remote source, or even from a local terminal, onto an IC card. First, the source of the application must be authenticated as the proper originator so that applications which may contain viruses or simply take up the limited storage memory in an IC card are not allowed to be loaded onto an IC card. Second, the application and associated data may contain private or trade secret information which needs to be encrypted so other people cannot view the contents of the encrypted application code and data. A portion of the application code and data may be secret while other portions are not. These concerns of authentication and protecting the contents of some or all of the application and associated data being loaded onto a card is addressed herein.

A number of encryption/decryption techniques are described herein. There are two basic types of encryption, symmetric encryption and asymmetric encryption. Symmetric encryption uses a secret key as part of a mathematical formula which encrypts data by transforming the data using the formula and key.

After the data is encrypted, another party can decrypt the encrypted data using the same secret key with a related decryption algorithm. Thus the same key is used for encryption and decryption so the technique is symmetric. A conventional example of a symmetric algorithm is DES.

Asymmetric encryption techniques use two different keys of a pair for encrypting and decrypting information. The two keys are normally referred to as a private or secret key and a public key. When data is encrypted with one key of the pair, the other key is used to decrypt the data. If a sender of data signs the data with his secret key, anyone with the public key can verify the message. Since public keys are typically known to the public, the contents of a data signed with a secret key cannot be protected but the origination of the data can be verified by determining if a particular secret key signed the data. This authentication process is termed a digital signature. If person A wanted to authenticate a message he was sending to person B, the person A would sign the document with his secret key. When person B received the message, he would use person A's public key to decipher the message. If the message was readable after the public key was applied to it, person B would know that the document was signed with secret key of person A. Thus, the origin of the message has been authenticated.

The asymmetric key set can also be used to protect the contents of a message. If person A wanted to send an encrypted message to person B that no one else could read, he would encrypt the data or message with person B's public key and send it to person B. Now only the holder of B's secret key could decrypt the data. If a combination of keys is used, a person could both authenticate and

encrypt the message. The asymmetric pair of keys has some powerful applications with respect to card security and is more robust than symmetric encryption. However, asymmetric encryption is more processor costly that symmetric encryption. A example of an asymmetric encryption method is RSA.

A hybrid of symmetric encryption which makes the encryption method more powerful is to encrypt data using two symmetric keys. This technique is called triple DES which encodes data with symmetric key 1, decodes the data using symmetric key 2 (which in effect further encodes the data) and then further encodes the data using key 1 again. Once the data has arrived at its destination, key 1 is used to decode the data, key 2 is used to encode the data, and key 1 is used to decode the data. These extra steps of encoding and decoding make the technique more powerful and more difficult to properly decipher without both keys.

Figure 1 shows a block diagram of the entities used in a secure remote application loading process. The application provider 101 can be a card issuer, bank or other entity which provides application loading services. The application provider 101 initiates an application loading process onto IC card 103. Application Provider 101 is connected to data conduit 107 which is connected to interface device 105 (e.g., a terminal that communicates with an IC card). Data conduit 107 can be a telephone line, an intranet, the Internet, a satellite link or any other type of communications link. The application provider 101, which is remotely located from the IC card 103, desires to send and load an application to the IC card. However, because the data link is an open link and subject to third parties possibly intercepting or replacing applications being transmitted, security

measures which authenticate the application itself, the application provider and the IC card must be used to ensure the integrity of the system. The Certificate Authority 109 may also be used to help authenticate that some data being transferred is part of an identified system.

In Figure 1, the application provider sends an application load unit 111 to the interface device 105 and finally to IC card 103. The ALU includes the application itself and security data required to authenticate and protect the application code and associated data. The ALU is discussed specifically in Figure 2 and in connection with the other figures herein. The ALU 111 also preferably contains Application Load Certificate (ALC) 113 data which is sent from the Certification Authority (CA) 109 to the application provider 101. The Certification Authority manages the overall security of the system by providing an Application Load Certificate for each application which is to be loaded onto an IC card. The application provider 101 and the IC card 103 both have individual public/secret keys sets provided to them. The authentication and security processes will now be described.

Figure 2 shows a diagram illustrating the components of an Application Load Unit which is sent from the application loader to the IC card during the application load process. The Application Load Unit (ALU) 201 contains an Application Unit (AU) 203, an Application Unit Signature (AU_S) 205, a Key Transformation Unit (KTU) 207 and an Application Load Certificate (ALC) 209. The ALU 201 is formatted in a conventional format used during data transmission. AU 203 contains the application code and data which are to be stored

on the IC card, some or all of which is encrypted to protect a secret portion or portions of the code and/or data. AU 203 is described in further detail in connection with Figure 3.

AU_S 205 is the application code and data AU 203 digitally signed with the secret key of the application provider. The public key of the application provider is sent as part of the ALC 209 and is used to authenticate the application provider as the originator of the application. ALC 209 is made up of card identification information and the application provider's public key and is signed by the secret key of the certification authority. All these elements will be described in more detail below.

KTU 207 contains information relating to the encryption of the AU 203 (the code and data of the application) which allows the IC card to decrypt the designated portions so that the application and data can be accessed by the IC card but protects the data during transmission between the application provider and the IC card. KTU 207 is signed with a public key of the IC card for which the application is intended which ensures that only the intended IC card can decrypt the application code and data using the KTU information. This element will be described in connection with Figure 5.

Figure 3 shows a graphic representation of the Application Unit 203 which is part of the application load unit. The AU 203 contains both the program code and associated data which is to be loaded onto the IC card of the card user. The program code consists of a number of program instructions which will be executed by the microprocessor on the IC card. The program instructions can be

written in any programming language which the operating system stored on the IC card can interpret.

For example, in the MULTOS system the program can be written in MEL™ (MULTOS Executable Language). Most applications have associated data which must be loaded onto the card. For instance, data which identifies the card user such as a person's name or account number may be loaded in a secure manner with the credit/debit application. An application provider may provide electronic cash represented by data as a promotion when installing an electronic purse application. Some or all of this data is desired to be kept secret from third parties. Additionally, the application code itself may be considered proprietary and portions may be desired to be kept secret from others. The use of a Key Transformation Unit (KTU) will allow an application provider to designate and encrypt selected portions of its application as confidential and protect it from third parties.

Application Unit portion 305 indicates the program code which is to be transferred from the application provider to the IC card. Application Unit portion 307 indicates the associated data which is to be transferred as part of the application to be loaded onto the IC card. In this example, three discrete areas of the application unit are shown to be encrypted using either single DES or triple DES. Any number of variations regarding the portions encrypted and the type of encryption can be employed using the techniques described herein.

In this example, encrypted location 309 shows the first portion of the Application Unit 203 which has been encrypted using a triple DES technique. The encryption process as described above involves using a symmetrical key and the

conventionally known DES algorithm to transform the data. The data can later be recovered by applying the key to the known DES algorithm. Encrypted location 311 shows a second portion of the application unit 203 which has been encrypted using triple DES. Encrypted location 313 shows a third portion which is encrypted using single DES. Single DES requires less computation to decrypt and takes up less space as part of the KTU as described below. If the application unit were intercepted by a third party while it was being transmitted from the application loader to the IC card, the encrypted portions could not be read unless the third party had the correct keys. That information, therefore, is protected in the KTU.

The KTU is used to allow the IC card for which the application and associated data is intended to decrypt the encrypted portions of the Application Unit by describing which portions of the application unit are encrypted, which encryption algorithm was used and the key or keys to be used to decipher the text. This information is highly confidential between the application provider and the intended IC card and therefore is protected in a manner unique to the intended card. In order to encrypt the KTU which is part of the overall ALU being transmitted, an individual key set for the particular intended IC card is used. The key set and its generation will now be described.

One of the security operations performed at the CA is to generate an individualized key set for each IC card which is stored on the card. The keys are used for off-card verification (i.e., to verify that the card is an authentic card) and for secure data transportation. The key generation process is shown generally in Figure 4. The key set is made up of three different key data items: the card's

secret key which is known only to the card, the card's public key which is stored on the card and the card's public key certificate which is the card's public key signed by one of the CA's secret keys. The individual keys of the key set are described in more detail below.

Step 401 stores a card specific transport secret key for the individual IC card in the memory of the card. This secret key is generated by the CA and loaded onto the card via a card acceptance device. Once stored on the card, the CA deletes from its own memory any data relating to the secret key. Thus, only the card itself knows its secret key. The data element containing the secret key information in the card is called "mkd_sk" which stands for MULTOS key data secret key.

Step 403 stores a card specific transport public key for the individual IC card in the memory of the card. This public key is preferably generated by the CA from the asymmetric encryption technique used to produce the secret key in step 401. The data element containing the card's public key information is called "mkd_pk" which stands for MULTOS key data public key.

Step 405 stores a card specific transport public key certificate for the individual IC card in the memory of the card. The data element containing the card's public key certificate information is called "mkd_pk_c" which stands for MULTOS key data public key certificate. This public key certificate is preferably generated by encrypting the transport public key mkd_pk with the secret key of the CA, indicated as follows:

$$\text{mkd\_pk\_c} = [\text{mkd\_pk}]_{CA\_sk}$$

which means the individual card's public key certificate is formed by applying the CA's secret key to the individual card's public key. The process is carried out at the CA. The public key certificate is retained by the CA so that it can regenerate the public key as needed.

A terminal can read the public key certificate from the IC cards to verify that the CA had signed and therefore approved the individual IC card. This is accomplished by verifying the public key certificate with the public component of the CA key set used to sign the mkd_pk. The decrypted public key certificate can then be compared with the public key to verify that the key certificate was certified (signed) by the CA.

Figure 5 is a graphic depiction of the contents of KTU 207, which contains Header portion 501 and KTU Ciphertext portion 503. As shown in Figure 5, header information 501 includes, for example, identifier or permissions information 505 such as the application_id_no (application identification number), mcd_no (IC card no) and/or msm_control_data_date (the date the IC card was issued). Additional identifiers could also be included. These identifiers allow the system to verify that the IC card which receives the ALU is the intended IC card. The permissions data is discussed in detail in the above referenced related application.

KTU Ciphertext 503 corresponds to KTU Plaintext (not encrypted) encrypted with the public key mkd_pk of the intended IC card as shown in box 507. The KTU Plaintext in further described in Figure 6. The public key mkd_pk is obtained from the intended IC card by the application provider. The public key

of an IC card is freely available to anyone and can be obtained directly from the card or from the CA. By signing the KTU Plaintext with the IC card public key, only the intended IC card can use its secret key of the public/secret key pair to decrypt the KTU Ciphertext. This means that only the intended IC card can determine the contents of the KTU plaint text. identify the encrypted portions of the application being loaded and use the keys provided to decrypt and recover the entire application and associate data. Because no other entity has the secret key of the IC card, the security and integrity of the program code and data being transmitted in ensured.

Figure 6 is a graphic representation of KTU Plaintext 601. KTU Plaintext 601 preferably includes identifier field 603, no_area_discriptors field 605, alg_id field 607, area_start field 609, area_length 611, key_length field 613, key_data field 615 and additional area and key fields depending upon the number of encrypted areas present in the Application Unit. Identifiers 603 contain identifying information of the Application Unit to which the KTU applies.

No_area_descriptors 605 indicates how many different portions of the AU have been encrypted. In the example of Figure 3, the number or area descriptors would be three. Field 607 contains the algorithm identifier for the first area which has been encrypted. The algorithm could be DES or triple DES, for example. Field 609 indicates the start of the first encrypted area. This indication could be an offset from the start of the AU. For example, the offset could be 100 which means that the first area starts at the 100th byte of the Application Unit. Field 611 indicates the area length for the first encrypted portions. This field allows the microprocessor on the IC card to know how large an area has been encrypted and when coupled with the start of the area, allows the IC card microprocessor to decrypt the correct portion of the Application Unit. Filed 613 indicates the key length for the particular encrypted portion of the application unit. The length of the key will differ for different encryption techniques. The key length field allows the IC card to know the length of the key data. Field 615 indicates the key data for the particular encrypted portion. The key data is used with the algorithm identity and the location of the encoded portion to decode the encrypted portion. If more than one encrypted area is indicated, then additional data referring of the algorithm, start location, length, key length and key data will be present in the KTU Plaintext. While a number of fields have been described, not all the fields are necessary for the invention. The most important field, however, is the key data itself.

Figure 7 is a graphic representation of the Application Load Certificate (ALC) 209. ALC 209 includes a header 701 and the Application Provider Public Key 703. Header 701 and Application Provider Public Key 703 are then signed (encrypted) with the CA secret key. Thus, the ALC 209 must be provided by the CA to the application provider for each application loaded because only the CA knows the CA private key. Header 701 contains information regarding the application provider and the IC card for which the application is intended. The ALC 209 is placed in the correct ALU by the application provider which can use the identification information. Application Provider Public Key 703 is provided to the CA along with the identification data. The CA then signs this information after verifying its authenticity and returns the signed ALC to the application provider.

The IC card, when it receives the ALC 209 as part of the ALU 201, will open the ALC 209 with the public key of the CA. This ensures that the CA signed the application load certificate and that it is genuine. After decrypting the information, the header identification information 701 is checked and the application provider public key is recovered. This public key will be used to verify that the application and code which is to be loaded onto the IC card originated with the proper application provider.

Figure 8 is a graphic representation of the use of the application provider's public key to decrypt the signed AU 205 in order to verify that AU 203 was signed by the application provider. AU signed 205 is verified with the Application Provider Public Key 801. The recovered AU 803 is then compared with AU 203. If the data blocks match, then the IC card has verified that the application provider signed (encrypted) the application unit and the application is genuine. This authentication is valid because only the application provider has its own secret key. The IC card can process this information because the application provider's public key is provided to it as part of the application load certificate 209 which is signed by the CA. Therefore, it does not need to retrieve the public key from an external location to authenticate the application.

Figure 9 shows a flow chart of the steps for processing the Application Load Unit when it is received by the IC card. Prior to receiving the ALU, identity checks as to the identity of the IC card can be performed if desired. The ALU processing techniques provide a number of further verifications including verifying that the application being loaded is: (1) from the correct application

provider, (2) being loaded on the intended card and (3) certified by the CA. The ALU processing techniques also allow the transportation of transport decryption keys which enable the IC card to decrypt portions of the program code and associated data in a secure manner. In step 901, the IC card receives the ALU from the application provider. The ALU can be transmitted via a terminal connection, contactless connection, telephone, computer, intranet, Internet or any other communication means. The ALU is placed in the EEPROM of the IC card along with header information indicating the starting addresses of AU 203, AU signed 205, the KTU 207 and ALC 209. Alternatively, the IC card could determine the relative address locations of these four units.

Step 903 decrypts the ALC 209 with the CA public key. Each IC card preferably stores in its memory a copy of the CA public key because it is used in many transactions. Alternatively, the IC card could obtain the public key from a known storage location. If the CA public key successfully verifies the ALC 209, then the IC card has verified that the CA has signed the ALC 209 with its secret key and thus the Application Load Certificate is proper. If the IC card cannot verify the ALC successfully, then the ALC was not signed by the CA and the certificate is not proper. The application loading process would then end.

Step 905 then checks the identity of IC card against the identification information sent in the application load certificate to make sure the card is intended to receive the application. This permissions checking is described in the related patent application identified above. If there is no match of identification data, the application loading process ends. If the identification data does match, then the

process continues.

Step 907 uses the application providers public key which was recovered from the verified ALC to verify the AU signature 205. When the ALU was generated by the application provider, the application unit 203 was signed with the application provider's secret key. The application provider then provides its public key to IC card through the ALC. The IC card then verifies the AU signed 205. If the ALU is successfully verified, then it is accepted as having been generated by the application provider. Because the application provider's public key is part of the ALC which is signed by the CA, the CA can make sure that the proper public key has been provided to the IC card. This unique key interaction between the application provider, CA and the intended IC card ensures that no counterfeit or unapproved applications or data are loaded onto an IC card which is part of the secure system.

Step 911 then processes a KTU authentication check which further verifies that only the intended card has received the application. The KTU authentication check makes sure that if a third party does somehow intercept the ALU, the third party cannot read the enciphered portions of the AU and cannot retrieve the keys to decrypt the AU. This step is further explained in Figure 10.

Figure 10 shows the steps of the KTU Authentication process. Step 1001, which is shown in dashed lines because it is preferably optional, checks the identification of the IC card a second time. The identification information can be sent as part of the KTU data. However, this check is optional as it has already been performed once in step 905.

Step 1003 then decrypts KTU ciphertext 503 using the IC card's secret key (mkd_sk). The KTU Plaintext was previously encrypted using the intended card's public key (mkd_pk). This means that only the holder of the intended card's secret key could decrypt the encrypted message. The application provider obtains the intended IC card's public key either from the IC card itself (See Figure 4 and related text for a discussion of the mkd key set) or from a database holding the public keys. If the IC card cannot decrypt the KTU ciphertext properly then the KTU is not meant for that card and the application loading process halts. If the IC card does properly decipher the KTU ciphertext, then the process continues.

Step 1005 identifies an encrypted area of the application unit (AU). In the example of the KTU Plaintext described in connection with Figure 6, the IC card uses a relative starting address and area length field to determine the encrypted portion. Step 1005 also identifies which encryption technique was used to encrypt the identified portion so that the proper decryption technique can be used. For example, the technique could by single or triple DES. Alternatively, the technique could be a default technique used in the system and need not be identified.

Step 1007 then retrieves the key from KTU Plaintext and decrypts the identified portion with the identified decryption technique. This allows the IC card to have the decrypted portion of the AU which it will store in its static memory once all the encrypted portions have been decrypted.

Step 1009 checks if there are any other additional encrypted areas. In the example described in Figure 3, there are three encrypted areas. The number

of encrypted areas was a field in the example of Figure 6. However, the number of portions can be determined using other conventional means. If there are additional encrypted portions, the process jumps to step 1005. If there are no additional encrypted portions, then the process continues with step 1011.

Step 1011 then loads the decrypted AU into the memory of the IC card. The ALU has passed all of the authentication and decryption checks and the application can now properly reside on the IC card and be executed and used by the card user. While the different checks have been presented in a particular order in Figures 9 and 10, the checks can be performed in any order. While all of the described techniques used in conjunction with the ALU provide the best security, one or more of the individual techniques could be used for their individual purposes or combined with other conventional security techniques.

Figure 11 shows an example of a block diagram of an IC card chip upon which an ALU can be loaded and processed. An integrated circuit is located on an IC card for use. The IC card preferably includes a central processing unit 1101, a RAM 1103, an EEPROM 1105, a ROM 1107, a timer 1109, control logic unit 1111, an I/O port 1113 and security circuitry 1115, which are connected together by a conventional data bus.

Control logic 1111 in memory cards provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports. CPU 1101 with its control logic can perform calculations, access memory locations, modify memory contents, and manage input/output ports. Some cards have a coprocessor for handling complex computations like performing

cryptographic operations. Input/output ports 1113 are used under the control of a CPU and control logic, for communications between the card and a card interface device. Timer 1109 (which generates or provides a clock pulse) drives the control logic 1111 and CPU 1101 through the sequence of steps that accomplish memory access, memory reading or writing, processing, and data communication. A timer may be used to provide application features such as call duration. Security circuitry 1115 includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed ("blown") upon completion of testing to prevent later access. The AU data after the ALU has been authenticated and verified is stored in EEPROM 1105. The authentication process as described herein is performed by the CPU 1101.

Figure 11 also shows a possible configuration for the integrated circuit chip for the application provider and for the certification authority. CPU 1101 present in the IC chip for the application provider encrypts the necessary information using encryption techniques described herein and performs the necessary data operations. CPU 1101 at the certification authority is used to sign the Application Load Certificate as described herein.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

For example, while loading an application is discussed herein, the

same secure loading process can apply to transmitting other types of data such as data blocks, database files, word processing documents or any other type of data need to be transmitted in a secure manner.

The scope of the present disclosure includes any novel feature or combination of features disclosed therein either explicitly or implicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or mitigates any or all of the problems addressed by the present invention. The application hereby gives notice that new claims may be formulated to such features during the prosecution of this application or of any such further application derived therefrom. In particular, with reference to the appended claims, features from dependant claims may be combined with those of the independent claims in any appropriate manner and not merely in the specific combinations enumerated in the claims.

ANNEX A TO THE DESCRIPTION

ANNEX A

IC CARD TRANSPORTATION KEY SET

ANNEX A TO THE DESCRIPTION

BACKGROUND OF INVENTION

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. This allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application when it is manufactured and before it is given to a card user. That application, however, cannot be modified or changed after the card is issued even if the modification is desired by the card user or card issuer. Moreover, if a card user wanted a variety of application functions to be performed by IC cards issued to him or her, such as both an electronic purse and a credit/debit function, the card user would be required to carry multiple physical cards on his or her person, which would be quite

cumbersome and inconvenient. If an application developer or card user desired two different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal, making the transaction difficult, lengthy and inconvenient.

Therefore, it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card. Although multiple applications could be preselected and placed in the memory of the card during its production stage, it would also be beneficial to have the ability to load and delete applications for the card post-production as needed.

The increased flexibility and power of storing multiple applications on a single card create new challenges to be overcome concerning the integrity and security of the information (including application code and associated data) exchanged between the individual card and the application provider as well as within the entire system when loading and deleting applications. It would be beneficial to have the capability in the IC card system to exchange data among cards, card issuers, system operators and application providers securely and to load and delete applications securely at any time from a local terminal or remotely over a telephone line, Internet or intranet connection or other data conduit. Because

ANNEX A TO THE DESCRIPTION

these data transmission lines are not typically secure lines, a number of security and entity authentication techniques must be implemented to make sure that applications being sent over the transmission lines are not tampered with and are only loaded on the intended cards.

As mentioned, it is important -- particularly where there is a continuing wide availability of new applications to the cardholder -- that the system has the capability of adding applications onto the IC card subsequent to issuance. This is necessary to protect the longevity of the IC cards; otherwise, once an application becomes outdated, the card would be useless. It would be beneficial to allow the addition of applications from a remote location as well as from a direct connection to an application provider's terminal. For example, it would be beneficial for a card user to be able to plug his or her IC card into a home computer and download an application over the Internet. This type of remote loading of applications raises a number of security risks when transmitting the application code and related data over an unsecured communications line such as the Internet.

An entity which transmits an application or data to an IC card requires that only the intended IC card should receive the transmitted data. Third parties should not be able to intercept and view the data. Additionally, a transmitting entity will require verification that the IC card which has requested information is actually part of the overall IC card system and not simply posing as being part of the system. These concerns are raised by both remote application loading as well as local terminal application loading.

ANNEX A TO THE DESCRIPTION

Accordingly, it is an object of this invention to provide a secure transfer technique and specifically to provide a secure IC-card system that allows for the secure transfer of data including smart card applications which may be loaded onto IC cards.

SUMMARY OF THE INVENTION

These and other objectives are achieved by the present invention which provides an IC card method and apparatus for securely transporting data including an application onto an IC card including storing a secret and public key pair on the IC card, retrieving the stored public key from the IC card, encrypting at least a portion of the data to be transported using the public key, transmitting the encrypted data to the IC card and decrypting the encrypted data using the IC card's secret key.

In a preferred embodiment. a certification authority ("CA") or the entity that manages the overall security of the IC card system. encrypts (or digitally signs) a copy of the IC card's public key and the signed copy is also stored on the IC card. The entity transmitting the data to the IC card can verify that the CA has approved the card by retrieving using the IC card's signed public key and verifying the signed public key using the public key of the CA. If verification is successful. the entity has verified that the CA approved the IC card.

BRIEF DESCRIPTION OF THE DRAWINGS

Further objects, features and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

Fig. 1A is a block diagram of the secure data transfer system which securely transfers data from a transferring entity to an IC card.

Fig. 1B is block diagram of the application loading system which loads an application from an application provider to an IC card;

Fig. 2 is a graphic representation of the contents of an Application Loading Unit;

Fig. 3 is a graphic representation of an Application Unit;

Fig. 4 is a flow chart of the steps for providing an individual key set for an IC card;

Fig. 5 is a graphic representation of a Key Transformation Unit;

Fig. 6 is a graphic representation of a Key Transformation Unit plaintext;

Fig. 7 is a graphic representation of the Application Load Certificate;

Fig. 8 is a graphic representation of the Application Unit being decrypted;

Fig. 9 is a flowchart illustrating the steps undertaken in processing the Application Load Unit;

Fig. 10 is a flowchart illustrating the steps undertaken in processing the KTU; and

Fig. 11 is a block diagram showing the components of an IC card which can receive and process and Application Load Unit.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

DETAILED DESCRIPTION OF THE INVENTION

It is beneficial to have the capability to load applications onto IC cards containing multiple application operating systems at any time during the lifetime of the IC card. This flexibility allows a user of a card to periodically add new applications to the IC card and also allows older applications to be updated with newer versions of the application when they are released. For example, a card user may start with an IC card that contains a purse, or electronic cash application (e.g., MONDEX™), being stored on his IC card. Some time after the user has the card, he or she may load an additional application onto the card such as a credit/debit application. Some time after loading the credit/debit application on the card, a new version of the credit/debit application may become available and the

ANNEX A TO THE DESCRIPTION

card user should be able to erase the old application on his IC card and replace it with the new version of the credit/debit application which may contain additional features. Additionally, an IC card needs to receive data regarding personal information such as new credit card account numbers or updated information.

The flexibility of loading applications and transmitting data at different times during the IC card's life cycle creates security issues with the process of loading applications onto the card. In a multiple application operating system environment, it is beneficial to be able to load applications and data both at terminals, such as a bank ATM machine, as well as over remote communication links, such as telephone lines, cable lines, the Internet, satellite or other communications means. When loading applications and data onto an IC card, the application provider needs to provide security regarding the applications to be loaded. First, the application provider must make sure the application is only sent to the correct card user who is intended to receive the application. Second, the application and associated data may contain private or trade secret information which needs to be encrypted so entities other than the IC card cannot view the contents of the encrypted application code and data. A portion of the application code and data may be secret while other portions are not. These concerns of authentication and protecting the contents of some or all of the application and associated data being loaded onto a card is addressed herein.

A number of encryption/decryption techniques are described herein. There are two basic types of encryption, symmetric encryption and asymmetric encryption. Symmetric encryption uses a secret key as part of a mathematical

ANNEX A TO THE DESCRIPTION

formula which encrypts data by transforming the data using the formula and key. After the data is encrypted, another party can decrypt the encrypted data using the same secret key with a decryption algorithm. Thus the same key is used for encryption and decryption so the technique is symmetric. A conventional example of a symmetric algorithm is DES.

Asymmetric encryption techniques use two different keys of a pair for encrypting and decrypting information. The two keys are normally referred to as a private or secret key and a public key. When data is encrypted with one key of the pair, the other key is used to decrypt the data. If a sender of data signs the data with his secret key, anyone with the public key can verify the message. Since public keys are typically known to the public. the contents of a data signed with a secret key cannot be protected but the origination of the data can be verified by determining if a particular secret key signed the data. This authentication process is termed a digital signature. If person A wanted to authenticate a message he was sending to person B. the person A would sign the document with his secret key. When person B received the message, he would use person A's public key to verify the message. If the message was verified with the public key, person B would know that the document was signed with secret key of person A. Thus, the origin of the message has been authenticated.

The asymmetric key set can also be used to protect the contents of a message. If person A wanted to send an encrypted message to person B that no one else could read, he would encrypt the data or message with person B's public key and send it to person B. Now only the holder of B's secret key could decrypt the

ANNEX A TO THE DESCRIPTION

data. If a combination of keys is used, a person could both authenticate and encrypt the message. The asymmetric pair of keys has some powerful applications with respect to card security. However, asymmetric encryption is relatively processor costly (processor cost is associated with computation time) compared with symmetric encryption. An example of asymmetric encryption method is RSA®.

A hybrid of symmetric encryption which makes the encryption method more powerful is to encrypt data using two symmetric keys. This technique is called triple DES which encodes data with key 1, decodes the data using key 2 (which in effect further encodes the data) and then further encodes the data using key 1 again. Once the data has arrived at its destination, key 1 is used to decode the data, key 2 is used to encode the data, and key 1 is used to decode the data. These extra steps of encoding and decoding make the technique more powerful and more difficult to properly decipher without both keys.

Figure 1A shows a block diagram of the entities used in transporting data in a secure manner in an IC card system. The transmitting entity 1 can be a card issuer, bank, IC card or other entity which desires to transport data to an IC card 3. The transmitting entity 1 preferably initiates the data transfer process. Alternatively, the IC card 3 can initiate the data transfer process if the card requires data from the transmitting entity 1.

The transmitting entity 1 is connected to interface device 5 (e.g., a terminal that communicates with an IC card). Data conduit 7 can be a telephone line, an intranet, the Internet, a satellite link or any other type of communications link. In this example, the transmitting entity 1, which is remotely located from IC card 3, desires to send data in a secure manner to the IC card. However, because the data link is an "open" link (i.e. not a private link) and subject to third parties possibly intercepting or replacing data being transmitted, security measures are needed to guarantee that only the intended IC card will receive the transmitted data. The Certificate Authority 9 can also be used to authenticate that the IC card has been validated as part of the IC card system.

In Figure 1A, a private (or secret) key 19 and corresponding public key 15 is generated for IC card 3. The keys are preferably generated using an asymmetric encryption algorithm such as RSA®. The keys can be generated at the CA 9 or any other location because they are specific only to the IC card 3 and no other copies need to be kept. A third data item, the public key certificate 17, is also generated and stored on the IC card 3.

The public key certificate 17 is generated by signing the public key 15 with the private key of the CA 9. This allows a person with the public key of the CA 9 to verify that the CA digitally signed the IC card's public key in order to certify the IC card's individual key set. The public key certificate can be generated by the CA at the time the IC card private/public key set is generated or at a subsequent time.

When a data transfer is initiated by the transmitting entity 1, the IC card 3 is contacted through the interface device 5 and the IC card 3 sends its public key 15 and its public key certificate 17 to the transmitting entity 1. The transmitting entity then verifies the public key certificate with public key of the CA 13 (which is publicly available from the CA 9 and may be stored in the transmitting

ANNEX A TO THE DESCRIPTION

entity 1) thus determining if the CA 9 digitally signed the public key and verifying that the IC card is a valid card.

The transmitting entity 1 then encrypts the data to be transmitted with the IC card's public key. The transmitting entity 1 then transmits the encrypted data 11 to the interface device 5 and to the IC card 3. The IC card 3 decrypts the encrypted data with its corresponding private (also called secret) key 19. The data can then be processed by the IC card 3. Only the IC card 3 has a copy of its private key so only the intended IC card can access the encrypted data. This ensures that third parties cannot access the encrypted data and correspondingly that only the intended IC card will be able to read and process the data.

Figure 1B shows a secure method for loading applications onto an IC card. Figure 1B shows a block diagram of the entities used in a secure remote application loading process. The application provider 101 can be a card issuer, bank or other entity which provides application loading services. The application provider 101 initiates an application loading process onto IC card 103. IC card 103 is connected to data conduit 107 which is connected to interface device 105 (e.g., a terminal that communicates with an IC card). Data conduit 107 can be a telephone line, an intranet, the Internet, a satellite link or any other type of communications link. The application provider 101, which is remotely located from the IC card 103, desires to send and load an application to the IC card. However, because the data link is an open link and subject to third parties possibly intercepting or replacing applications being transmitted, security measures which authenticate the application itself, the application provider and the IC card must be used to ensure the integrity of the system. The CA 109 may also be used to help authenticate that some data being transferred is part of an identified system.

In Figure 1B, the application provider sends an application load unit 111 to the interface device 105 and finally to IC card 103. The ALU includes the application itself and security data required to authenticate and protect the application code and associated data. The ALU is discussed specifically in Figure 2 and in connection with the other figures herein. The ALU 111 also preferably contains Application Load Certificate (ALC) 113 data which is sent from the Certification Authority (CA) 109 to the application provider 101. The Certification Authority manages the overall security of the system by providing an Application Load Certificate for each application which is to be loaded onto an IC card. The application provider 101 and the IC card 103 both have individual public/secret keys sets. The authentication and security processes will now be described.

Figure 2 shows a diagram illustrating the components of an Application Load Unit which is sent from the application loader to the IC card during the application load process. The Application Load Unit (ALU) 201 contains an Application Unit (AU) 203, an Application Unit Signature (AU_S) 205, a Key Transformation Unit (KTU) 207 and an Application Load Certificate (ALC) 209. The ALU 201 is formatted in a conventional format used during data transmission. AU 203 contains the application code and data which are to be stored on the IC card, some or all of which is encrypted to protect a secret portion or portions of the code and/or data. AU 203 is described in further detail in connection with Figure 3.

ANNEX A TO THE DESCRIPTION

AU_S 205 is the application code and data AU 203 digitally signed with the secret key of the application provider. The public key of the application provider is sent as part of the ALC 209 and is used to authenticate the application provider as the originator of the application. ALC 209 is made up of card identification information and the application provider's public key and is signed by the secret key of the certification authority. All these elements will be described in more detail below.

KTU 207 contains information relating to the encryption of the AU 203 (the code and data of the application) which allows the IC card to decrypt the designated portions so that the application and data can be accessed by the IC card but protects the data during transmission between the application provider and the IC card. KTU 207 is encrypted with the public key of the IC card for which the application is intended which ensures that only the intended IC card can decrypt the application code and data using the KTU information. This element will be described in connection with Figure 5.

Figure 3 shows a graphic representation of the Application Unit 203 which is part of the application load unit. The AU 203 contains both the program code and associated data which is to be loaded onto the IC card of the card user. The program code consists of a number of program instructions which will be executed by the microprocessor on the IC card. The program instructions can be written in any programming language which the operating system stored on the IC card can interpret.

For example, in the MULTOS system the program can be written in

ANNEX A TO THE DESCRIPTION

MEL™ (MULTOS Executable Language). Most applications have associated data which must be loaded onto the card. For instance, data which identifies the card user such as a person's name or account number may be loaded in a secure manner with the credit/debit application. An application provider may provide electronic cash represented by data as a promotion when installing an electronic purse application. Some or all of this data is desired to be kept secret from third parties. Additionally, the application code itself may be considered proprietary and portions may be desired to be kept secret from others. The use of a Key Transformation Unit (KTU) will allow an application provider to designate and encrypt selected portions of its application as confidential and protect it from third parties.

Application Unit portion 305 indicates the program code which is to be transferred from the application provider to the IC card. Application Unit portion 307 indicates the associated data which is to be transferred as part of the application to be loaded onto the IC card. In this example, three discrete areas of the application unit are shown to be encrypted using either single DES or triple DES. Any number of variations regarding the portions encrypted and the type of encryption can be employed using the techniques described herein.

In this example, encrypted location 309 shows the first portion of the Application Unit 203 which has been encrypted using a triple DES technique. The encryption process as described above involves using a symmetric key and the conventionally known DES-based algorithm to transform the data. The data can later be recovered by applying the key to a conventionally known DES-based decryption algorithm. Encrypted location 311 shows a second portion of the

ANNEX A TO THE DESCRIPTION

application unit 203 which has been encrypted using triple DES. Encrypted location 313 shows a third portion which is encrypted using single DES. Single DES requires less computation to decrypt and takes up less space as part of the KTU as described below. If the application unit were intercepted by a third party while it was being transmitted from the application loader to the IC card, the encrypted portions could not be read unless the third party had the correct keys and decryption algorithm. That information, therefore, is protected in the KTU.

The KTU is used to allow the IC card for which the application and associated data is intended to decrypt the encrypted portions of the Application Unit by describing which portions of the application unit are encrypted, which encryption algorithm was used and the key or keys to be used to decipher the text. This information is highly confidential between the application provider and the intended IC card and therefore is protected in a manner unique to the intended card. In order to encrypt the KTU which is part of the overall ALU being transmitted, an individual key set for the particular intended IC card is used. The key set and its generation will now be described.

In accordance with the present invention, one of the security operations performed at the CA is to generate an individualized key set for each IC card which is stored on the card. The keys are used for off-card verification (i.e., to verify that the card is an authentic card) and for secure data transportation. The key generation process is shown generally in Figure 4. The key set is made up of three different key data items: the card's secret key which is known only to the card, the card's public key which is stored on the card and the card's public key

ANNEX A TO THE DESCRIPTION

certificate which is the card's public key signed by the CA's secret key. The individual keys of the key set are described in more detail below.

Step 401 stores a card specific transport secret key for the individual IC card in the memory of the card. This secret key is generated by the CA from a standard asymmetric encryption technique such as RSA® and loaded onto the card via a card acceptance device. Once stored on the card, the CA deletes from its own memory any data relating to the secret key. Thus, only the card itself knows its secret key. The data element containing the secret key information in the card is called "mkd_sk" which stands for MULTOS key data secret key.

Step 403 stores a card specific transport public key for the individual IC card in the memory of the card. This public key is preferably generated by the CA from the asymmetric encryption technique used to produce the secret key in step 401. As with the secret key, once the public key is stored on the card, the CA (or other key provider) deletes from its systems the public key data so that the only copy of the public key is kept in the card. The data element containing the card's public key information is called "mkd_pk" which stands for MULTOS key data public key.

Step 405 stores a card specific transport public key certificate for the individual IC card in the memory of the card. The data element containing the card's public key certificate information is called "mkd_pk_c" which stands for MULTOS key data public key certificate. This public key certificate is preferably generated by signing the transport public key mkd_pk with the secret key of the CA, indicated as follows:

ANNEX A TO THE DESCRIPTION

$$mkd\_pk\_c = [mkd\_pk]_{CA\_sk}$$

which means the individual card's public key certificate is formed by applying the CA's secret key to the individual card's public key. The process is carried out at the CA. The public key certificate is retained by the CA so that it can regenerate the public key as needed.

A terminal can read the public key certificate from the IC cards to verify that the CA had signed and therefore approved the individual IC card. This is accomplished by verifying the public key certificate with the public component of the CA key set used to sign the mkd_pk.

Figure 5 is a graphic depiction of the contents of KTU 207, which contains Header portion 501 and KTU Ciphertext portion 503. As shown in Figure 5, header information 501 includes, for example, identifier or permissions information 505 such as the application_id_no (application identification number), mcd_no (IC card no) and/or msm_control_data_date (the date the IC card was issued). Additional identifiers could also be included. These identifiers allow the system to verify that the IC card which receives the ALU is the intended IC card. The permissions data is discussed in detail in the above referenced related application.

KTU Ciphertext 503 corresponds to KTU Plaintext (not encrypted) encrypted with the public key mkd_pk of the intended IC card as shown in box 507. The KTU Plaintext in further described in Figure 6. The public key mkd_pk is obtained from the intended IC card by the application provider. The public key of an IC card is freely available to anyone and can be obtained directly from the card or from the CA. By encrypting the KTU Plaintext with the IC card public key, only the intended IC card can use its secret key of the public/secret key pair to decrypt the KTU Ciphertext. This means that only the intended IC card can determine the contents of the KTU plaint text. identify the encrypted portions of the application being loaded and use the keys to decrypt and recover the entire application and associate data. Because no other entity has the secret key of the IC card, the security and integrity of the program code and data being transmitted in ensured.

Figure 6 is a graphic representation of KTU Plaintext 601. KTU Plaintext 601 preferably includes identifier field 603. no_area_discriptors field 605, alg_id field 607, area_start field 609, area_length 611. key_length field 613, key_data field 615 and additional area and key fields depending upon the number of encrypted areas present in the Application Unit. Identifiers 603 contain identifying information of the Application Unit to which the KTU applies.

No_area_descriptors 605 indicates how many different portions of the AU have been encrypted. In the example of Figure 3. the number or area descriptors would be three. Field 607 contains the algorithm identifier for the first area which has been encrypted. The algorithm could be DES or triple DES, for example. Field 609 indicates the start of the first encrypted area. This indication could be an offset from the start of the AU. For example, the offset could by 100 which means that the first area starts at the 100th byte of the Application Unit. Field 611 indicates the area length for the first encrypted portions. This field allows the microprocessor on the IC card to know how large an area has been encrypted and when coupled with the start of the area, allows the IC card microprocessor to decrypt the correct portion of the Application Unit. Filed 613 indicates the key length for the particular encrypted portion of the application unit. The length of the key will differ for different encryption techniques. The key length field allows the IC card to know the length of the key data. Field 615 indicates the key data for the particular encrypted portion. The key data is used with the algorithm identity and the location of the encoded portion to decode the encrypted portion. If more than one encrypted area is indicated, then additional data referring to the algorithm, start location, length, key length and key data will be present in the KTU Plaintext. While a number of fields have been described, not all the fields are necessary for the invention. The most important field, however, is the key data itself.

Figure 7 is a graphic representation of the Application Load Certificate (ALC) 209. ALC 209 includes a header 701 and the Application Provider Public Key 703. Header 701 and Application Provider Public Key 703 are then signed (encrypted) with the CA secret key. Thus, the ALC 209 must be provided by the CA to the application provider for each application loaded because only the CA knows the CA private key. Header 701 contains information regarding the application provider and the IC card for which the application is intended. The ALC 209 is placed in the correct ALU by the application provider which can use the identification information. Application Provider Public Key 703 is provided to the CA along with the identification data. The CA then signs this information after verifying its authenticity and returns the signed ALC to the application provider. The IC card, when it receives the ALC 209 as part of the ALU 201, will verify the

ANNEX A TO THE DESCRIPTION

ALC 209 with the public key of the CA. This ensures that the CA signed the Application Load Certificate and that it is genuine. After verifying the information, the header identification information 701 is checked and the application provider public key is recovered. This public key will be used to verify that the application and code which is to be loaded onto the IC card originated with the proper application provider.

Figure 8 is a graphic representation of the use of the application provider's public key to verify the signature of the AU 205 in order to verify that AU 203 was signed by the application provider. AU signature 205 is verified with the Application Provider Public Key 801 and compared with AU 203. If the data blocks match, then the IC card has verified that the application provider signed (encrypted) the application unit and the application is genuine. This authentication is valid because only the application provider has its own secret key. The IC card can process this information efficiently because the application provider's public key is provided to it as part of the Application Load Certificate 209 which is signed by the CA. Therefore, it does not need to retrieve the public key from an external location to authenticate the application.

Figure 9 shows a flow chart of the steps for processing the Application Load Unit when it is received by the IC card. Prior to receiving the ALU, identity checks as to the identity of the IC card can be performed if desired. The ALU processing techniques provide a number of further verifications including verifying that the application being loaded is: (1) from the correct application provider, (2) being loaded on the intended card and (3) certified by the CA. The

ALU processing techniques also allow the transportation of transport decryption keys which enable the IC card to decrypt portions of the program code and associated data in a secure manner. In step 901, the IC card receives the ALU from the application provider. The ALU can be transmitted via a terminal connection, contactless connection, telephone, computer, intranet, Internet or any other communication means. The ALU is placed in an I/O buffer of the IC card along with header information indicating the starting addresses of AU 203, AU signed 205, the KTU 207 and ALC 209. Alternatively, the IC card could determine the relative address locations of these four units.

Step 903 verifies the ALC 209 with the CA public key. Each IC card preferably stores in its memory a copy of the CA public key because it is used in many transactions. Alternatively, the IC card could obtain the public key from a known storage location. If the CA public key verifies the ALC 209 properly, then the IC card has verified that the CA has signed the ALC 209 with its secret key and thus the Application Load Certificate is proper. If the IC card cannot verify the ALC properly, then the ALC was not signed by the CA and the certificate is not proper. The application loading process would then end.

Step 905 then checks the identity of IC card against the identification information sent in the Application Load Certificate to make sure the card is intended to receive the application. This permissions checking is described in the related patent application identified above. If there is no match of identification data, the application loading process ends. If the identification data does match, then the process continues.

ANNEX A TO THE DESCRIPTION

Step 907 uses the application providers public key which was recovered from the verified ALC to verify AU signature 205. When the ALU was generated by the application provider, the application unit 203 was signed with the application provider's secret key to authenticate that the application was provided by the correct application provider. The application provider then provides its public key to IC card through the ALC. The IC card then verifies the AU signature 205. If the two data blocks match, then the ALU is verified as being generated by the application provider. Because the application provider's public key is part of the ALC which is signed by the CA, the CA can make sure that the proper public key has been provided to the IC card. This unique key interaction between the application provider, CA and the intended IC card ensures that no counterfeit or unapproved applications or data are loaded onto an IC card which is part of the secure system.

Step 911 then processes a KTU authentication check which further verifies that only the intended card has received the application. The KTU authentication check makes sure that if a third party does somehow intercept the ALU, the third party cannot read the enciphered portions of the AU and cannot retrieve the keys to decrypt the AU. This step is further explained in Figure 10.

Figure 10 shows the steps of the KTU Authentication process. Step 1001, which is shown in dashed lines because it is preferably optional, checks the identification of the IC card a second time. The identification information can be sent as part of the KTU data. However, this check is optional as it has already been performed once in step 905.

ANNEX A TO THE DESCRIPTION

Step 1003 then decrypts KTU ciphertext 503 using the IC card's secret key (mkd_sk). The KTU Plaintext was previously encrypted using the intended card's public key (mkd_pk). This means that only the holder of the intended card's secret key could decrypt the encrypted message. The application provider obtains the intended IC card's public key either from the IC card itself (See Figure 4 and related text for a discussion of the mkd key set) or from a database holding the public keys. If the IC card cannot decrypt the KTU ciphertext properly then the KTU is not meant for that card and the application loading process halts. If the IC card does properly decipher the KTU ciphertext, then the process continues.

Step 1005 identifies an encrypted area of the application unit (AU). In the example of the KTU Plaintext described in connection with Figure 6, the IC card uses a relative starting address and area length field to determine the encrypted portion. Step 1005 also identifies which encryption technique was used to encrypt the identified portion so that the proper decryption technique can be used. For example, the technique could by single or triple DES. Alternatively, the technique could be a default technique used in the system and need not be identified.

Step 1007 then retrieves the key from KTU Plaintext and decrypts the identified portion with the identified decryption technique. This allows the IC card to have the decrypted portion of the AU which it will store in its EEPROM once all the encrypted portions have been decrypted.

Step 1009 checks if there are any other additional encrypted areas. In the example described in Figure 3, there are three encrypted areas. The number

of encrypted areas was a field in the example of Figure 6. However, the number of portions can be determined using other conventional means. If there are additional encrypted portions, the process jumps to step 1005. If there are no additional encrypted portions, then the process continues with step 1011.

Step 1011 then loads the decrypted AU into the memory of the IC card. The ALU has passed all of the authentication and decryption checks and the application can now properly reside on the IC card and be executed and used by the card user. While the different checks have been presented in a particular order in Figures 9 and 10, the checks can be performed in any order. While all of the described techniques used in conjunction with the ALU provide the best security, one or more of the individual techniques could be used for their individual purposes or combined with other conventional security techniques.

Figure 11 shows an example of a block diagram of an IC card chip upon which an ALU can be loaded and processed. An integrated circuit is located on an IC card for use. The IC card preferably includes a central processing unit 1101, a RAM 1103, an EEPROM 1105, a ROM 1107, a timer 1109, control logic 1111, an I/O port 1113 and security circuitry 1115, which are connected together by a conventional data bus.

Control logic 1111 in memory cards provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports. CPU 1101 with its control logic can perform calculations, access memory locations, modify memory contents, and manage input/output ports. Some cards have a coprocessor for handling complex computations like cryptographic operations. Input/output ports 1113 are used under the control of a CPU and control logic, for communications between the card and a card interface device. Timer 1109 (which generates or provides a clock pulse) drives the control logic 1111 and CPU 1101 through the sequence of steps that accomplish memory access, memory reading or writing, processing, and data communication. A timer may be used to provide application features such as call duration. Security circuitry 1115 includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed ("blown") upon completion of testing to prevent later access. The AU data after the ALU has been authenticated and verified is stored in EEPROM 1105. The IC card private key will be stored in a secure memory location. The IC card public key and public key certificate is preferably stored in EEPROM 1105. The authentication process as described herein is performed by the CPU 1101.

Figure 11 also shows a possible configuration for the application provider, transmitting entity and for the CA. CPU 1101 present in the application provider encrypts the necessary information using encryption techniques described herein and performs the necessary data operations. CPU 1101 present in the certification authority is used to sign the Application Load Certificate and the public key certificate as described herein.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of

ANNEX A TO THE DESCRIPTION

the invention.

For example, while loading an application is discussed herein, the same secure loading processes can apply to transmitting other types of data such as data blocks, database files, word processing documents or any other type of data need to be transmitted in a secure manner.

ANNEX A TO THE DESCRIPTION

WE CLAIM:

1. A method for securely transporting data onto an integrated circuit card by using an individualized key set for said card, comprising the steps of:

storing a private key and public key pair unique to said integrated circuit card in said memory located on said integrated circuit card;

retrieving said stored public key from said integrated circuit card;

encrypting at least a portion of said data to be transported onto said card, using said retrieved public key;

transmitting said encrypted data to said integrated circuit card; and

decrypting said encrypted data using said integrated circuit card's private key to recover said transported data.

2. The method of claim 1, further including the step of storing said decrypted data on said integrated circuit card.

3. The method of claim 1, wherein a certification authority digitally signs said integrated circuit card's public key to produce a public key certificate unique to said card and stored thereon, and wherein said public key certificate is verified prior to said transmitting step.

ANNEX A TO THE DESCRIPTION

4. The method of claim 3, wherein said public key certificate is verified with said certification authority's stored public key prior to said transmitting steps.

5. The method of claim 4, wherein said retrieved public key certificate is recovered and compared with said stored public key.

6. The method of claim 5, wherein said integrated circuit card's public and private keys are provided using an asymmetric technique.

7. The method of claim 6, wherein said asymmetric technique is RSA.

8. A method performed by an integrated circuit card for processing incoming data transmission to said integrated circuit card by using an individualized key set for the card, comprising the steps of:

receiving said data transmission comprising data encrypted with a public key stored on said integrated circuit card, said public key forming part of said individualized key set;

retrieving a unique private key for said integrated circuit card which is part of said individualized key set; and

decrypting said encrypted data with said unique private key to recover said data.

9. The method of claim 8, further including the step of storing said decrypted data on said integrated circuit card.

10. The method of claim 8, wherein said individualized key set is generated by asymmetric encryption.

11. The method of claim 8, wherein a certification authority digitally signs said integrated circuit card's public key to produce a public key certificate unique to said card and stored thereon, and wherein said public key certificate is verified prior to said transmitting step.

12. The method of claim 11, wherein said public key certificate is retrieved prior to said transmitting steps.

13. The method of claim 12, wherein said retrieved public key certificate is verified using said certification authority's stored public key.

14. An apparatus located on an integrated circuit card by using an individualized key set for said card for processing an incoming secure data transmission comprising:

means for receiving said data transmission comprising data encrypted with a public key stored on said integrated circuit card, said public key forming part of said individualized key set;

means for retrieving a unique public key for said integrated circuit card which is part of said individualized key set; and

means for decrypting said encrypted data with said unique private key to recover said data.

15. The apparatus of claim 14, further comprising means for storing said data on said integrated circuit card.

16. The apparatus of claim 14, further including means for retrieving a public key certificate which is generated by a certificate authority digitally signing said unique public key.

17. The apparatus of claim 16, further including means for transmitting said public key certificate prior to said receiving means receiving.

18. The apparatus of claim 17, wherein said transmitted public key certificate is verified using said certification authority's stored public key.

19. A method of securely transporting data onto an integrated circuit card by using an individualized key set for the card, comprising the steps of:

providing a first unique private and public key pair for a certification authority;

storing a second unique private and public key pair which

form said individualized key set for said integrated circuit card in a memory located on said integrated circuit card;

encrypting said second public key with said first certification authority's private key to form a public key certificate:

storing said public key certificate on said integrated circuit card;

retrieving said stored public key certificate from said integrated circuit card;

verifying said public key certificate with said first public key to ensure that said public key certificate is valid;

encrypting at least a portion of said data using said retrieved second public key;

transporting said encrypted data to said integrated circuit card; and

decrypting said encrypted data using said second private key to retrieve said data.

20. The method of claim 19, wherein said data comprises an application.

ANNEX A TO THE DESCRIPTION

ABSTRACT OF THE DISCLOSURE

Method and apparatus for securely transporting data onto an IC card. The method is used, for example, to transport data, including application programs, in a secure manner from a source located outside the IC card. At least a portion of the data is encrypted using the public key of a public/secret key pair of the intended IC card unit. The encrypted data is then sent to the IC card and the IC card verifies the key transformation unit using its unique secret key. The data can then be stored on the IC card. A copy of the public key signed by a certification authority can be used to verify that the card is authorized to be part of the overall authorized system.

ANNEX B TO THE DESCRIPTION

ANNEX B

MULTI-APPLICATION IC CARD SYSTEM

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. This allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application at its personalization stage. That application, however, cannot be modified or changed after the card is issued even if the modification is desired by the card user or card issuer. Moreover, if a card user wanted a variety of application functions to be performed by IC cards issued to him or her, such as

ANNEX B TO THE DESCRIPTION

both an electronic purse and a credit/debit function, the card user would be required to carry multiple physical cards on his or her person, which would be quite cumbersome and inconvenient. If an application developer or card user desired two different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal, making the transaction difficult, lengthy and inconvenient.

The Applicant has recognised therefore, that it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card. Although multiple applications could be pre-selected and placed in the memory of the card during is production stage, it would also be beneficial to have the ability to load and delete applications for card post-production as needed.

The increased flexibility and power of storing multiple applications on a single card create new challenges to be overcome concerning the integrity and security of the information (including application code and associated data) exchanged between the individual card and the application provider as well as within the entire system when loading and deleting applications. The Applicant has further recognised that it would be beneficial to have the capability of the IC card system to exchange data among cards, card issuers, system operators and application

ANNEX B TO THE DESCRIPTION

providers securely and to load and delete applications securely at any time from either a terminal or remotely over a telephone line, internet or intranet connection or other data conduit. Because these data transmission lines are not typically secure lines, a number of security and entity-authentication techniques must be implemented to make sure that applications being sent over the transmission lines are only loaded on the intended cards.

As mentioned, it is important -- particularly where there is a continuing wide availability of new applications to the cardholder -- that the system has the capability of adding applications onto the IC card subsequent to issuance. This is highly advantageous since it protects the longevity of the IC cards; otherwise, once an application becomes outdated, the card would be useless. In this regard, to protect against the improper or undesired loading of applications onto IC cards, the Applicant has further recognised that it would be beneficial for the IC card system to have the capability of controlling the loading process and restricting, when necessary or desirable, the use of certain applications to a limited group or number of cards such that the applications are "selectively available" to the IC-cards in the system. This "selective capability" would allow the loading and deleting of applications at, for example, a desired point in time in the card's life cycle. It would also allow the loading of an application only to those cards chosen to receive the selected application.

Accordingly, it is an advantage of a preferred embodiment of the invention that it provides these important features and specifically a secure IC-card system that allows for selective availability of smart card applications which may be loaded onto IC cards.

ANNEX 6 TO THE DESCRIPTION

These and other advantages are achieved by an embodiment of the present invention which proves an IC card system comprising at least one IC card and an application to be loaded onto the card wherein the IC card contains card personalization date and the application is assigned application permissions data designating which IC card or group of IC cards upon which the application may be loaded. The system checks to determine whether the card's personalization data falls within the permissible set indicated by the application's permissions data. If it does, the application may be loaded onto the card.

In a preferred embodiment, the card personalization data is transferred onto the card by the personalization bureau after the card is manufactured. The data preferably includes data representing the card number, the issuer, product class (i.e., such as gold or platinum cards), and the date on which the card was personalized. The card further preferably contains enablement data indicating whether or not the card has been enabled with personalized data.

In a further preferred embodiment, the IC card secure system checks the enablement data prior to loading an application to determine whether or not the card has been enabled. Preferably, if the card has been enabled, the system checks if the card number, the issuer, the product class and/or the date on which the card was personalized are within the acceptable set indicated by the application's permissions data. If so, the application may be loaded onto the IC card.

ANNEX B TO THE DESCRIPTION

In yet another preferred embodiment, the application's permissions data may contain data representative of a blanket permission such that all cards would pass for application loading.

Further aspects, features and advantages of embodiments of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

Fig. 1 is block diagram illustrating the three stages in the life of a multi-application IC card in a secure system;

Fig. 2 is a block diagram illustrating the steps of the card manufacture process;

Fig. 3 is a flow diagram illustrating the steps involved in enabling each of the IC cards in the secure system;

Fig. 4 is a block diagram of an IC card chip which can be used in accordance with an embodiment of the invention;

Fig. 5 is a block diagram illustrating the data stored on the IC card as indicated in block 307 of Fig. 3;

Fig. 5A is a schematic of the data structures residing in an IC card and representing personalization data;

Fig. 6 is a flowchart illustrating the steps of loading an application onto an IC card in the secure system;

Fig. 7 is a flow chart illustrating the checking steps as indicated in block 601 of Fig. 6;

Fig. 8 is a flowchart illustrating the steps undertaken in determining if loading of an application may proceed;

Fig. 9 is a block diagram showing the components of the system architecture for the enablement process of an IC card in a secure multi-application IC card system; and

Fig. 10 is a system diagram of entities involved with the use of the IC card once it has been personalized.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

ANNEX 6 TO THE DESCRIPTION

An embodiment of the present invention provides an IC card system and process which allow the flexibility to load and delete selected applications over the lifetime of a multi-application IC card in response to the needs or desires of the card user, card issuers and/or application developers. A card user who has such a card can selectively load and delete applications as desired if allowed by the card issuer in conjunction with the system operator or Certification Authority ("CA") which controls the loading and deleting process by certifying the transfer of information relating to the process.

By allowing applications to be selectively loaded and deleted from the card, a card issuer can extend additional functionality to an individual IC card without having to issue new cards. Moreover, application developers can replace old applications with new enhanced versions, and applications residing on the same card using a common multiple application operating system may interact and exchange data in a safe and secure manner. For example, a frequent flyer loyalty program may automatically credit one frequent flyer mile to a card user's internal account for every dollar spent with an electronic purse such as the Mondex purse or with a credit/debit application. By allowing the ability to selectively load and delete applications, the card user, subject to the requirements of the card issuer, also has the option of changing loyalty programs as desired.

A card issuer or application developer may intend that a particular application be loaded on only one card for a particular card user in a card system. A regional bank may desire to have a proprietary application reside only on the cards which

ANNEX B TO THE DESCRIPTION

the bank issues. Embodiments in accordance with the present invention would allow for this selective loading and specifically allow for the prevention of loading proprietary applications onto unauthorized cards issued by others.

To achieve these desired objectives, embodiments of the present invention give each card a specific indentity by storing "card personalization data" on the card. Morover, each application to be loaded or deleted on one or more cards in the system is assigned "application permissions data" which specify the cards upon which the applications may be loaded.

The type of personalized data can vary depending upon the needs and requirements of the card system. In the preferred embodiment, described in greater detail below, the personalization data include unique card identification designation data, the card issuer, the product class or type (which is defined by the card issuer) and the date of personalization. However, not all of these data elements are required to be used and additional elements could also be included.

The application permissions data associated with an application, also described in greater detail below, can be a single value in an identity field or could include multiple values in the identity field. For example, the application permissions data in the card issuer field could represent both product class A and product class B from a certain Bank X, indicating that the application could be loaded onto cards designated as product classes A and B issued by Bank X (as indicated in the card product ID field of the card's personalization data).

ANNEX B TO THE DESCRIPTION

In addition, a "global value" could be stored in the issuer field (or other field) of the application permissions data indicating that all IC cards in the system regardless of who issued the card would match this permissions field. In this case, for example, a data value of zero stored in the application permissions card-issuer field will match all of the cards' personalization card-issuer fields.

Figure 1 shows the three steps involved in providing an operational multi-application IC card in a secure system. The first step is the card manufacturing step 101. The second step is the personalization step 103 where card personalization data (also called entity authentication data) is loaded onto the card. The third step is the application loading step 105 which checks to see if a card is qualified to receive an application, i.e., when the personalization data is checked against the application permissions data associated with the application to be loaded. Each of these three steps is described in detail below.

Card Manufacture

Figure 2 shows the steps necessary in manufacturing an IC card in a secure system. Step 201 manufactures the physical IC card by creating the integrated circuit on silicon and placing it on the card. The integrated circuit chip will include RAM, ROM and EEPROM memories. When the card is first manufactured, a global public key of the system operator (in this case called the Certification Authority (CA)) is stored on each card in ROM in step 203. This will allow the card to authenticate that the source of any message to it is from the CA since the public key on the card will be matched to the CA's secret key.

ANNEX 8 TO THE DESCRIPTION

More specifically, this public key stored on the card will allow the individual card to verify data signed with the CA's private key. The public key of the CA, which is stored on the card, is used only for determining if the data sent to the card was signed with the proper CA private key. This allows the card to verify the source of any message coming from the CA.

Step 205 inserts a card enablement key in a secure portion of EEPROM in the card to facilitate card specific confidentiality during enablement, and step 207 inserts a card identifier in EEPROM of the card. The identifier, which can be accessed by any terminal, will allow the system to determine the identity of the card in later processes. The identifier is freely available and will not be used to authenticate messages.

Step 209 stores the operating system code in ROM on the card including any primitives which are called or supported by the operating system. The primitives are written in native language code (e.g., assembly language) and are stored in ROM. The primitives are subroutines which may be called by the operating system or by applications residing on the card such as mathematic functions (multiply or divide), data retrieval, data manipulation or cryptographic algorithms. The primitives can be executed very quickly because they are written in the native language of the processor.

After the IC cards are manufactured, they are sent to a personalization bureau ("PB") to enable and personalize the card by storing card personalization data in the memory of the card. The terms enablement and personalization are used interchangeably herein to indicate the preparatory steps taken to allow the card to be loaded securely with

an application. The individual cards are preferably manufactured in batches and are sent to a personalization bureau in a group for processing.

Card Enablement/Personalization

Figure 3 shows the steps of the card enablement process when the card arrives at a personalization bureau. The personalization bureau may be the card issuer (e.g., a bank or other financial institution) or may be a third party that performs the service for the card issuer. The personalization bureau configures the card to a specific user or user class.

Figure 3 specifically shows the steps taken to enable and personalize each IC card which will work within the system. The cards can be placed in a terminal which communicates with IC cards and which reads the card identifier data (previously placed on the card during the manufacturing process -- see step 207). This card identification data is read from the card in step 301. The terminal will effectively send a "get identification data" command to the card and the card will return the identification data to the terminal.

The PB typically processes a group of cards at the same time, and will first compile a list of IC card identification data for the group of cards it is personalizing. The PB then sends electronically (or otherwise) this list of identification data to the Certification Authority ("CA") which creates a personalization (or enablement) data block for each card identifier. The data block includes the card personalization data organized in a number of identity fields and an individual key set for the card, discussed below. These data blocks are then encrypted and sent to the PB in step 302. By using the

ANNEX B TO THE DESCRIPTION

card identification data, the PB then matches the cards with the encrypted data blocks and separately loads each data block onto the matched card. To insure that the CA controls the identity of the card and the integrity of the system, the PB never obtains knowledge of the content of the data blocks transferred. Some aspects of the personalization are requested by the card issuer to the CA in order to affect their preferred management of the cards they issue. The following additional steps are performed.

Step 303 first checks to see if an enablement bit stored in EEPROM of the card has been already set. If it already has been set, the card has already been configured and personalized and the enablement process will end as shown in step 304. A card cannot be enabled and personalized twice. If the bit has not been set, then the process continues with step 305.

In step 305, the individualized card key set for the card being enabled (which key set is generated at the CA) is stored on the card. The keys can be used later in off-card verification (i.e., to verify that the card is an authentic card). This verification is necessary to further authenticate the card as the one for which the application was intended.

Step 307 generates four different MULTOS Security Manager (MSM) characteristic data elements (otherwise referred to herein as personalization data) for the card at the CA which are used for securely and correctly loading and deleting applications from a particular card. The MSM characteristics also allow for the loading of applications on specific classes of identified cards. (These MSM characteristics are further desczibed in connection with Figure 5.)

ANNEX B TO THE DESCRIPTION

Other data can also be stored on the card at this time as needed by the system design such as an address table or further subroutines.

Step 311 sets the enablement bit in EEPROM of the card which indicates that the enablement process has been completed for the particular card. When this bit is set, another enablement process cannot occur on the card. This ensures that only one personalization and enablement process will occur to the card thus preventing illegal tampering of the card or altering the card by mistake. In the preferred embodiment, the enablement bit is initially not set when the card is manufactured and is set at the end of the enablement process.

Figure 4 shows an example of a block diagram of an IC card chip which has been manufactured and personalized. The IC card chip is located on an IC card for use. The IC card preferably includes a central processing unit 401, a RAM 403, a EEPROM 405, a ROM 407, a timer 409, control logic 411, an I/O ports 413 and security circuitry 415, which are connected together by a conventional data bus.

Control logic 411 in memory cards provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports. CPU 401 with its control logic can perform calculations, access memory locations, modify memory contents, and manage input/output ports. Some cards have a coprocessor for handling complex computations like cryptographic algorithms. Input/output ports 413 are used under the control of a CPU and control logic alone, for communications between the card and a card acceptance device. Timer 409 (which generates or provides a clock pulse) drives the control logic 411 and CPU 401 through the sequence of steps that

ANNEX B TO THE DESCRIPTION

accomplish memory access, memory reading or writing, processing, and data communication. A timer may be used to provide application features such as call duration. Security circuitry 415 includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed ("blown") upon completion of testing to prevent later access. The personalization data to qualify the card is stored in a secured location of EEPROM 405. The comparing of the personalization data to applications permissions data is performed by the CPU 401.

Figure 5 shows the steps of generating and loading the four elements of the card personalization data into the memory of the IC cards, and Fig. 5A shows a schematic of bit maps for each identity field residing in the memory of an IC card containing personalization data in accordance with the present invention. Each data structure for each identity field has its own descriptor code. Step 501 loads the data structure for the identity field "card ID" called "msm_mcd_permissions_mcd_no." This nomenclature stands for MULTOS system manager _ MULTOS card device _ permissions_ MULTOS card device number. Although this number is typically 8 bytes long as shown in Fig. 5A, the data could be any length that indicates a unique number for the card. In the preferred embodiment, 2 bytes are dedicated as a signal indicator, 2 bytes comprise a MULTOS Injection Security Module ID (MISM ID) indicating which security module injected the card with its injected keys when it was manufactured, and 4 bytes comprise an Integrated Circuit Card (ICC) serial number which identifies the individual card produced at the particular MISM.

ANNEX B TO THE DESCRIPTION

Step 503 loads the data structure for the identity field "issuer ID" called "msm_mcd_permissions_ mcd_issuer_id." This nomenclature stands for a MULTOS card device issuer identification number. Each card issuer (such as a particular bank, financial institution or other company involved with an application) will be assigned a unique number in the card system. Each IC card in the MULTOS system will contain information regarding the card issuer which personalized the card or is responsible for the card. A card issuer will order a certain number of cards from a manufacturer and perform or have performed the personalization process as described herein. For example, a regional bank may order 5,000 cards to be distributed to its customers. The "mcd_issuer_id" data structure on these cards will indicate which issuer issued the cards. In the preferred embodiment, the data structure is 4 bytes long (as shown in Fig. 5A at 503A) to allow for many different issuers in the system although the length of the data structure can vary with the needs of the card system.

Step 505 loads the data structure for the identity field "product ID" called "msm_mcd_permissions_mcd_ issuer_product_id." This nomenclature stands for MULTOS card device issuer product identification number. Each card issuer may have different classes of products or cards which it may want to differentiate. For example, a bank could issue a regular credit card with one product ID, a gold credit card with another product ID and a platinum card with still another product ID. The card issuer may wish to load certain applications onto only one class of credit cards. A gold credit card user who pays an annual fee may be entitled to a greater variety of applications than a regular credit card user who pays no annual fee. The product ID field identifies the card as a

ANNEX 6 TO THE DESCRIPTION

particular class and will later allow the card issuer to check the product ID and only load applications onto cards which match the desired class.

Another way to differentiate products is by application type, such as by categorizing the application as financial, legal, medical and/or recreational, or by assigning particular applications to a group of cards. For example, one card issuer may have different loyalty programs available with different companies to different sets of card users. For example, a bank may have an American Airlines® loyalty program and a British Airways® loyalty program for different regions of the country dependent on where the airlines fly. The product type allows the issuer to fix the product classification of the card during the personalization process. When loading applications onto the card, the product type identification number on each card will be checked to make sure it matches the type of card onto which the issuer desires to load. The product type data structure is preferably an indexing mechanism (unlike the other personalization data structure) of 8 bits (as shown at 505A in Fig. 5A) but could be any length depending upon the needs of the card system. In the illustrated embodiment, the resulting instruction would be to locate the second bit (since the byte's indicated value is 2) in the array to be searched (see discussion of step 809 below).

Step 507 loads the data structure for the identity field data called "msm_mcd_permissions_mcd_ controls_data_ date." This nomenclature stands for the MULTOS card device controls data date or, in other words, the date on which the card was personalized so that, for example, the application loader can load cards dated only after a certain date, load cards before a certain date (e.g., for application updates) or load

ANNEX 6 TO THE DESCRIPTION

cards with a particular data date. The information can include the year, month and day of personalization or may include less information, if desired. The data_date data structure is preferably 1 byte in length (see 507A in Fig. 5A) although it could be any length depending upon the needs of the particular card system used.

Once all of the personalization data structures are loaded and stored in the card, the card has been identified by issuer, product class, date and identification number (and other data fields, if desired), and the card cannot change its identity; these fields cannot be changed in the memory of the card. If a card user wants to change the product_id stored in the card to gain access to different applications available to another product type, a new card will have to be issued to the user containing the correct personalization data. This system is consistent with a gold card member receiving a new card when the classification is changed to platinum.

After the card has been enabled and personalized by storing its individual card key set, MSM personalization characteristics and enablement bit as described in Fig. 3, the card is ready to have applications loaded into its memory.

Loading Applications

The application loading process contains a number of security and card configuration checks to ensure the secure and proper loading of an application onto the intended IC card. The application loading process is preferably performed at the personalization bureau so that the card will contain one or more applications when the card is issued. The card may contain certain common applications which will be present on every card the issuer sends out, such as an electronic purse application or a credit/debit

ANNEX 6 TO THE DESCRIPTION

application. Alternatively, the personalization bureau could send the enabled cards to a third party for the process of loading applications. The multiple application operating system stored in the ROM of each card and the card MSM personalization data is designed to allow future loading and deleting of applications after the card has been issued depending upon the desires of the particular card user and the responsible card issuer. Thus, an older version of an application stored on the IC card could be replaced with a new version of the application. An additional loyalty application could also be added to the card after it has been initially sent to the card user because the application is newly available or the user desires to use the new application. These loading and deleting functions for applications can be performed directly by a terminal or may be performed over telephone lines, data lines, a network such as the Internet or any other way of transmitting data between two entities. In the present IC card system, the process of transmitting the application program and data ensures that only IC cards containing the proper personalization data and which fit on application permissions profile will be qualified and receive the corresponding application program and data.

Figure 6 shows the preferred steps performed in loading an application onto an IC card in the MULTOS IC card system. For this example, the personalization bureau is loading an application from a terminal which enabled the same card. Step 601 performs an "open command" initiated by the terminal which previews the card to make sure the card is qualified to accept the loading of a specific application. The open command provides the card with the application's permissions data, the application's size, and instructs the card to determine (1) if the enablement bit is set indicating the card

ANNEX 6 TO THE DESCRIPTION

has been personalized; (2) whether the application code and associated data will fit in the existing memory space on the card; and (3) whether the personalization data assigned to the application to be loaded allows for the loading of the application onto the particular card at issue. The open command could also make additional checks as required by the card system. These checking steps during the open command execution will be described in detail in conjunction with Figure 7.

After the open command has been executed, the application loader via the terminal will be advised if the card contains the proper identification personalization data and if enough room exists in the memory of the card for the application code and related data. If there is insufficient memory, then a negative response is returned by the card and the process is abended (abnormally ended). If the identification personalization data does not match the applications permissions data, a warning response is given in step 603, but the process continues to the load and create steps. Alternatively, if there is no match, the process may automatically be abended. If a positive response is returned by the card to the terminal in step 605, the application loader preferably proceeds to next steps. The open command allows the application to preview the card before starting any transfer of the code and data.

Step 607 then loads the application code and data onto the IC card into EEPROM. The actual loading occurs in conjunction with create step 609 which completes the loading process and enables the application to execute on the IC card after it is loaded. The combination of the open, load and create commands are sent by the terminal, or another application provider source, to the IC card to perform the application

ANNEX 6 TO THE DESCRIPTION

loading process. The operating system in the IC cards is programmed to perform a specific set of instructions with respect to each of these commands so that the IC card will communicate with and properly carry out the instructions from the terminal.

Step 609 performs the create command which at least: (1) checks if an application load certificate is signed (encrypted) by the CA and therefore authenticates the application as a proper application for the system; and (2) checks the card personalization data stored on the card against the permissions profile for the application to be loaded to qualify the card for loading. It may do other checks as required. If one of the checks fails, then a failure response 610 is given and the process aborts. The application after it has passed these checks will be loaded into the memory of the card.

Figure 7 shows the various steps of the open step 601 of Fig. 6 in more detail. Step 701 determines if the enablement (i.e., control) bit is set. This bit is set when the card has completed its personalization process and has been assigned its personalization data. An application can be loaded on an IC card in the card system only if the card contains the personalization data. If the enablement bit is not set, the card has not been personalized and therefore the card returns a negative response 703 to the terminal. If the enablement bit is set, then the card has been enabled and the test conditions continue with step 711.

Step 711 checks if there is sufficient space in the memory on the card to store the application code and its associated data. Applications will typically have associated data related to their functions. This data will be used and manipulated when the application is run. Storage space in the memory of an IC card is a continuing concern

due to the relatively large physical space required for EEPROM and how it fits in the integrated circuit which is desired to be small enough to fit on a credit card sized card. An example of the size of a preset EEPROM on an IC card is 16K bytes although the actual size varies. Applications can range from 1K byte or less for a very simple application up to the size of available memory for a more sophisticated application. The data associated with an application can range from no data being stored in the card memory to a size constrained by the amount of available memory. These varied sizes of application code and data continually increase as applications become more advanced and diverse.

MULTOS as an operating system is not limited by the number of applications and associated data it can store on the card. Thus, if five applications can fit in the available memory of the card, the card user will have greatly increased functionality than if one or two applications were stored on the card. Once a card's memory is filled to its capacity, however, a new application cannot be loaded onto the card unless another application including its code and data of sufficient size can be deleted. Therefore, checking the amount of available space on the card is an important step. If there is not sufficient space, then an insufficient space response 713 will be returned to the terminal. The application loader can then decide if another existing application on the card should be deleted to make room for the new application. Deletion depends upon the card issuer having an application delete certificate from the CA. If there is sufficient space on the card, then the process continues with step 715.

ANNEX 6 TO THE DESCRIPTION

An example of the testing of memory spaces in step 711 is now described. The numbers used in this example in no way limit the scope of the invention but are used only to illustrate memory space requirements. An IC card may have 16K available EEPROM when it is first manufactured. The operating system data necessary for the operating system may take up 2K of memory space. Thus, 14K would remain. An electronic purse application's code is stored in EEPROM and may take up 8K of memory space. The purse application's required data may take up an additional 4K of memory space in EEPROM. The memory space which is free for other applications would thus be 2K (16K-2K-8K-4K=2K). If a card issuer wants to load a credit/debit application whose code is 6K bytes in size onto the card in this example, the application will not fit in the memory of the IC card. Therefore, the application cannot load the new application without first removing the purse application from the card. If a new credit/debit application was loaded into EEPROM of the IC card, then it would have to overwrite other application's code or data. The application loader is prevented from doing this.

Figure 8 shows the steps performed in determining whether the card's personalization data falls within the permissible set of cards onto which the application at issue may be loaded. These steps are preferably performed during the execution of the "create" command. However, these steps may be performed at any time during the loading or deleting of an application. As described previously, the card is personalized by storing data specific to the card (MSM personalization data) including: a card ID designation specific to an individual card, the card issuer number indicating the issuer of the card, the product type of the card, such as a gold or platinum card, and the date the

ANNEX 6 TO THE DESCRIPTION

card was personalized. This data uniquely identifies the card apart from all other IC cards in the system.

Accordingly, applications can be selectively stored on individual cards in the IC card system on virtually any basis, including the following. An application can be loaded selectively to cards containing one or more specific card numbers. An application can be selectively loaded on one or more cards containing a specified card issuer ID. Moreover, an application can be loaded only upon one type of product specified by the particular card issuer, and/or the application can be loaded only on cards which have a specified date or series of dates of personalization. Each of the personalization data allows an application to be selectively loaded onto certain cards or groups of cards and also ensures that cards without the proper permissions will not receive the application. Personalization data types in addition to the four described can also be used as needed.

The selection of IC cards upon which a particular application may be loaded is made possible by the use of "applications permissions data" which is assigned to the application and represents at least one set of cards upon which the application may be loaded. The set may be based on virtually any factor, including one or more of the following: card numbers, card issuers, product types or personalization dates. Although the individual card's personalization data typically identify one specific number, one card issuer, one product type and one date, the application's permissions data may indicate a card numbers or a blanket permission, a card issuer or a blanket permission, and a number of product types and dates.

For example, a frequent loyalty program may be configured to allow its loading and use on cards in different product classes belonging to one card issuer. In addition, the application permissions data may indicate that the loyalty program can be used on gold and platinum product types if the card was issued after May, 1998. Thus, the MSM permissions check will determine if the card's individual personalization data is included in the allowed or permissible set of cards upon which the application may be loaded. If it is, the application will be loaded.

To expedite the comparison process, an alternative embodiment may include setting one or more permissions data at zero representing a blanket permission for that particular data. For instance, by placing a zero for the "card number" entry in the application permissions data or some other value indicating that all cards may be loaded regardless of their number, the system knows not to deny any cards based on their card number. Moreover, if a zero is placed in the application's permissions data "issuer ID," then all cards similarly will pass the "issuer" test comparison. This feature allows greater flexibility in selecting groups of cards. The zero indicator could also be used for other permissions data, as required.

Referring to Figure 8, each of the permissions data is checked in the order shown, but other orders could be followed because if any one of the permissions fails, the application will be prevented from being loaded on the IC card being checked. The permissions are preferably checked in the order shown. Step 801 checks if the application permissions product type set encompasses the card's product type number stored in the memory of the card. Each card product type is assigned a number by the

ANNEX B TO THE DESCRIPTION

system operator. The product types are specified for each card issuer because different card issuers will have different product types. The cards are selectively checked to ensure that applications are loaded only on cards of authorized product type. The application permissions product type set can be 32 bytes long which includes multiple acceptable product types or can be a different length depending upon the needs of the system. Using data structure 505A as an example, the operating system would check bit number 2 in the 256 bit array (32 bytes x 8 bits per byte) resulting from the 32 byte long application permissions data structure. If the permissions check fails, then the card returns a failure message to the terminal in step 803. If the product type check passes (for example, the value of bit no. 2 being 1), then the process continues with step 805.

Step 805 checks if the application permissions allowable card issuer number set encompasses the card's issuer number stored in the memory of the card or if the application permissions issuer data is zero (indicating all cards pass this individual permissions check). Each card issuer is assigned a number by the system operator and the cards are selectively checked to ensure that applications are loaded only on cards distributed by authorized card issuers. The application permissions card issuer number set can be 4 bytes long if one issuer is designated or can be longer depending upon the needs of the system. If the issuer check fails, then the card returns a failure message to the terminal in step 807. If the check passes, then the process continues with step 809.

Step 809 checks if the application permissions date set encompasses the card's data date stored in the memory of the card. The date that the IC card was personalized will be stored and will preferably include at least the month and year. The

ANNEX 6 TO THE DESCRIPTION

cards are selectively checked to ensure that applications are loaded only on cards with the authorized personalization date. The application permissions date set can be 32 bytes long which includes multiple dates or can be a different length depending upon the needs of the system. If the date permissions check fails, then the card returns a failure message to the terminal in step 811. If the date check passes, then the process continues with step 813.

Step 813 checks if the application permissions allowable card number set encompasses the card's ID number stored in the card memory or if the application permissions allowable card number data is zero (indicating all cards pass this individual permissions check). The testing of the permissions is performed on the card during the execution of the open, load and create commands. The application permissions card number data set can be 8 bytes long if one number is designated or can be longer depending upon the needs of the system. If the card number check fails, then the card returns a failure message to the terminal in step 815. If the check passes, then the process continues with step 817.

Summary of IC Card System's Process

Figure 9 shows the components of the system architecture for the card initialization process of an IC card in a secure multiple application IC card system. The system includes a card manufacturer 102, a personalization bureau 104, an application loader 106, the IC card 107 being initialized, the card user 109 and the certification authority 111 for the entire multiple application secure system. The card user 131 is the

ANNEX B TO THE DESCRIPTION

person or entity who will use the stored applications on the IC card. For example, a card user may prefer an IC card that contains both an electronic purse containing electronic cash (such as MONDEX™) and a credit/debit application (such as the MasterCard® EMV application) on the same IC card. The following is a description of one way in which the card user would obtain an IC card containing the desired applications in a secure manner.

The card user would contact a card issuer 113, such as a bank which distributes IC cards, and request an IC card with the two applications both residing in memory of a single IC card. The integrated circuit chip for the IC card would be manufactured by manufacturer 102 and sent to the card issuer 113 (or an entity acting on its behalf) in the form of an IC chip on a card. As discussed above (see steps 201-209), during the manufacturing process, data is transmitted 115 via a data conduit from the manufacturer 102 to card 107 and stored in IC card 107's memory. (Any of the data conduits described in this figure could be a telephone line, Internet connection or any other transmission medium.) The certification authority 111, which maintains encryption/decryption keys for the entire system, transmits 117 security data (i.e., global public key) to the manufacturer over a data conduit which is placed on the card by the manufacturer along with other data, such as the card enablement key and card identifier. The card's multiple application operating system is also stored in ROM and placed on the card by the manufacturer. After the cards have been initially processed, they are sent to the card issuer for personalization and application loading.

ANNEX 6 TO THE DESCRIPTION

The card issuer 113 performs, or has performed by another entity, two separate functions. First, the personalization bureau 104 personalizes the IC card 107 in the ways described above, and second, the application loader 106 loads the application provided the card is qualified, as described.

Regarding personalization, an individualized card key set is generated by the CA and stored on the card (see Fig. 3). The card is further given a specific identity using MSM personalization (see Fig. 3, step 307 and Fig. 5) including a card ID number, an issuer ID number identifying the card issuer which processed the card, a card product type number which is specified by the card issuer and the date upon which the personalization took place. After the card has been personalized, applications need to be loaded onto the card so that the card can perform desired functions.

The application loader 106, which could use the same terminal or data conduit as personalization bureau 104, first needs to have determined if the card is qualified to accept the application. This comparison process takes place on the card itself (as instructed by its operating system) using the permissions information. The card, if it is qualified, thus selectively loads the application onto itself based upon the card's identity and the card issuer's instructions. The application loader communicates 119 with the IC card via a terminal or by some other data conduit. After the applications have been loaded on the card, the card is delivered to the card user 109 for use.

The secure multiple application IC card system described herein allows for selective loading and deleting of applications at any point in the life cycle of the IC card after the card has been personalized. Thus, a card user could also receive a personalized

ANNEX 6 TO THE DESCRIPTION

card with no applications and then select a desired application over a common transmission line such as a telephone line or Internet connection.

Figure 10 is a system diagram of entities involved with the use of an IC card once it has been personalized. The system includes an IC card 151, a terminal 153, an application load/delete entity 155, the certification authority 157, a card issuer 171 and other IC cards 159 in the system. The arrows indicate communication between the respective entities. The CA 157 facilitates loading and deleting of applications. After providing the MSM permissions data and card specific keyset to the card during card enablements, the CA allows applications to be later loaded and deleted preferably by issuing an application certificate. Application specific keys are required to authenticate communication between a card and terminal. The IC card 151 also can communicate with other IC cards 159. Card issuer 171 is involved with all decisions of loading and deleting applications for a card which it issued. All communications are authenticated and transmitted securely in the system.

For instance, IC card 151 will use the following procedure to load a new application onto the card. IC card 101 is connected to terminal 153 and the terminal requests that an application be loaded. Terminal 153 contacts application load/delete entity 155 which, as a result and in conjunction with card issuer 171, sends the application code, data and application permissions data (along with any other necessary data) to terminal 153. Terminal 153 then queries card 151 to ensure it is the correct card onto which the application may be loaded. If IC card passes the checks discussed above, the application is loaded onto card 151. The CA 157 provides the application load or

delete certificate that enables the application to be loaded or deleted from the card. This example shows one way to load the application, but other variations using the same principles could be performed, such as directly loading the application at the application load/delete entity 155.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

For example, it will be appreciated that the MSM personalization and permissions data may not only be used for loading applications onto IC cards but also for deleting applications from said cards. The same checks involving MSM permissions and loading applications are made for deleting applications. A delete certificate from the CA authorizing the deletion of an application will control from which cards the application may be deleted. This is accomplished through the personalization data stored on each IC card and the permissions check as described herein.

Moreover, the data may also be applicable to personal computers or other units onto which applications may be loaded which are not physically loaded on cards. In addition, the application's permissions data may actually include data representative of a set or sets of cards to be excluded, instead of included -- cards that cannot be loaded with the application.

The scope of the present disclosure includes any novel feature or combination of features disclosed therein either explicitly or implicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or mitigates any or all of the problems addressed by the present invention. The applicant hereby gives notice that new claims may be formulated to such features during the prosecution of this application or of any such further application derived therefrom. In particular, with reference to the appended claims, features from dependent claims may be combined with those of the independent claims in any appropriate manner and not merely in the specific combinations enumerated in the claims.

ANNEX B TO THE DESCRIPTION

CLAIMS:

1. An IC card system comprising at least one IC card, an application to be loaded onto said card and means for determining whether said card is qualified to accept the loading of said application onto said card.

2. The IC card system of claim 1, wherein said IC card contains card personalization data, and said application is assigned application permissions data representing at least one set of IC cards upon which said application may be loaded.

3. The IC card system of claim 2, wherein said determining means compares said card personalization data with said application permissions data.

4. The IC card system of claim 3, wherein whether said application is loaded onto said IC card depends on the result of said comparison, such that in the event the card personalization data matches said permissions data set the card is qualified and the application is loaded.

5. The IC card system of any of claims 2 to claim 4, wherein said personalization data comprises data representative of a unique card identification designation.

6. The IC card system of any of claims 2 to claim 5, wherein said personalization data comprises data representative of a card issuer.

7. The IC card system of any of claims 2 to claim 6, wherein said personalization data comprises data representative of a product class.

8. The IC card system of any of claims 2 to claim 7, wherein said personalization data comprises data representative of a date.

9. An IC card system comprising at least one IC card and an application, wherein said IC card contains personalization data representative of that card and said application is assigned a permissions data set representing at least one IC card upon which said application may be loaded, said system further comprising means for determining whether said personalization data falls within said permissions data set.

10. The IC card system of claim 9 wherein said application is loaded onto said IC card in the event said determining means determines that said personalization data falls within said set.

11. The IC card system of claim 9 or claim 10 wherein said personalization data comprises data representing a card identification designation, and an issuer of said card.

12. The IC card system of any of claims 9 to claim 11 wherein said personalization data comprises data representing a product class and a date.

13. The IC card system of any of claims 9 to 12 wherein said permissions data set includes a plurality of card identification designations.

14. The IC card system of any of claims 9 to 13 wherein said permissions data set includes one or more issuers of IC cards.

15. The IC card system of any of claims 9 to 14 wherein said permissions data set includes one or more product classes.

16. The IC card system of any of claims 9 to 15 wherein said permissions data set includes a plurality range of dates.

17. The IC card system of any of claims 9 to 16 wherein said permissions data set includes all IC cards which attempt to load the application.

18. An IC card system comprising at least one IC card, an application to be loaded onto said card and means for enabling said card to be loaded with said application.

ANNEX B TO THE DESCRIPTION

19. The IC card system of claim 18 wherein said enabling means comprises means for storing personalization data onto said card.

20. The IC card system of claim 18 wherein said enabling means comprises means for setting an enablement bit.

21. The IC card system of claim 19 wherein said enabling means comprises means for setting an enablement bit.

22. The IC card system of claim 20 further comprising means for checking the enablement bit prior to enabling said IC card to determine whether or not said card has already been enabled.

23. The IC card system of claim 21 further comprising means for checking the enablement bit prior to enabling said IC card to determine whether or not said card has already been enabled.

24. A process for loading an application onto an IC card comprising the step of determining whether said IC card is qualified to accept the loading of said application onto said card.

ANNEX B TO THE DESCRIPTION

25. The process of claim 24 wherein said determining step includes the steps of: providing said card with personalization data;

assigning to said application permissions data representing at least one set of IC cards upon which said application may be loaded;

comparing said personalization data with said permissions data; and

loading said application onto said IC card provided said personalization data falls within said set of cards upon which said application may be loaded.

26. The process of claim 25, wherein said personalization data comprises data representative of a card identification designation.

27. The process of claim 25 or claim 26, wherein said personalization data comprises data representative of a card issuer.

28. The process of any of claims 25 to claim 27, wherein said personalization data comprises data representative of a product class.

29. The process of any of claims 25 to claim 28. wherein said personalization data comprises data representative of a date.

30. The process of any of claims 25 to claim 29 further comprising the first step of enabling said card to be loaded with said application.

31. The process of claim 30 wherein said enabling step includes the step of storing personalization data onto said card.

32. The process of claim 30 wherein said enabling step includes the step of setting an enablement bit indicating that the card has been enabled.

33. The process of claim 31 wherein said enabling step further includes the step of setting an enablement bit indicating that the card has been enabled.

34. The process of claim 32 wherein prior to said enabling step a checking step is performed to determine whether said card has been enabled.

35. The process of claim 33 wherein prior to said enabling step a checking step is performed to determine whether said card has been enabled.

36. A process for deleting an application from an IC card comprising the step of determining whether said IC card is qualified to delete said application based upon permissions data associated with said application.

ANNEX 6 TO THE DESCRIPTION

37. The process of claim 36 wherein said determining step includes the steps of:

providing said card with personalization data;

assigning to said application permissions data representing at least one set of IC cards from which said application may be deleted;

comparing said personalization data with said permissions data; and

deleting said application from said IC card provided said personalization data falls within said set of cards from which said application may be deleted.

38. The process of claim 37, wherein said personalization data comprises data representative of a card identification designation.

39. The process of claim 37 or claim 38, wherein said personalization data comprises data representative of a card issuer.

40. The process of any of claims 37 to claim 39, wherein said personalization data comprises data representative of a product class.

41. The process of any of claims 37 to claim 40, wherein said personalization data further comprises data representative of a date.

ANNEX B TO THE DESCRIPTION

42. An IC card system comprising at least one IC card, an application to be deleted from said card and means for determining whether said card is qualified to delete said application from said card.

43. The IC card system of claim 42, wherein said IC card contains card personalization data, and said application is assigned application permissions data set representing at least one set of IC cards from which said application may be deleted.

44. The IC card system of claim 43, wherein said determining means compares said card personalization data with said application permissions data.

45. The IC card system of claim 44, wherein whether said application is deleted from said IC card depends on the result of said comparison, such that in the event the card personalization data matches said permissions data set the card is qualified and the application is deleted.

ABSTRACT

Multi-Application IC Card System

A multi-application IC card system is disclosed having selective application loading and deleting capability. Prior to loading an application onto an IC card a test is conducted to determine if the card is qualified to receive the application using personalization data stored on the card and comparing it with permissions data associated with the application indicating one or more sets of cards upon which the application may be loaded. If the personalization data of the card falls within the allowable set of permissions for that application then the card may be loaded with the application. Preferably, the personalization data includes data representative of the card number, issuer, a product class and the date on which the card is personalized.

I CLAIM:

1. A method for loading an application onto an IC card comprising the steps of:

providing a secret key and public key pair for said IC card;

encrypting at least a portion of said application using a transfer key;

encrypting said transfer key using said IC card's public key to form a key transformation unit;

transmitting said encrypted application and said key transformation unit to said IC card;

decrypting said key transformation unit using said IC card's secret key to recover said transfer key; and

decrypting said encrypted application using said recovered transfer key.

2. The method of claim 1, further including the step of storing said decrypted application on said IC card.

3. The method of claim 1 or claim 2, wherein said encryption technique using said transfer key is symmetric.

4. The method of claim 3, wherein said symmetric technique is DES.

5. The method of any of claims 1 to 4, wherein said IC card's public and private keys are provided using an asymmetric technique.

6. The method of claim 5, wherein said asymmetric technique is RSA.

7. The method of any preceding claim, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

8. The method of any preceding claim, further including the steps of enciphering a second portion of said application exclusive of said at least a portion of said application.

9. The method of claim 8, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

10. The method of claim 8 or claim 9, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

11. The method of any of claims 8 to 10, wherein said key transformation unit indicates the location of said second portion of said application.

12. The method of any preceding claim, wherein said key transformation unit indicates the location of said at least a portion of said application.

13. The method of any preceding claim, wherein said key transformation unit indicates the number of encrypted portions of said application.

14. The method of any preceding claim, further including the steps of providing a public key and secret key set for an application provider; providing a public and secret key set for a certification authority; encrypting said application provider's public key using said certificate authorities' secret key to produce an application load certificate; further signing said encrypted application using said application provider's secret key to produce a signed application and transmitting said signed application and said application load certificate to said IC card.

15. The method of claim 14, further including the step of the IC card verifying said application load certificate with said certification authority's public key.

16. The method of claim 15, further including the steps of verifying the signed encrypted application using the application provider's public key from said decrypted application load certificate.

17. The method of claim 16, wherein said verified application signature is compared to sent encrypted application to determine if they are equivalent.

18. An IC card system comprising:

at least one IC card;

an application provider for providing an application to said at least one IC card;

a communications link coupled to said at least one IC card and said application provider;

a public key and secret key set generated for said IC card;

a transport key generated for use by said applications provider; and

an application, wherein at least a portion of said application is encrypted by said application provider using said transport key; said transport key is encrypted using said IC card's public key to form a key transformation unit; wherein said encrypted application and said key transformation unit are then transmitted to said IC card over said communications link; said transmitted key transformation unit is decrypted using said IC card's private key to recover said transport key; and said transmitted application is decrypted using said recovered transport key to recover said application.

19. The system of claim 18, wherein said recovered application is stored on said card.

20. The system of claim 18 or 19, wherein said encryption technique using said transfer key is symmetric.

21. The system of claim 20, wherein said symmetric technique is DES.

22. The system of any of claims 18 to 21, wherein said IC card's public and private keys are provided using an asymmetric technique.

23. The system of claim 22, wherein said asymmetric technique is RSA.

24. The system of any of claims 18 to 23, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

25. The system of any of claims 18 to 24, further including the steps of enciphering a second portion of said application independently of said at least a portion of said application.

26. The system of claim 25, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

27. The system of claim 25 or claim 26, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

28. The system of any of claims 25 to 27, wherein said key transformation unit indicates the location of said second portion of said application.

29. The system of any of claims 18 to 28, wherein said key transformation unit indicates the location of at least a portion of said application.

30. The system of any of claims 18 to 29, wherein said key transformation unit indicates the number of encrypted portions of said application.

31. The system of any of claims 18 to 30, further including a certification authority, wherein a public key and secret key set is provided for an application provider; a public and secret key set is provided for said certification authority; said certificate authority's secret key is used to sign said application provider's public key to produce an application load certificate; said application provider's secret key is used to further sign said encrypted application to produce a signed encrypted application and said signed encrypted application and said application load certificate is transmitted to said IC card.

32. The system of claim 31, wherein the IC card verifies said application load certificate with said certification authority's public key.

33. The system of claim 32, wherein said IC card verifies the signed encrypted application using the application provider's public key from said verified application load certificate.

34. The system of claim 33, wherein said verified application signature is compared to said encrypted application to determine if they are equivalent.

35. A method for transmitting data from a first microprocessor based device to a second microprocessor based device, comprising the steps of:

encrypting at least a portion of said data at said first device using a transfer key;

encrypting said transfer key with a second key at said first device to form a key transformation unit;

transmitting said encrypted data and said key transformation unit to said second device;

decrypting said key transformation unit at said second device to recover said transfer key; and

decrypting said encrypted data using said recovered transfer key.

36. The method of claim 35, further including the step of storing said decrypted data in said second device.

37. The method of claim 35 or claim 36, wherein said second key is from a public key and private key set used in asymmetric encryption.

38. The method of any of claims 35 to 37, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

39. The method of any of claims 35 to 38, further including the steps of enciphering a second portion of said application independently of said at least a portion of said application.

40. The method of claim 39, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

41. The method of claim 39 or claim 40, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

42. The method of claim 39, wherein said key transformation unit indicates the location of said second portion of said application.

43. The method of any of claims 35 to 42, wherein said key transformation unit indicates the location of said at least a portion of said application.

44. The method of any of claims 35 to 43, further including the steps of providing a public key and secret key set for an application provider; providing a public and secret key set for a certification authority; signing said application provider's public key using said certificate authority's secret key to produce an application load certificate; further signing said encrypted application using said application provider's secret key to produce a signed encrypted application and transmitting said signed application and said application load certificate to said IC card.

45. A method for processing a data transmission comprising the steps of:

receiving said data transmission comprising an application including at least a portion encrypted with a first key and a key transformation unit encrypted with a second key, wherein said key transformation unit comprises said first key;

decrypting said key transformation unit to recover said first key;

decrypting said encrypted application using said first key; and

storing said decrypted application.

46. The method of claim 45, wherein said second key is from a public key and private key set used in asymmetric encryption.

47. The method of claim 45 or claim 46, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

48. The method of any of claims 45 to 47, further including the steps of enciphering a second portion of said application independently of said at least a portion of said application.

49. The method of claim 48, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

50. The method of claim 48 or claim 49, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

51. The method of claim 48, wherein said key transformation unit indicates the location of said second portion of said application.

52. The method of any of claims 45 to 51, wherein said key transformation unit indicates the location of said at least a portion of said application.

53. The method of any of claims 45 to 52, further including the steps of providing a public key and secret key set for an application provider; providing a public and secret key set for a certification authority; signing said application provider's public key using said certificate authorities' secret key to produce an application load certificate; further encrypting said encrypted application using said application provider's secret key to produce a signed encrypted application and transmitting said signed application and said application load certificate to said IC card.

54. The method of claim 53, further including the step of the IC card verifying said application load certificate with said certification authority's public key.

55. The method of claim 54, further including the steps of verifying the signed encrypted application using the application provider's public key from said verified application load certificate.

56. The method of claim 55, wherein said verified application signature is compared to said encrypted application to determine if they are equivalent.

57. An apparatus for processing a data transmission comprising the steps of:

means for receiving said data transmission comprising an application including at least a portion encrypted with a first key and a key transformation unit encrypted with a second key, wherein said key transformation unit comprises said first key;

means for decrypting said key transformation unit to recover said first key;

means for decrypting said encrypted application using said first key; and

means for storing said decrypted application.

58. The apparatus of claim 57, wherein said second key is from a public key and private key set used in asymmetric encryption.

59. The apparatus of claim 57 or claim 58, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

60. The apparatus of any of claims 57 to 59, further including means for enciphering a second portion of said application exclusive of said at least a portion of said application.

61. The apparatus of claim 60, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

62. The apparatus of claim 60 or claim 61, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

63. The apparatus of any of claims 60 to 62, wherein said key transformation unit indicates the location of said second portion of said application.

64. The apparatus of any of claims 57 to 63, wherein said key transformation unit indicates the location of said at least a portion of said application.

65. The apparatus of any of claims 60 to 64, further including means for verifying an application load certificate with said certification authority's public key.

66. The apparatus of claim 65, further including means for verifying the signed encrypted application using an application provider's public key located in said verified application load certificate.

67. The apparatus of claim 66, wherein said verified application signature is compared to the said encrypted application to determine if they are equivalent.

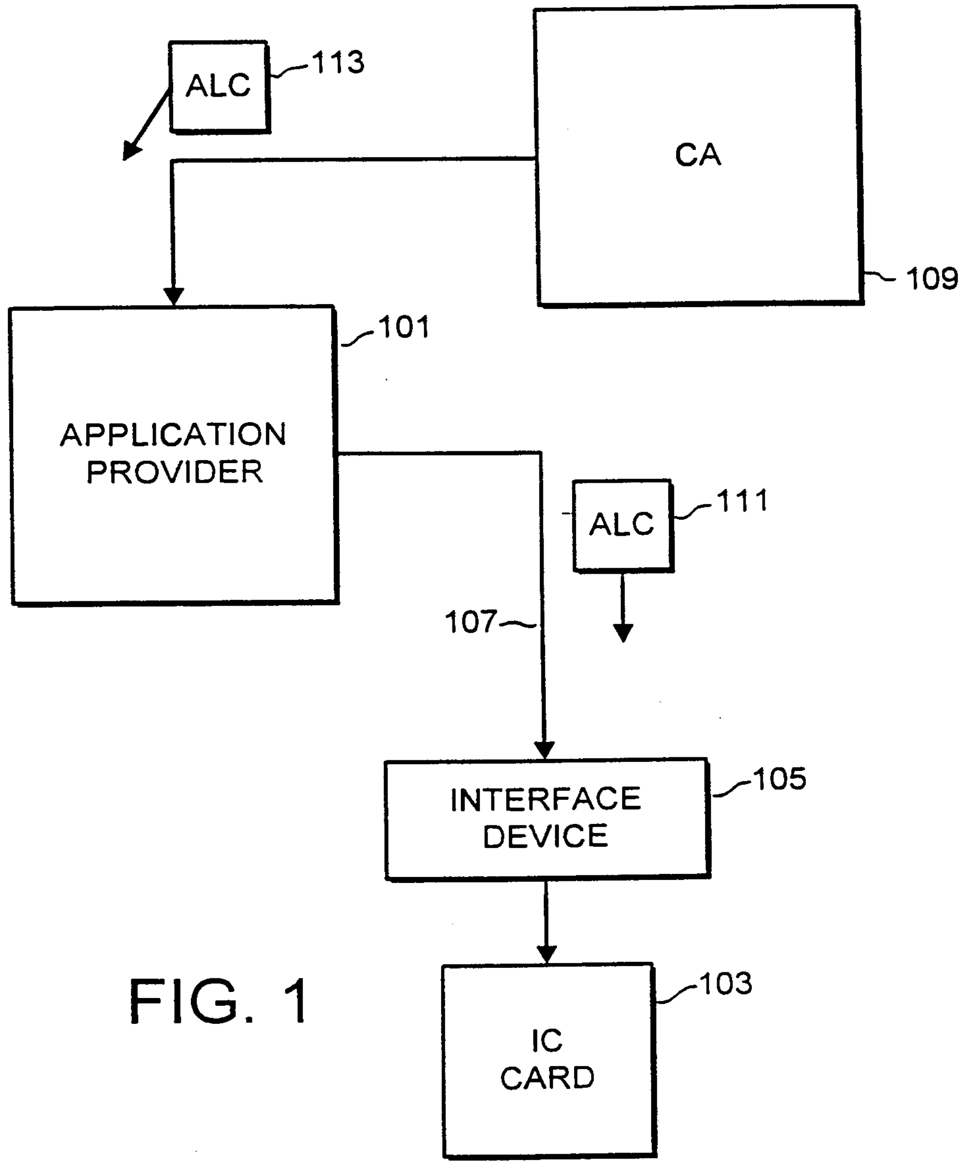

ALC
113
CA
109
101
APPLICATION PROVIDER
ALC
111
107
INTERFACE DEVICE
105
103
IC CARD

FIG. 1

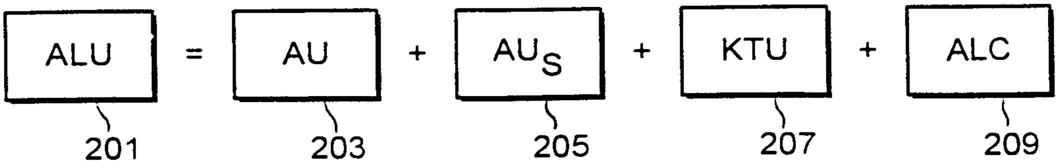

ALU
=
AU
+
AU$_S$
+
KTU
+
ALC
201
203
205
207
209

FIG. 2

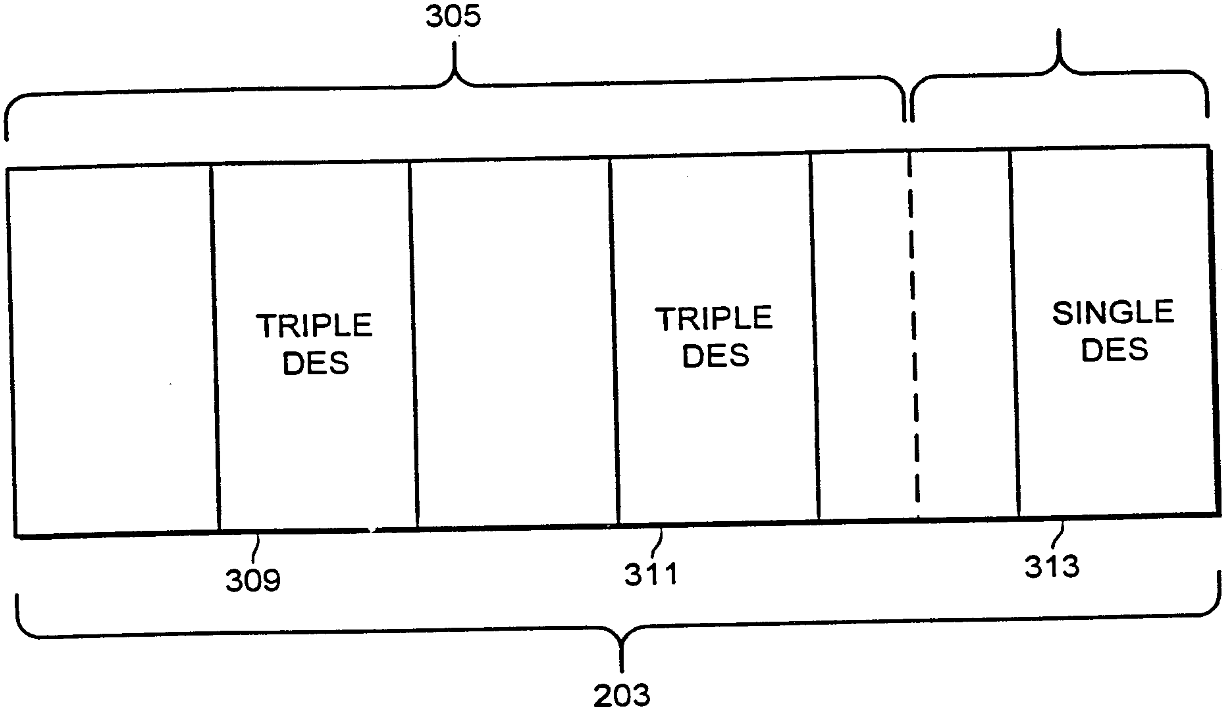

305
TRIPLE DES
TRIPLE DES
SINGLE DES
309
311
313
203

FIG. 3

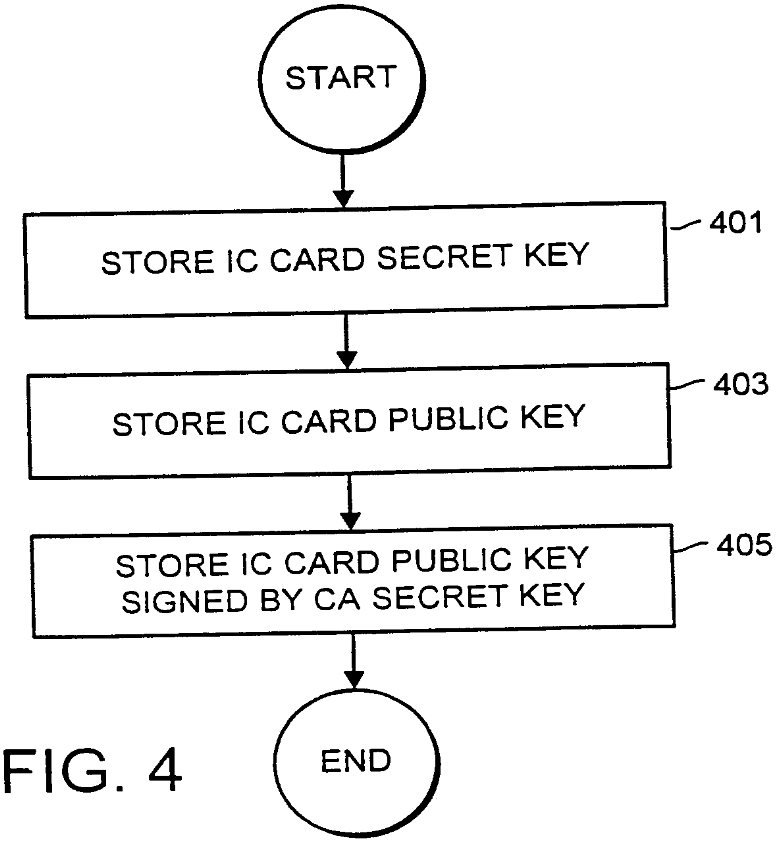

START
STORE IC CARD SECRET KEY
401
STORE IC CARD PUBLIC KEY
403
STORE IC CARD PUBLIC KEY SIGNED BY CA SECRET KEY
405
END

FIG. 4

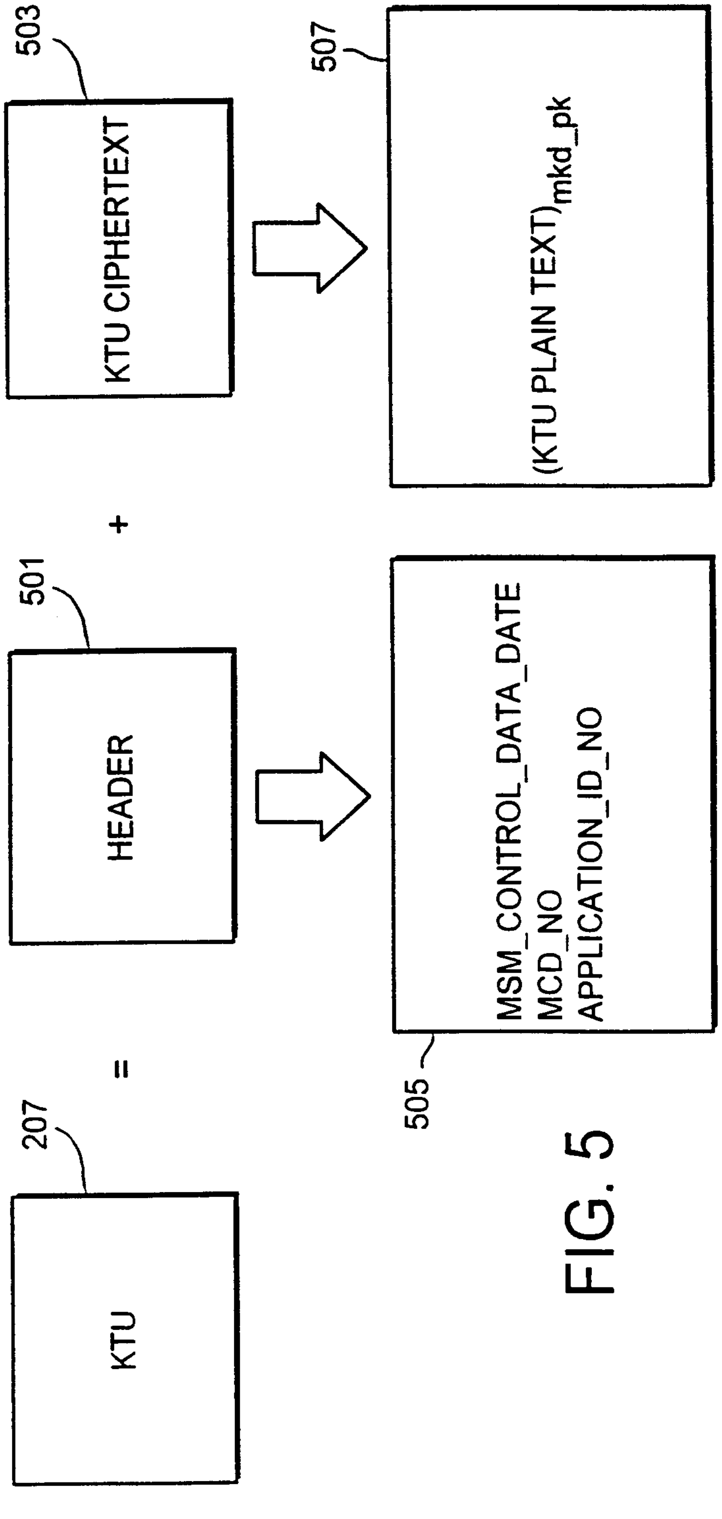

KTU
207
=
HEADER
501
+
KTU CIPHERTEXT
503
MSM_CONTROL_DATA_DATE
MCD_NO
APPLICATION_ID_NO
505
(KTU PLAIN TEXT)mkd_pk
507

FIG. 5

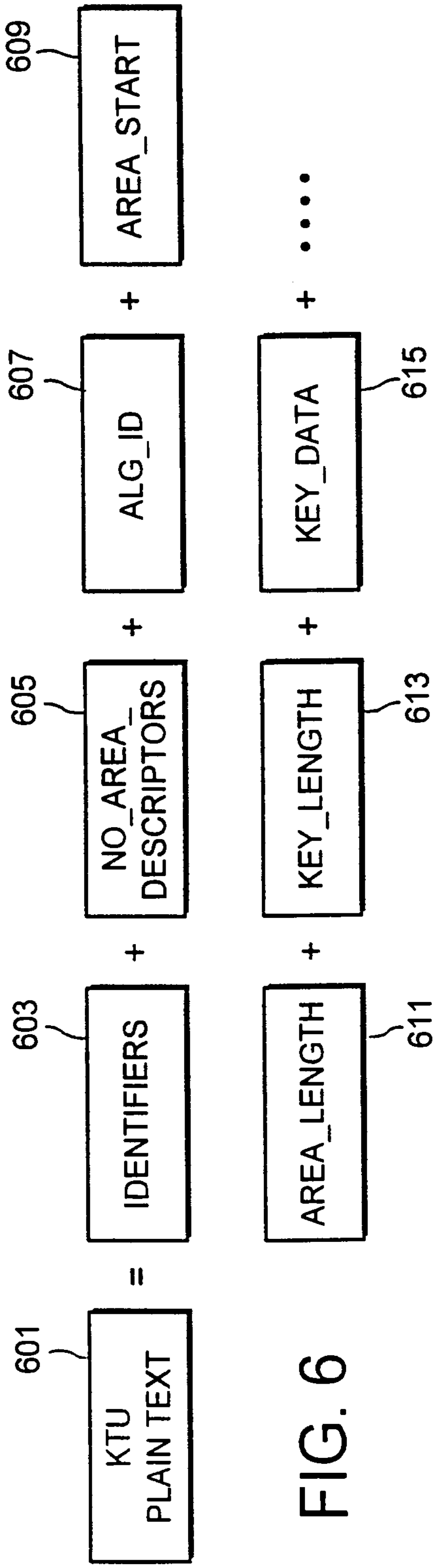

KTU
PLAIN TEXT
601
=
IDENTIFIERS
603
+
NO_AREA_
DESCRIPTORS
605
+
ALG_ID
607
+
AREA_START
609
AREA_LENGTH
611
+
KEY_LENGTH
613
+
KEY_DATA
615
+
• • • •

FIG. 6

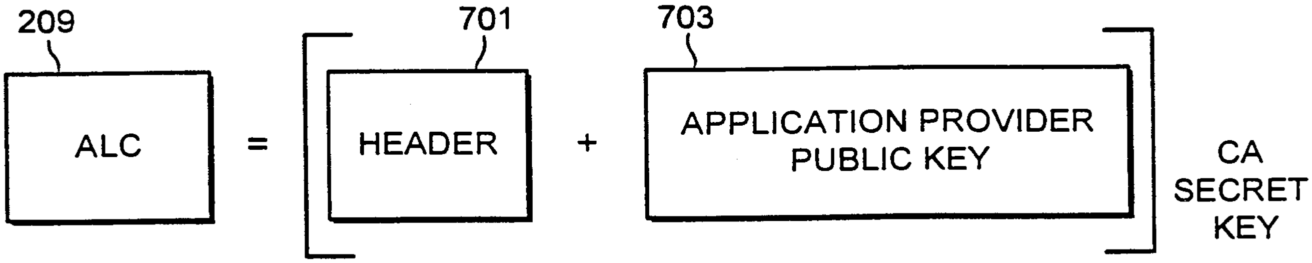


FIG. 7

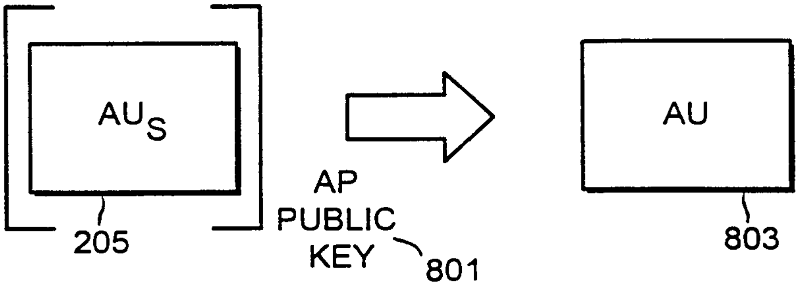


FIG. 8

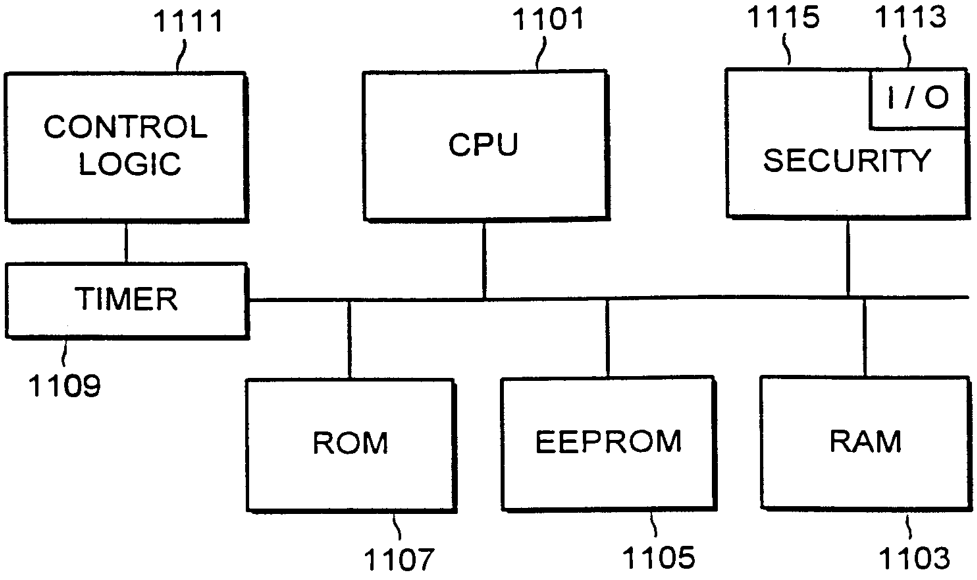


FIG. 11

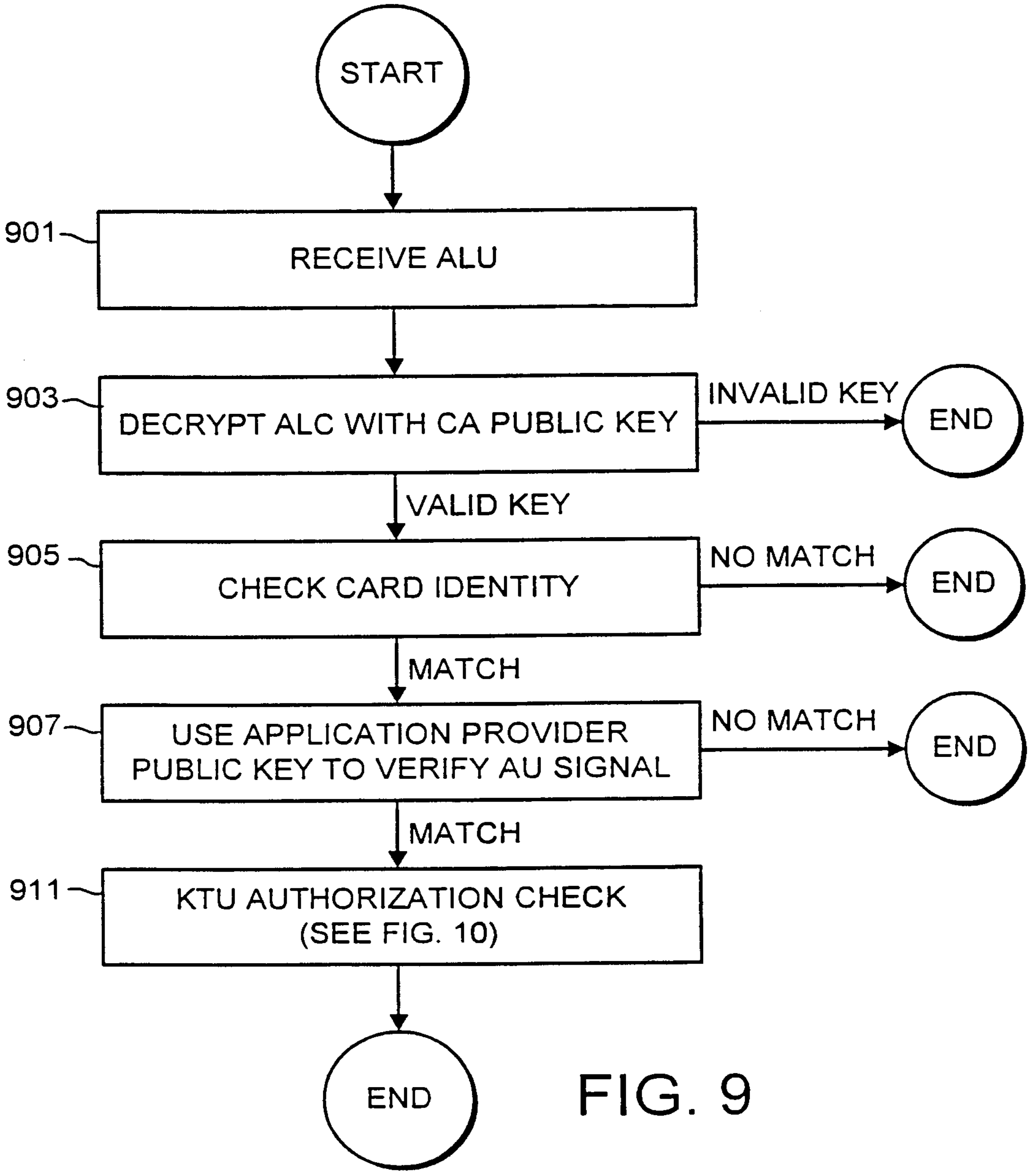

START
901
RECEIVE ALU
903
DECRYPT ALC WITH CA PUBLIC KEY
INVALID KEY
END
VALID KEY
905
CHECK CARD IDENTITY
NO MATCH
END
MATCH
907
USE APPLICATION PROVIDER PUBLIC KEY TO VERIFY AU SIGNAL
NO MATCH
END
MATCH
911
KTU AUTHORIZATION CHECK (SEE FIG. 10)
END

FIG. 9

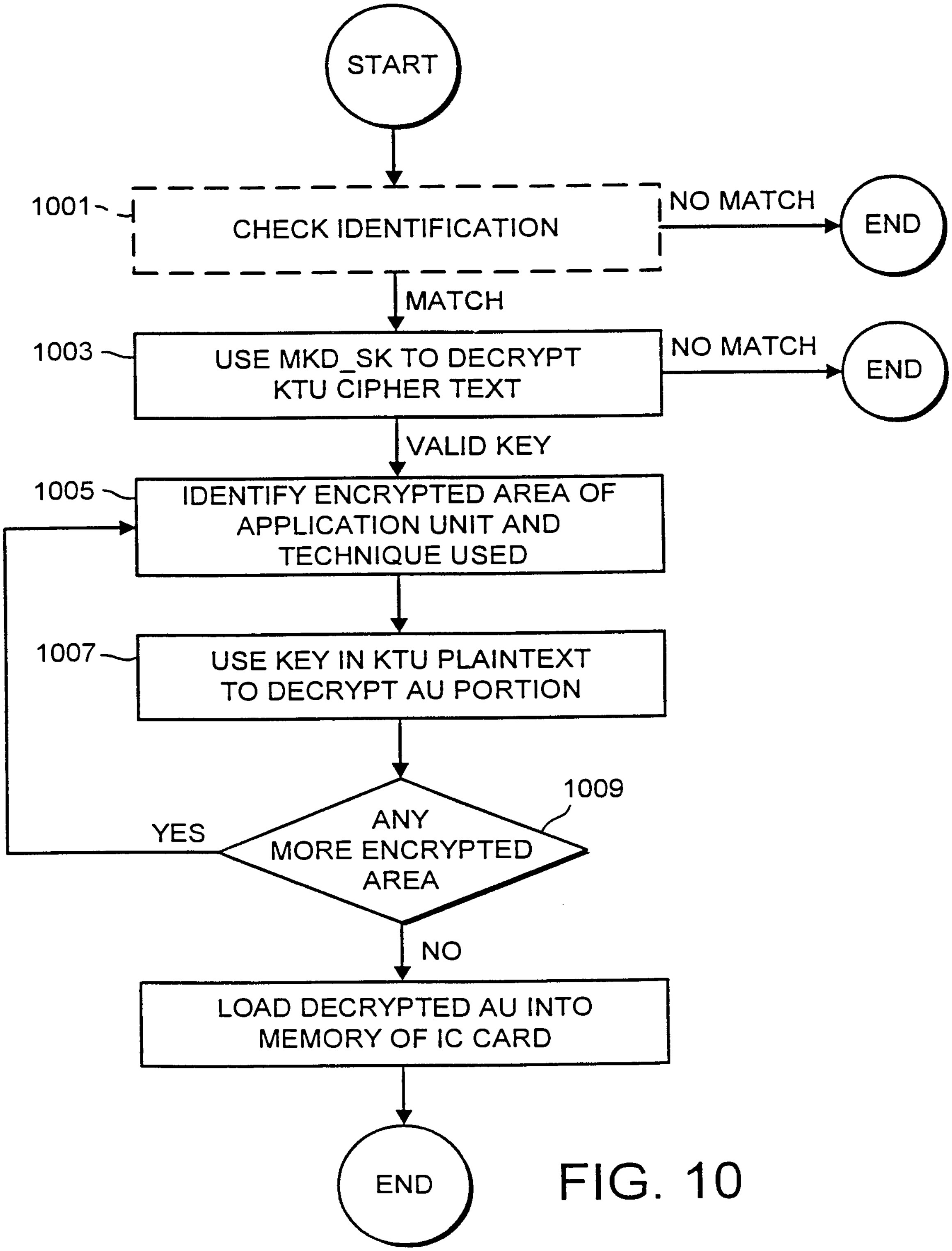

START
1001
CHECK IDENTIFICATION
NO MATCH
END
MATCH
1003
USE MKD_SK TO DECRYPT KTU CIPHER TEXT
NO MATCH
END
VALID KEY
1005
IDENTIFY ENCRYPTED AREA OF APPLICATION UNIT AND TECHNIQUE USED
1007
USE KEY IN KTU PLAINTEXT TO DECRYPT AU PORTION
1009
ANY MORE ENCRYPTED AREA
YES
NO
LOAD DECRYPTED AU INTO MEMORY OF IC CARD
END

FIG. 10

ANNEX A TO THE DRAWINGS

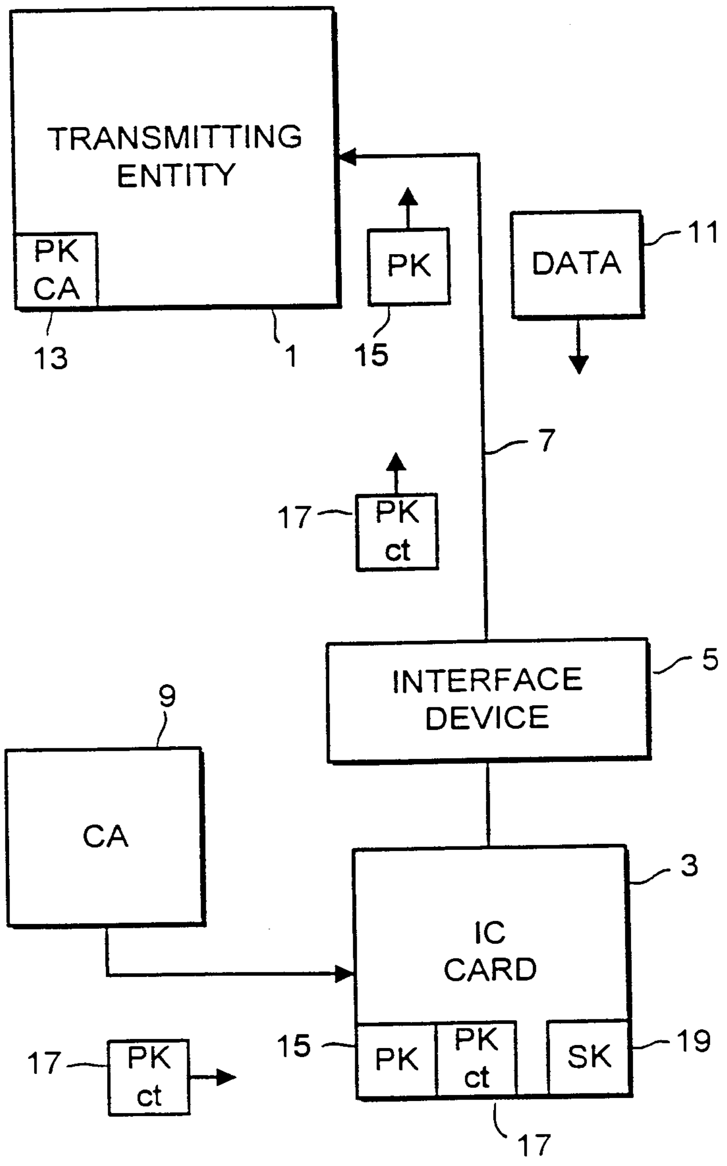


FIG. 1A

ANNEX A TO THE DRAWINGS

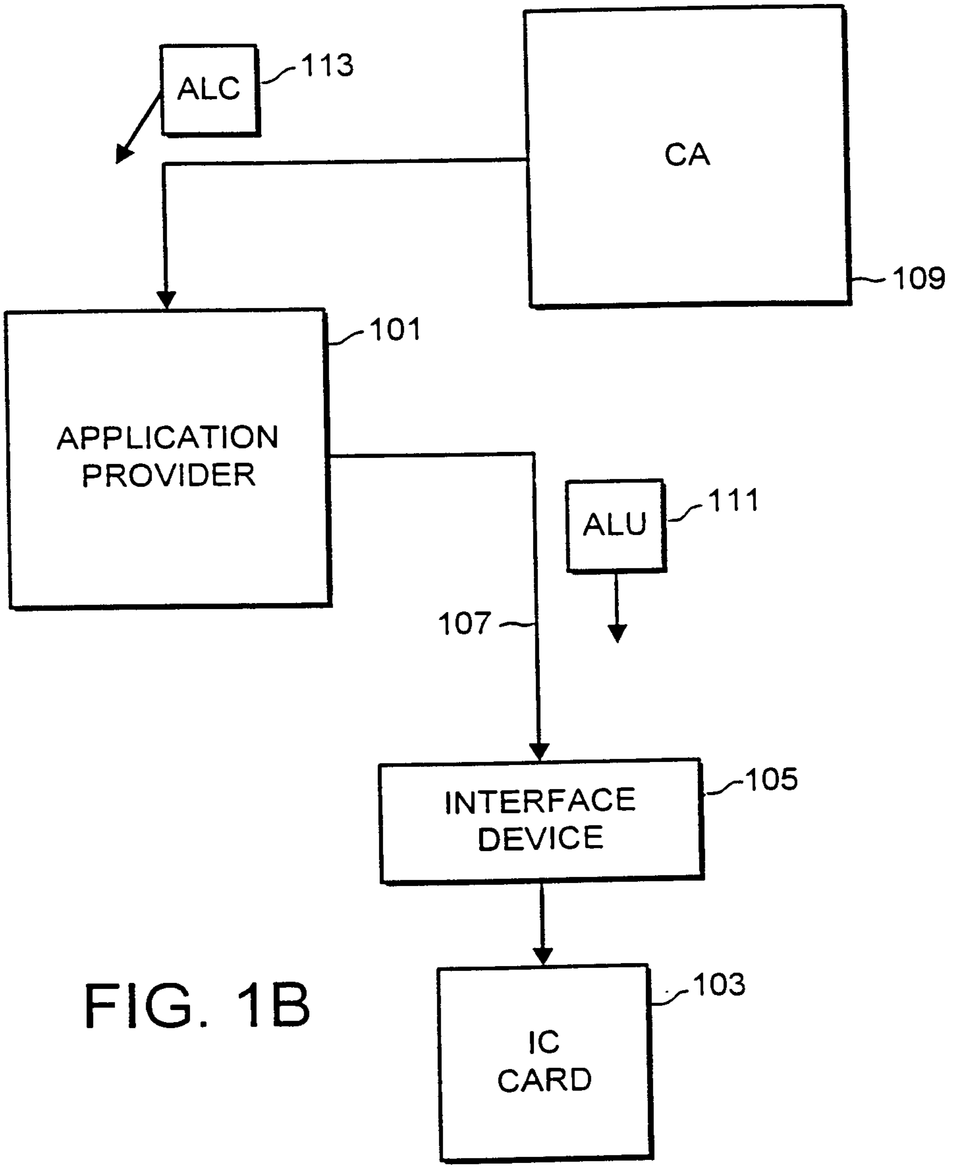


FIG. 1B

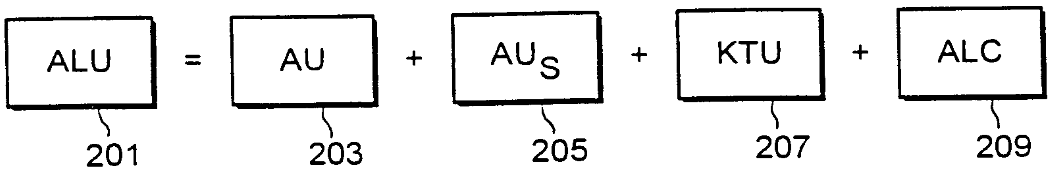


FIG. 2

ANNEX A TO THE DRAWINGS

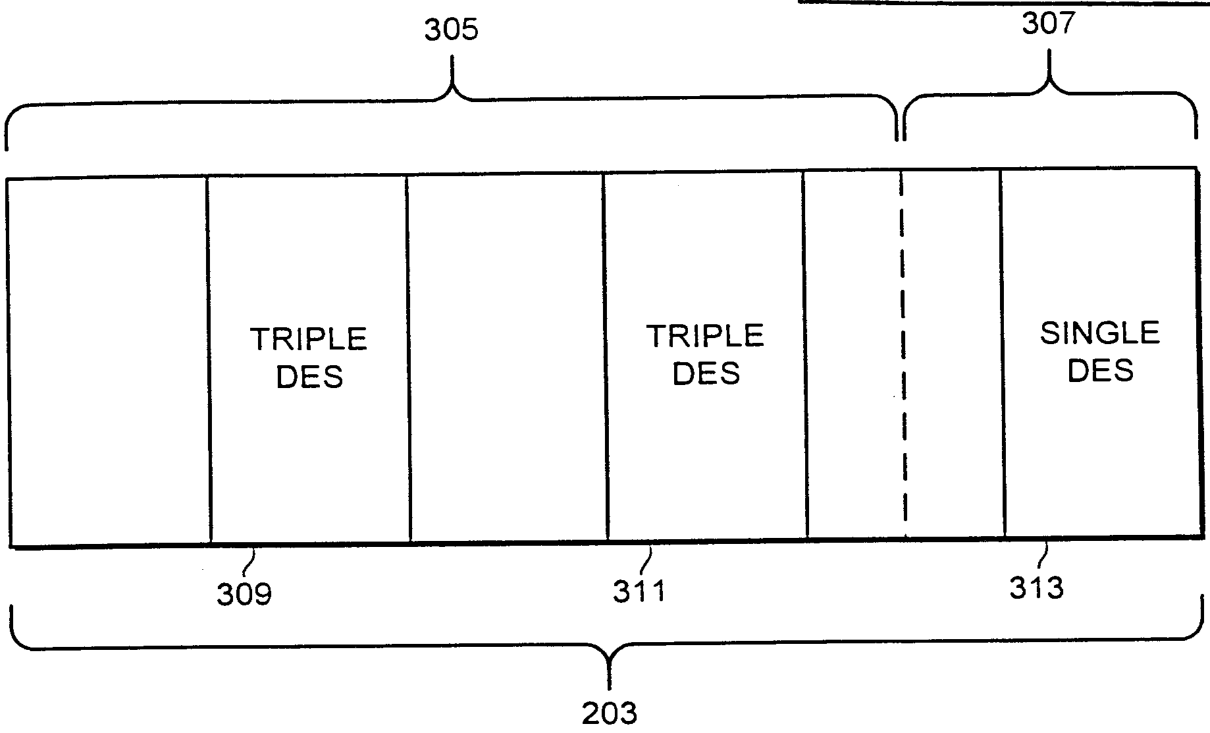


FIG. 3

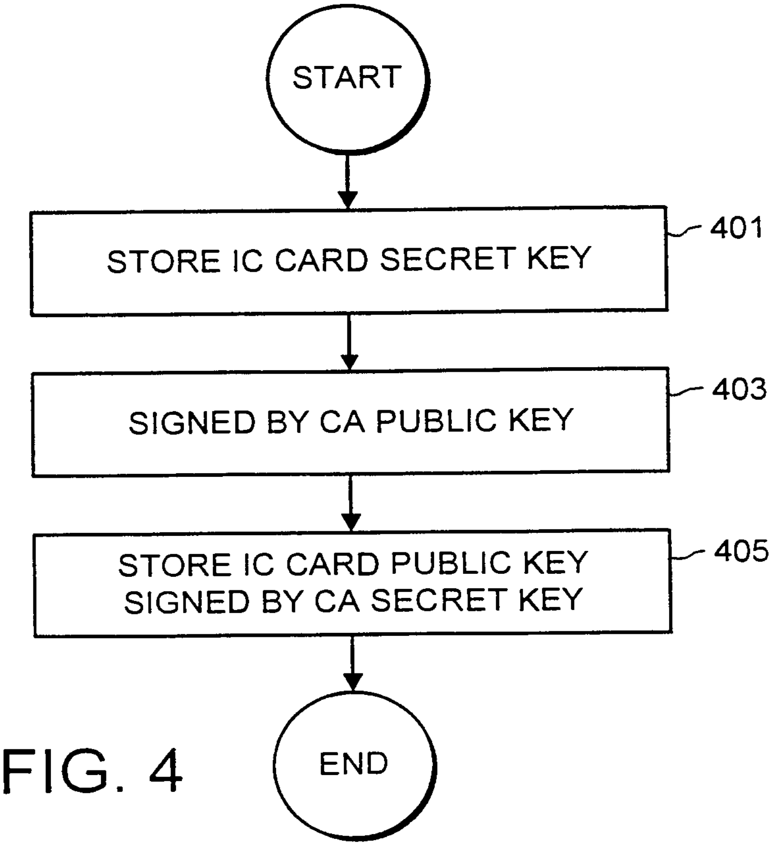


FIG. 4

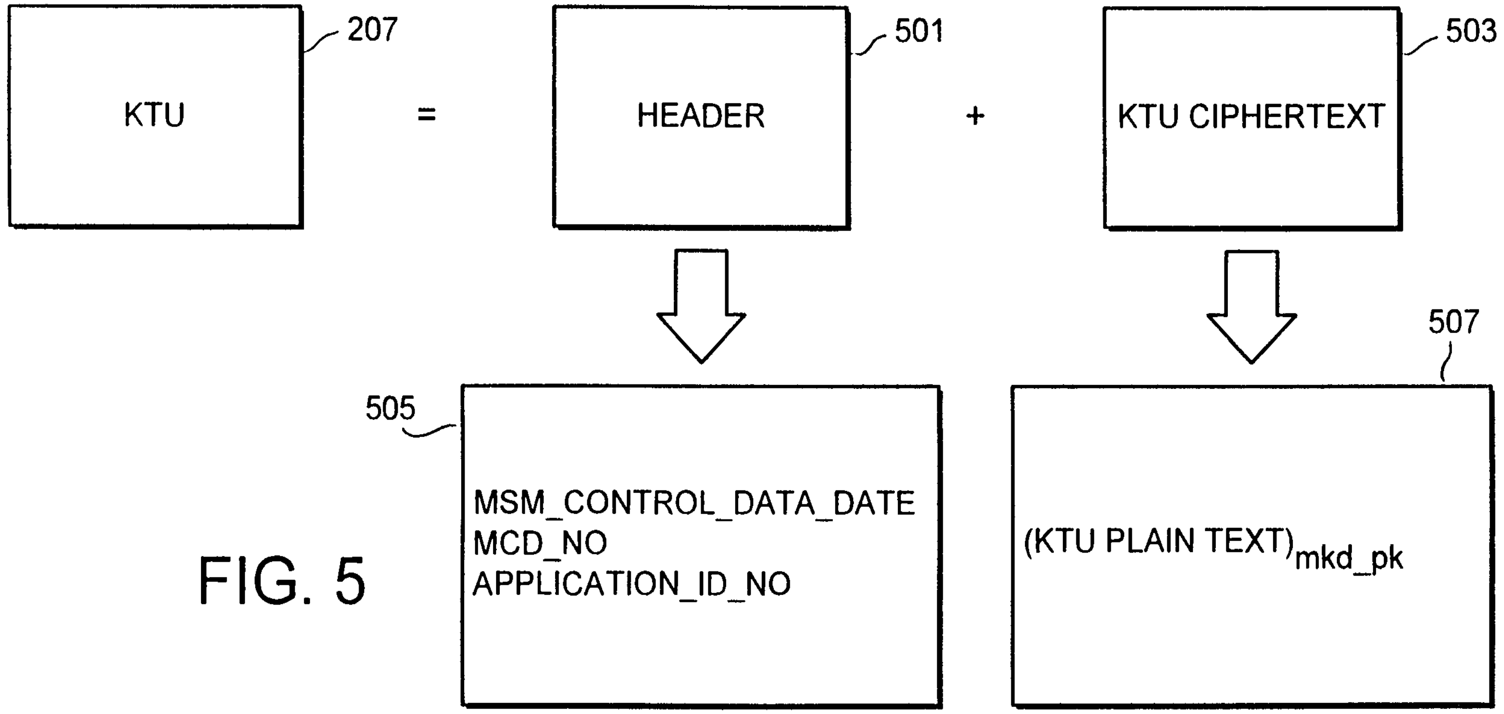

KTU
207
=
HEADER
501
+
KTU CIPHERTEXT
503
505
MSM_CONTROL_DATA_DATE
MCD_NO
APPLICATION_ID_NO
507
(KTU PLAIN TEXT)mkd_pk

FIG. 5

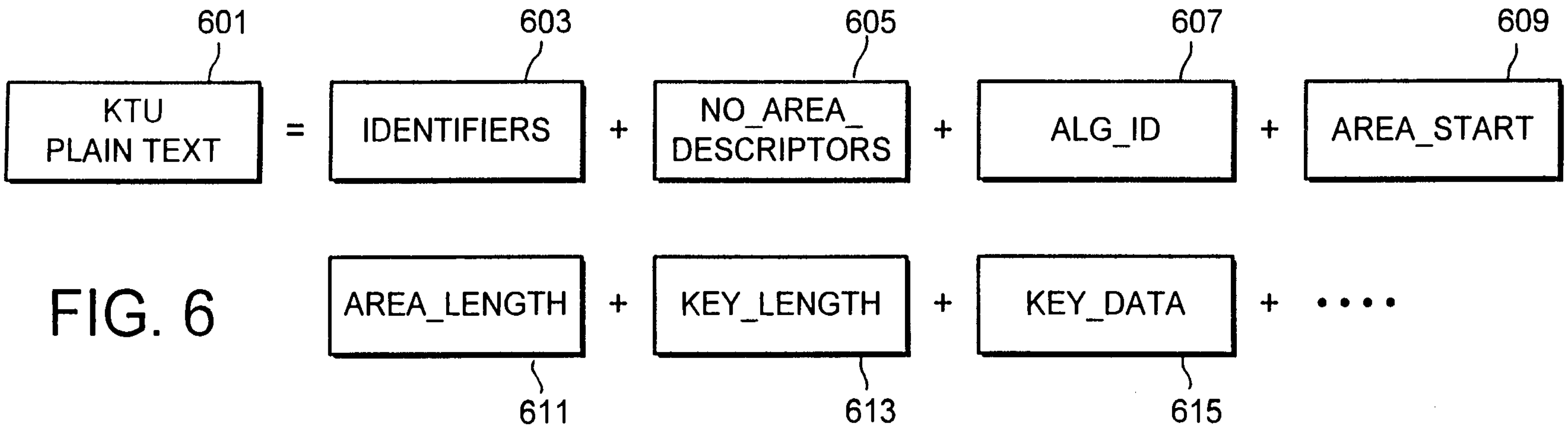

601
KTU PLAIN TEXT
=
603
IDENTIFIERS
+
605
NO_AREA_ DESCRIPTORS
+
607
ALG_ID
+
609
AREA_START
AREA_LENGTH
611
+
KEY_LENGTH
613
+
KEY_DATA
615
+ • • • •

FIG. 6

ANNEX A TO THE DRAWINGS

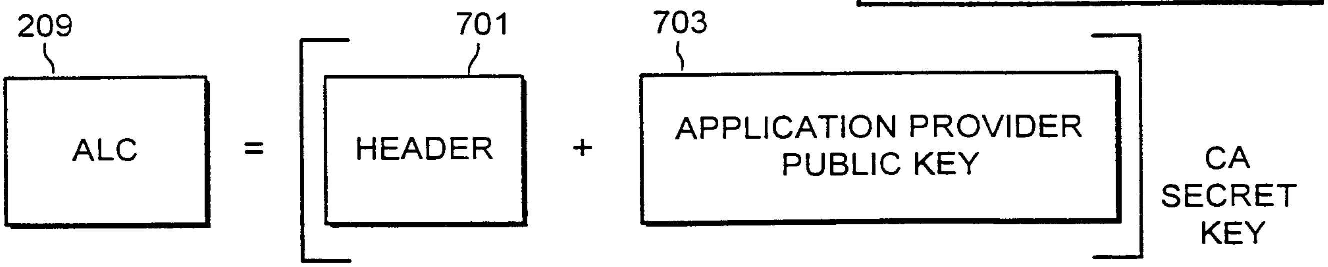


FIG. 7

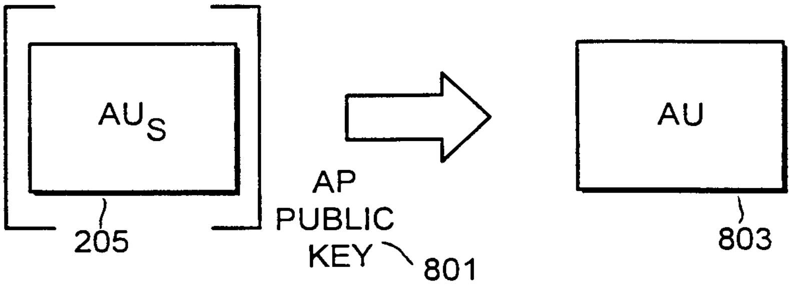


FIG. 8

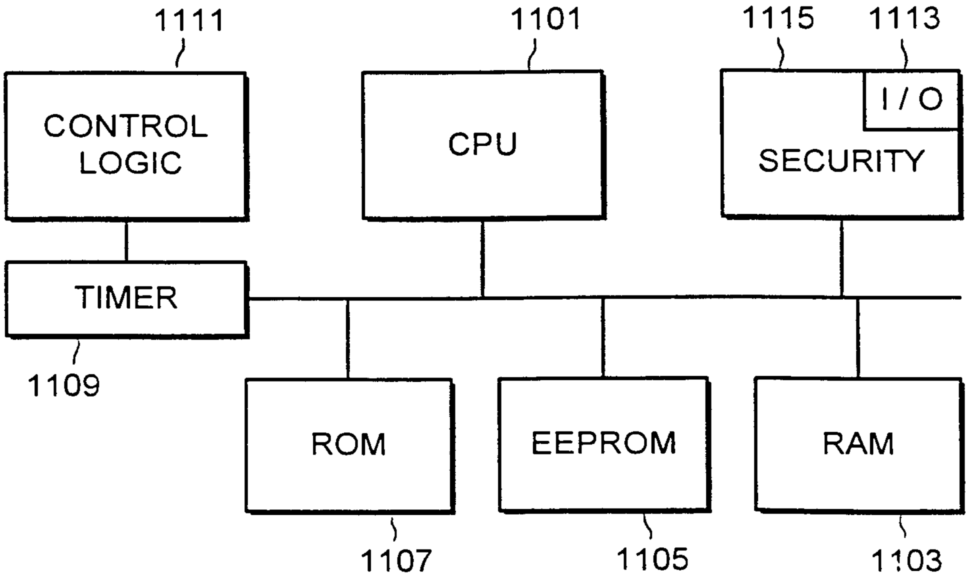


FIG. 11

ANNEX A TO THE DRAWINGS

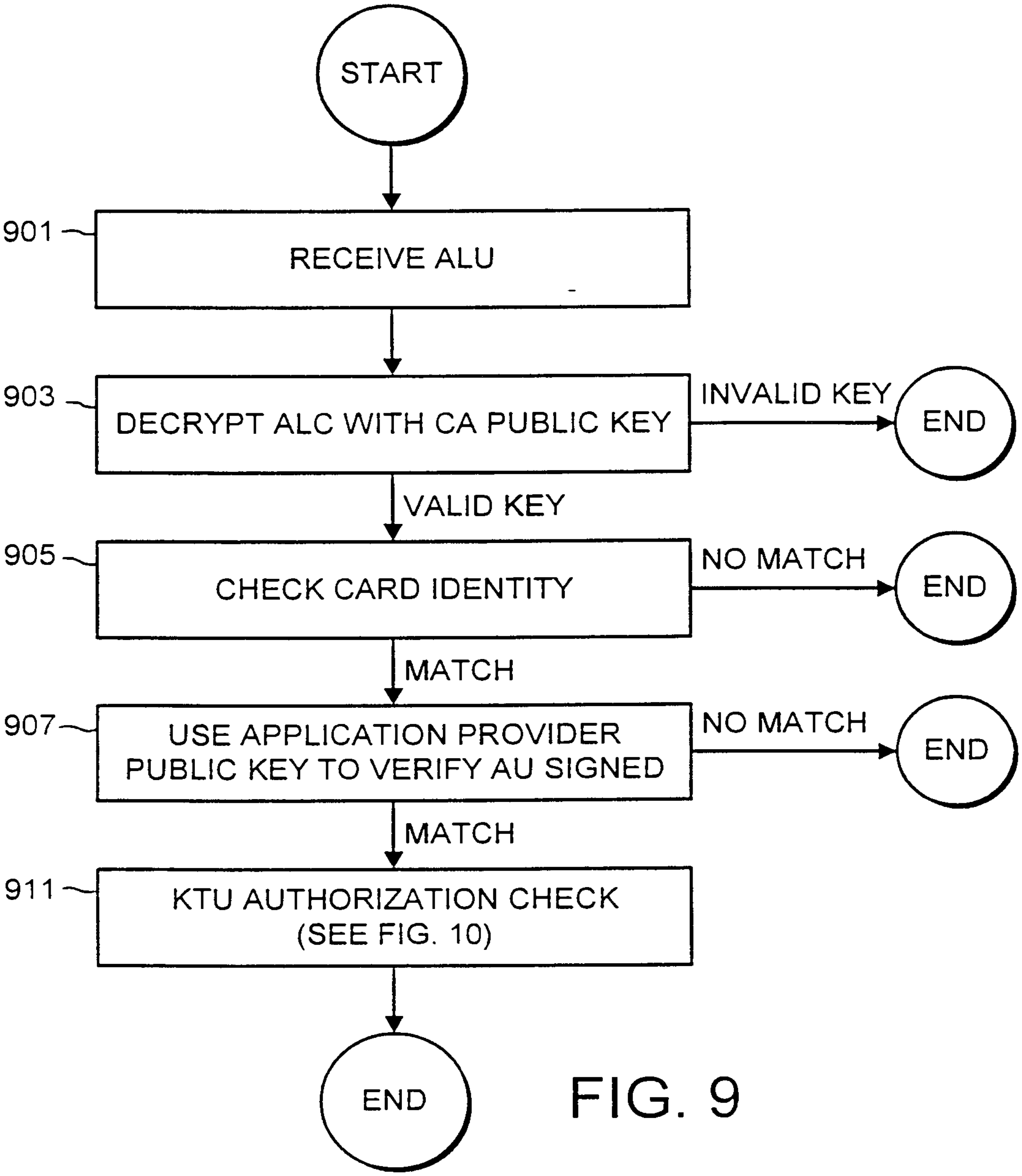

START
901
RECEIVE ALU
903
DECRYPT ALC WITH CA PUBLIC KEY
INVALID KEY
END
VALID KEY
905
CHECK CARD IDENTITY
NO MATCH
END
MATCH
907
USE APPLICATION PROVIDER PUBLIC KEY TO VERIFY AU SIGNED
NO MATCH
END
MATCH
911
KTU AUTHORIZATION CHECK (SEE FIG. 10)
END

FIG. 9

ANNEX A TO THE DRAWINGS

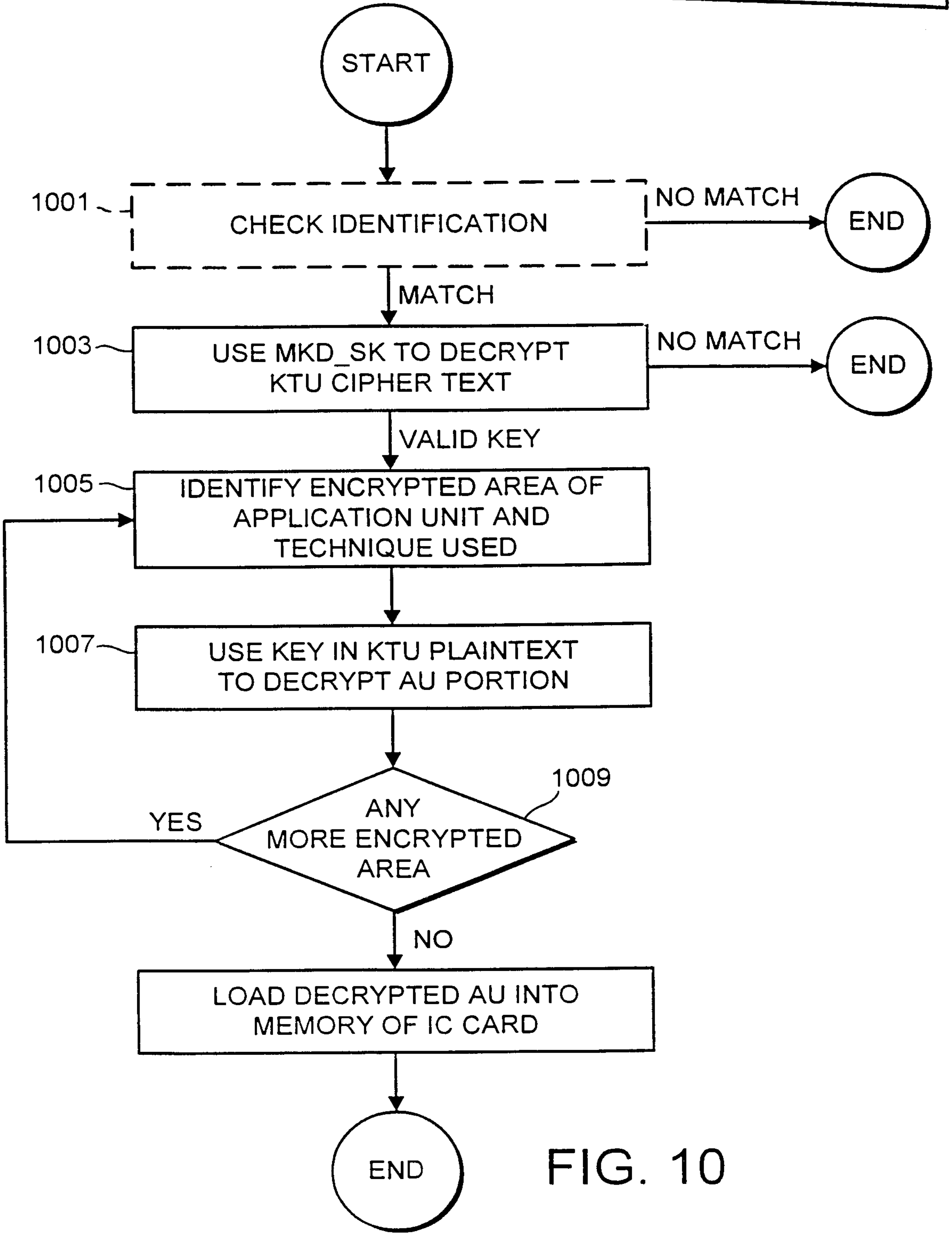

START
1001
CHECK IDENTIFICATION
NO MATCH
END
MATCH
1003
USE MKD_SK TO DECRYPT KTU CIPHER TEXT
NO MATCH
END
VALID KEY
1005
IDENTIFY ENCRYPTED AREA OF APPLICATION UNIT AND TECHNIQUE USED
1007
USE KEY IN KTU PLAINTEXT TO DECRYPT AU PORTION
1009
ANY MORE ENCRYPTED AREA
YES
NO
LOAD DECRYPTED AU INTO MEMORY OF IC CARD
END

FIG. 10

ANNEX B TO THE DRAWINGS

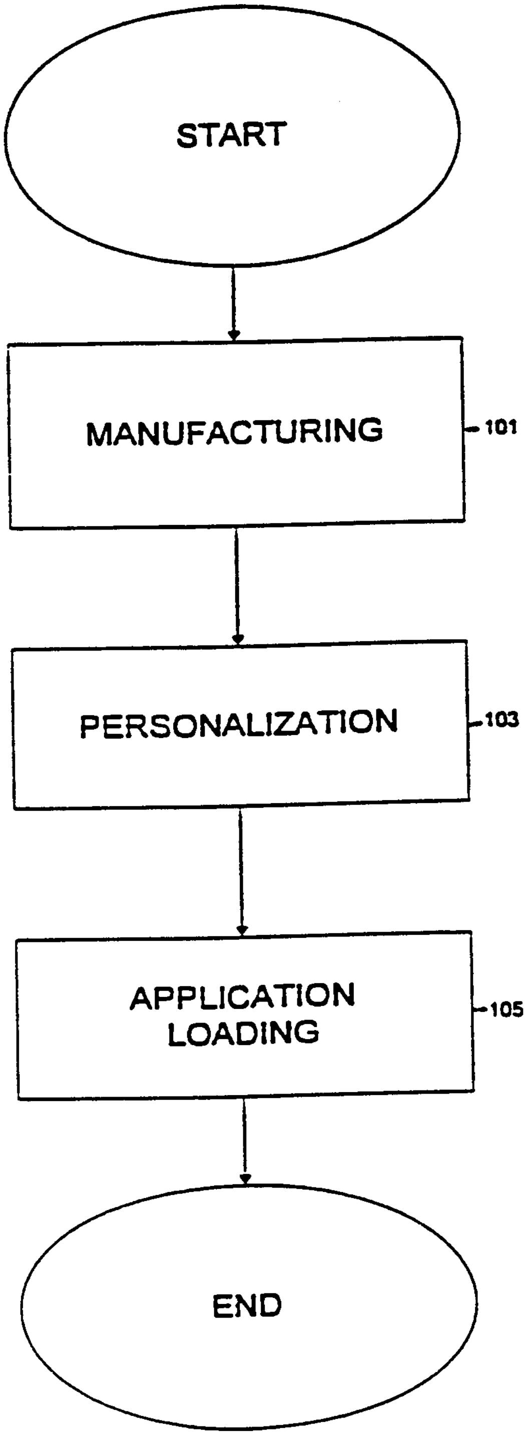


FIG. 1

ANNEX B TO THE DRAWINGS

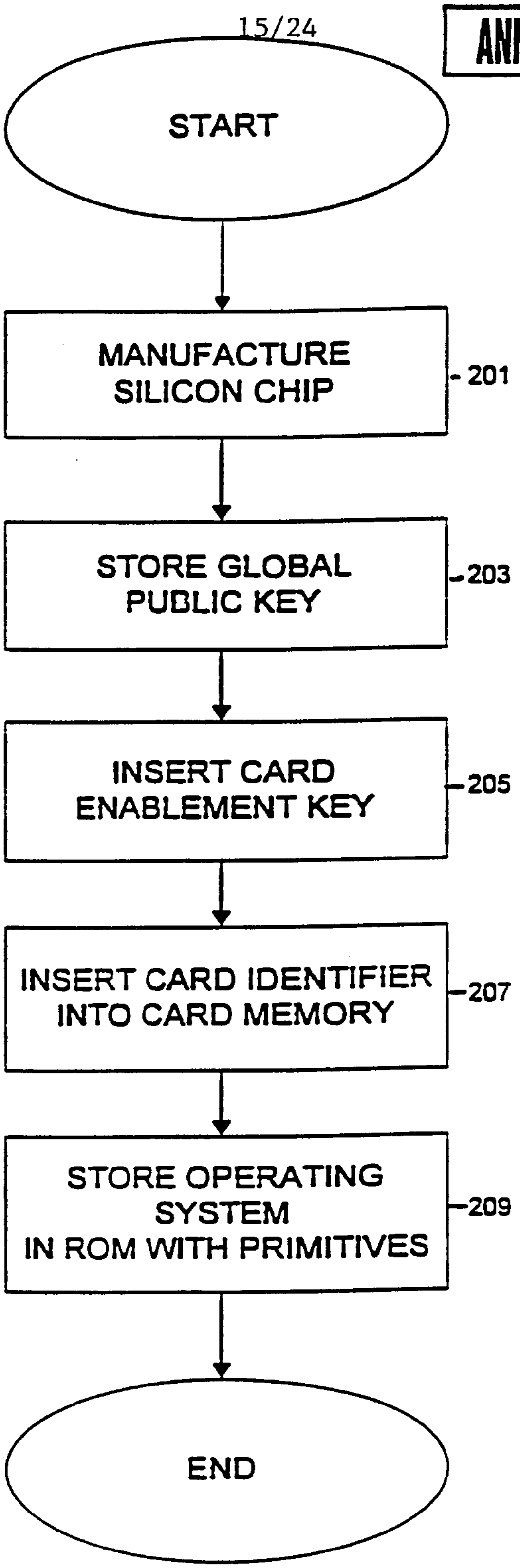

START
MANUFACTURE SILICON CHIP
201
STORE GLOBAL PUBLIC KEY
203
INSERT CARD ENABLEMENT KEY
205
INSERT CARD IDENTIFIER INTO CARD MEMORY
207
STORE OPERATING SYSTEM IN ROM WITH PRIMITIVES
209
END

FIG. 2

ANNEX B TO THE DRAWINGS

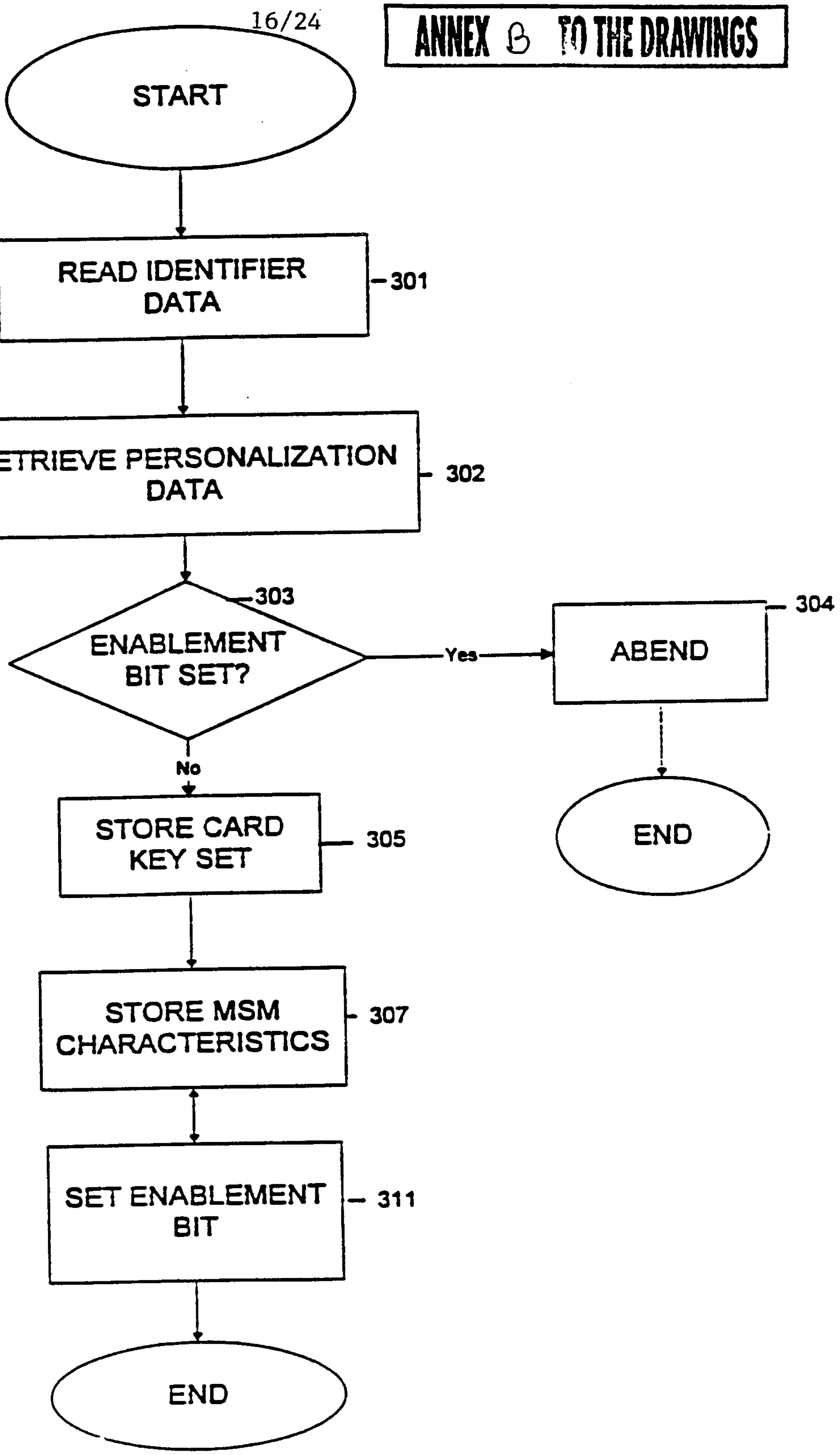


FIG. 3

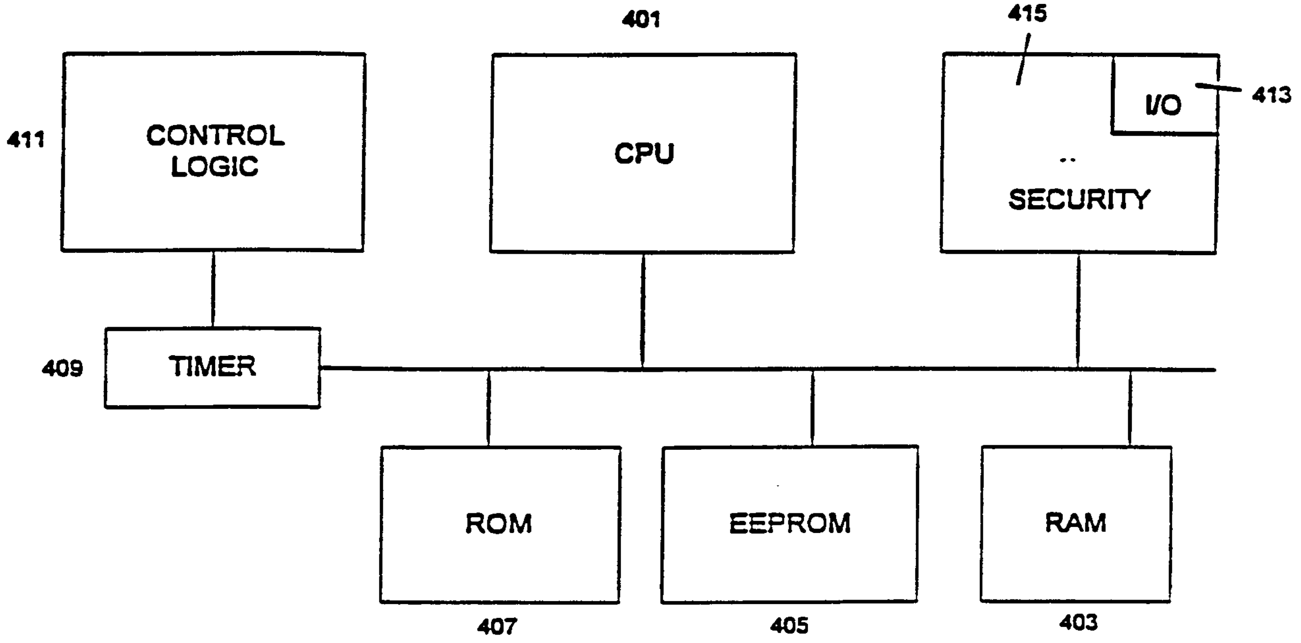

401
415
411
CONTROL LOGIC
CPU
I/O
413
SECURITY
409
TIMER
ROM
EEPROM
RAM
407
405
403

FIG. 4

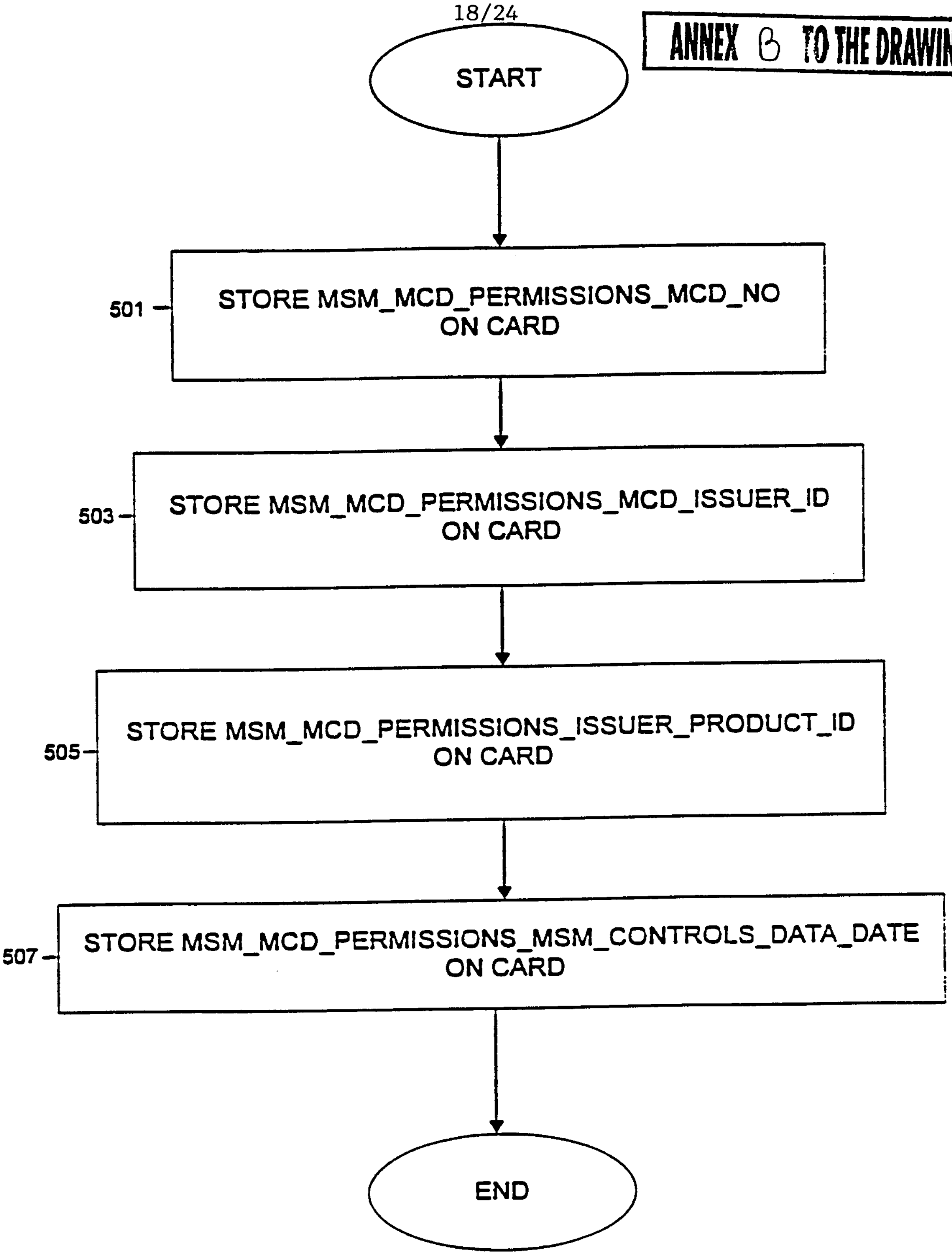

START
501
STORE MSM_MCD_PERMISSIONS_MCD_NO ON CARD
503
STORE MSM_MCD_PERMISSIONS_MCD_ISSUER_ID ON CARD
505
STORE MSM_MCD_PERMISSIONS_ISSUER_PRODUCT_ID ON CARD
507
STORE MSM_MCD_PERMISSIONS_MSM_CONTROLS_DATA_DATE ON CARD
END

FIG.. 5

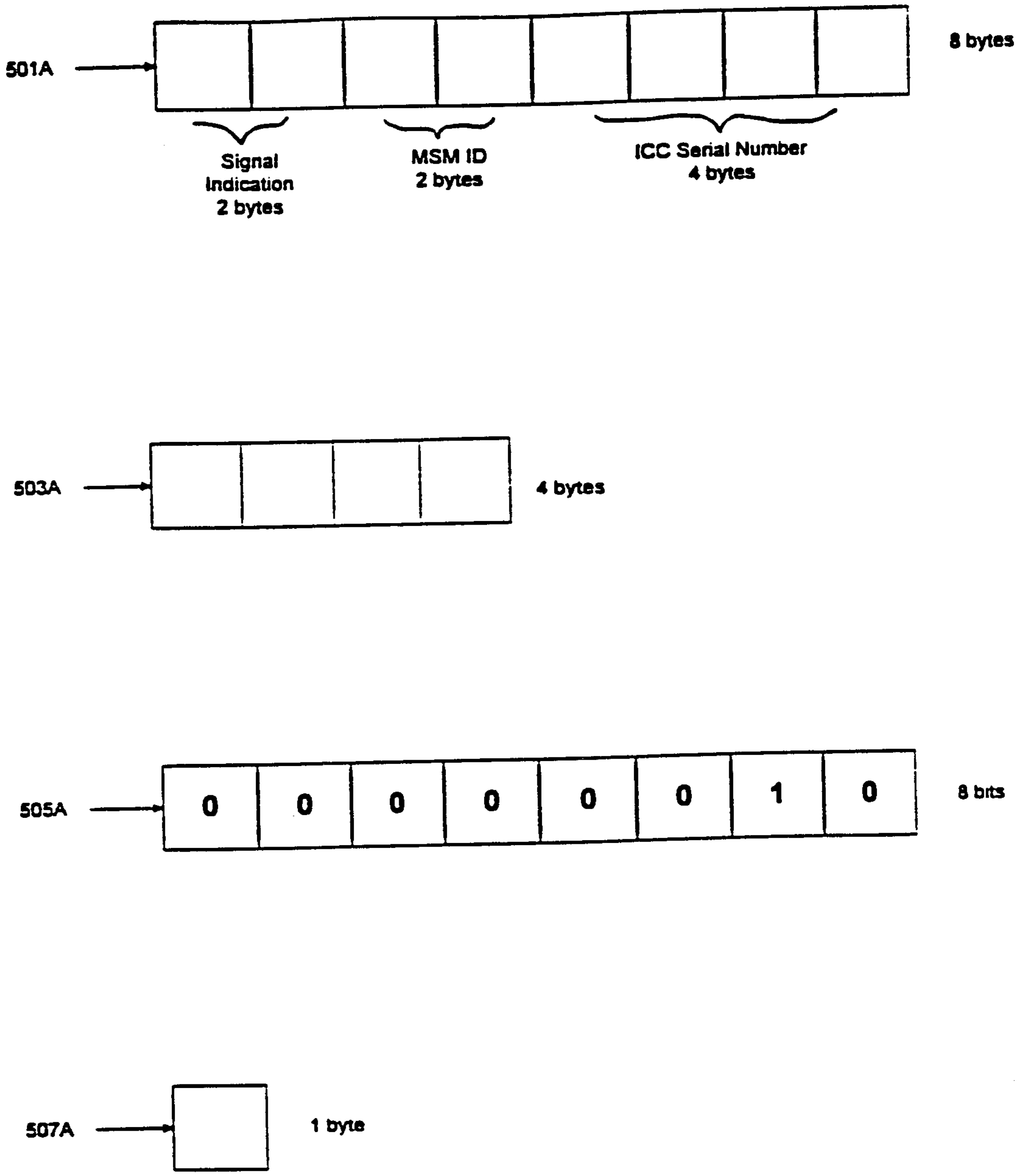

501A
8 bytes
Signal Indication 2 bytes
MSM ID 2 bytes
ICC Serial Number 4 bytes
503A
4 bytes
505A
0
0
0
0
0
0
1
0
8 bits
507A
1 byte

FIG. 5A

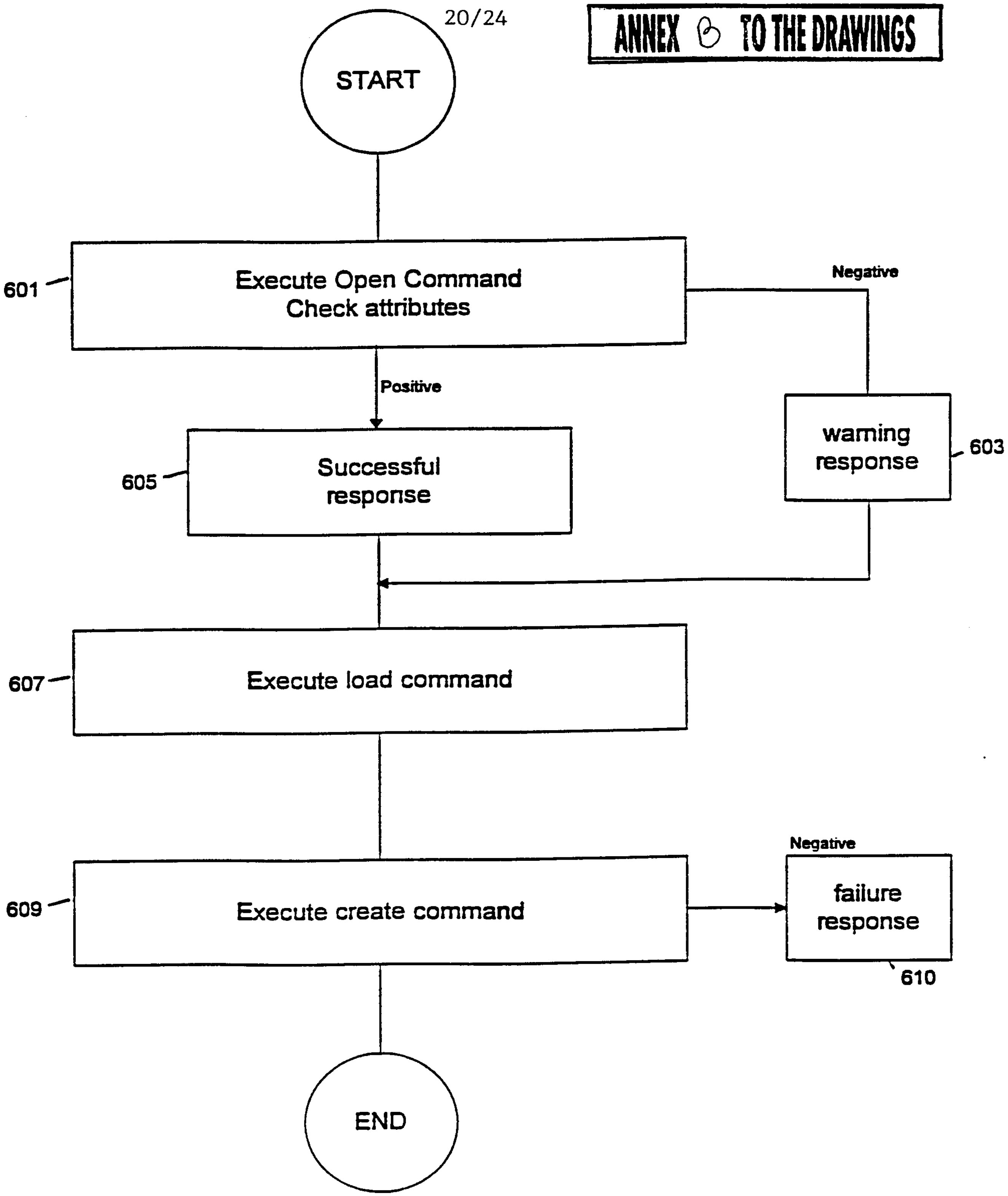

START
601
Execute Open Command
Check attributes
Negative
Positive
605
Successful
response
warning
response
603
607
Execute load command
609
Execute create command
Negative
failure
response
610
END

FIG. 6

ANNEX B TO THE DRAWINGS

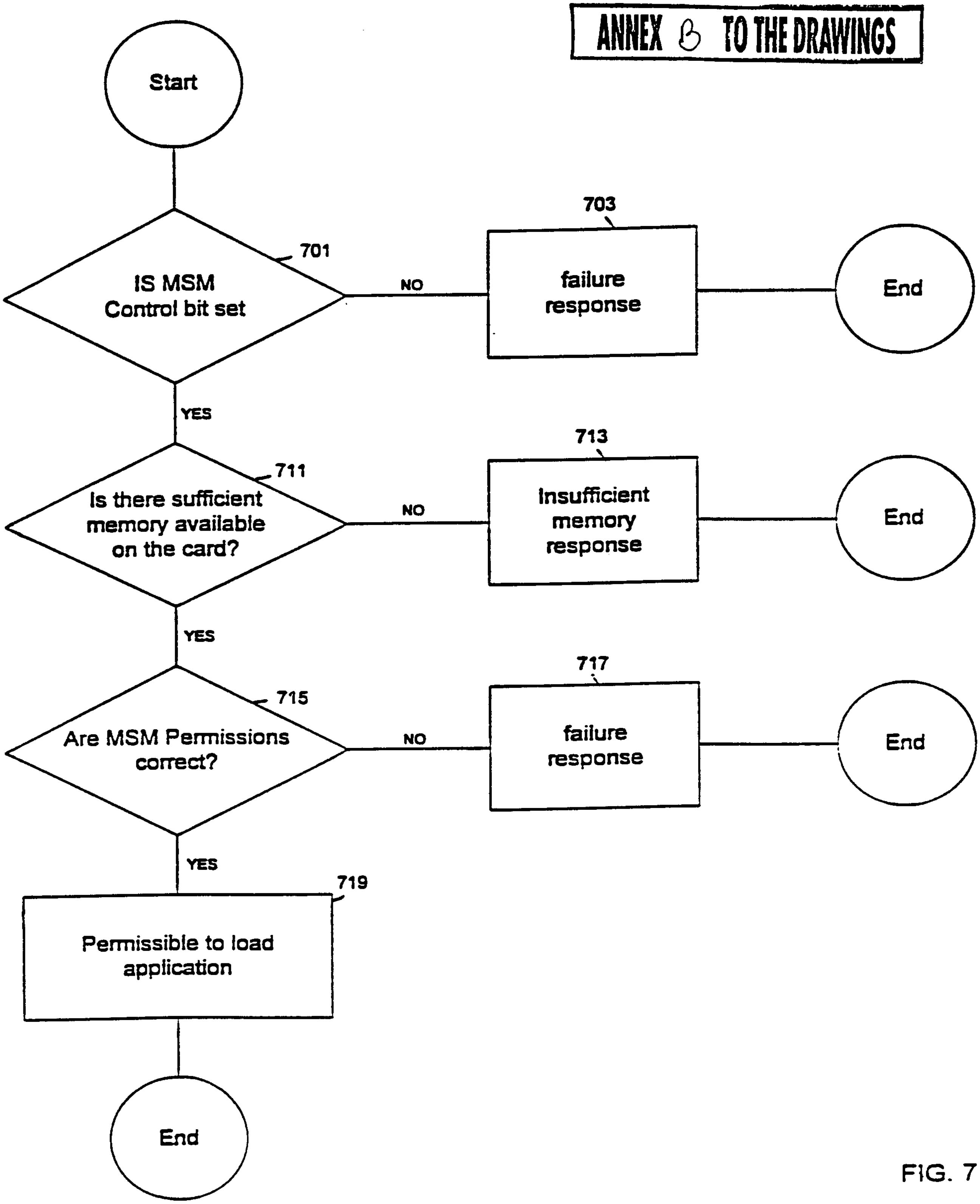

Start
IS MSM Control bit set
701
NO
703
failure response
End
YES
Is there sufficient memory available on the card?
711
NO
713
Insufficient memory response
End
YES
Are MSM Permissions correct?
715
NO
717
failure response
End
YES
719
Permissible to load application
End

FIG. 7

ANNEX 6 TO THE DRAWINGS

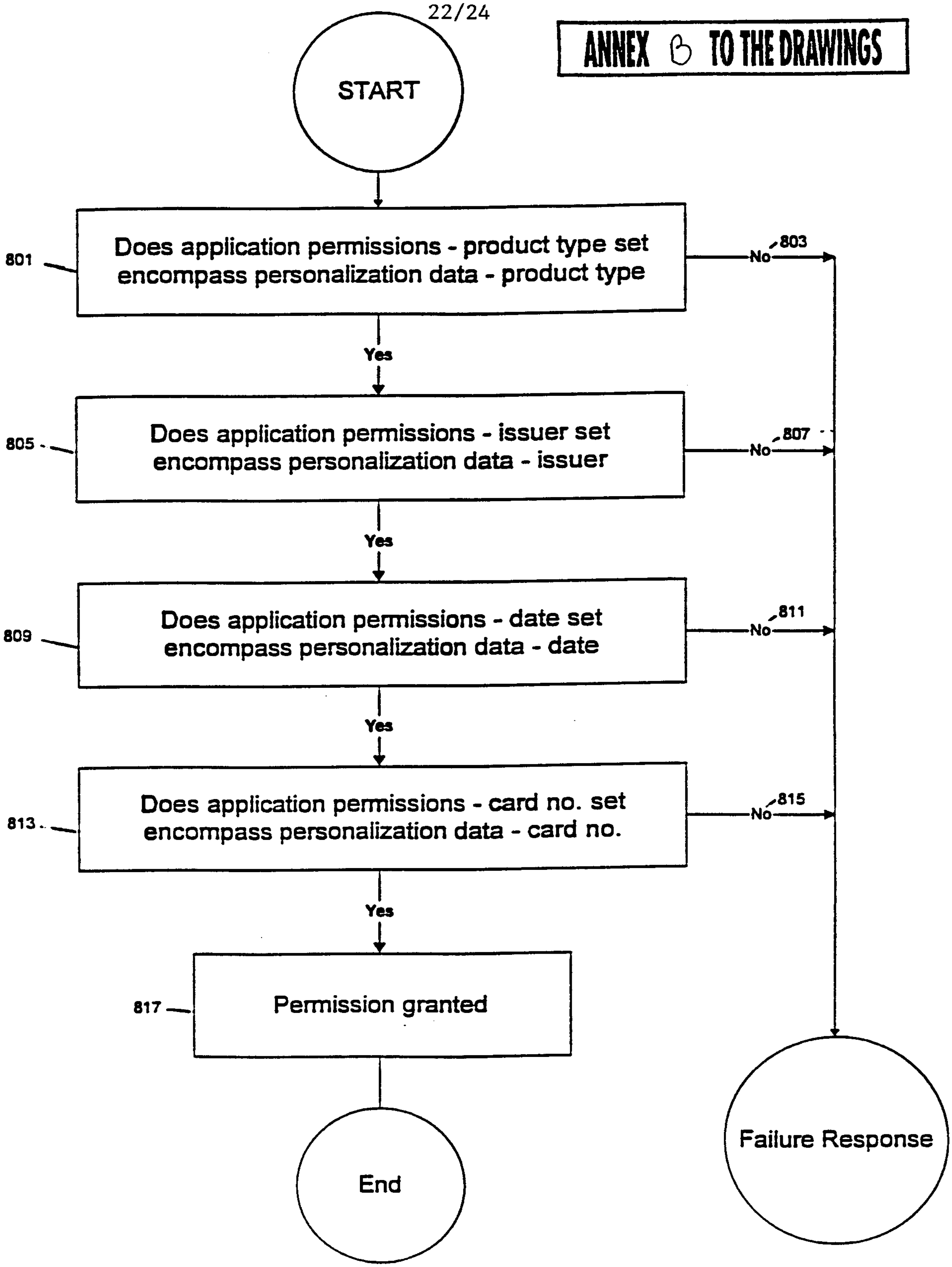

START
801
Does application permissions - product type set encompass personalization data - product type
No
803
Yes
805
Does application permissions - issuer set encompass personalization data - issuer
No
807
Yes
809
Does application permissions - date set encompass personalization data - date
No
811
Yes
813
Does application permissions - card no. set encompass personalization data - card no.
No
815
Yes
817
Permission granted
End
Failure Response

FIG. 8

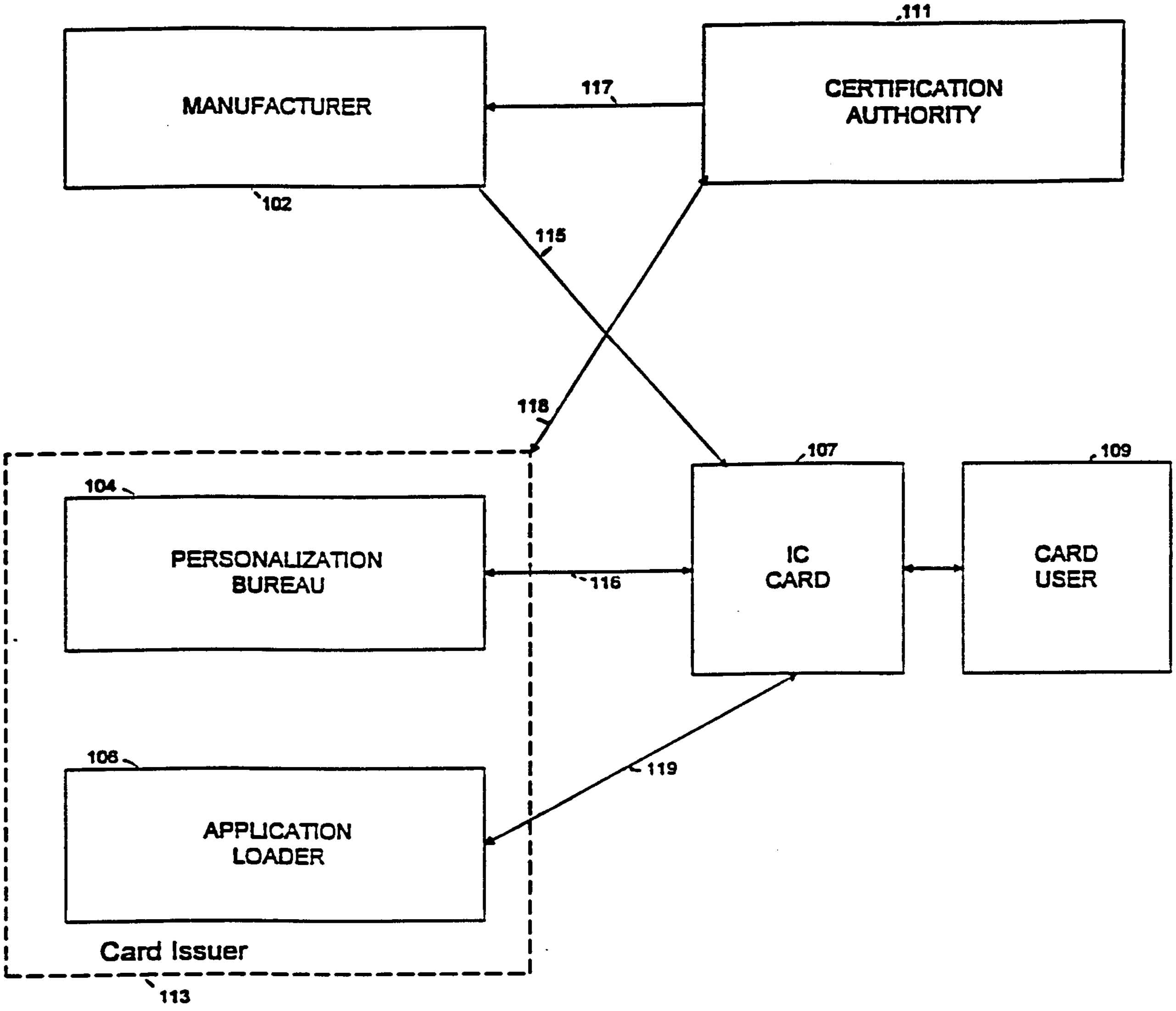

MANUFACTURER
102
CERTIFICATION AUTHORITY
111
117
115
118
104
PERSONALIZATION BUREAU
116
IC CARD
107
CARD USER
109
106
APPLICATION LOADER
119
Card Issuer
113

FIG. 9

ANNEX 6 TO THE DRAWINGS

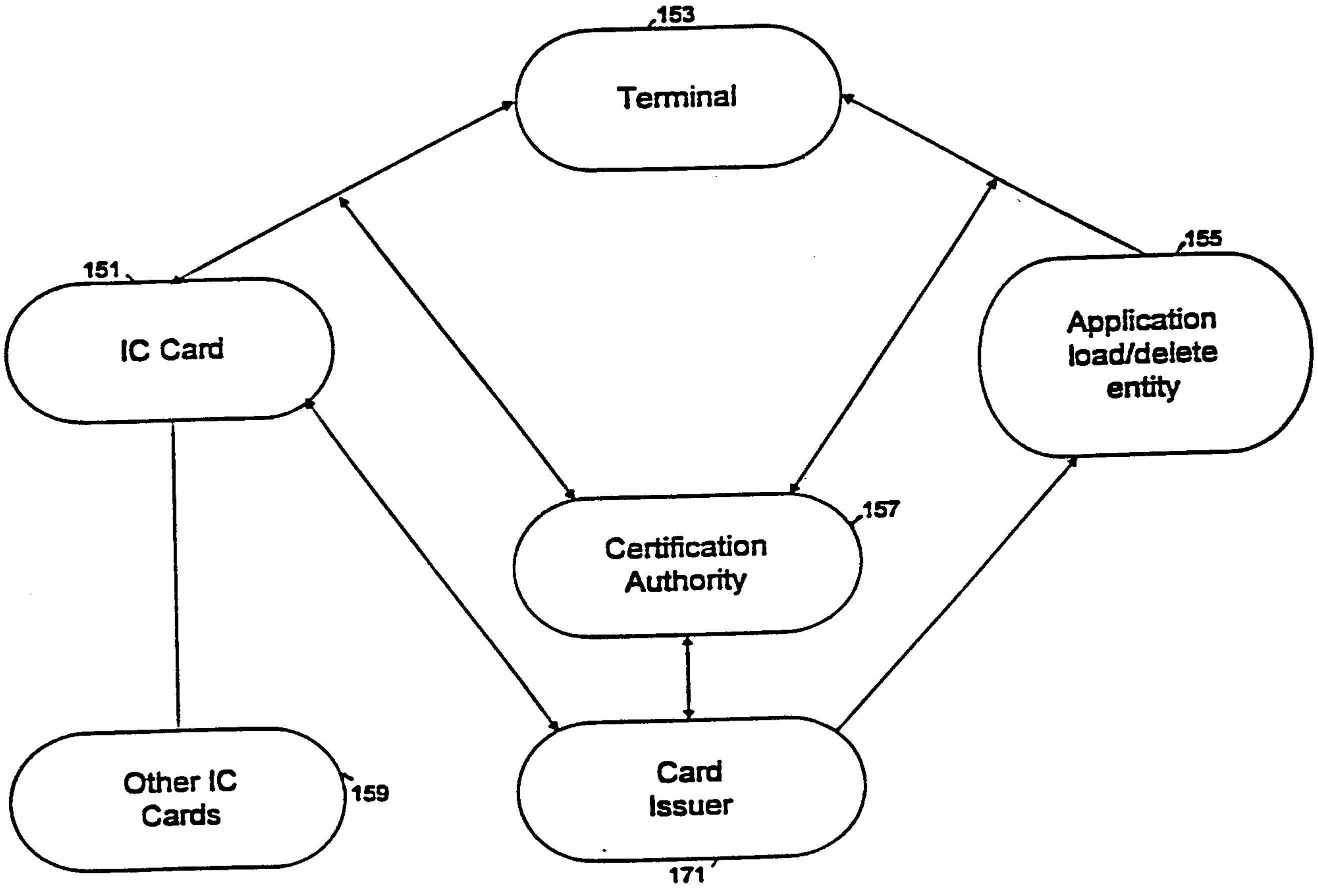


FIG. 10

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : **G07F 7/10**

A2

(11) International Publication Number: **WO 98/52162**

(43) International Publication Date: 19 November 1998 (19.11.98)

(21) International Application Number: PCT/GB98/01402

(22) International Filing Date: 14 May 1998 (14.05.98)

(30) Priority Data:

60/046,514	15 May 1997 (15.05.97)	US
60/036,543	15 May 1997 (15.05.97)	US
09/023,057	12 February 1998 (12.02.98)	US
09/076,551	12 May 1998 (12.05.98)	US

(71) Applicant: MONDEX INTERNATIONAL LIMITED [GB/GB]; 47-53 Cannon Street, London EC4M 5SQ (GB).

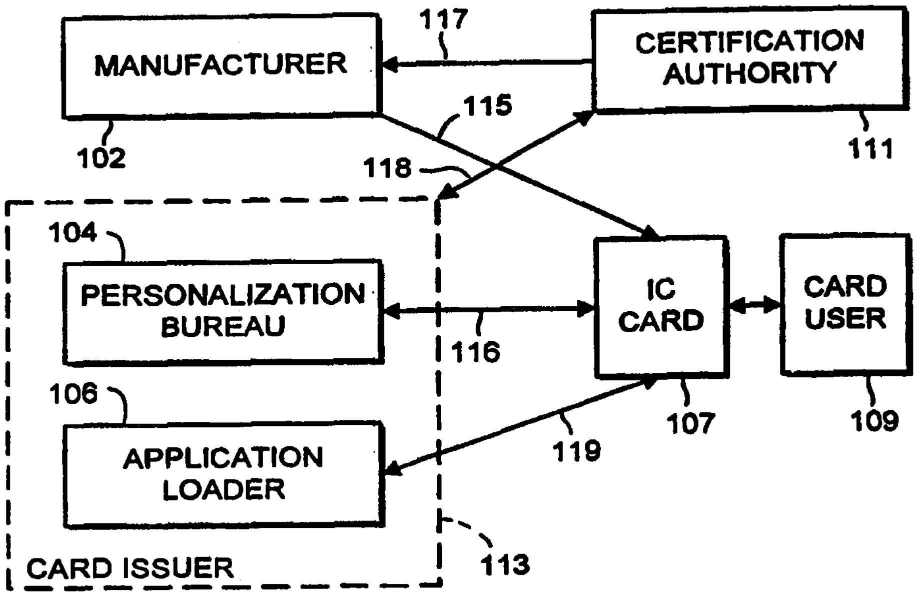
(72) Inventors: EVERETT, David, Barrington; 31 Ashdown Avenue, Saltdean, Brighton, East Sussex BN2 8AH (GB). MILLER, Stuart, James; 9 Woodford Green, The Warren, Bracknell, Berks RG12 9YQ (GB). PEACHAM, Anthony, David; 4 Lynwood, Groombridge, Tunbridge Wells, Kent TN3 9LX (GB). SIMMONS, Ian, Stephens; The Elms, School Road, Broughton, Cambs PE17 3AT (GB). RICHARDS, Timothy, Philip; 32 Craig Mount, Radlett, Herts WD7 7LW (GB). VINER, John, Charles; Hydes, Woodlands Lane, Windlesham (GB).

(74) Agent: POTTER, Julian, Mark; D. Young & Co., 21 New Fetter Lane, London EC4A 1DA (GB).

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

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

(54) Title: SECURE MULTIPLE APPLICATION CARD SYSTEM AND PROCESS



(57) Abstract

A secure multiple application card system and process are provided having secure loading and deleting capability by use of a Certification Authority and Personalization Bureau. The certification authority maintains the security of the system by requiring IC cards to be injected with its public key and a card identifier for uniquely identifying each card, by providing a personalization data block for each card, and by signing with its private key all applications to be loaded or deleted from the IC card.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

SECURE MULTIPLE APPLICATION CARD SYSTEM AND PROCESS

BACKGROUND OF INVENTION

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. This allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application at its personalization stage. That application, however, cannot be modified or changed after the card is issued even if the modification is desired by the card user or card issuer. Moreover, if a card user wanted a variety of application functions to be performed by IC cards issued to him or her, such as both an electronic purse and a credit/debit function, the card user would be required to carry multiple physical cards on his or her person, which would be quite cumbersome and inconvenient. If an application developer or card user desired two different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal, making the transaction difficult, lengthy and inconvenient.

Therefore, it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card. Although multiple applications could be pre-selected and placed in the memory of the card during its production stage, it would also be beneficial to have the ability to load and delete applications for the card post-production as needed.

The increased flexibility and power of storing multiple applications on a single card create new technical challenges to be overcome concerning the integrity and security of the information (including application code and associated data) exchanged between the individual card and the application provider as well as within the entire system when loading and deleting applications. It would be beneficial to have the capability in the IC card system to exchange data among cards, card issuers, system operators and application providers securely and to load and delete applications securely at any time from either a terminal or remotely over a telephone line, internet or intranet connection or other data conduit. Because these data transmission lines are not typically secure lines, a number of security and entity-authentication techniques must be implemented to make sure that applications being sent over the transmission lines are only loaded on the intended cards.

As mentioned, it is important -- particularly where there is a continuing wide availability of new applications to the cardholder -- that the system has the capability of adding applications onto the IC card subsequent to issuance. This is necessary to protect the longevity of the IC cards; otherwise, once an application becomes outdated, the card would be useless. In this regard, to protect against the improper or undesired loading of applications onto IC cards, it would be beneficial for the IC card system to have the capability of controlling the loading process and restricting, when necessary or desirable, the use of certain applications to a limited group or number of cards such that the applications are "selectively available" to the IC-cards in the system. This "selective capability" would allow the loading and deleting of applications at, for example, a desired point in time in the card's life cycle. It would also allow the loading of an application only to those cards chosen to receive the selected application.

Accordingly, it is an object of embodiments of this invention to provide these important features and specifically an IC-card system having improved security that allows for selective availability of smart card applications which may be loaded onto IC cards.

SUMMARY OF THE INVENTION

These and other objectives are achieved by embodiments in accordance with the present invention which provide an IC card system comprising at least one integrated circuit card and having a certification authority and a personalization bureau. The certification authority ("CA") maintains encryption and decryption keys for the entire system and provides the card manufacturer with security data to be placed on the card at manufacture. Thus, there is advantageously provided a secure multiple application card system.

Specifically, in a preferred embodiment, an IC card is injected at manufacture with the public key of the CA and a card identifier for uniquely identifying each of the cards. Subsequent to manufacturer, the cards are preferably provided to a personalization bureau ("PB") which could be a card issuer, for enabling the cards. The PB obtains from the cards the identifiers and forwards a list of card identifiers to the CA.

The CA in turn creates a personalization data block for each card identifier, and each data block preferably includes card personalization data and an individual key set. The data block is encrypted and forwarded back to the PB. By using the card identifier, the PB then matches the cards with the encrypted data blocks and separately loads each data block onto the matched card, and preferably sets an enablement bit indicating that the card has been enabled and is ready for application loading.

The application loading process is preferably performed at the PB. At first, the system checks to see whether the card to be loaded is qualified (as defined below) to accept the loading of a specific application. The application loader via a terminal will be advised if the card is qualified and, if so, a check will be done using the CA's public key to determine whether the application to be loaded has been signed by the CA's secret key indicating that the application to be loaded has been allowed by the CA.

BRIEF DESCRIPTION OF THE DRAWINGS

Further objects, features and advantages of embodiments in accordance with the invention will become apparent from the following detailed description taken by way of example only and in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

Fig. 1 is block diagram illustrating the three stages in the life of a multi-application IC card in a secure system;

Fig. 2 is a block diagram illustrating the steps of the card manufacture process;

Fig. 3 is a flow diagram illustrating the steps involved in enabling each of the IC cards in the secure system;

Fig. 4 is a block diagram of an IC card chip which can be used in accordance with the invention;

Fig. 5 is a block diagram illustrating the data stored on the IC card as indicated in block 307 of Fig. 3;

Fig. 5A is a schematic of the data structures residing in an IC card and representing personalization data;

Fig. 6 is a flowchart illustrating the steps of loading an application onto an IC card in the secure system;

Fig. 7 is a flow chart illustrating the checking steps as indicated in block 601 of Fig. 6;

Fig. 8 is a flowchart illustrating the steps undertaken in determining if loading of an application may proceed;

Fig. 9 is a block diagram showing the components of the system architecture for the enablement process of an IC card in a secure multi-application IC card system; and

Fig. 10 is a system diagram of entities involved with the use of the IC card once it has been personalized.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

DETAILED DESCRIPTION OF THE INVENTION

An embodiment in accordance with the present invention provides an IC card system and process which allow the flexibility to load and delete selected applications over the lifetime of a multi-application IC card in response to the needs or desires of the card user, card issuers and/or application developers. A card user who has such a card can selectively load and delete applications as desired if allowed by the card issuer in conjunction with the system operator or Certification Authority ("CA") which controls the loading and deleting process by certifying the transfer of information relating to the process.

By allowing applications to be selectively loaded and deleted from the card, a card issuer can extend additional functionality to an individual IC card without having to issue new cards. Moreover, application developers can replace old applications with new enhanced versions, and applications residing on the same card using a common multiple application operating system may interact and exchange data in a safe and secure manner. For example, a frequent flyer loyalty program may automatically credit one frequent flyer mile to a card user's internal account for every dollar spent with the Mondex purse or with a credit/debit application. By allowing the ability to selectively load and delete applications, the card user, subject to the requirements of the card issuer, also has the option of changing loyalty programs as desired.

A card issuer or application developer may intend that a particular application be loaded on only one card for a particular card user in a card system. A regional bank may desire to have a proprietary application reside only on the cards which the bank issues. Embodiments of the present invention would allow for this selective loading and specifically allow for the prevention of loading proprietary applications onto unauthorized cards issued by others.

To achieve these desired objectives, an embodiment in accordance with the present invention gives each card a specific identity by storing "card personalization data" on the card. Moreover, each application to be loaded or deleted on one or more cards in the system is assigned "application permissions data" which specify the cards upon which the applications may be loaded.

The type of personalized data can vary depending upon the needs and requirements of the card system. In the preferred embodiment, described in greater detail below, the personalization data include unique card identification designation data, the card issuer, the product class or type (which is defined by the card issuer) and the date of personalization. However, not all of these data elements are required to be used and additional elements could also be included.

The application permissions data associated with an application, also described in greater detail below, can be a single value in an identity field or could include multiple values in the identity field. For example, the application permissions data in the card issuer field could represent both product class A and product class B from a certain Bank X, indicating that the application could be loaded onto cards designated as product classes A and B issued by Bank X (as indicated in the card product ID field of the card's personalization data).

In addition, a "global value" could be stored in the issuer field (or other field) of the application permissions data indicating that all IC cards in the system regardless of who issued the card would match this permissions field. In this case, for example, a data value of zero stored in the application permissions card-issuer field will match all of the cards' personalization card-issuer fields.

Figure 1 shows the three steps involved in providing an operational multi-application IC card in a secure system. The first step is the card manufacturing step 101. The second step is the personalization step 103 where card personalization data (also called entity authentication data) is loaded onto the card. The third step is the application loading step 105 which checks to see if a card is qualified to receive an application, i.e., when the personalization data is checked against the application permissions data associated with the application to be loaded. Each of these three steps is described in detail below.

Card Manufacture

Figure 2 shows the steps necessary in manufacturing an IC card in a secure system. Step 201 manufactures the physical IC card by creating the integrated circuit on silicon and placing it on the card. The integrated circuit chip will include RAM, ROM and EEPROM memories. When the card is first manufactured, a global public key of the system operator (in this case called the Certification Authority (CA)) is stored on each card in ROM in step 203. This will allow the card to authenticate that the source of any message to it is from the CA since the public key on the card will be matched to the CA's secret key.

More specifically, this public key stored on the card will allow the individual card to verify data signed with the CA's private key. The public key of the CA, which is stored on the card, is used only for determining if the data sent to the card was signed with the proper CA private key. This allows the card to verify the source of any message coming from the CA.

Step 205 inserts a card enablement key in a secure portion of EEPROM in the card to facilitate card specific confidentiality during enablement, and step 207 inserts a card identifier in EEPROM of the card. The identifier, which can be accessed by any terminal, will allow the system to determine the identity of the card in later processes. The identifier is freely available and will not be used to authenticate messages.

Step 209 stores the operating system code in ROM on the card including any primitives which are called or supported by the operating system. The primitives are written in native language code (e.g., assembly language) and are stored in ROM. The primitives are subroutines which may be called by the operating system or by applications residing on the card such as mathematic functions (multiply or divide), data retrieval, data manipulation or cryptographic algorithms. The primitives can be executed very quickly because they are written in the native language of the processor. After the IC cards are manufactured, they are sent to a personalization bureau ("PB") to enable and personalize the card by storing card personalization data in the memory of the card. The terms enablement and personalization are used interchangeably herein to indicate the preparatory steps taken to allow the card to be loaded securely with an application. The individual cards are preferably manufactured in batches and are sent to a personalization bureau in a group for processing.

Card Enablement/Personalization

Figure 3 shows the steps of the card enablement process when the card arrives at a personalization bureau. The personalization bureau may be the card issuer (e.g., a bank or other financial institution) or may be a third party that performs the service for the card issuer. The personalization bureau configures the card to a specific user or user class.

Figure 3 specifically shows the steps taken to enable and personalize each IC card which will work within the system. The cards can be placed in a terminal which communicates with IC cards and which reads the card identifier data (previously placed on the card during the manufacturing process -- see step 207). This card identification data is read from the card in step 301. The terminal will effectively send a "get identification data" command to the card and the card will return the identification data to the terminal.

The PB typically processes a group of cards at the same time, and will first compile a list of IC card identification data for the group of cards it is personalizing. The PB then sends electronically (or otherwise) this list of identification data to the Certification Authority ("CA") which creates a personalization (or enablement) data block for each card identifier. The data block includes the card personalization data organized in a number of identity fields and an individual key set for the card, discussed below. These data blocks are then encrypted and sent to the PB in step 302. By using the card identification data, the

PB then matches the cards with the encrypted data blocks and separately loads each data block onto the matched card. To insure that the CA controls the identity of the card and the integrity of the system, the PB never obtains knowledge of the content of the data blocks transferred. Some aspects of the personalization are requested by the card issuer to the CA in order to affect their preferred management of the cards they issue. The following additional steps are performed.

Step 303 first checks to see if an enablement bit stored in EEPROM of the card has been already set. If it already has been set, the card has already been configured and personalized and the enablement process will end as shown in step 304. A card cannot be enabled and personalized twice. If the bit has not been set, then the process continues with step 305.

In step 305, the individualized card key set for the card being enabled (which key set is generated at the CA) is stored on the card. The keys can be used later in off-card verification (i.e., to verify that the card is an authentic card). This verification is necessary to further authenticate the card as the one for which the application was intended.

Step 307 generates four different MULTOS Security Manager (MSM) characteristic data elements (otherwise referred to herein as personalization data) for the card at the CA which are used for securely and correctly loading and deleting applications from a particular card. The MSM characteristics also allow for the loading of applications on specific classes of identified cards. (These MSM characteristics are further described in connection with Figure 5.)

Other data can also be stored on the card at this time as needed by the system design such as an address table or further subroutines.

Step 311 sets the enablement bit in EEPROM of the card which indicates that the enablement process has been completed for the particular card. When this bit is set, another enablement process cannot occur on the card. This ensures that only one personalization and enablement process will occur to the card thus inhibiting illegal tampering of the card or altering the card by mistake. In the preferred embodiment, the enablement bit is initially not set when the card is manufactured and is set at the end of the enablement process.

Figure 4 shows an example of a block diagram of an IC card chip which has been manufactured and personalized. The IC card chip is located on an IC card for use. The IC card preferably includes a central processing unit 401, a RAM 403, a EEPROM 405, a ROM 407, a timer 409, control logic 411, an I/O ports 413 and security circuitry 415, which are connected together by a conventional data bus.

Control logic 411 in memory cards provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports. CPU 401 with its control logic can perform calculations, access memory locations, modify memory contents, and manage input/output ports. Some cards have a coprocessor for handling complex computations like cryptographic algorithms. Input/output ports 413 are used under the control of a CPU and control logic alone, for communications between the card and a card acceptance device. Timer 409 (which generates or provides a clock pulse) drives the control logic 411 and CPU 401 through the sequence of steps that accomplish memory access, memory reading or writing, processing, and data communication. A timer may be used to provide application features such as call duration. Security circuitry 415 includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed ("blown") upon completion of testing to prevent later access. The personalization data to qualify the card is stored in a secured location of EEPROM 405. The comparing of the personalization data to applications permissions data is performed by the CPU 401.

Figure 5 shows the steps of generating and loading the four elements of the card personalization data into the memory of the IC cards, and Fig. 5A shows a schematic of bit maps for each identity field residing in the memory of an IC card containing personalization data in accordance with the present invention. Each data structure for each identity field has its own descriptor code. Step 501 loads the data structure for the identity field "card ID" called "msm_mcd_permissions_mcd_no." This nomenclature stands for MULTOS system manager _ MULTOS card device _ permissions_ MULTOS card device number. Although this number is typically 8 bytes long as shown in Fig. 5A, the data could be any length that indicates a unique number for the card. In the preferred embodiment, 2 bytes are dedicated as a signal indicator, 2 bytes comprise a MULTOS Injection Security Module ID (MISM ID) indicating which security module injected the card with its injected keys when it was manufactured, and 4 bytes comprise an Integrated Circuit Card (ICC) serial number which identifies the individual card produced at the particular MISM.

Step 503 loads the data structure for the identity field "issuer ID" called "msm_mcd_permissions_ mcd_issuer_id." This nomenclature stands for a MULTOS card device issuer identification number. Each card issuer (such as a particular bank, financial institution or other company involved with an application) will be assigned a unique number in the card system. Each IC card in the MULTOS system will contain information regarding the card issuer which personalized the card or is responsible for the card. A card issuer will order a certain number of cards from a manufacturer and perform or have performed the personalization process as described herein. For example, a regional bank may order 5,000 cards to be distributed to its customers. The "mcd_issuer_id" data structure on these cards will indicate which issuer issued the cards. In the preferred embodiment, the data structure is 4 bytes long (as shown in Fig. 5A at 503A) to allow for many different issuers in the system although the length of the data structure can vary with the needs of the card system.

Step 505 loads the data structure for the identity field "product ID" called "msm_mcd_permissions_mcd_ issuer_product_id." This nomenclature stands for MULTOS card device issuer product identification number. Each card issuer may have different classes of products or cards which it may want to differentiate. For example, a bank could issue a regular credit card with one product ID, a gold credit card with another product ID and a platinum card with still another product ID. The card issuer may wish to load certain applications onto only one class of credit cards. A gold credit card user who pays an annual fee may be entitled to a greater variety of applications than a regular credit card user who pays no annual fee. The product ID field identifies the card as a particular class and will later allow the card issuer to check the product ID and only load applications onto cards which match the desired class.

Another way to differentiate products is by application type, such as by categorizing the application as financial, legal, medical and/or recreational, or by assigning particular applications to a group of cards. For example, one card issuer may have different loyalty programs available with different companies to different sets of card users. For example, a bank may have an American Airlines® loyalty program and a British Airways® loyalty program for different regions of the country dependent on where the airlines fly. The product type allows the issuer to fix the product classification of the card during the personalization process. When loading applications onto the card, the product type identification number on each card will be checked to make sure it matches the type of card onto which the issuer desires to load. The product type data structure is preferably an indexing mechanism (unlike the other personalization data structure) of 8 bits (as shown at 505A in Fig. 5A) but could be any length depending upon the needs of the card system. In the illustrated embodiment, the resulting instruction would be to locate the second bit (since the byte's indicated value is 2) in the array to be searched (see discussion of step 809 below).

Step 507 loads the data structure for the identity field data called "msm_mcd_permissions_mcd_ controls_data_ date." This nomenclature stands for the MULTOS card device controls data date or, in other words, the date on which the card was personalized so that, for example, the application loader can load cards dated only after a certain date, load cards before a certain date (e.g., for application updates) or load cards with a particular data date. The information can include the year, month and day of personalization or may include less information, if desired. The data_date data structure is preferably 1 byte in length (see 507A in Fig. 5A) although it could be any length depending upon the needs of the particular card system used.

Once all of the personalization data structures are loaded and stored in the card, the card has been identified by issuer, product class, date and identification number (and other data fields, if desired), and the card cannot change its identity; these fields cannot be changed in the memory of the card. If a card user wants to change the product_id stored in the card to gain access to different applications available to another product type, a new card will have to be issued to the user containing the correct personalization data. This system is consistent with a gold card member receiving a new card when the classification is changed to platinum.

After the card has been enabled and personalized by storing its individual card key set, MSM personalization characteristics and enablement bit as described in Fig. 3, the card is ready to have applications loaded into its memory.

Loading Applications

The application loading process contains a number of security and card configuration checks to ensure the secure and proper loading of an application onto the intended IC card. The application loading process is preferably performed at the personalization bureau so that the card will contain one or more applications when the card is issued. The card may contain certain common applications which will be present on every card the issuer sends out, such as an electronic purse application or a credit/debit application. Alternatively, the personalization bureau could send the enabled cards to a third party for the process of loading applications. The multiple application operating system stored in the ROM of each card and the card MSM personalization data is designed to allow future loading and deleting of applications after the card has been issued depending upon the desires of the particular card user and the responsible card issuer. Thus, an older version of an application stored on the IC card could be replaced with a new version of the application. An additional loyalty application could also be added to the card after it has been initially sent to the card user because the application is newly available or the user desires to use the new application. These loading and deleting functions for applications can be performed directly by a terminal or may be performed over telephone lines, data lines, a network such as the Internet or any other way of transmitting data between two entities. In the present IC card system, the process of transmitting the application program and data ensures that only IC cards containing the proper personalization data and which fit on application permissions profile will be qualified and receive the corresponding application program and data.

Figure 6 shows the preferred steps performed in loading an application onto an IC card in the MULTOS IC card system. For this example, the personalization bureau is loading an application from a terminal which enabled the same card. Step 601 performs an “open command” initiated by the terminal which previews the card to make sure the card is qualified to accept the loading of a specific application. The open command provides the card with the application's permissions data, the application's size, and instructs the card to determine (1) if the enablement bit is set indicating the card has been personalized; (2) whether the application code and associated data will fit in the existing memory space on the card; and (3) whether the personalization data assigned to the application to be loaded allows for the loading of the application onto the particular card at issue. The open command could also make additional checks as required by the card system. These checking steps during the open command execution will be described in detail in conjunction with Figure 7.

After the open command has been executed, the application loader via the terminal will be advised if the card contains the proper identification personalization data and if enough room exists in the memory of the card for the application code and related data. If there is insufficient memory, then a negative response is returned by the card and the process is abended (abnormally ended). If the identification personalization data does not match the applications permissions data, a warning response is given in step 603, but the process continues to the load and create steps. Alternatively, if there is no match, the process may automatically be abended. If a positive response is returned by the card to the terminal in step 605, the application loader preferably proceeds to next steps. The open command allows the application to preview the card before starting any transfer of the code and data.

Step 607 then loads the application code and data onto the IC card into EEPROM. The actual loading occurs in conjunction with create step 609 which completes the loading process and enables the application to execute on the IC card after it is loaded. The combination of the open, load and create commands are sent by the terminal, or another application provider source, to the IC card to perform the application loading process. The operating system in the IC cards is programmed to perform a specific set of instructions with respect to each of these commands so that the IC card will communicate with and properly carry out the instructions from the terminal.

Step 609 performs the create command which at least: (1) checks if an application load certificate is signed (encrypted) by the CA and therefore authenticates the application as a proper application for the system; and (2) checks the card personalization data stored on the card against the permissions profile for the application to be loaded to qualify the card for loading. It may do other checks as required. If one of the checks fails, then a failure response 610 is given and the process aborts. The application after it has passed these checks will be loaded into the memory of the card.

Figure 7 shows the various steps of the open step 601 of Fig. 6 in more detail. Step 701 determines if the enablement (i.e., control) bit is set. This bit is set when the card has completed its personalization process and has been assigned its personalization data. An application can be loaded on an IC card in the card system only if the card contains the personalization data. If the enablement bit is not set, the card has not been personalized and therefore the card returns a negative response 703 to the terminal. If the enablement bit is set, then the card has been enabled and the test conditions continue with step 711.

Step 711 checks if there is sufficient space in the memory on the card to store the application code and its associated data. Applications will typically have associated data related to their functions. This data will be used and manipulated when the application is run. Storage space in the memory of an IC card is a continuing concern due to the relatively large physical space required for EEPROM and how it fits in the integrated circuit which is desired to be small enough to fit on a credit card sized card. An example of the size of a preset EEPROM on an IC card is 16K bytes although the actual size varies. Applications can range from 1K byte or less for a very simple application up to the size of available memory for a more sophisticated application. The data associated with an application can range from no data being stored in the card memory to a size constrained by the amount of available memory. These varied sizes of application code and data continually increase as applications become more advanced and diverse.

MULTOS as an operating system is not limited by the number of applications and associated data it can store on the card. Thus, if five applications can fit in the available memory of the card, the card user will have greatly increased functionality than if one or two applications were stored on the card. Once a card's memory is filled to its capacity, however, a new application cannot be loaded onto the card unless another application including its code and data of sufficient size can be deleted. Therefore, checking the amount of available space on the card is an important step. If there is not sufficient space, then an insufficient space response 713 will be returned to the terminal. The application loader can then decide if another existing application on the card should be deleted to make room for the new application. Deletion depends upon the card issuer having an application delete certificate from the CA. If there is sufficient space on the card, then the process continues with step 715.

An example of the testing of memory spaces in step 711 is now described. The numbers used in this example in no way limit the scope of the invention but are used only to illustrate memory space requirements. An IC card may have 16K available EEPROM when it is first manufactured. The operating system data necessary for the operating system may take up 2K of memory space. Thus, 14K would remain. An electronic purse application's code is stored in EEPROM and may take up 8K of memory space. The purse application's required data may take up an additional 4K of memory space in EEPROM. The memory space which is free for other applications would thus be 2K (16K-2K-8K-4K=2K). If a card issuer wants to load a credit/debit application whose code is 6K bytes in size onto the card in this example, the application will not fit in the memory of the IC card. Therefore, the application cannot load the new application without first removing the purse application from the card. If a new credit/debit application was loaded into EEPROM of the IC card, then it would have to overwrite other application's code or data. The application loader is prevented from doing this.

Figure 8 shows the steps performed in determining whether the card's personalization data falls within the permissible set of cards onto which the application at issue may be loaded. These steps are preferably performed during the execution of the "create" command. However, these steps may be performed at any time during the loading or deleting of an application. As described previously, the card is personalized by storing data specific to the card (MSM personalization data) including: a card ID designation specific to an individual card, the card issuer number indicating the issuer of the card, the product type of the card, such as a gold or platinum card, and the date the card was personalized. This data uniquely identifies the card apart from all other IC cards in the system.

Accordingly, applications can be selectively stored on individual cards in the IC card system on virtually any basis, including the following. An application can be loaded selectively to cards containing one or more specific card numbers. An application can be selectively loaded on one or more cards containing a specified card issuer ID. Moreover, an application can be loaded only upon one type of product specified by the particular card issuer, and/or the application can be loaded only on cards which have a specified date or series of dates of personalization. Each of the personalization data allows an application to be selectively loaded onto certain cards or groups of cards and also ensures that cards without the proper permissions will not receive the application. Personalization data types in addition to the four described can also be used as needed.

The selection of IC cards upon which a particular application may be loaded is made possible by the use of "applications permissions data" which is assigned to the application and represents at least one set of cards upon which the application may be loaded. The set may be based on virtually any factor, including one or more of the following: card numbers, card issuers, product types or personalization dates. Although the individual card's personalization data typically identify one specific number, one card issuer, one product type and one date, the application's permissions data may indicate a card number or a blanket permission, a card issuer or a blanket permission, and a number of product types and dates.

For example, a frequent loyalty program may be configured to allow its loading and use on cards in different product classes belonging to one card issuer. In addition, the application permissions data may indicate that the loyalty program can be used on gold and platinum product types if the card was issued after May, 1998. Thus, the MSM permissions check will determine if the card's individual personalization data is included in the allowed or permissible set of cards upon which the application may be loaded. If it is, the application will be loaded.

To expedite the comparison process, an alternative embodiment may include setting one or more permissions data at zero representing a blanket permission for that particular data. For instance, by placing a zero for the "card number" entry in the application permissions data or some other value indicating that all cards may be loaded regardless of their number, the system knows not to deny any cards based on their card number. Moreover, if a zero is placed in the application's permissions data "issuer ID," then all cards similarly will pass the "issuer" test comparison. This feature allows greater flexibility in selecting groups of cards. The zero indicator could also be used for other permissions data, as required.

Referring to Figure 8, each of the permissions data is checked in the order shown, but other orders could be followed because if any one of the permissions fails, the application will be prevented from being loaded on the IC card being checked. The permissions are preferably checked in the order shown. Step 801 checks if the application permissions product type set encompasses the card's product type number stored in the memory of the card. Each card product type is assigned a number by the system operator. The product types are specified for each card issuer because different card issuers will have different product types. The cards are selectively checked to ensure that applications are loaded only on cards of authorized product type. The application permissions product type set can be 32 bytes long which includes multiple acceptable product types or can be a different length depending upon the needs of the system. Using data structure 505A as an example, the operating system would check bit number 2 in the 256 bit array (32 bytes x 8 bits per byte) resulting from the 32 byte long application permissions data structure. If the permissions check fails, then the card returns a failure message to the terminal in step 803. If the product type check passes (for example, the value of bit no. 2 being 1), then the process continues with step 805.

Step 805 checks if the application permissions allowable card issuer number set encompasses the card's issuer number stored in the memory of the card or if the application permissions issuer data is zero (indicating all cards pass this individual permissions check). Each card issuer is assigned a number by the system operator and the cards are selectively checked to ensure that applications are loaded only on cards distributed by authorized card issuers. The application permissions card issuer number set can be 4 bytes long if one issuer is designated or can be longer depending upon the needs of the system. If the issuer check fails, then the card returns a failure message to the terminal in step 807. If the check passes, then the process continues with step 809.

Step 809 checks if the application permissions date set encompasses the card's data date stored in the memory of the card. The date that the IC card was personalized will be stored and will preferably include at least the month and year. The cards are selectively checked to ensure that applications are loaded only on cards with the authorized personalization date. The application permissions date set can be 32 bytes long which includes multiple dates or can be a different length depending upon the needs of the system. If the date permissions check fails, then the card returns a failure message to the terminal in step 811. If the date check passes, then the process continues with step 813.

Step 813 checks if the application permissions allowable card number set encompasses the card's ID number stored in the card memory or if the application permissions allowable card number data is zero (indicating all cards pass this individual permissions check). The testing of the permissions is performed on the card during the execution of the open, load and create commands. The application permissions card number data set can be 8 bytes long if one number is designated or can be longer depending upon the needs of the system. If the card number check fails, then the card returns a failure message to the terminal in step 815. If the check passes, then the process continues with step 817.

Summary of IC Card System's Process

Figure 9 shows the components of the system architecture for the card initialization process of an IC card in a secure multiple application IC card system. The system includes a card manufacturer 102, a personalization bureau 104, an application loader 106, the IC card 107 being initialized, the card user 109 and the certification authority 111 for the entire multiple application secure system. The card user 131 is the person or entity who will use the stored applications on the IC card. For example, a card user may prefer an IC card that contains both an electronic purse containing electronic cash (such as MONDEX™) and a credit/debit application (such as the MasterCard® EMV application) on the same IC card. The following is a description of one way in which the card user would obtain an IC card containing the desired applications in a secure manner.

The card user would contact a card issuer 113, such as a bank which distributes IC cards, and request an IC card with the two applications both residing in memory of a single IC card. The integrated circuit chip for the IC card would be manufactured by manufacturer 102 and sent to the card issuer 113 (or an entity acting on its behalf) in the form of an IC chip on a card. As discussed above (see steps 201-209), during the manufacturing process, data is transmitted 115 via a data conduit from the manufacturer 102 to card 107 and stored in IC card 107's memory. (Any of the data conduits described in this figure could be a telephone line, Internet connection or any other transmission medium.) The certification authority 111, which maintains encryption/decryption keys for the entire system, transmits 117 security data (i.e., global public key) to the manufacturer over a data conduit which is placed on the card by the manufacturer along with other data, such as the card enablement key and card identifier. The card's multiple application operating system is also stored in ROM and placed on the card by the manufacturer.

After the cards have been initially processed, they are sent to the card issuer for personalization and application loading.

The card issuer 113 performs, or has performed by another entity, two separate functions. First, the personalization bureau 104 personalizes the IC card 107 in the ways described above, and second, the application loader 106 loads the application provided the card is qualified, as described.

Regarding personalization, an individualized card key set is generated by the CA and stored on the card (see Fig. 3). The card is further given a specific identity using MSM personalization (see Fig. 3, step 307 and Fig. 5) including a card ID number, an issuer ID number identifying the card issuer which processed the card, a card product type number which is specified by the card issuer and the date upon which the personalization took place. After the card has been personalized, applications need to be loaded onto the card so that the card can perform desired functions.

The application loader 106, which could use the same terminal or data conduit as personalization bureau 104, first needs to have determined if the card is qualified to accept the application. This comparison process takes place on the card itself (as instructed by its operating system) using the permissions information. The card, if it is qualified, thus selectively loads the application onto itself based upon the card's identity and the card issuer's instructions. The application loader communicates 119 with the IC card via a terminal or by some other data conduit. After the applications have been loaded on the card, the card is delivered to the card user 109 for use.

The secure multiple application IC card system described herein allows for selective loading and deleting of applications at any point in the life cycle of the IC card after the card has been personalized. Thus, a card user could also receive a personalized card with no applications and then select a desired application over a common transmission line such as a telephone line or Internet connection.

Figure 10 is a system diagram of entities involved with the use of an IC card once it has been personalized. The system includes an IC card 151, a terminal 153, an application load/delete entity 155, the certification authority 157, a card issuer 171 and other IC cards 159 in the system. The arrows indicate communication between the respective entities. The CA 157 facilitates loading and deleting of applications. After providing the MSM permissions data and card specific keyset to the card during card enablements, the CA allows applications to be later loaded and deleted preferably by issuing an application certificate. Application specific keys are required to authenticate communication between a card and terminal. The IC card 151 also can communicate with other IC cards 159. Card issuer 171 is involved with all decisions of loading and deleting applications for a card which it issued. All communications are authenticated and transmitted securely in the system.

For instance, IC card 151 will use the following procedure to load a new application onto the card. IC card 101 is connected to terminal 153 and the terminal requests that an application be loaded. Terminal 153 contacts application load/delete entity 155 which, as a result and in conjunction with card issuer 171, sends the application code, data and application permissions data (along with any other necessary data) to terminal 153. Terminal 153 then queries card 151 to ensure it is the correct card onto which the application may be loaded. If IC card passes the checks discussed above, the application is loaded onto card 151. The CA 157 provides the application load or delete certificate that enables the application to be loaded or deleted from the card. This example shows one way to load the application, but other variations using the same principles could be performed, such as directly loading the application at the application load/delete entity 155.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

For example, it will be appreciated that the MSM personalization and permissions data may not only be used for loading applications onto IC cards but also for deleting applications from said cards. The same checks involving MSM permissions and loading applications are made for deleting applications. A delete certificate from the CA authorizing the deletion of an application will control from which cards the application may be deleted. This is accomplished through the personalization data stored on each IC card and the permissions check as described herein.

Moreover, the data may also be applicable to personal computers or other units onto which applications may be loaded which are not physically loaded on cards. In addition, the application's permissions data may actually include data representative of a set or sets of cards to be excluded, instead of included -- cards that cannot be loaded with the application.

The scope of the present disclosure includes any novel feature or combination of features disclosed therein either explicitly or implicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or mitigates any or all of the problems addressed by the present invention. The application hereby gives notice that new claims may be formulated to such features during the prosecution of this application or of any such further application derived therefrom. In particular, with reference to the appended claims, features from dependant claims may be combined with those of the independent claims in any appropriate manner and not merely in the specific combinations enumerated in the claims.

ANNEX A

MULTI-APPLICATION IC CARD SYSTEM

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. This allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application at its personalization stage. That application, however, cannot be modified or changed after the card is issued even if the modification is desired by the card user or card issuer. Moreover, if a card user wanted a variety of application functions to be performed by IC cards issued to him or her, such as

ANNEX A TO THE DESCRIPTION

both an electronic purse and a credit/debit function, the card user would be required to carry multiple physical cards on his or her person, which would be quite cumbersome and inconvenient. If an application developer or card user desired two different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal, making the transaction difficult, lengthy and inconvenient.

The Applicant has recognised therefore, that it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card. Although multiple applications could be pre-selected and placed in the memory of the card during is production stage, it would also be beneficial to have the ability to load and delete applications for card post-production as needed.

The increased flexibility and power of storing multiple applications on a single card create new challenges to be overcome concerning the integrity and security of the information (including application code and associated data) exchanged between the individual card and the application provider as well as within the entire system when loading and deleting applications. The Applicant has further recognised that it would be beneficial to have the capability of the IC card system to exchange data among cards, card issuers, system operators and application

ANNEX A TO THE DESCRIPTION

providers securely and to load and delete applications securely at any time from either a terminal or remotely over a telephone line, internet or intranet connection or other data conduit. Because these data transmission lines are not typically secure lines, a number of security and entity-authentication techniques must be implemented to make sure that applications being sent over the transmission lines are only loaded on the intended cards.

As mentioned, it is important -- particularly where there is a continuing wide availability of new applications to the cardholder -- that the system has the capability of adding applications onto the IC card subsequent to issuance. This is highly advantageous since it protects the longevity of the IC cards; otherwise, once an application becomes outdated, the card would be useless. In this regard, to protect against the improper or undesired loading of applications onto IC cards, the Applicant has further recognised that it would be beneficial for the IC card system to have the capability of controlling the loading process and restricting, when necessary or desirable, the use of certain applications to a limited group or number of cards such that the applications are "selectively available" to the IC-cards in the system. This "selective capability" would allow the loading and deleting of applications at, for example, a desired point in time in the card's life cycle. It would also allow the loading of an application only to those cards chosen to receive the selected application.

Accordingly, it is an advantage of a preferred embodiment of the invention that it provides these important features and specifically a secure IC-card system that allows for selective availability of smart card applications which may be loaded onto IC cards.

ANNEX A TO THE DESCRIPTION

These and other advantages are achieved by an embodiment of the present invention which proves an IC card system comprising at least one IC card and an application to be loaded onto the card wherein the IC card contains card personalization date and the application is assigned application permissions data designating which IC card or group of IC cards upon which the application may be loaded. The system checks to determine whether the card's personalization data falls within the permissible set indicated by the application's permissions data. If it does, the application may be loaded onto the card.

In a preferred embodiment, the card personalization data is transferred onto the card by the personalization bureau after the card is manufactured. The data preferably includes data representing the card number, the issuer, product class (i.e., such as gold or platinum cards), and the date on which the card was personalized. The card further preferably contains enablement data indicating whether or not the card has been enabled with personalized data.

In a further preferred embodiment, the IC card secure system checks the enablement data prior to loading an application to determine whether or not the card has been enabled. Preferably, if the card has been enabled, the system checks if the card number, the issuer, the product class and/or the date on which the card was personalized are within the acceptable set indicated by the application's permissions data. If so, the application may be loaded onto the IC card.

ANNEX A TO THE DESCRIPTION

In yet another preferred embodiment, the application's permissions data may contain data representative of a blanket permission such that all cards would pass for application loading.

Further aspects, features and advantages of embodiments of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

Fig. 1 is block diagram illustrating the three stages in the life of a multi-application IC card in a secure system;

Fig. 2 is a block diagram illustrating the steps of the card manufacture process;

Fig. 3 is a flow diagram illustrating the steps involved in enabling each of the IC cards in the secure system;

Fig. 4 is a block diagram of an IC card chip which can be used in accordance with an embodiment of the invention;

Fig. 5 is a block diagram illustrating the data stored on the IC card as indicated in block 307 of Fig. 3;

Fig. 5A is a schematic of the data structures residing in an IC card and representing personalization data;

Fig. 6 is a flowchart illustrating the steps of loading an application onto an IC card in the secure system;

Fig. 7 is a flow chart illustrating the checking steps as indicated in block 601 of Fig. 6;

Fig. 8 is a flowchart illustrating the steps undertaken in determining if loading of an application may proceed;

Fig. 9 is a block diagram showing the components of the system architecture for the enablement process of an IC card in a secure multi-application IC card system; and

Fig. 10 is a system diagram of entities involved with the use of the IC card once it has been personalized.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

ANNEX A TO THE DESCRIPTION

An embodiment of the present invention provides an IC card system and process which allow the flexibility to load and delete selected applications over the lifetime of a multi-application IC card in response to the needs or desires of the card user, card issuers and/or application developers. A card user who has such a card can selectively load and delete applications as desired if allowed by the card issuer in conjunction with the system operator or Certification Authority ("CA") which controls the loading and deleting process by certifying the transfer of information relating to the process.

By allowing applications to be selectively loaded and deleted from the card, a card issuer can extend additional functionality to an individual IC card without having to issue new cards. Moreover, application developers can replace old applications with new enhanced versions, and applications residing on the same card using a common multiple application operating system may interact and exchange data in a safe and secure manner. For example, a frequent flyer loyalty program may automatically credit one frequent flyer mile to a card user's internal account for every dollar spent with an electronic purse such as the Mondex purse or with a credit/debit application. By allowing the ability to selectively load and delete applications, the card user, subject to the requirements of the card issuer, also has the option of changing loyalty programs as desired.

A card issuer or application developer may intend that a particular application be loaded on only one card for a particular card user in a card system. A regional bank may desire to have a proprietary application reside only on the cards which

the bank issues. Embodiments in accordance with the present invention would allow for this selective loading and specifically allow for the prevention of loading proprietary applications onto unauthorized cards issued by others.

To achieve these desired objectives, embodiments of the present invention give each card a specific indentity by storing "card personalization data" on the card. Morover, each application to be loaded or deleted on one or more cards in the system is assigned "application permissions data" which specify the cards upon which the applications may be loaded.

The type of personalized data can vary depending upon the needs and requirements of the card system. In the preferred embodiment, described in greater detail below, the personalization data include unique card identification designation data, the card issuer, the product class or type (which is defined by the card issuer) and the date of personalization. However, not all of these data elements are required to be used and additional elements could also be included.

The application permissions data associated with an application, also described in greater detail below, can be a single value in an identity field or could include multiple values in the identity field. For example, the application permissions data in the card issuer field could represent both product class A and product class B from a certain Bank X, indicating that the application could be loaded onto cards designated as product classes A and B issued by Bank X (as indicated in the card product ID field of the card's personalization data).

In addition, a "global value" could be stored in the issuer field (or other field) of the application permissions data indicating that all IC cards in the system regardless of who issued the card would match this permissions field. In this case, for example, a data value of zero stored in the application permissions card-issuer field will match all of the cards' personalization card-issuer fields.

Figure 1 shows the three steps involved in providing an operational multi-application IC card in a secure system. The first step is the card manufacturing step 101. The second step is the personalization step 103 where card personalization data (also called entity authentication data) is loaded onto the card. The third step is the application loading step 105 which checks to see if a card is qualified to receive an application, i.e., when the personalization data is checked against the application permissions data associated with the application to be loaded. Each of these three steps is described in detail below.

Card Manufacture

Figure 2 shows the steps necessary in manufacturing an IC card in a secure system. Step 201 manufactures the physical IC card by creating the integrated circuit on silicon and placing it on the card. The integrated circuit chip will include RAM, ROM and EEPROM memories. When the card is first manufactured, a global public key of the system operator (in this case called the Certification Authority (CA)) is stored on each card in ROM in step 203. This will allow the card to authenticate that the source of any message to it is from the CA since the public key on the card will be matched to the CA's secret key.

ANNEX A TO THE DESCRIPTION

More specifically, this public key stored on the card will allow the individual card to verify data signed with the CA's private key. The public key of the CA, which is stored on the card, is used only for determining if the data sent to the card was signed with the proper CA private key. This allows the card to verify the source of any message coming from the CA.

Step 205 inserts a card enablement key in a secure portion of EEPROM in the card to facilitate card specific confidentiality during enablement, and step 207 inserts a card identifier in EEPROM of the card. The identifier, which can be accessed by any terminal, will allow the system to determine the identity of the card in later processes. The identifier is freely available and will not be used to authenticate messages.

Step 209 stores the operating system code in ROM on the card including any primitives which are called or supported by the operating system. The primitives are written in native language code (e.g., assembly language) and are stored in ROM. The primitives are subroutines which may be called by the operating system or by applications residing on the card such as mathematic functions (multiply or divide), data retrieval, data manipulation or cryptographic algorithms. The primitives can be executed very quickly because they are written in the native language of the processor.

After the IC cards are manufactured, they are sent to a personalization bureau ("PB") to enable and personalize the card by storing card personalization data in the memory of the card. The terms enablement and personalization are used interchangeably herein to indicate the preparatory steps taken to allow the card to be loaded securely with

ANNEX A TO THE DESCRIPTION

an application. The individual cards are preferably manufactured in batches and are sent to a personalization bureau in a group for processing.

Card Enablement/Personalization

Figure 3 shows the steps of the card enablement process when the card arrives at a personalization bureau. The personalization bureau may be the card issuer (e.g., a bank or other financial institution) or may be a third party that performs the service for the card issuer. The personalization bureau configures the card to a specific user or user class.

Figure 3 specifically shows the steps taken to enable and personalize each IC card which will work within the system. The cards can be placed in a terminal which communicates with IC cards and which reads the card identifier data (previously placed on the card during the manufacturing process -- see step 207). This card identification data is read from the card in step 301. The terminal will effectively send a "get identification data" command to the card and the card will return the identification data to the terminal.

The PB typically processes a group of cards at the same time, and will first compile a list of IC card identification data for the group of cards it is personalizing. The PB then sends electronically (or otherwise) this list of identification data to the Certification Authority ("CA") which creates a personalization (or enablement) data block for each card identifier. The data block includes the card personalization data organized in a number of identity fields and an individual key set for the card, discussed below. These data blocks are then encrypted and sent to the PB in step 302. By using the

ANNEX A TO THE DESCRIPTION

card identification data, the PB then matches the cards with the encrypted data blocks and separately loads each data block onto the matched card. To insure that the CA controls the identity of the card and the integrity of the system, the PB never obtains knowledge of the content of the data blocks transferred. Some aspects of the personalization are requested by the card issuer to the CA in order to affect their preferred management of the cards they issue. The following additional steps are performed.

Step 303 first checks to see if an enablement bit stored in EEPROM of the card has been already set. If it already has been set, the card has already been configured and personalized and the enablement process will end as shown in step 304. A card cannot be enabled and personalized twice. If the bit has not been set, then the process continues with step 305.

In step 305, the individualized card key set for the card being enabled (which key set is generated at the CA) is stored on the card. The keys can be used later in off-card verification (i.e., to verify that the card is an authentic card). This verification is necessary to further authenticate the card as the one for which the application was intended.

Step 307 generates four different MULTOS Security Manager (MSM) characteristic data elements (otherwise referred to herein as personalization data) for the card at the CA which are used for securely and correctly loading and deleting applications from a particular card. The MSM characteristics also allow for the loading of applications on specific classes of identified cards. (These MSM characteristics are further described in connection with Figure 5.)

Other data can also be stored on the card at this time as needed by the system design such as an address table or further subroutines.

Step 311 sets the enablement bit in EEPROM of the card which indicates that the enablement process has been completed for the particular card. When this bit is set, another enablement process cannot occur on the card. This ensures that only one personalization and enablement process will occur to the card thus preventing illegal tampering of the card or altering the card by mistake. In the preferred embodiment, the enablement bit is initially not set when the card is manufactured and is set at the end of the enablement process.

Figure 4 shows an example of a block diagram of an IC card chip which has been manufactured and personalized. The IC card chip is located on an IC card for use. The IC card preferably includes a central processing unit 401, a RAM 403, a EEPROM 405, a ROM 407, a timer 409, control logic 411, an I/O ports 413 and security circuitry 415, which are connected together by a conventional data bus.

Control logic 411 in memory cards provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports. CPU 401 with its control logic can perform calculations, access memory locations, modify memory contents, and manage input/output ports. Some cards have a coprocessor for handling complex computations like cryptographic algorithms. Input/output ports 413 are used under the control of a CPU and control logic alone, for communications between the card and a card acceptance device. Timer 409 (which generates or provides a clock pulse) drives the control logic 411 and CPU 401 through the sequence of steps that

accomplish memory access, memory reading or writing, processing, and data communication. A timer may be used to provide application features such as call duration. Security circuitry 415 includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed ("blown") upon completion of testing to prevent later access. The personalization data to qualify the card is stored in a secured location of EEPROM 405. The comparing of the personalization data to applications permissions data is performed by the CPU 401.

Figure 5 shows the steps of generating and loading the four elements of the card personalization data into the memory of the IC cards, and Fig. 5A shows a schematic of bit maps for each identity field residing in the memory of an IC card containing personalization data in accordance with the present invention. Each data structure for each identity field has its own descriptor code. Step 501 loads the data structure for the identity field "card ID" called "msm_mcd_permissions_mcd_no." This nomenclature stands for MULTOS system manager _ MULTOS card device _ permissions_ MULTOS card device number. Although this number is typically 8 bytes long as shown in Fig. 5A, the data could be any length that indicates a unique number for the card. In the preferred embodiment, 2 bytes are dedicated as a signal indicator, 2 bytes comprise a MULTOS Injection Security Module ID (MISM ID) indicating which security module injected the card with its injected keys when it was manufactured, and 4 bytes comprise an Integrated Circuit Card (ICC) serial number which identifies the individual card produced at the particular MISM.

Step 503 loads the data structure for the identity field "issuer ID" called "msm_mcd_permissions_ mcd_issuer_id." This nomenclature stands for a MULTOS card device issuer identification number. Each card issuer (such as a particular bank, financial institution or other company involved with an application) will be assigned a unique number in the card system. Each IC card in the MULTOS system will contain information regarding the card issuer which personalized the card or is responsible for the card. A card issuer will order a certain number of cards from a manufacturer and perform or have performed the personalization process as described herein. For example, a regional bank may order 5,000 cards to be distributed to its customers. The "mcd_issuer_id" data structure on these cards will indicate which issuer issued the cards. In the preferred embodiment, the data structure is 4 bytes long (as shown in Fig. 5A at 503A) to allow for many different issuers in the system although the length of the data structure can vary with the needs of the card system.

Step 505 loads the data structure for the identity field "product ID" called "msm_mcd_permissions_mcd_ issuer_product_id." This nomenclature stands for MULTOS card device issuer product identification number. Each card issuer may have different classes of products or cards which it may want to differentiate. For example, a bank could issue a regular credit card with one product ID, a gold credit card with another product ID and a platinum card with still another product ID. The card issuer may wish to load certain applications onto only one class of credit cards. A gold credit card user who pays an annual fee may be entitled to a greater variety of applications than a regular credit card user who pays no annual fee. The product ID field identifies the card as a

ANNEX A TO THE DESCRIPTION

particular class and will later allow the card issuer to check the product ID and only load applications onto cards which match the desired class.

Another way to differentiate products is by application type, such as by categorizing the application as financial, legal, medical and/or recreational, or by assigning particular applications to a group of cards. For example, one card issuer may have different loyalty programs available with different companies to different sets of card users. For example, a bank may have an American Airlines® loyalty program and a British Airways® loyalty program for different regions of the country dependent on where the airlines fly. The product type allows the issuer to fix the product classification of the card during the personalization process. When loading applications onto the card, the product type identification number on each card will be checked to make sure it matches the type of card onto which the issuer desires to load. The product type data structure is preferably an indexing mechanism (unlike the other personalization data structure) of 8 bits (as shown at 505A in Fig. 5A) but could be any length depending upon the needs of the card system. In the illustrated embodiment, the resulting instruction would be to locate the second bit (since the byte's indicated value is 2) in the array to be searched (see discussion of step 809 below).

Step 507 loads the data structure for the identity field data called "msm_mcd_permissions_mcd_ controls_data_ date." This nomenclature stands for the MULTOS card device controls data date or, in other words, the date on which the card was personalized so that, for example, the application loader can load cards dated only after a certain date, load cards before a certain date (e.g., for application updates) or load

ANNEX A TO THE DESCRIPTION

cards with a particular data date. The information can include the year, month and day of personalization or may include less information, if desired. The data_date data structure is preferably 1 byte in length (see 507A in Fig. 5A) although it could be any length depending upon the needs of the particular card system used.

Once all of the personalization data structures are loaded and stored in the card, the card has been identified by issuer, product class, date and identification number (and other data fields, if desired), and the card cannot change its identity: these fields cannot be changed in the memory of the card. If a card user wants to change the product_id stored in the card to gain access to different applications available to another product type, a new card will have to be issued to the user containing the correct personalization data. This system is consistent with a gold card member receiving a new card when the classification is changed to platinum.

After the card has been enabled and personalized by storing its individual card key set, MSM personalization characteristics and enablement bit as described in Fig. 3, the card is ready to have applications loaded into its memory.

Loading Applications

The application loading process contains a number of security and card configuration checks to ensure the secure and proper loading of an application onto the intended IC card. The application loading process is preferably performed at the personalization bureau so that the card will contain one or more applications when the card is issued. The card may contain certain common applications which will be present on every card the issuer sends out, such as an electronic purse application or a credit/debit

application. Alternatively, the personalization bureau could send the enabled cards to a third party for the process of loading applications. The multiple application operating system stored in the ROM of each card and the card MSM personalization data is designed to allow future loading and deleting of applications after the card has been issued depending upon the desires of the particular card user and the responsible card issuer. Thus, an older version of an application stored on the IC card could be replaced with a new version of the application. An additional loyalty application could also be added to the card after it has been initially sent to the card user because the application is newly available or the user desires to use the new application. These loading and deleting functions for applications can be performed directly by a terminal or may be performed over telephone lines, data lines, a network such as the Internet or any other way of transmitting data between two entities. In the present IC card system, the process of transmitting the application program and data ensures that only IC cards containing the proper personalization data and which fit on application permissions profile will be qualified and receive the corresponding application program and data.

Figure 6 shows the preferred steps performed in loading an application onto an IC card in the MULTOS IC card system. For this example, the personalization bureau is loading an application from a terminal which enabled the same card. Step 601 performs an "open command" initiated by the terminal which previews the card to make sure the card is qualified to accept the loading of a specific application. The open command provides the card with the application's permissions data, the application's size, and instructs the card to determine (1) if the enablement bit is set indicating the card

has been personalized; (2) whether the application code and associated data will fit in the existing memory space on the card; and (3) whether the personalization data assigned to the application to be loaded allows for the loading of the application onto the particular card at issue. The open command could also make additional checks as required by the card system. These checking steps during the open command execution will be described in detail in conjunction with Figure 7.

After the open command has been executed, the application loader via the terminal will be advised if the card contains the proper identification personalization data and if enough room exists in the memory of the card for the application code and related data. If there is insufficient memory, then a negative response is returned by the card and the process is abended (abnormally ended). If the identification personalization data does not match the applications permissions data, a warning response is given in step 603, but the process continues to the load and create steps. Alternatively, if there is no match, the process may automatically be abended. If a positive response is returned by the card to the terminal in step 605, the application loader preferably proceeds to next steps. The open command allows the application to preview the card before starting any transfer of the code and data.

Step 607 then loads the application code and data onto the IC card into EEPROM. The actual loading occurs in conjunction with create step 609 which completes the loading process and enables the application to execute on the IC card after it is loaded. The combination of the open, load and create commands are sent by the terminal, or another application provider source, to the IC card to perform the application

loading process. The operating system in the IC cards is programmed to perform a specific set of instructions with respect to each of these commands so that the IC card will communicate with and properly carry out the instructions from the terminal.

Step 609 performs the create command which at least: (1) checks if an application load certificate is signed (encrypted) by the CA and therefore authenticates the application as a proper application for the system; and (2) checks the card personalization data stored on the card against the permissions profile for the application to be loaded to qualify the card for loading. It may do other checks as required. If one of the checks fails, then a failure response 610 is given and the process aborts. The application after it has passed these checks will be loaded into the memory of the card.

Figure 7 shows the various steps of the open step 601 of Fig. 6 in more detail. Step 701 determines if the enablement (i.e., control) bit is set. This bit is set when the card has completed its personalization process and has been assigned its personalization data. An application can be loaded on an IC card in the card system only if the card contains the personalization data. If the enablement bit is not set, the card has not been personalized and therefore the card returns a negative response 703 to the terminal. If the enablement bit is set, then the card has been enabled and the test conditions continue with step 711.

Step 711 checks if there is sufficient space in the memory on the card to store the application code and its associated data. Applications will typically have associated data related to their functions. This data will be used and manipulated when the application is run. Storage space in the memory of an IC card is a continuing concern

due to the relatively large physical space required for EEPROM and how it fits in the integrated circuit which is desired to be small enough to fit on a credit card sized card. An example of the size of a preset EEPROM on an IC card is 16K bytes although the actual size varies. Applications can range from 1K byte or less for a very simple application up to the size of available memory for a more sophisticated application. The data associated with an application can range from no data being stored in the card memory to a size constrained by the amount of available memory. These varied sizes of application code and data continually increase as applications become more advanced and diverse.

MULTOS as an operating system is not limited by the number of applications and associated data it can store on the card. Thus, if five applications can fit in the available memory of the card, the card user will have greatly increased functionality than if one or two applications were stored on the card. Once a card's memory is filled to its capacity, however, a new application cannot be loaded onto the card unless another application including its code and data of sufficient size can be deleted. Therefore, checking the amount of available space on the card is an important step. If there is not sufficient space, then an insufficient space response 713 will be returned to the terminal. The application loader can then decide if another existing application on the card should be deleted to make room for the new application. Deletion depends upon the card issuer having an application delete certificate from the CA. If there is sufficient space on the card, then the process continues with step 715.

ANNEX A TO THE DESCRIPTION

An example of the testing of memory spaces in step 711 is now described. The numbers used in this example in no way limit the scope of the invention but are used only to illustrate memory space requirements. An IC card may have 16K available EEPROM when it is first manufactured. The operating system data necessary for the operating system may take up 2K of memory space. Thus, 14K would remain. An electronic purse application's code is stored in EEPROM and may take up 8K of memory space. The purse application's required data may take up an additional 4K of memory space in EEPROM. The memory space which is free for other applications would thus be 2K (16K-2K-8K-4K=2K). If a card issuer wants to load a credit/debit application whose code is 6K bytes in size onto the card in this example, the application will not fit in the memory of the IC card. Therefore, the application cannot load the new application without first removing the purse application from the card. If a new credit/debit application was loaded into EEPROM of the IC card, then it would have to overwrite other application's code or data. The application loader is prevented from doing this.

Figure 8 shows the steps performed in determining whether the card's personalization data falls within the permissible set of cards onto which the application at issue may be loaded. These steps are preferably performed during the execution of the "create" command. However, these steps may be performed at any time during the loading or deleting of an application. As described previously, the card is personalized by storing data specific to the card (MSM personalization data) including: a card ID designation specific to an individual card, the card issuer number indicating the issuer of the card, the product type of the card, such as a gold or platinum card, and the date the

ANNEX A TO THE DESCRIPTION

card was personalized. This data uniquely identifies the card apart from all other IC cards in the system.

Accordingly, applications can be selectively stored on individual cards in the IC card system on virtually any basis, including the following. An application can be loaded selectively to cards containing one or more specific card numbers. An application can be selectively loaded on one or more cards containing a specified card issuer ID. Moreover, an application can be loaded only upon one type of product specified by the particular card issuer, and/or the application can be loaded only on cards which have a specified date or series of dates of personalization. Each of the personalization data allows an application to be selectively loaded onto certain cards or groups of cards and also ensures that cards without the proper permissions will not receive the application. Personalization data types in addition to the four described can also be used as needed.

The selection of IC cards upon which a particular application may be loaded is made possible by the use of "applications permissions data" which is assigned to the application and represents at least one set of cards upon which the application may be loaded. The set may be based on virtually any factor, including one or more of the following: card numbers, card issuers, product types or personalization dates. Although the individual card's personalization data typically identify one specific number, one card issuer, one product type and one date, the application's permissions data may indicate a card numbers or a blanket permission, a card issuer or a blanket permission, and a number of product types and dates.

ANNEX A TO THE DESCRIPTION

For example, a frequent loyalty program may be configured to allow its loading and use on cards in different product classes belonging to one card issuer. In addition, the application permissions data may indicate that the loyalty program can be used on gold and platinum product types if the card was issued after May, 1998. Thus, the MSM permissions check will determine if the card's individual personalization data is included in the allowed or permissible set of cards upon which the application may be loaded. If it is, the application will be loaded.

To expedite the comparison process, an alternative embodiment may include setting one or more permissions data at zero representing a blanket permission for that particular data. For instance, by placing a zero for the "card number" entry in the application permissions data or some other value indicating that all cards may be loaded regardless of their number, the system knows not to deny any cards based on their card number. Moreover, if a zero is placed in the application's permissions data "issuer ID," then all cards similarly will pass the "issuer" test comparison. This feature allows greater flexibility in selecting groups of cards. The zero indicator could also be used for other permissions data, as required.

Referring to Figure 8, each of the permissions data is checked in the order shown, but other orders could be followed because if any one of the permissions fails, the application will be prevented from being loaded on the IC card being checked. The permissions are preferably checked in the order shown. Step 801 checks if the application permissions product type set encompasses the card's product type number stored in the memory of the card. Each card product type is assigned a number by the

system operator. The product types are specified for each card issuer because different card issuers will have different product types. The cards are selectively checked to ensure that applications are loaded only on cards of authorized product type. The application permissions product type set can be 32 bytes long which includes multiple acceptable product types or can be a different length depending upon the needs of the system. Using data structure 505A as an example, the operating system would check bit number 2 in the 256 bit array (32 bytes x 8 bits per byte) resulting from the 32 byte long application permissions data structure. If the permissions check fails, then the card returns a failure message to the terminal in step 803. If the product type check passes (for example, the value of bit no. 2 being 1), then the process continues with step 805.

Step 805 checks if the application permissions allowable card issuer number set encompasses the card's issuer number stored in the memory of the card or if the application permissions issuer data is zero (indicating all cards pass this individual permissions check). Each card issuer is assigned a number by the system operator and the cards are selectively checked to ensure that applications are loaded only on cards distributed by authorized card issuers. The application permissions card issuer number set can be 4 bytes long if one issuer is designated or can be longer depending upon the needs of the system. If the issuer check fails, then the card returns a failure message to the terminal in step 807. If the check passes, then the process continues with step 809.

Step 809 checks if the application permissions date set encompasses the card's data date stored in the memory of the card. The date that the IC card was personalized will be stored and will preferably include at least the month and year. The

cards are selectively checked to ensure that applications are loaded only on cards with the authorized personalization date. The application permissions date set can be 32 bytes long which includes multiple dates or can be a different length depending upon the needs of the system. If the date permissions check fails, then the card returns a failure message to the terminal in step 811. If the date check passes, then the process continues with step 813.

Step 813 checks if the application permissions allowable card number set encompasses the card's ID number stored in the card memory or if the application permissions allowable card number data is zero (indicating all cards pass this individual permissions check). The testing of the permissions is performed on the card during the execution of the open, load and create commands. The application permissions card number data set can be 8 bytes long if one number is designated or can be longer depending upon the needs of the system. If the card number check fails, then the card returns a failure message to the terminal in step 815. If the check passes, then the process continues with step 817.

Summary of IC Card System's Process

Figure 9 shows the components of the system architecture for the card initialization process of an IC card in a secure multiple application IC card system. The system includes a card manufacturer 102, a personalization bureau 104, an application loader 106, the IC card 107 being initialized, the card user 109 and the certification authority 111 for the entire multiple application secure system. The card user 131 is the

ANNEX A TO THE DESCRIPTION

person or entity who will use the stored applications on the IC card. For example, a card user may prefer an IC card that contains both an electronic purse containing electronic cash (such as MONDEX™) and a credit/debit application (such as the MasterCard® EMV application) on the same IC card. The following is a description of one way in which the card user would obtain an IC card containing the desired applications in a secure manner.

The card user would contact a card issuer 113, such as a bank which distributes IC cards, and request an IC card with the two applications both residing in memory of a single IC card. The integrated circuit chip for the IC card would be manufactured by manufacturer 102 and sent to the card issuer 113 (or an entity acting on its behalf) in the form of an IC chip on a card. As discussed above (see steps 201-209), during the manufacturing process, data is transmitted 115 via a data conduit from the manufacturer 102 to card 107 and stored in IC card 107's memory. (Any of the data conduits described in this figure could be a telephone line, Internet connection or any other transmission medium.) The certification authority 111, which maintains encryption/decryption keys for the entire system, transmits 117 security data (i.e., global public key) to the manufacturer over a data conduit which is placed on the card by the manufacturer along with other data, such as the card enablement key and card identifier. The card's multiple application operating system is also stored in ROM and placed on the card by the manufacturer. After the cards have been initially processed, they are sent to the card issuer for personalization and application loading.

ANNEX A TO THE DESCRIPTION

The card issuer 113 performs, or has performed by another entity, two separate functions. First, the personalization bureau 104 personalizes the IC card 107 in the ways described above, and second, the application loader 106 loads the application provided the card is qualified, as described.

Regarding personalization, an individualized card key set is generated by the CA and stored on the card (see Fig. 3). The card is further given a specific identity using MSM personalization (see Fig. 3, step 307 and Fig. 5) including a card ID number, an issuer ID number identifying the card issuer which processed the card, a card product type number which is specified by the card issuer and the date upon which the personalization took place. After the card has been personalized, applications need to be loaded onto the card so that the card can perform desired functions.

The application loader 106, which could use the same terminal or data conduit as personalization bureau 104, first needs to have determined if the card is qualified to accept the application. This comparison process takes place on the card itself (as instructed by its operating system) using the permissions information. The card, if it is qualified, thus selectively loads the application onto itself based upon the card's identity and the card issuer's instructions. The application loader communicates 119 with the IC card via a terminal or by some other data conduit. After the applications have been loaded on the card, the card is delivered to the card user 109 for use.

The secure multiple application IC card system described herein allows for selective loading and deleting of applications at any point in the life cycle of the IC card after the card has been personalized. Thus, a card user could also receive a personalized

ANNEX A TO THE DESCRIPTION

card with no applications and then select a desired application over a common transmission line such as a telephone line or Internet connection.

Figure 10 is a system diagram of entities involved with the use of an IC card once it has been personalized. The system includes an IC card 151, a terminal 153, an application load/delete entity 155, the certification authority 157, a card issuer 171 and other IC cards 159 in the system. The arrows indicate communication between the respective entities. The CA 157 facilitates loading and deleting of applications. After providing the MSM permissions data and card specific keyset to the card during card enablements, the CA allows applications to be later loaded and deleted preferably by issuing an application certificate. Application specific keys are required to authenticate communication between a card and terminal. The IC card 151 also can communicate with other IC cards 159. Card issuer 171 is involved with all decisions of loading and deleting applications for a card which it issued. All communications are authenticated and transmitted securely in the system.

For instance, IC card 151 will use the following procedure to load a new application onto the card. IC card 101 is connected to terminal 153 and the terminal requests that an application be loaded. Terminal 153 contacts application load/delete entity 155 which, as a result and in conjunction with card issuer 171, sends the application code, data and application permissions data (along with any other necessary data) to terminal 153. Terminal 153 then queries card 151 to ensure it is the correct card onto which the application may be loaded. If IC card passes the checks discussed above, the application is loaded onto card 151. The CA 157 provides the application load or

ANNEX A TO THE DESCRIPTION

delete certificate that enables the application to be loaded or deleted from the card. This example shows one way to load the application, but other variations using the same principles could be performed, such as directly loading the application at the application load/delete entity 155.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

For example, it will be appreciated that the MSM personalization and permissions data may not only be used for loading applications onto IC cards but also for deleting applications from said cards. The same checks involving MSM permissions and loading applications are made for deleting applications. A delete certificate from the CA authorizing the deletion of an application will control from which cards the application may be deleted. This is accomplished through the personalization data stored on each IC card and the permissions check as described herein.

Moreover, the data may also be applicable to personal computers or other units onto which applications may be loaded which are not physically loaded on cards. In addition, the application's permissions data may actually include data representative of a set or sets of cards to be excluded, instead of included -- cards that cannot be loaded with the application.

ANNEX A TO THE DESCRIPTION

The scope of the present disclosure includes any novel feature or combination of features disclosed therein either explicitly or implicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or mitigates any or all of the problems addressed by the present invention. The applicant hereby gives notice that new claims may be formulated to such features during the prosecution of this application or of any such further application derived therefrom. In particular, with reference to the appended claims, features from dependent claims may be combined with those of the independent claims in any appropriate manner and not merely in the specific combinations enumerated in the claims.

ANNEX A TO THE DESCRIPTION

CLAIMS:

1. An IC card system comprising at least one IC card, an application to be loaded onto said card and means for determining whether said card is qualified to accept the loading of said application onto said card.

2. The IC card system of claim 1, wherein said IC card contains card personalization data, and said application is assigned application permissions data representing at least one set of IC cards upon which said application may be loaded.

3. The IC card system of claim 2, wherein said determining means compares said card personalization data with said application permissions data.

4. The IC card system of claim 3, wherein whether said application is loaded onto said IC card depends on the result of said comparison, such that in the event the card personalization data matches said permissions data set the card is qualified and the application is loaded.

5. The IC card system of any of claims 2 to claim 4, wherein said personalization data comprises data representative of a unique card identification designation.

ANNEX A TO THE DECRIPTION

6. The IC card system of any of claims 2 to claim 5, wherein said personalization data comprises data representative of a card issuer.

7. The IC card system of any of claims 2 to claim 6, wherein said personalization data comprises data representative of a product class.

8. The IC card system of any of claims 2 to claim 7, wherein said personalization data comprises data representative of a date.

9. An IC card system comprising at least one IC card and an application, wherein said IC card contains personalization data representative of that card and said application is assigned a permissions data set representing at least one IC card upon which said application may be loaded, said system further comprising means for determining whether said personalization data falls within said permissions data set.

10. The IC card system of claim 9 wherein said application is loaded onto said IC card in the event said determining means determines that said personalization data falls within said set.

11. The IC card system of claim 9 or claim 10 wherein said personalization data comprises data representing a card identification designation, and an issuer of said card.

ANNEX A TO THE DESCRIPTION

12. The IC card system of any of claims 9 to claim 11 wherein said personalization data comprises data representing a product class and a date.

13. The IC card system of any of claims 9 to 12 wherein said permissions data set includes a plurality of card identification designations.

14. The IC card system of any of claims 9 to 13 wherein said permissions data set includes one or more issuers of IC cards.

15. The IC card system of any of claims 9 to 14 wherein said permissions data set includes one or more product classes.

16. The IC card system of any of claims 9 to 15 wherein said permissions data set includes a plurality range of dates.

17. The IC card system of any of claims 9 to 16 wherein said permissions data set includes all IC cards which attempt to load the application.

18. An IC card system comprising at least one IC card, an application to be loaded onto said card and means for enabling said card to be loaded with said application.

19. The IC card system of claim 18 wherein said enabling means comprises means for storing personalization data onto said card.

20. The IC card system of claim 18 wherein said enabling means comprises means for setting an enablement bit.

21. The IC card system of claim 19 wherein said enabling means comprises means for setting an enablement bit.

22. The IC card system of claim 20 further comprising means for checking the enablement bit prior to enabling said IC card to determine whether or not said card has already been enabled.

23. The IC card system of claim 21 further comprising means for checking the enablement bit prior to enabling said IC card to determine whether or not said card has already been enabled.

24. A process for loading an application onto an IC card comprising the step of determining whether said IC card is qualified to accept the loading of said application onto said card.

ANNEX A TO THE DESCRIPTION

25. The process of claim 24 wherein said determining step includes the steps of: providing said card with personalization data;

assigning to said application permissions data representing at least one set of IC cards upon which said application may be loaded;

comparing said personalization data with said permissions data; and

loading said application onto said IC card provided said personalization data falls within said set of cards upon which said application may be loaded.

26. The process of claim 25, wherein said personalization data comprises data representative of a card identification designation.

27. The process of claim 25 or claim 26, wherein said personalization data comprises data representative of a card issuer.

28. The process of any of claims 25 to claim 27, wherein said personalization data comprises data representative of a product class.

29. The process of any of claims 25 to claim 28. wherein said personalization data comprises data representative of a date.

ANNEX A TO THE DESCRIPTION

30. The process of any of claims 25 to claim 29 further comprising the first step of enabling said card to be loaded with said application.

31. The process of claim 30 wherein said enabling step includes the step of storing personalization data onto said card.

32. The process of claim 30 wherein said enabling step includes the step of setting an enablement bit indicating that the card has been enabled.

33. The process of claim 31 wherein said enabling step further includes the step of setting an enablement bit indicating that the card has been enabled.

34. The process of claim 32 wherein prior to said enabling step a checking step is performed to determine whether said card has been enabled.

35. The process of claim 33 wherein prior to said enabling step a checking step is performed to determine whether said card has been enabled.

36. A process for deleting an application from an IC card comprising the step of determining whether said IC card is qualified to delete said application based upon permissions data associated with said application.

ANNEX A TO THE DESCRIPTION

37. The process of claim 36 wherein said determining step includes the steps of:

providing said card with personalization data;

assigning to said application permissions data representing at least one set of IC cards from which said application may be deleted;

comparing said personalization data with said permissions data; and

deleting said application from said IC card provided said personalization data falls within said set of cards from which said application may be deleted.

38. The process of claim 37, wherein said personalization data comprises data representative of a card identification designation.

39. The process of claim 37 or claim 38, wherein said personalization data comprises data representative of a card issuer.

40. The process of any of claims 37 to claim 39, wherein said personalization data comprises data representative of a product class.

41. The process of any of claims 37 to claim 40, wherein said personalization data further comprises data representative of a date.

ANNEX A TO THE DESCRIPTION

42. An IC card system comprising at least one IC card, an application to be deleted from said card and means for determining whether said card is qualified to delete said application from said card.

43. The IC card system of claim 42, wherein said IC card contains card personalization data, and said application is assigned application permissions data set representing at least one set of IC cards from which said application may be deleted.

44. The IC card system of claim 43, wherein said determining means compares said card personalization data with said application permissions data.

45. The IC card system of claim 44, wherein whether said application is deleted from said IC card depends on the result of said comparison, such that in the event the card personalization data matches said permissions data set the card is qualified and the application is deleted.

ABSTRACT ANNEX A TO THE DESCRIPTION

Multi-Application IC Card System

A multi-application IC card system is disclosed having selective application loading and deleting capability. Prior to loading an application onto an IC card a test is conducted to determine if the card is qualified to receive the application using personalization data stored on the card and comparing it with permissions data associated with the application indicating one or more sets of cards upon which the application may be loaded. If the personalization data of the card falls within the allowable set of permissions for that application then the card may be loaded with the application. Preferably, the personalization data includes data representative of the card number, issuer, a product class and the date on which the card is personalized.

WE CLAIM:

1. A multiple application card system comprising:

a certification authority for which a public and private key pair are generated;

at least one integrated circuit card including at manufacture said public key of said certification authority and a card identifier for uniquely identifying each said card;

means for creating at said certification authority a personalization data block for at least one card identifier, means for encrypting said personalization data block and forwarding said encrypted data block to a personalization bureau;

means for loading at said personalization bureau said encrypted data block on said card having the card identifier matching said encrypted personalization data block;

means for determining based at least on said encrypted personalization data block whether one of said integrated circuit cards is qualified to accept the loading of a specific application;

means for authenticating said application for loading onto said card by using said public key of said certification authority; and

loading means responsive to said determining and authenticating means for securely loading said application onto said card.

2. The system of claim 1, further comprising personalization means for enabling at least one of said cards at said personalization bureau.

3. The system of claim 1 or claim 2, wherein said at least one integrated circuit card further comprises memory means for storing an operating system for instructing said determining means, authentication means and said loading means.

4. The system of any of claims 1 to 3 wherein said at least one integrated circuit card further comprises a card enablement key for facilitating card specific confidentiality.

5. The system of claim 2 or any preceding claim dependent on claim 2 wherein said personalization means comprises means for compiling a list of said card identifiers and means for forwarding said list to said authority.

6. The system of any of claims 1 to claim 5 wherein said personalization data block comprises card personalization data and an individual key set.

7. The system of any preceding claim dependent on claim 4 claim 6 further including means for checking whether said card enablement key has been set, and wherein said means for loading said encrypted data block only loads said block in the event said enablement key has not been set, and wherein

said card enablement key is set upon loading said encrypted data block.

8. A multiple application card system comprising:

one or more integrated circuit cards each including at manufacture a public key for authenticating the source of any message to it from an authority holding a corresponding secret key, a card enablement key for facilitating card specific confidentiality, a card identifier for uniquely identifying each card, and memory storing an operating system;

personalization means for enabling said card at a personalization bureau, said personalization means including means for compiling a list of said card identifiers and means for forwarding said list to said authority;

means for creating at said authority a personalization data block for each card identifier forwarded to said authority, said data block including card personalization data and an individual key set for each of said cards;

means for encrypting each of said data blocks and means for forwarding said encrypted data blocks to said personalization bureau;

means for checking whether said card enablement key has been set and, if not, for matching said card identifiers with said encrypted data blocks, loading said encrypted data block on its matched corresponding card, and setting said enablement key;

means for determining whether said card is qualified to accept the loading of a specific application; checking means for authenticating said specific application to be loaded by checking whether said application has been signed by said authority; and

means responsive to said determining and checking means for loading said one or more specific applications.

9. A method for loading one or more applications on an integrated circuit card comprising the steps of:

transmitting security data including a public key of a certification authority onto an integrated circuit card;

creating at said certification authority a personalization data block for said card, encrypting said data block and forwarding said encrypted data block to a personalization bureau;

loading said encrypted data block onto said card;

determining based at least on said encrypted data block whether said card is qualified to accept the loading of a specific application;

authenticating said application for loading onto said card by using said public key;

loading said application in the event said card is qualified and said application is authenticated.

10. A method for deleting one or more applications from an integrated circuit card comprising the steps of:

transmitting security data including a public key of a certification authority onto an integrated circuit card;

creating at said certification authority a personalization data block for said card, encrypting said data block and forwarding said encrypted data block to a personalization bureau;

loading said encrypted data block onto said card;

determining based at least on said encrypted data block whether said card is qualified to accept the deleting of a specific application;

deleting said application in the event said card is qualified.

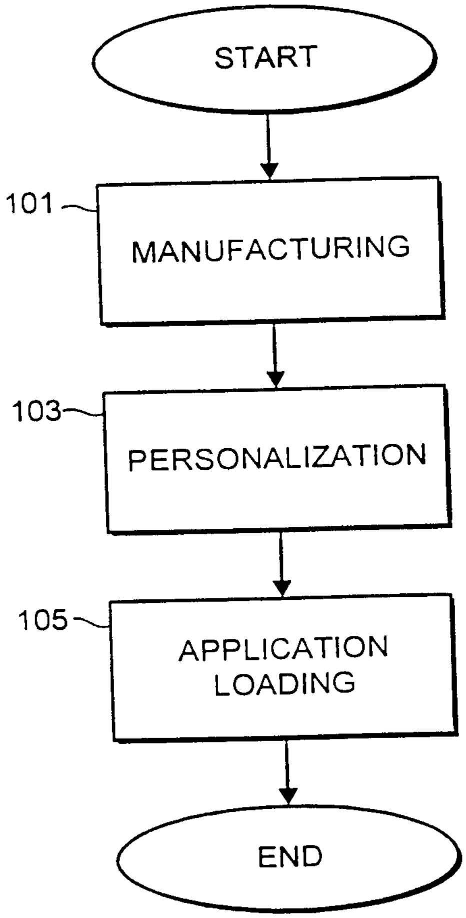

START
101
MANUFACTURING
103
PERSONALIZATION
105
APPLICATION LOADING
END

FIG. 1

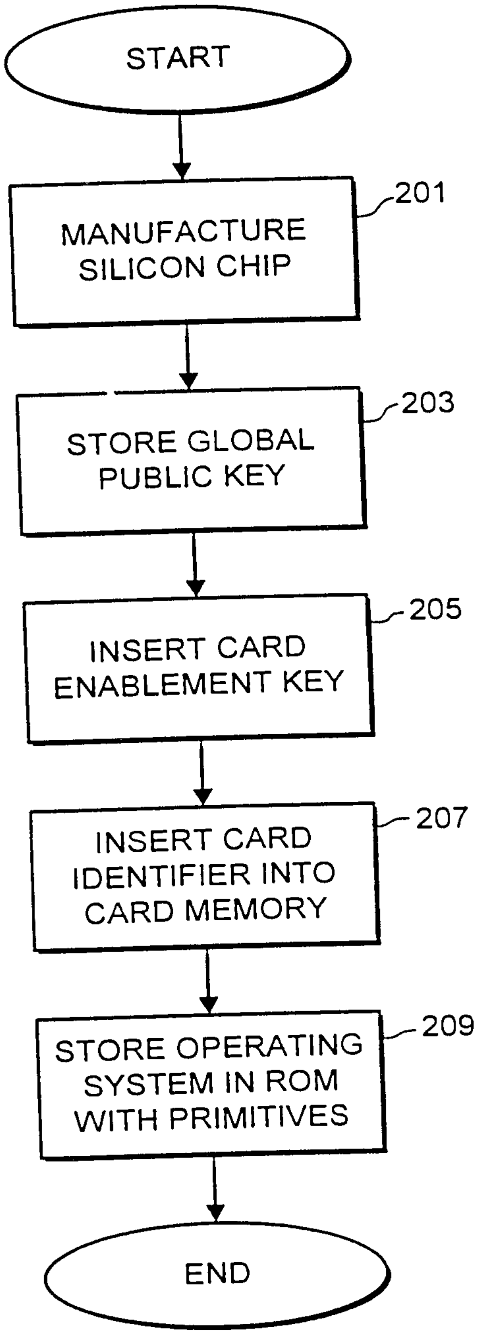

START
MANUFACTURE SILICON CHIP
201
STORE GLOBAL PUBLIC KEY
203
INSERT CARD ENABLEMENT KEY
205
INSERT CARD IDENTIFIER INTO CARD MEMORY
207
STORE OPERATING SYSTEM IN ROM WITH PRIMITIVES
209
END

FIG. 2

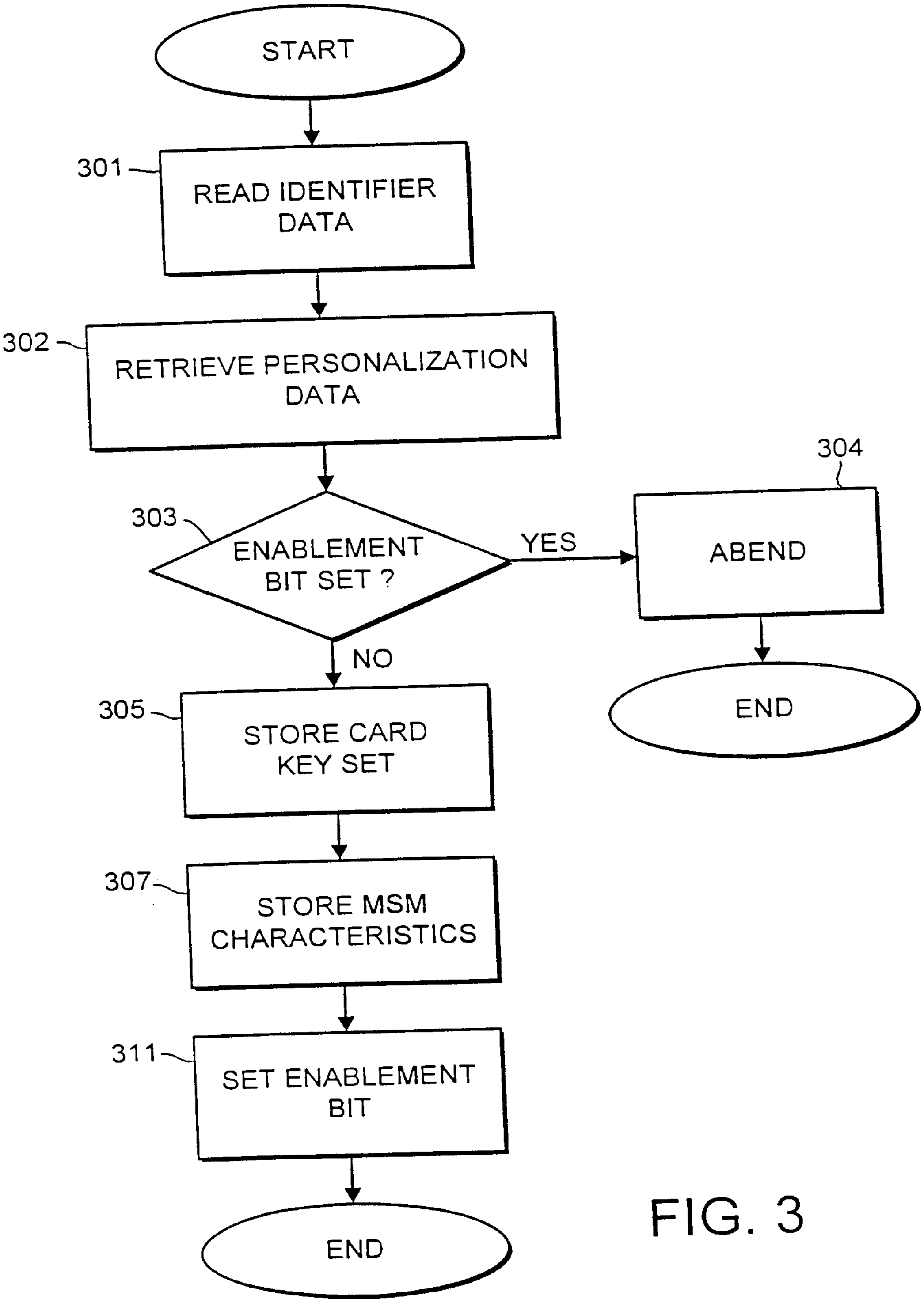

START
301
READ IDENTIFIER DATA
302
RETRIEVE PERSONALIZATION DATA
303
ENABLEMENT BIT SET ?
YES
304
ABEND
END
NO
305
STORE CARD KEY SET
307
STORE MSM CHARACTERISTICS
311
SET ENABLEMENT BIT
END

FIG. 3

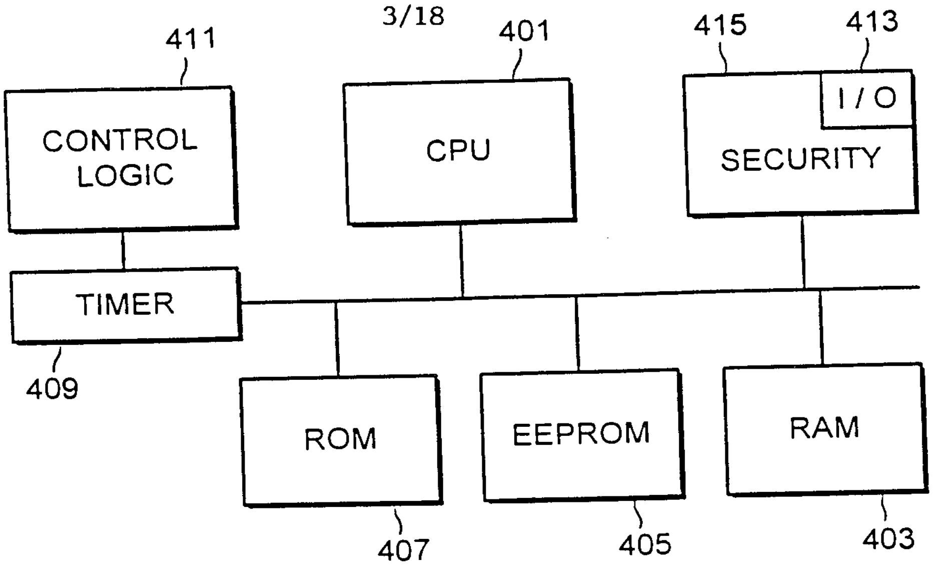

411
CONTROL LOGIC
401
CPU
415
413
I / O
SECURITY
TIMER
409
ROM
407
EEPROM
405
RAM
403

FIG. 4

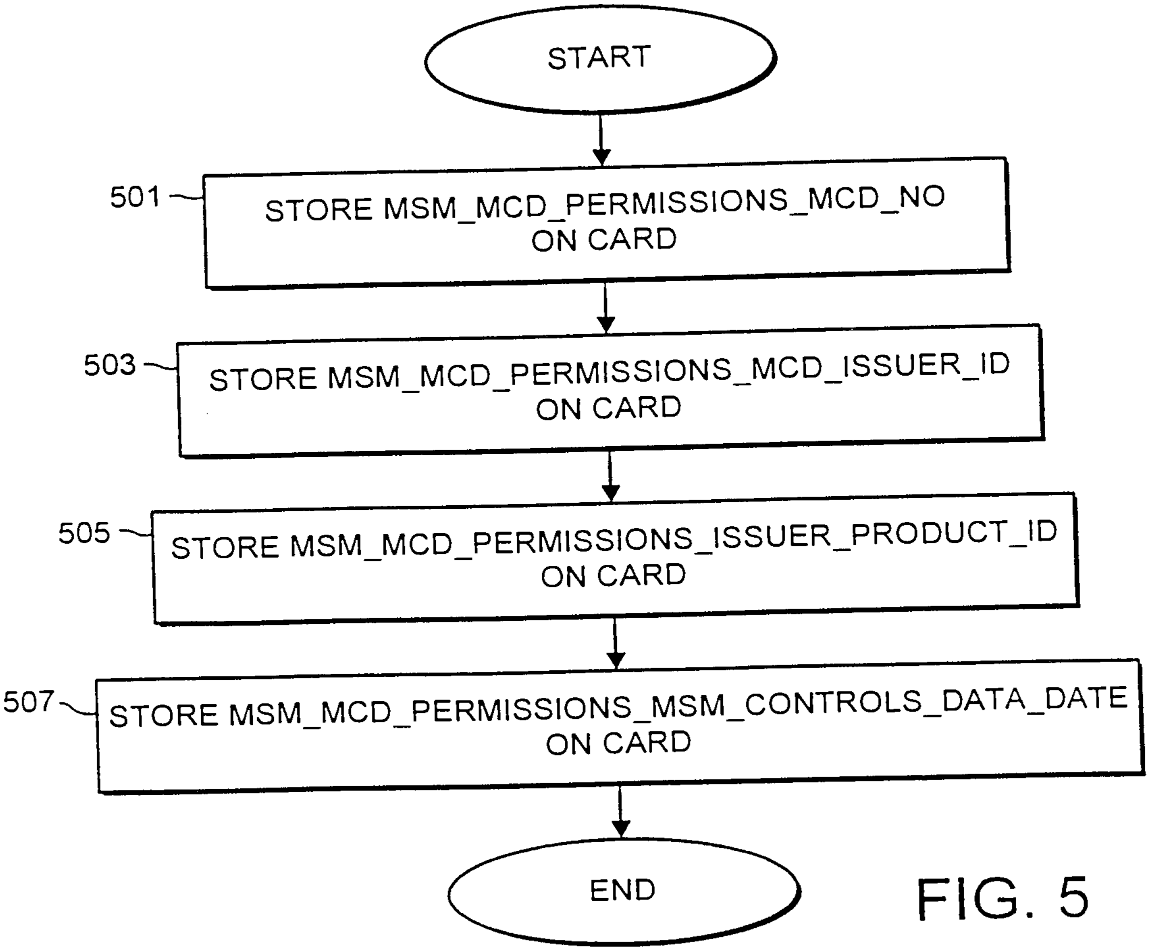

START
501
STORE MSM_MCD_PERMISSIONS_MCD_NO ON CARD
503
STORE MSM_MCD_PERMISSIONS_MCD_ISSUER_ID ON CARD
505
STORE MSM_MCD_PERMISSIONS_ISSUER_PRODUCT_ID ON CARD
507
STORE MSM_MCD_PERMISSIONS_MSM_CONTROLS_DATA_DATE ON CARD
END

FIG. 5

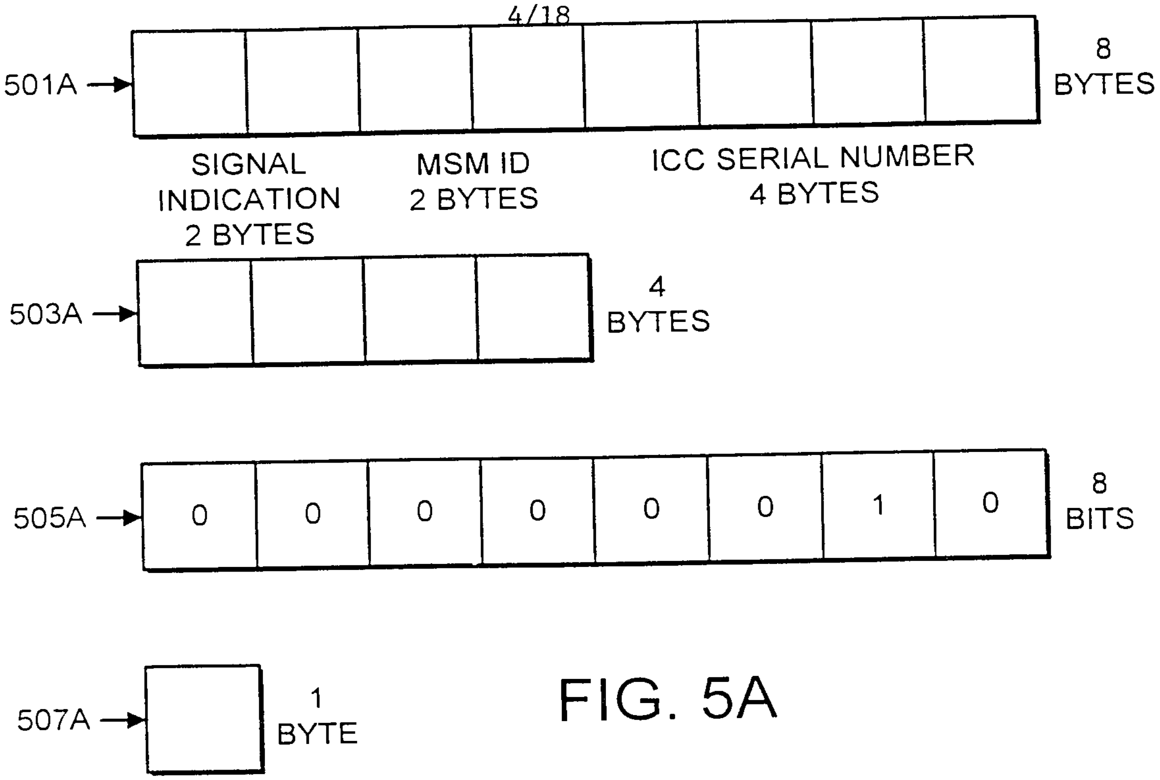

501A
8 BYTES
SIGNAL INDICATION 2 BYTES
MSM ID 2 BYTES
ICC SERIAL NUMBER 4 BYTES
503A
4 BYTES
505A
0
0
0
0
0
0
1
0
8 BITS
507A
1 BYTE

FIG. 5A

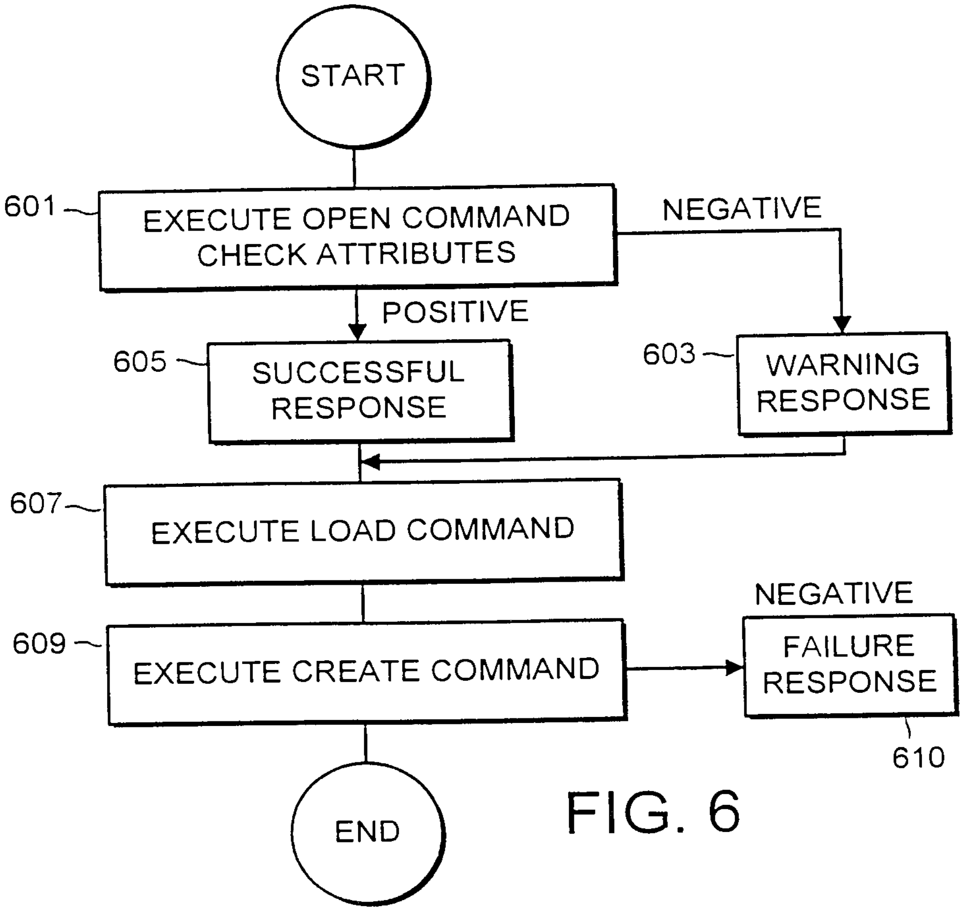

START
601
EXECUTE OPEN COMMAND CHECK ATTRIBUTES
NEGATIVE
POSITIVE
605
SUCCESSFUL RESPONSE
603
WARNING RESPONSE
607
EXECUTE LOAD COMMAND
NEGATIVE
609
EXECUTE CREATE COMMAND
FAILURE RESPONSE
610
END

FIG. 6

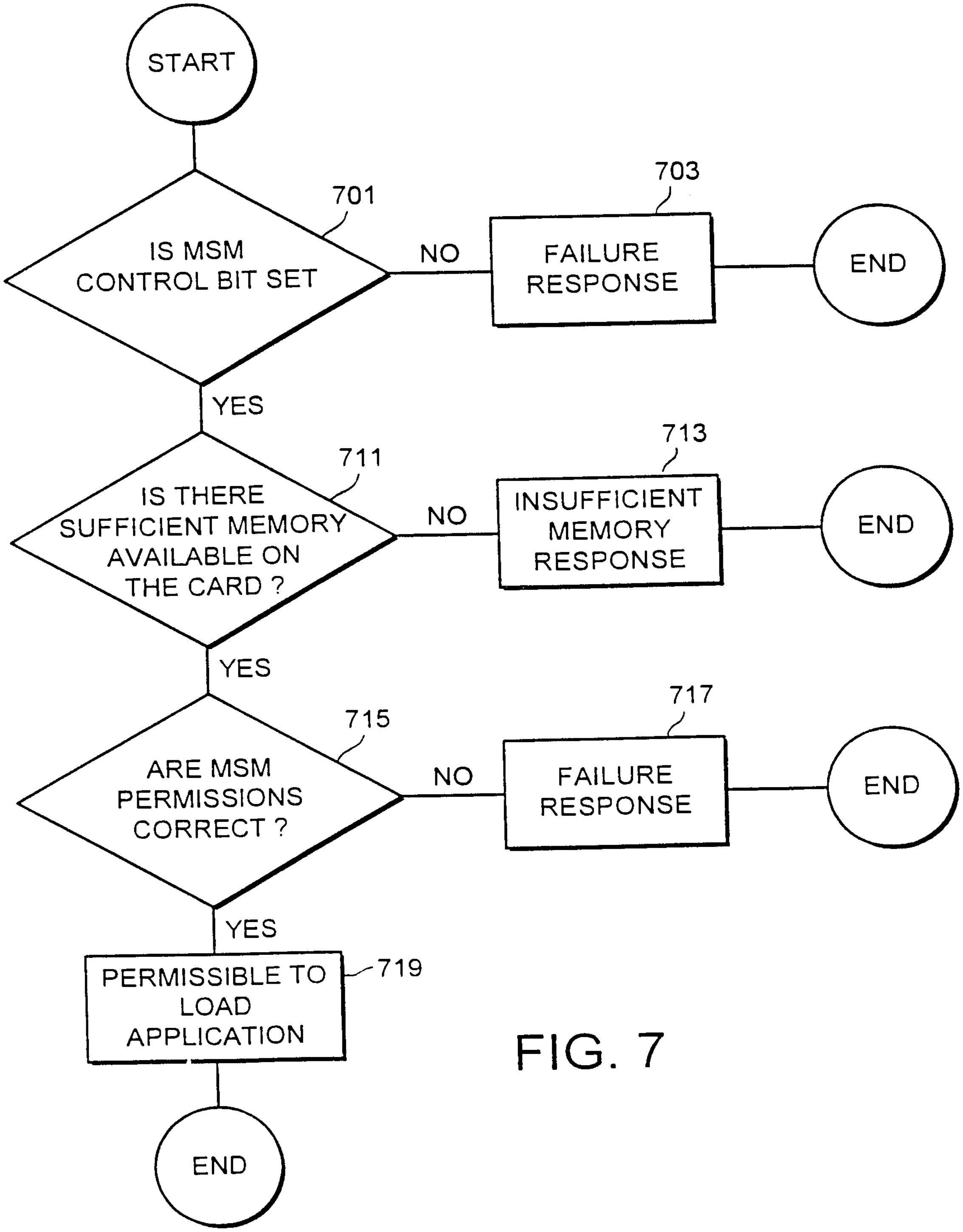

START
IS MSM CONTROL BIT SET
701
NO
FAILURE RESPONSE
703
END
YES
IS THERE SUFFICIENT MEMORY AVAILABLE ON THE CARD ?
711
NO
INSUFFICIENT MEMORY RESPONSE
713
END
YES
ARE MSM PERMISSIONS CORRECT ?
715
NO
FAILURE RESPONSE
717
END
YES
PERMISSIBLE TO LOAD APPLICATION
719
END

FIG. 7

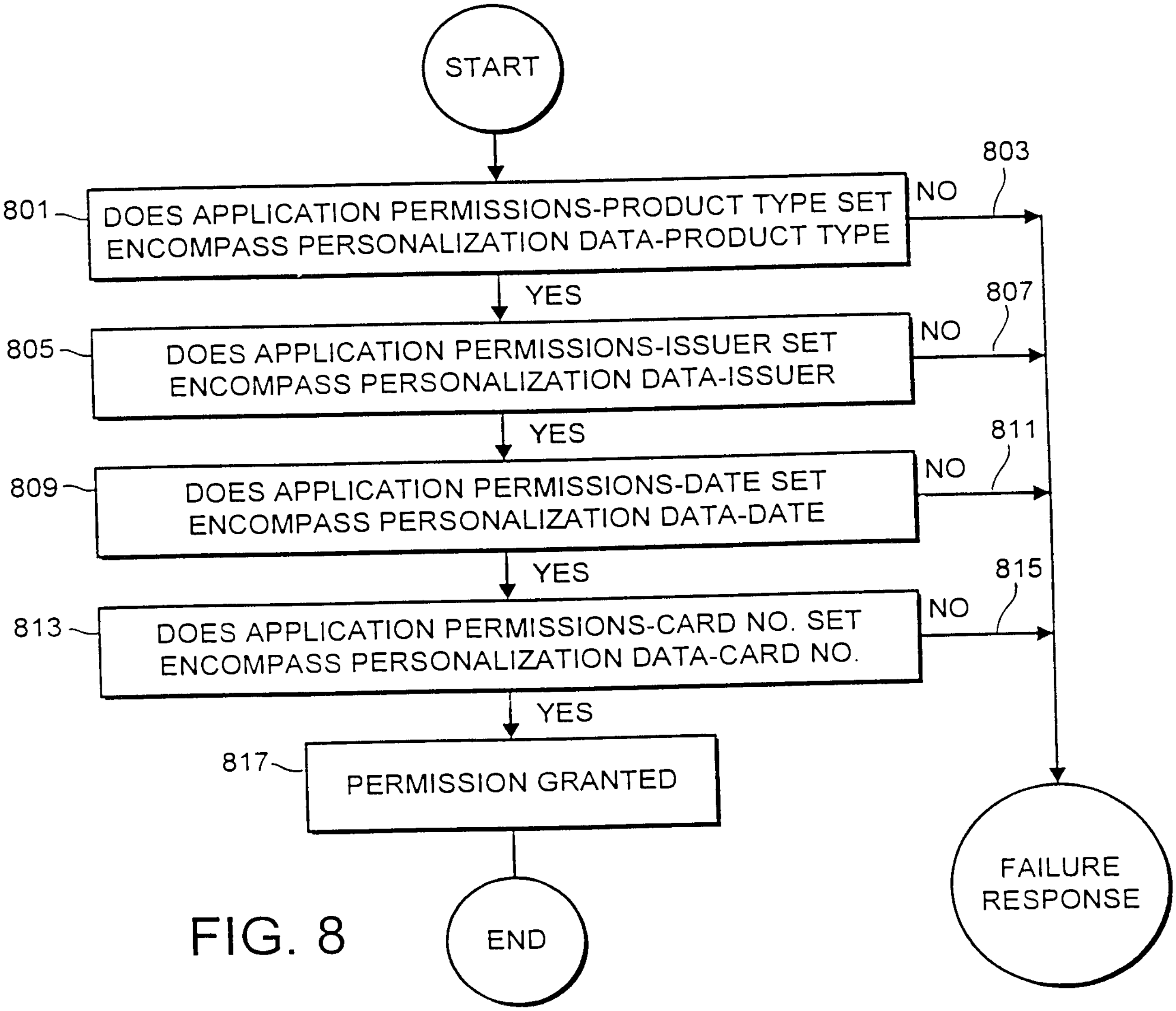

START
801
DOES APPLICATION PERMISSIONS-PRODUCT TYPE SET ENCOMPASS PERSONALIZATION DATA-PRODUCT TYPE
NO
803
YES
805
DOES APPLICATION PERMISSIONS-ISSUER SET ENCOMPASS PERSONALIZATION DATA-ISSUER
NO
807
YES
809
DOES APPLICATION PERMISSIONS-DATE SET ENCOMPASS PERSONALIZATION DATA-DATE
NO
811
YES
813
DOES APPLICATION PERMISSIONS-CARD NO. SET ENCOMPASS PERSONALIZATION DATA-CARD NO.
NO
815
YES
817
PERMISSION GRANTED
END
FAILURE RESPONSE

FIG. 8

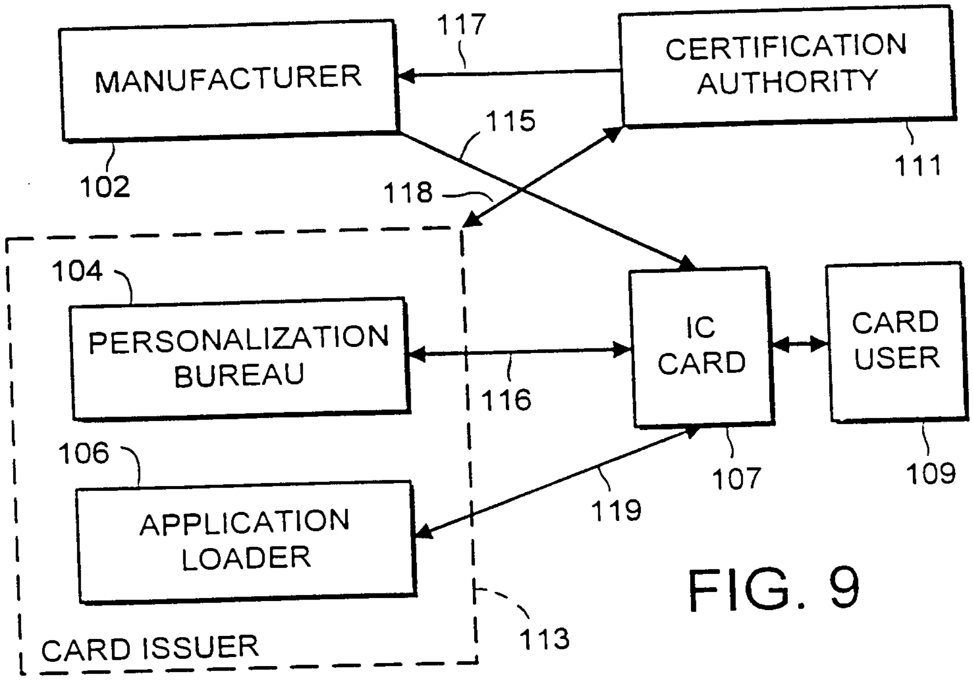

MANUFACTURER
102
117
CERTIFICATION AUTHORITY
111
115
118
104
PERSONALIZATION BUREAU
116
IC CARD
107
CARD USER
109
106
APPLICATION LOADER
119
CARD ISSUER
113

FIG. 9

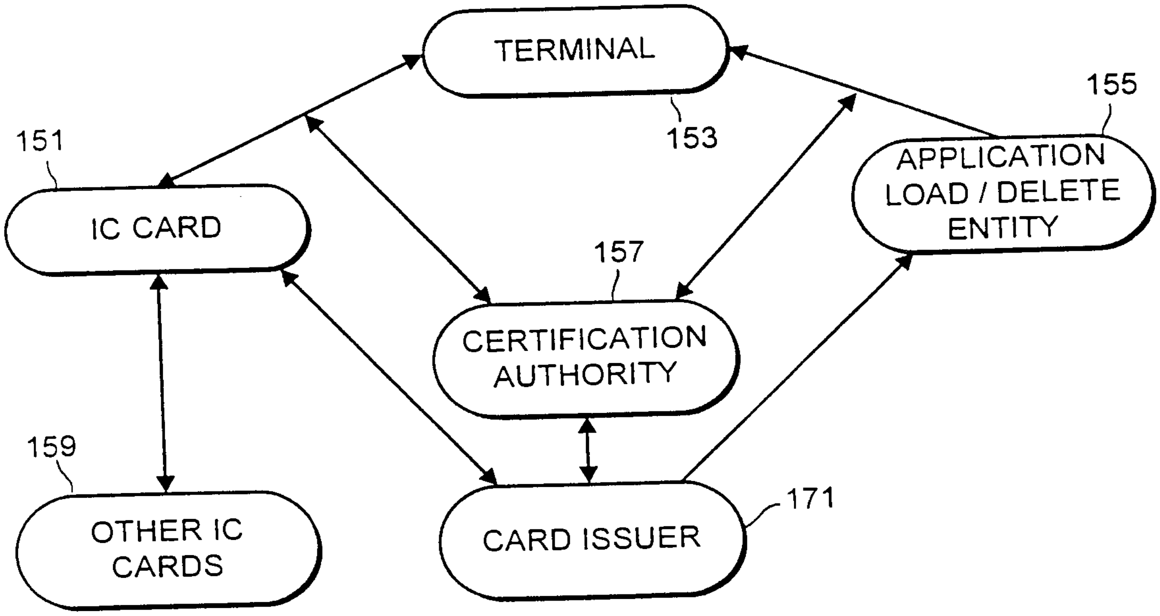

TERMINAL
153
151
IC CARD
155
APPLICATION LOAD / DELETE ENTITY
157
CERTIFICATION AUTHORITY
159
OTHER IC CARDS
CARD ISSUER
171

FIG. 10

ANNEX A TO THE DRAWINGS

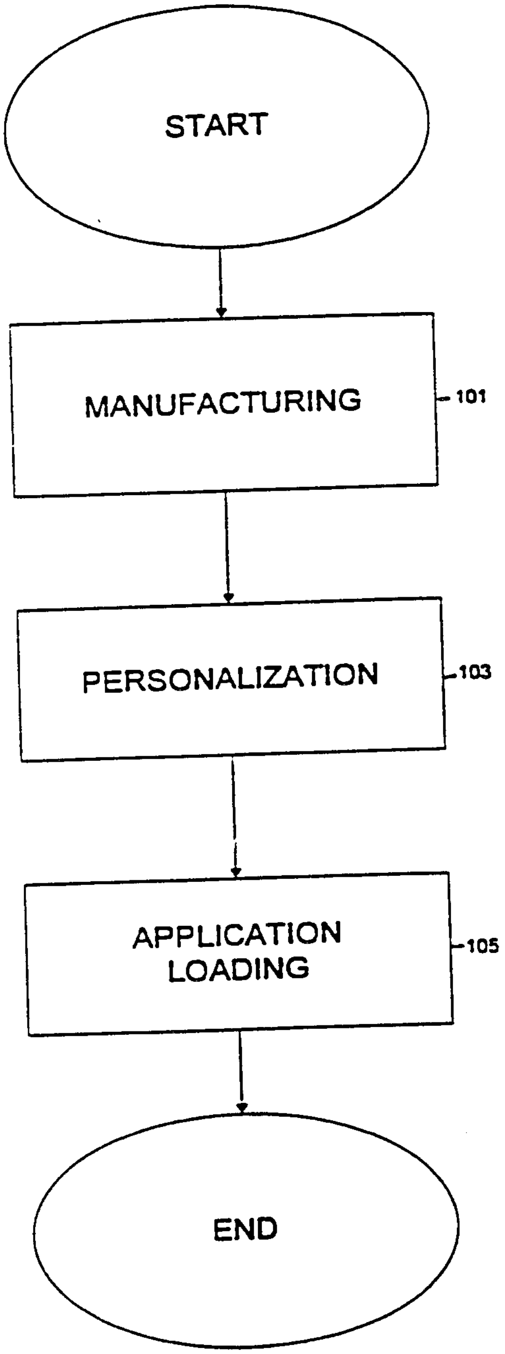


FIG. 1

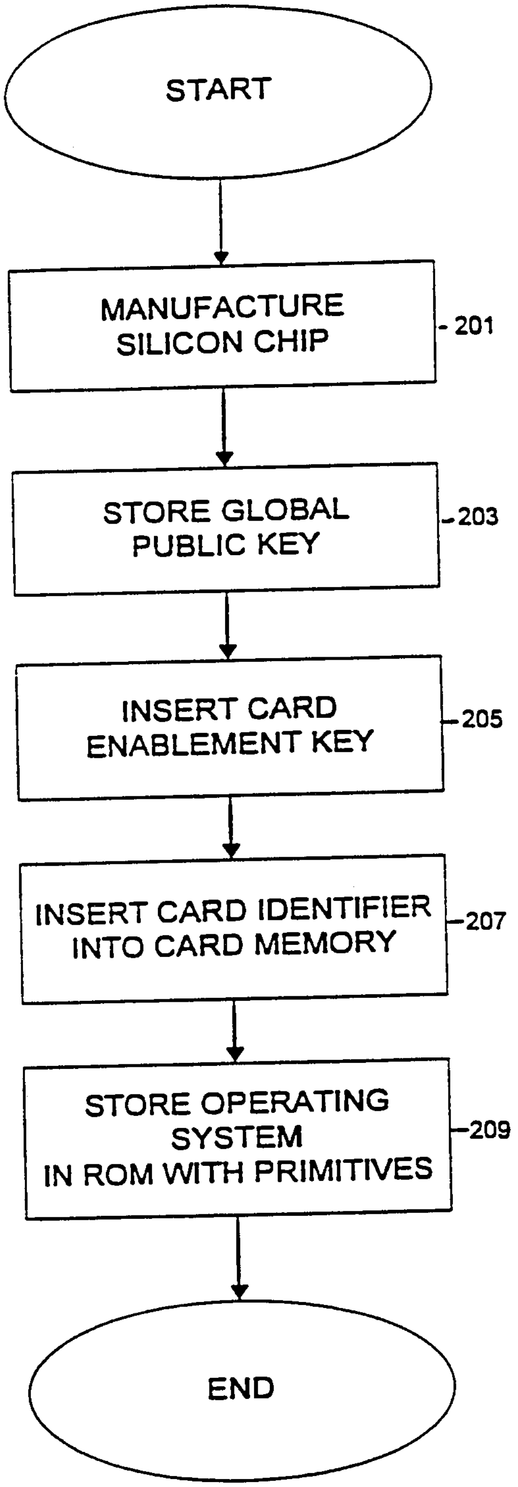

START
MANUFACTURE SILICON CHIP
201
STORE GLOBAL PUBLIC KEY
203
INSERT CARD ENABLEMENT KEY
205
INSERT CARD IDENTIFIER INTO CARD MEMORY
207
STORE OPERATING SYSTEM IN ROM WITH PRIMITIVES
209
END

FIG. 2

ANNEX A TO THE DRAWINGS

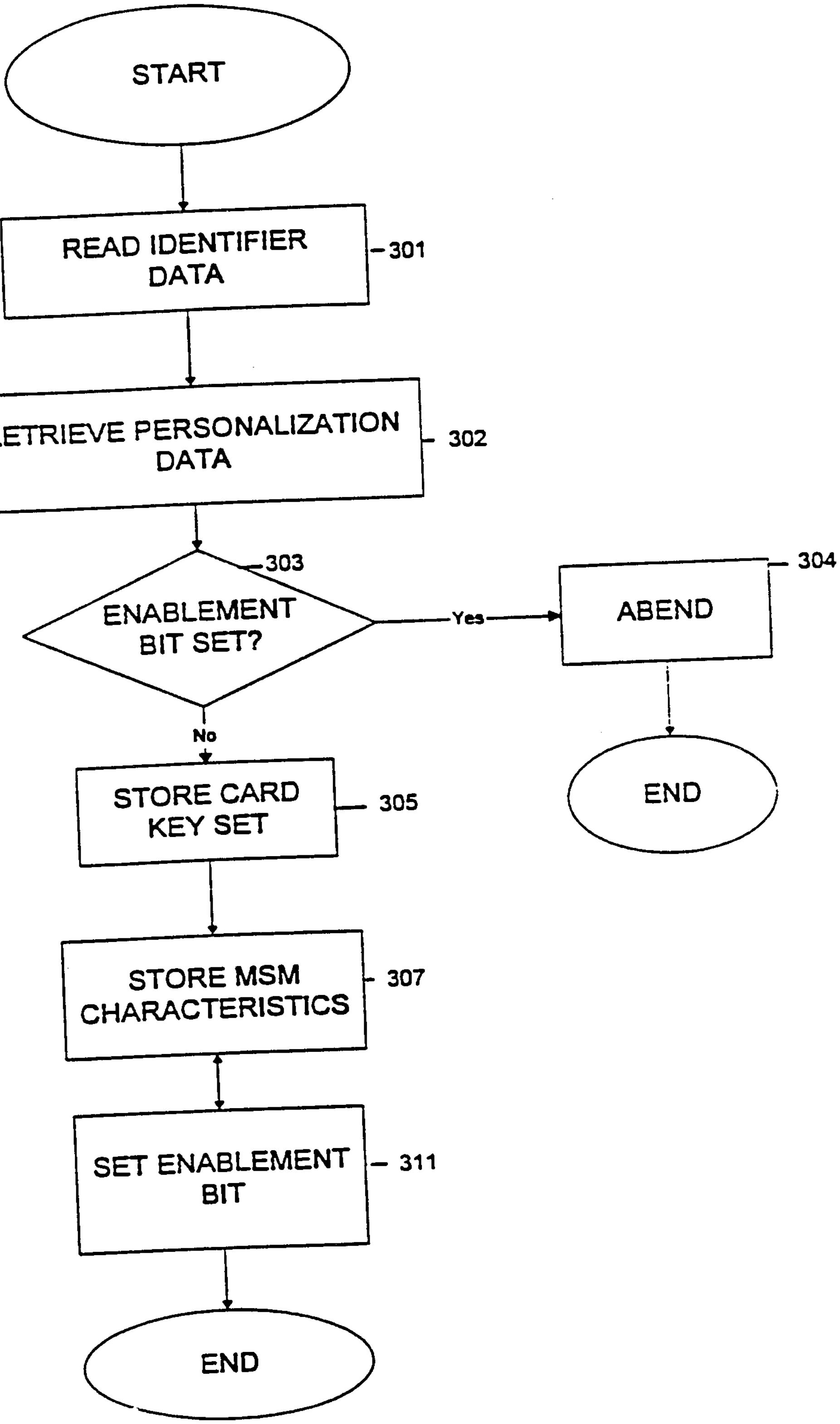


FIG. 3

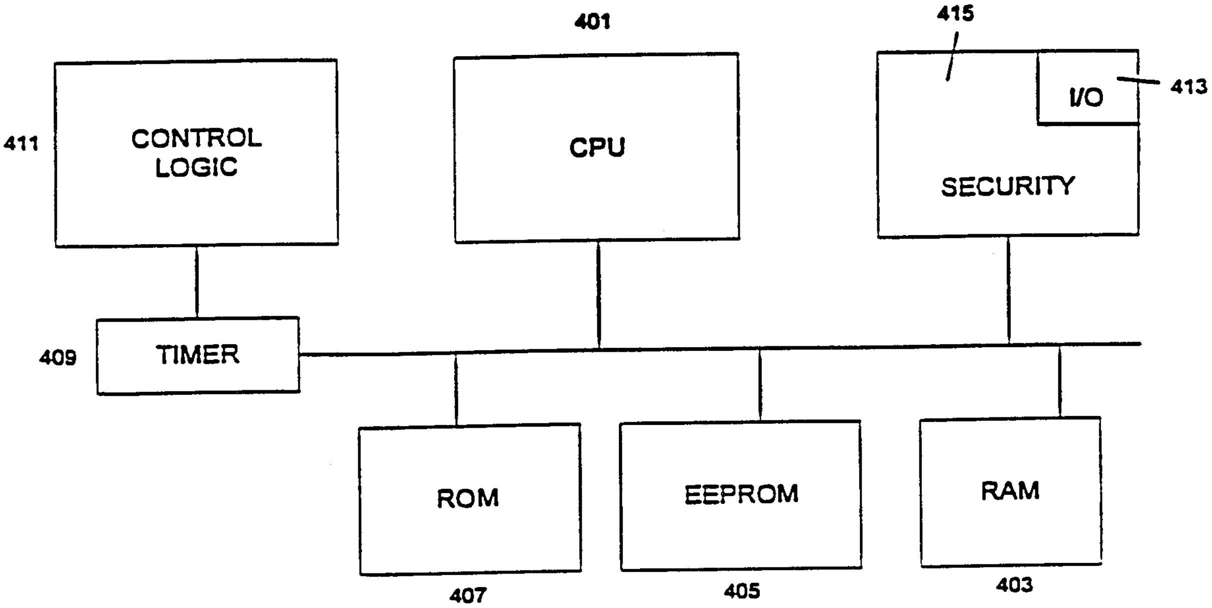

401
415
411
CONTROL LOGIC
CPU
I/O
413
SECURITY
409
TIMER
ROM
EEPROM
RAM
407
405
403

FIG. 4

ANNEX A TO THE DRAWINGS

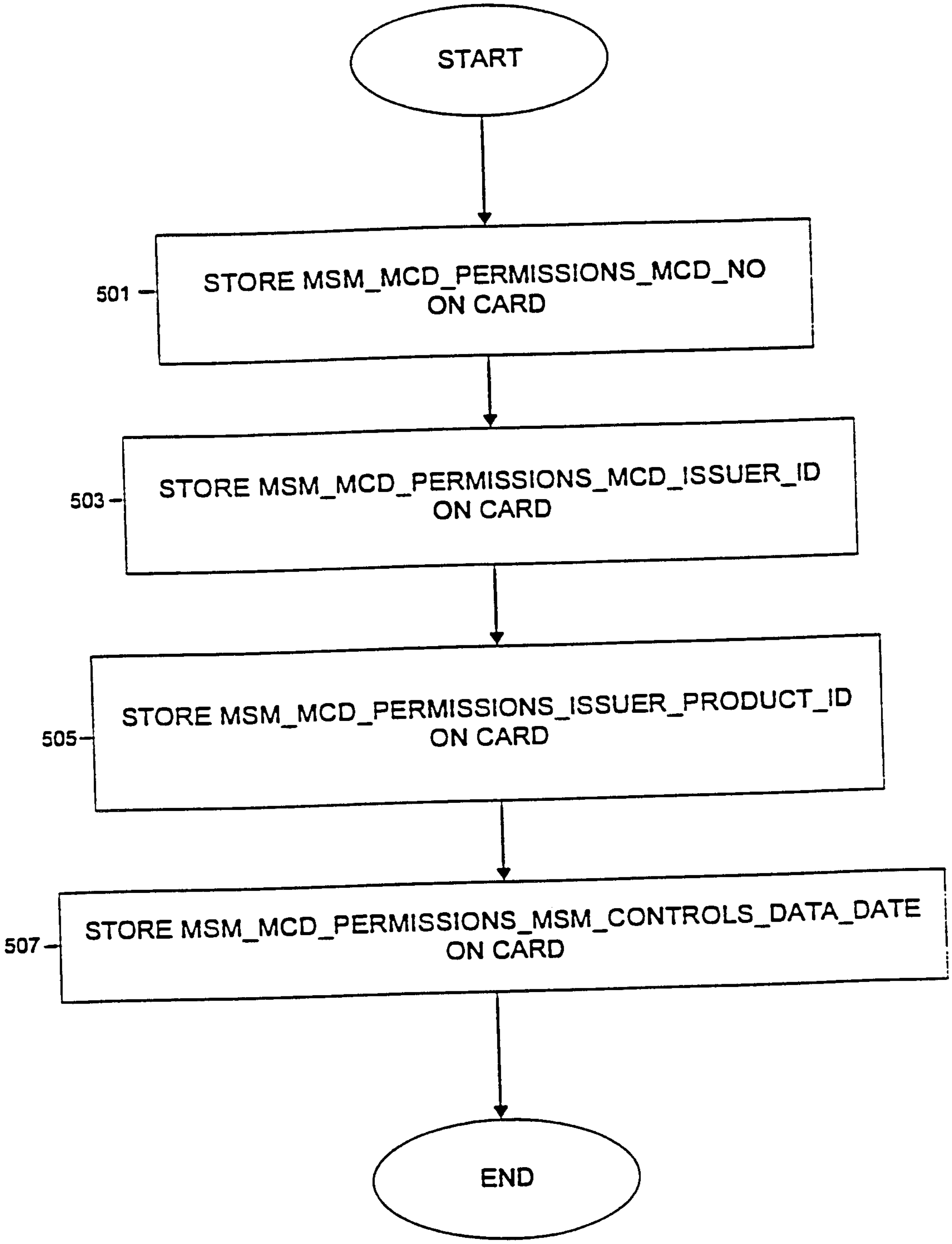

START
501
STORE MSM_MCD_PERMISSIONS_MCD_NO ON CARD
503
STORE MSM_MCD_PERMISSIONS_MCD_ISSUER_ID ON CARD
505
STORE MSM_MCD_PERMISSIONS_ISSUER_PRODUCT_ID ON CARD
507
STORE MSM_MCD_PERMISSIONS_MSM_CONTROLS_DATA_DATE ON CARD
END

FIG..5

ANNEX A TO THE DRAWINGS

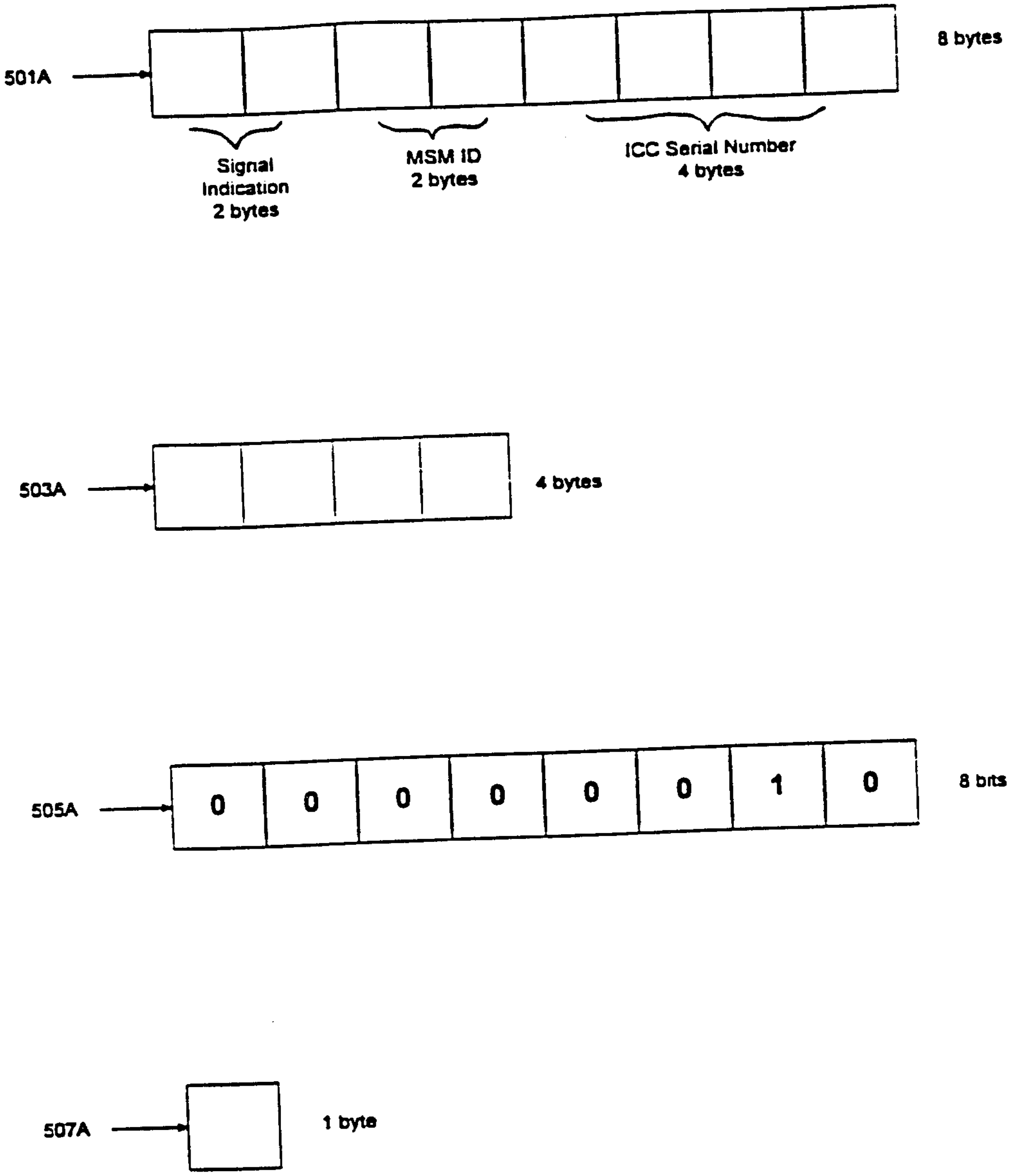


FIG. 5A

ANNEX A TO THE DRAWINGS

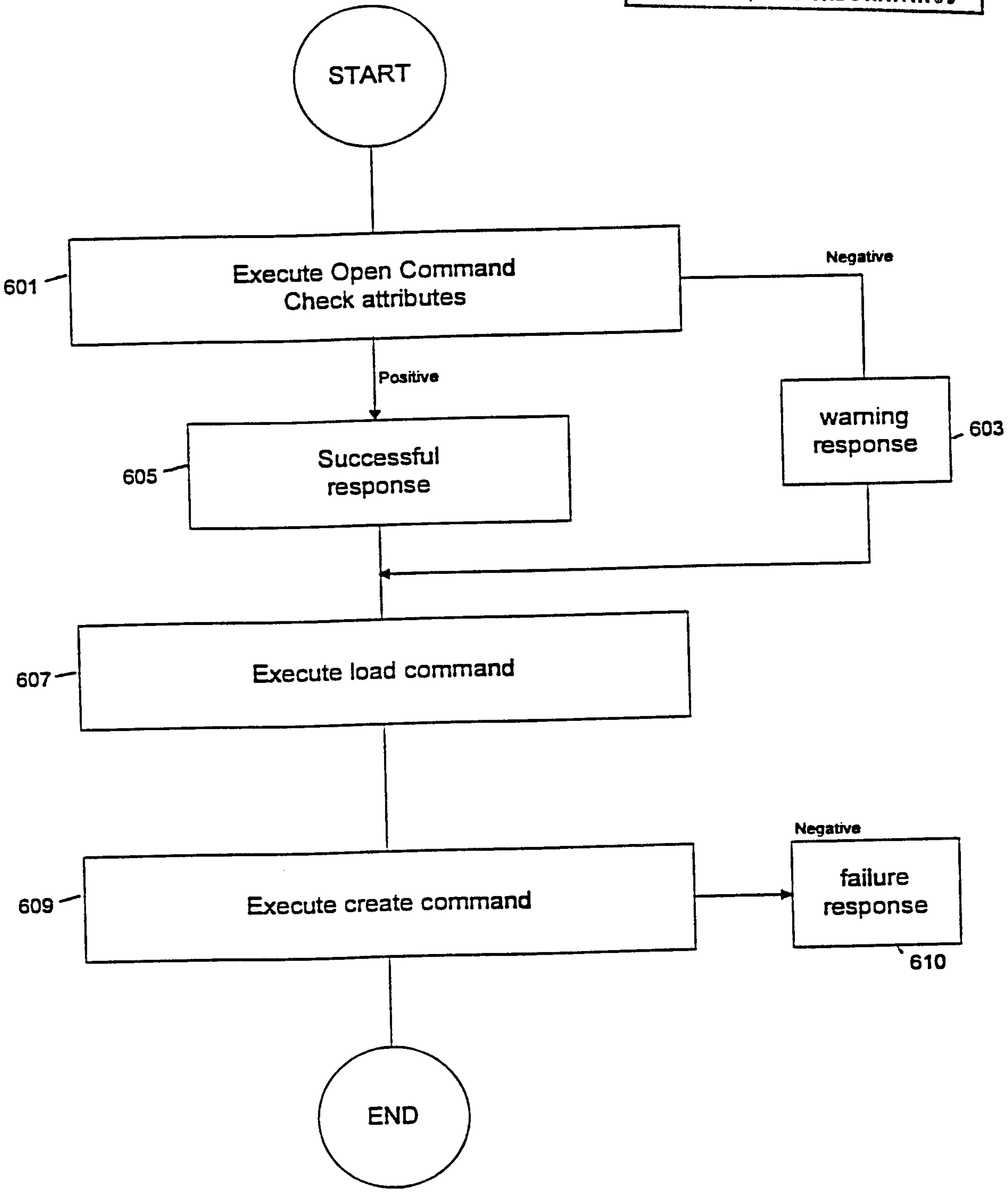


FIG. 6

ANNEX A TO THE DRAWINGS

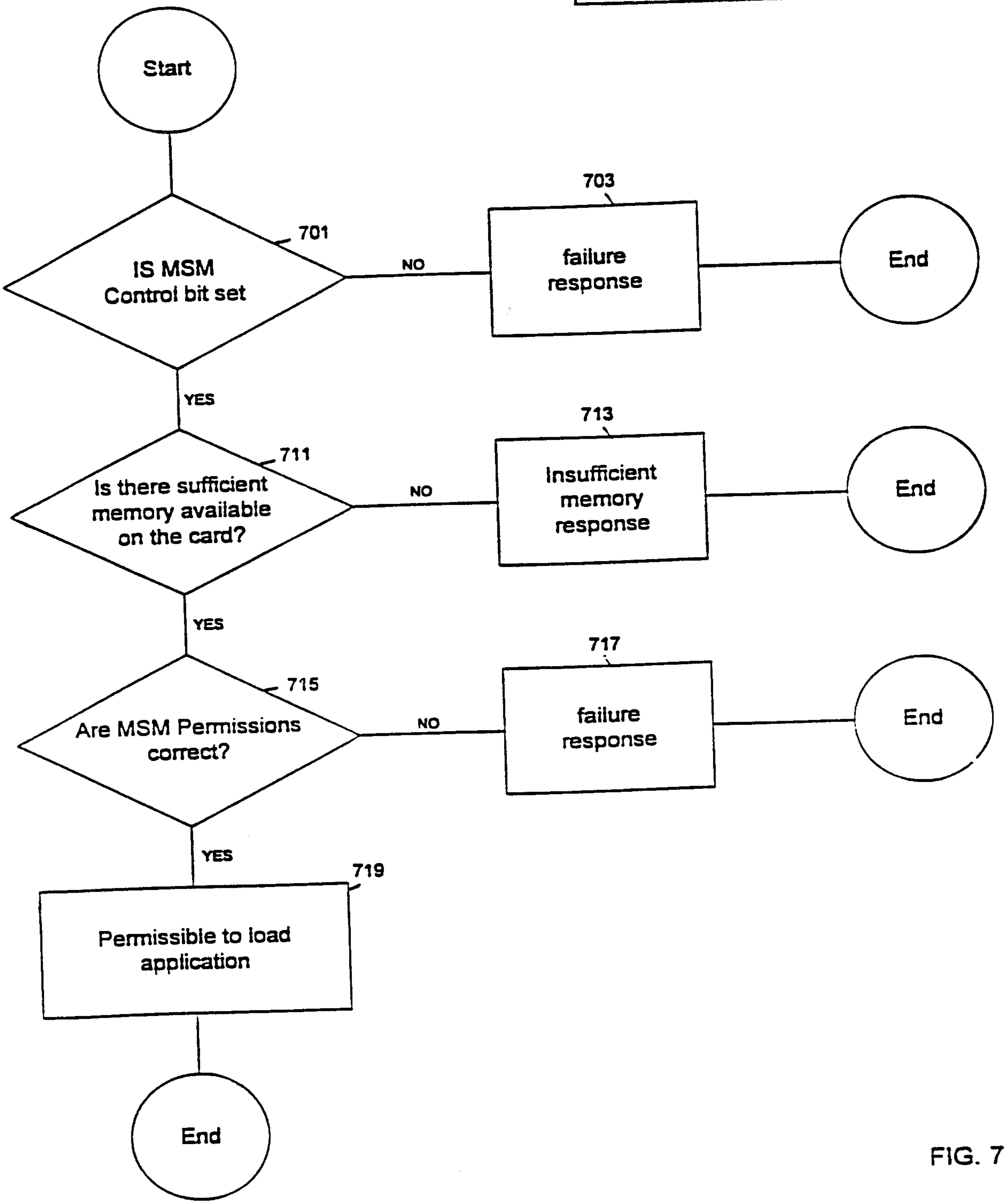

Start
IS MSM Control bit set
701
NO
703
failure response
End
YES
Is there sufficient memory available on the card?
711
NO
713
Insufficient memory response
End
YES
Are MSM Permissions correct?
715
NO
717
failure response
End
YES
719
Permissible to load application
End

FIG. 7

ANNEX A TO THE DRAWINGS

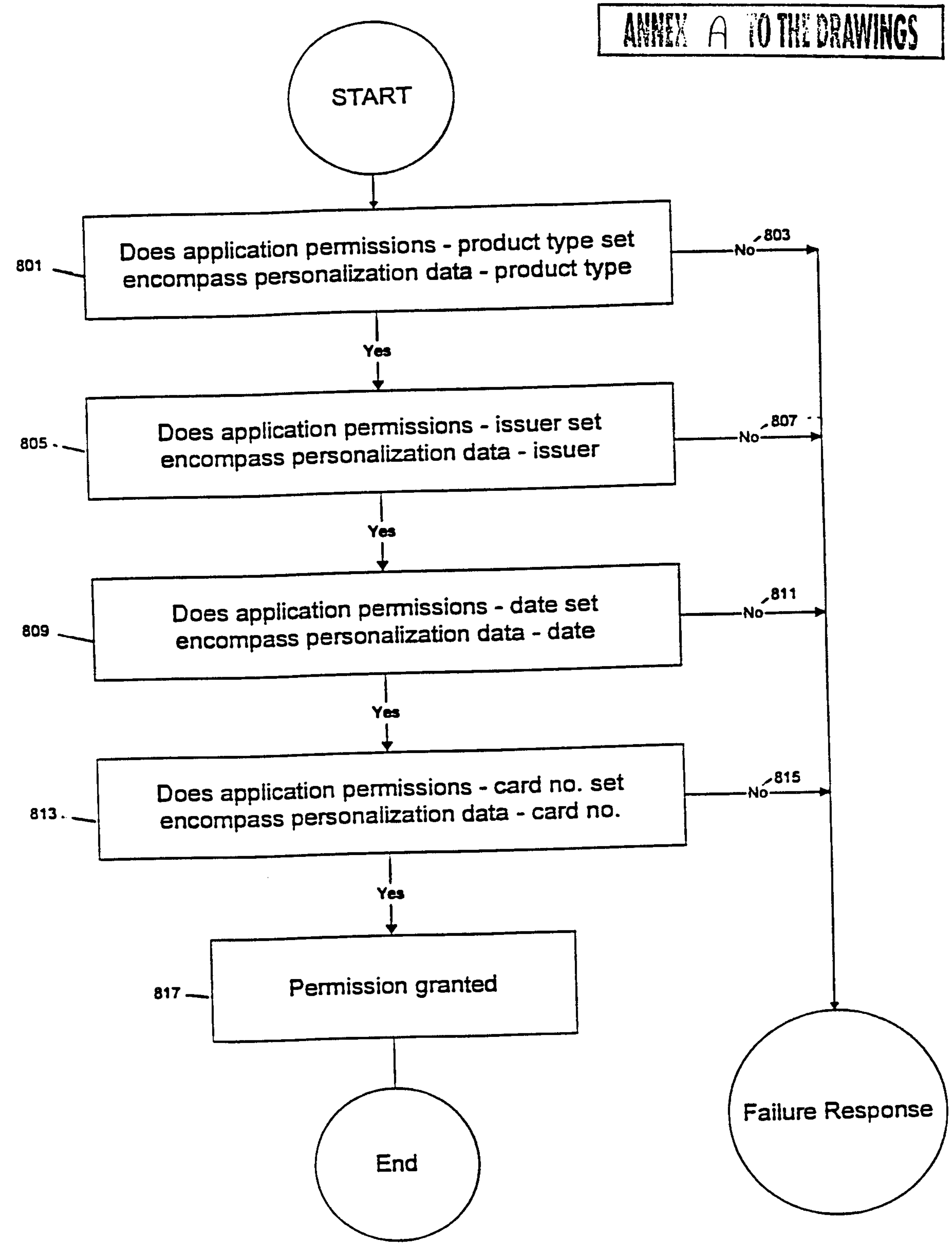

START
801
Does application permissions - product type set encompass personalization data - product type
No
803
Yes
805
Does application permissions - issuer set encompass personalization data - issuer
No
807
Yes
809
Does application permissions - date set encompass personalization data - date
No
811
Yes
813
Does application permissions - card no. set encompass personalization data - card no.
No
815
Yes
817
Permission granted
End
Failure Response

FIG. 8

ANNEX A TO THE DRAWINGS

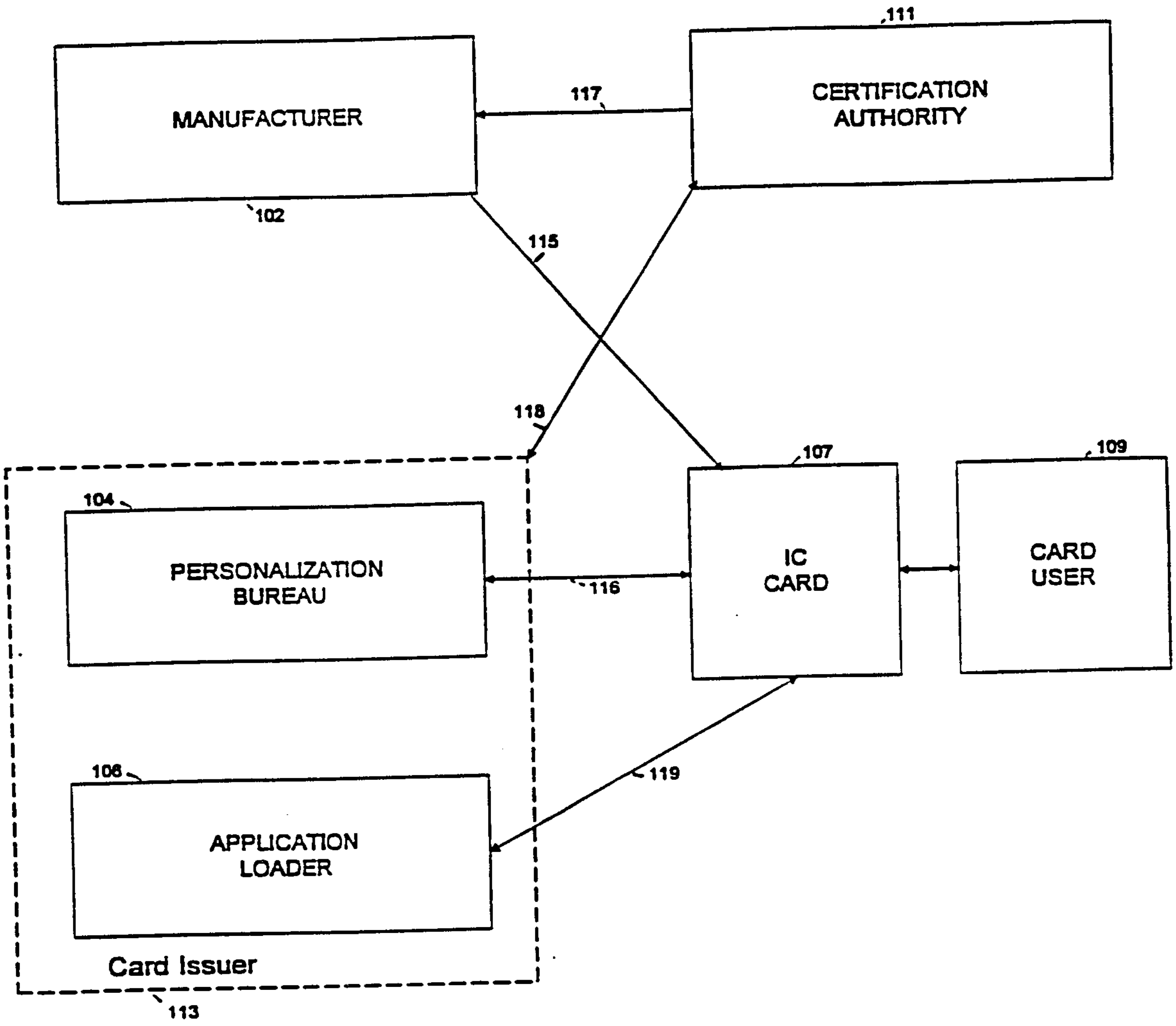


FIG. 9

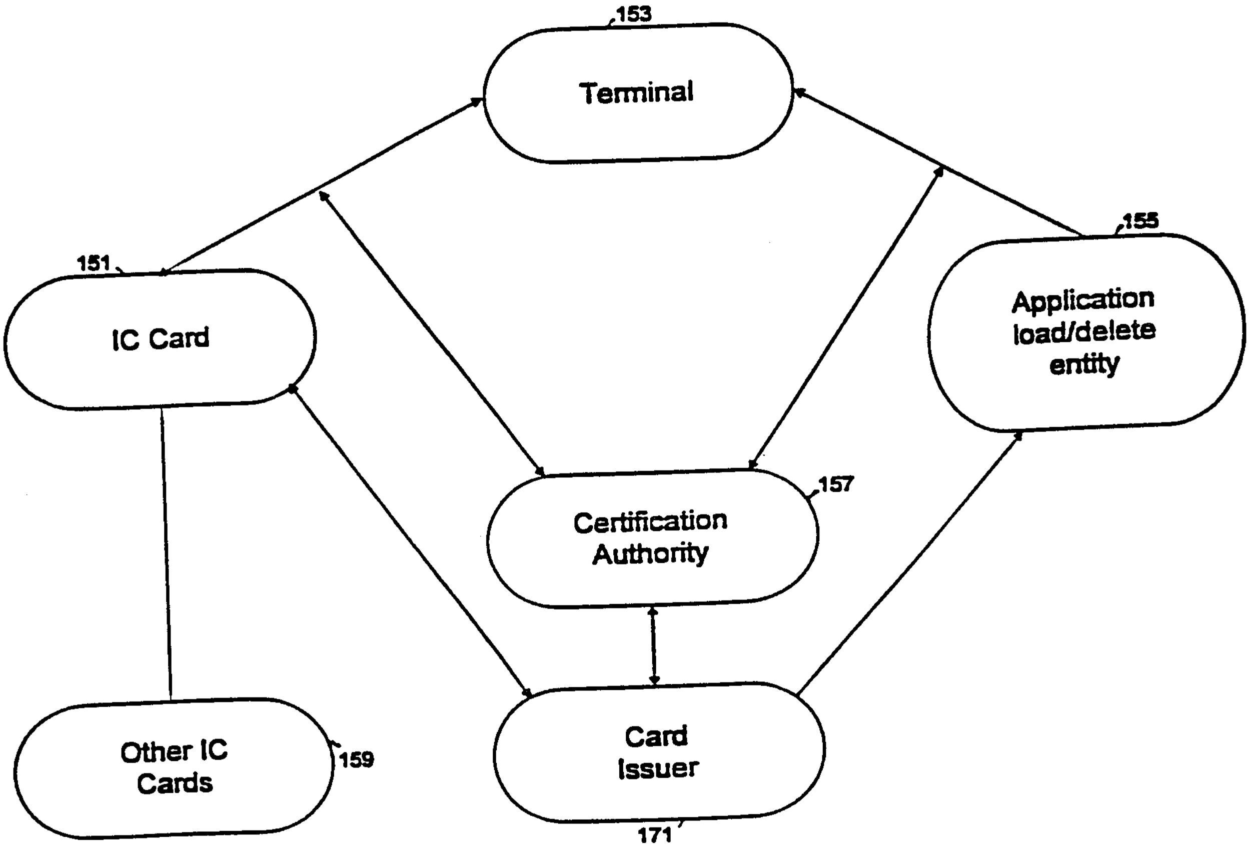

153
Terminal
151
IC Card
155
Application load/delete entity
157
Certification Authority
Other IC Cards
159
Card Issuer
171

FIG. 10

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) **International Patent Classification** 6 :
G07F 7/10

A2

(11) **International Publication Number:** **WO 98/52163**

(43) **International Publication Date:** 19 November 1998 (19.11.98)

(21) **International Application Number:** PCT/GB98/01405

(22) **International Filing Date:** 14 May 1998 (14.05.98)

(30) **Priority Data:**

60/046,514	15 May 1997 (15.05.97)	US
09/075,973	11 May 1998 (11.05.98)	US

(71) **Applicant:** MONDEX INTERNATIONAL LIMITED [GB/GB]; 47–53 Cannon Street, London EC4M 5SQ (GB).

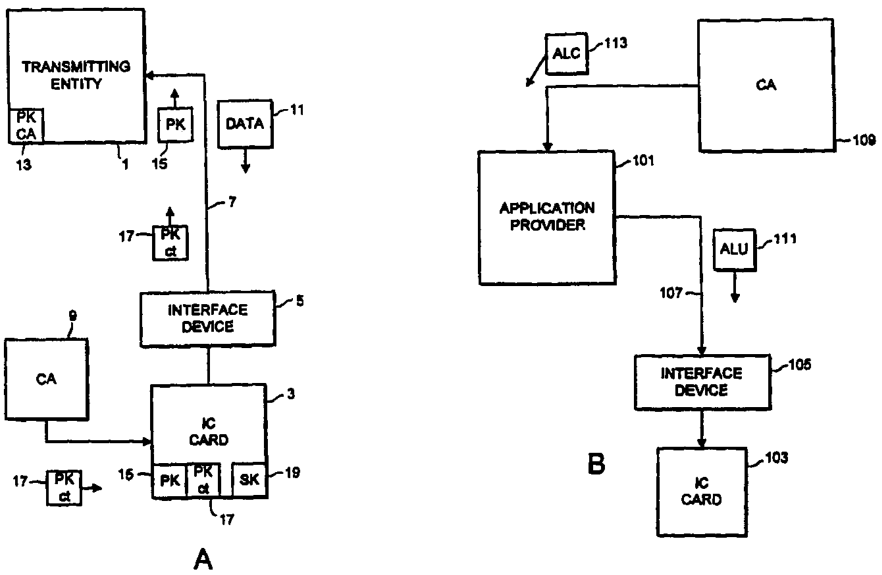
(72) **Inventors:** RICHARDS, Timothy, Philip; 32 Craig Mount, Radlett, Herts. WD7 7LW (GB). EVERETT, David, Barrington; 31 Ashdown Avenue, Saltdean, Brighton, Easr Sussex BN2 8AH (GB). VINER, John, Charles; Hydes, Woodlands Lane, Windlesham (GB).

(74) **Agent:** POTTER, Julian, Mark; D. Young & Co., 21 New Fetter Lane, London EC4A 1DA (GB).

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

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

(54) **Title:** IC CARD TRANSPORTATION KEY SET



(57) **Abstract**

Method and apparatus for securely transporting data onto an IC card. The method is used, for example, to transport data, including application programs, in a secure manner from a source located outside the IC card. At least a portion of the data is encrypted using the public key of a public/secret key pair of the intended IC card unit. The encrypted data is then sent to the IC card and the IC card verifies the key transformation unit using its unique secret key. The data can then be stored on the IC card. A copy of the public key signed by a certification authority can be used to verify that the card is authorized to be part of the overall authorized system.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

IC CARD TRANSPORTATION KEY SET

BACKGROUND OF INVENTION

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. This allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application when it is manufactured and before it is given to a card user. That application, however, cannot be modified or changed after the card is issued even if the modification is desired by the card user or card issuer. Moreover, if a card user wanted a variety of application functions to be performed by IC cards issued to him or her, such as both an electronic purse and a credit/debit function, the card user would be required to carry multiple physical cards on his or her person, which would be quite

cumbersome and inconvenient. If an application developer or card user desired two different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal, making the transaction difficult, lengthy and inconvenient.

Therefore, it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card. Although multiple applications could be preselected and placed in the memory of the card during its production stage, it would also be beneficial to have the ability to load and delete applications for the card post-production as needed.

The increased flexibility and power of storing multiple applications on a single card create new technical challenges to be overcome concerning the integrity and security of the information (including application code and associated data) exchanged between the individual card and the application provider as well as within the entire system when loading and deleting applications. It would be beneficial to have the capability in the IC card system to exchange data among cards, card issuers, system operators and application providers securely and to load and delete applications securely at any time from a local terminal or remotely over a telephone line, Internet or intranet connection or other data conduit. Because these data transmission lines are not typically secure lines, a number of security and entity authentication techniques must be implemented to make sure that applications being sent over the transmission lines are not tampered with and are only loaded on the intended cards.

As mentioned, it is important -- particularly where there is a continuing wide availability of new applications to the cardholder -- that the system has the capability of adding applications onto the IC card subsequent to issuance. This is necessary to protect the longevity of the IC cards; otherwise, once an application becomes outdated. the card would be useless. It would be beneficial to allow the addition of applications from a remote location as well as from a direct connection to an application provider's terminal. For example, it would be beneficial for a card user to be able to plug his or her IC card into a home computer and download an application over the Internet. This type of remote loading of applications raises a number of security risks when transmitting the application code and related data over an unsecured communications line such as the Internet.

An entity which transmits an application or data to an IC card requires that only the intended IC card should receive the transmitted data. Third parties should not be able to intercept and view the data. Additionally, a transmitting entity will require verification that the IC card which has requested information is actually part of the overall IC card system and not simply posing as being part of the system. These concerns are raised by both remote application loading as well as local terminal application loading.

Accordingly, it is an object of embodiments of this invention to provide a transfer technique having improved security and specifically to provide an IC-card system that allows for the transfer of data with improved security including smart card applications which may be loaded onto IC cards.

SUMMARY OF THE INVENTION

These and other objectives are achieved by an embodiment of the present invention which provides an IC card method and apparatus for securely transporting data including an application onto an IC card including storing a secret and public key pair on the IC card, retrieving the stored public key from the IC card, encrypting at least a portion of the data to be transported using the public key, transmitting the encrypted data to the IC card and decrypting the encrypted data using the IC card's secret key.

In a preferred embodiment, a certification authority ("CA") or the entity that manages the overall security of the IC card system, encrypts (or digitally signs) a copy of the IC card's public key and the signed copy is also stored on the IC card. The entity transmitting the data to the IC card can verify that the CA has approved the card by retrieving using the IC card's signed public key and verifying the signed public key using the public key of the CA. If verification is successful, the entity has verified that the CA approved the IC card.

BRIEF DESCRIPTION OF THE DRAWINGS

Further objects, features and advantages of embodiments of the invention will become apparent from the following detailed description taken by way of example only in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

Fig. 1A is a block diagram of the secure data transfer system which securely transfers data from a transferring entity to an IC card.

Fig. 1B is block diagram of the application loading system which loads an application from an application provider to an IC card;

Fig. 2 is a graphic representation of the contents of an Application Loading Unit;

Fig. 3 is a graphic representation of an Application Unit;

Fig. 4 is a flow chart of the steps for providing an individual key set for an IC card;

Fig. 5 is a graphic representation of a Key Transformation Unit;

Fig. 6 is a graphic representation of a Key Transformation Unit plaintext;

Fig. 7 is a graphic representation of the Application Load Certificate;

Fig. 8 is a graphic representation of the Application Unit being decrypted;

Fig. 9 is a flowchart illustrating the steps undertaken in processing the Application Load Unit;

Fig. 10 is a flowchart illustrating the steps undertaken in processing the KTU; and

Fig. 11 is a block diagram showing the components of an IC card which can receive and process and Application Load Unit.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

DETAILED DESCRIPTION OF THE INVENTION

It is beneficial to have the capability to load applications onto IC cards containing multiple application operating systems at any time during the lifetime of the IC card. This flexibility allows a user of a card to periodically add new applications to the IC card and also allows older applications to be updated with newer versions of the application when they are released. For example, a card user may start with an IC card that contains a purse, or electronic cash application (e.g., MONDEX™), being stored on his IC card. Some time after the user has the card, he or she may load an additional application onto the card such as a credit/debit application. Some time after loading the credit/debit application on the card, a new version of the credit/debit application may become available and the card user should be able to erase the old application on his IC card and replace it with the new version of the credit/debit application which may contain additional features. Additionally, an IC card needs to receive data regarding personal information such as new credit card account numbers or updated information.

The flexibility of loading applications and transmitting data at different times during the IC card's life cycle creates security issues with the process of loading applications onto the card. In a multiple application operating system environment, it is beneficial to be able to load applications and data both at terminals, such as a bank ATM machine, as well as over remote communication links, such as telephone lines, cable lines, the Internet, satellite or other communications means. When loading applications and data onto an IC card, the application provider needs to provide security regarding the applications to be loaded. First, the application provider must make sure the application is only sent to the correct card user who is intended to receive the application. Second, the application and associated data may contain private or trade secret information which needs to be encrypted so entities other than the IC card cannot view the contents of the encrypted application code and data. A portion of the application code and data may be secret while other portions are not. These concerns of authentication and protecting the contents of some or all of the application and associated data being loaded onto a card is addressed herein.

A number of encryption/decryption techniques are described herein. There are two basic types of encryption, symmetric encryption and asymmetric encryption. Symmetric encryption uses a secret key as part of a mathematical formula which encrypts data by transforming the data using the formula and key. After the data is encrypted, another party can decrypt the encrypted data using the same secret key with a decryption algorithm. Thus the same key is used for encryption and decryption so the technique is symmetric. A conventional example of a symmetric algorithm is DES.

Asymmetric encryption techniques use two different keys of a pair for encrypting and decrypting information. The two keys are normally referred to as a private or secret key and a public key. When data is encrypted with one key of the pair, the other key is used to decrypt the data. If a sender of data signs the data with his secret key, anyone with the public key can verify the message. Since public keys are typically known to the public, the contents of a data signed with a secret key cannot be protected but the origination of the data can be verified by determining if a particular secret key signed the data. This authentication process is termed a digital signature. If person A wanted to authenticate a message he was sending to person B, the person A would sign the document with his secret key. When person B received the message, he would use person A's public key to verify the message. If the message was verified with the public key, person B would know that the document was signed with secret key of person A. Thus, the origin of the message has been authenticated.

The asymmetric key set can also be used to protect the contents of a message. If person A wanted to send an encrypted message to person B that no one else could read, he would encrypt the data or message with person B's public key and send it to person B. Now only the holder of B's secret key could decrypt the data. If a combination of keys is used, a person could both authenticate and encrypt the message. The asymmetric pair of keys has some powerful applications with respect to card security. However, asymmetric encryption is relatively processor costly (processor cost is associated with computation time) compared with symmetric encryption. An example of asymmetric encryption method is RSA®.

A hybrid of symmetric encryption which makes the encryption method more powerful is to encrypt data using two symmetric keys. This technique is called triple DES which encodes data with key 1, decodes the data using key 2 (which in effect further encodes the data) and then further encodes the data using key 1 again. Once the data has arrived at its destination, key 1 is used to decode the data, key 2 is used to encode the data, and key 1 is used to decode the data. These extra steps of encoding and decoding make the technique more powerful and more difficult to properly decipher without both keys.

Figure 1A shows a block diagram of the entities used in transporting data in a secure manner in an IC card system. The transmitting entity 1 can be a card issuer, bank, IC card or other entity which desires to transport data to an IC card 3. The transmitting entity 1 preferably initiates the data transfer process. Alternatively, the IC card 3 can initiate the data transfer process if the card requires data from the transmitting entity 1.

The transmitting entity 1 is connected to interface device 5 (e.g., a terminal that communicates with an IC card). Data conduit 7 can be a telephone line, an intranet, the Internet, a satellite link or any other type of communications link. In this example, the transmitting entity 1, which is remotely located from IC card 3, desires to send data in a secure manner to the IC card. However, because the data link is an "open" link (i.e. not a private link) and subject to third parties possibly intercepting or replacing data being transmitted, security measures are needed to guarantee that only the intended IC card will receive the transmitted data. The Certificate Authority 9 can also be used to authenticate that the IC card has been validated as part of the IC card system.

In Figure 1A, a private (or secret) key 19 and corresponding public key 15 is generated for IC card 3. The keys are preferably generated using an asymmetric encryption algorithm such as RSA®. The keys can be generated at the CA 9 or any other location because they are specific only to the IC card 3 and no other copies need to be kept. A third data item, the public key certificate 17, is also generated and stored on the IC card 3.

The public key certificate 17 is generated by signing the public key 15 with the private key of the CA 9. This allows a person with the public key of the CA 9 to verify that the CA digitally signed the IC card's public key in order to certify the IC card's individual key set. The public key certificate can be generated by the CA at the time the IC card private/public key set is generated or at a subsequent time.

When a data transfer is initiated by the transmitting entity 1, the IC card 3 is contacted through the interface device 5 and the IC card 3 sends its public key 15 and its public key certificate 17 to the transmitting entity 1. The transmitting entity then verifies the public key certificate with public key of the CA

13 (which is publicly available from the CA 9 and may be stored in the transmitting entity 1) thus determining if the CA 9 digitally signed the public key and verifying that the IC card is a valid card.

The transmitting entity 1 then encrypts the data to be transmitted with the IC card's public key. The transmitting entity 1 then transmits the encrypted data 11 to the interface device 5 and to the IC card 3. The IC card 3 decrypts the encrypted data with its corresponding private (also called secret) key 19. The data can then be processed by the IC card 3. Only the IC card 3 has a copy of its private key so only the intended IC card can access the encrypted data. This ensures that third parties cannot access the encrypted data and correspondingly that only the intended IC card will be able to read and process the data.

Figure 1B shows a secure method for loading applications onto an IC card. Figure 1B shows a block diagram of the entities used in a secure remote application loading process. The application provider 101 can be a card issuer, bank or other entity which provides application loading services. The application provider 101 initiates an application loading process onto IC card 103. IC card 103 is connected to data conduit 107 which is connected to interface device 105 (e.g., a terminal that communicates with an IC card). Data conduit 107 can be a telephone line, an intranet, the Internet, a satellite link or any other type of communications link. The application provider 101, which is remotely located from the IC card 103, desires to send and load an application to the IC card. However, because the data link is an open link and subject to third parties possibly intercepting or replacing applications being transmitted, security measures which authenticate the application itself, the application provider and the IC card must be used to ensure the integrity of the system. The CA 109 may also be used to help authenticate that some data being transferred is part of an identified system.

In Figure 1B, the application provider sends an application load unit 111 to the interface device 105 and finally to IC card 103. The ALU includes the application itself and security data required to authenticate and protect the application code and associated data. The ALU is discussed specifically in Figure 2 and in connection with the other figures herein. The ALU 111 also preferably contains Application Load Certificate (ALC) 113 data which is sent from the Certification Authority (CA) 109 to the application provider 101. The Certification Authority manages the overall security of the system by providing an Application Load Certificate for each application which is to be loaded onto an IC card. The application provider 101 and the IC card 103 both have individual public/secret keys sets. The authentication and security processes will now be described.

Figure 2 shows a diagram illustrating the components of an Application Load Unit which is sent from the application loader to the IC card during the application load process. The Application Load Unit (ALU) 201 contains an Application Unit (AU) 203, an Application Unit Signature (AU_S) 205, a Key Transformation Unit (KTU) 207 and an Application Load Certificate (ALC) 209. The ALU 201 is formatted in a conventional format used during data transmission. AU 203 contains the application code and data which are to be stored on the IC card, some or all of which is encrypted to protect a secret portion or portions of the code and/or data. AU 203 is described in further detail in connection with Figure 3.

AU_S 205 is the application code and data AU 203 digitally signed with the secret key of the application provider. The public key of the application provider is sent as part of the ALC 209 and is used to authenticate the application provider as the originator of the application. ALC 209 is made up of card identification information and the application provider's public key and is signed by the secret key of the certification authority. All these elements will be described in more detail below.

KTU 207 contains information relating to the encryption of the AU 203 (the code and data of the application) which allows the IC card to decrypt the designated portions so that the application and data can be accessed by the IC card but protects the data during transmission between the application provider and the IC card. KTU 207 is encrypted with the public key of the IC card for which the application is intended which ensures that only the intended IC card can decrypt the application code and data using the KTU information. This element will be described in connection with Figure 5.

Figure 3 shows a graphic representation of the Application Unit 203 which is part of the application load unit. The AU 203 contains both the program code and associated data which is to be loaded onto the IC card of the card user. The program code consists of a number of program instructions which will be executed by the microprocessor on the IC card. The program instructions can be written in any programming language which the operating system stored on the IC card can interpret.

For example, in the MULTOS system the program can be written in MEL™ (MULTOS Executable Language). Most applications have associated data which must be loaded onto the card. For instance, data which identifies the card user such as a person's name or account number may be loaded in a secure manner with the credit/debit application. An application provider may provide electronic cash represented by data as a promotion when installing an electronic purse application. Some or all of this data is desired to be kept secret from third parties. Additionally, the application code itself may be considered proprietary and portions may be desired to be kept secret from others. The use of a Key Transformation Unit (KTU) will allow an application provider to designate and encrypt selected portions of its application as confidential and protect it from third parties.

Application Unit portion 305 indicates the program code which is to be transferred from the application provider to the IC card. Application Unit portion 307 indicates the associated data which is to be transferred as part of the application to be loaded onto the IC card. In this example, three discrete areas of the application unit are shown to be encrypted using either single DES or triple DES. Any number of variations regarding the portions encrypted and the type of encryption can be employed using the techniques described herein.

In this example, encrypted location 309 shows the first portion of the Application Unit 203 which has been encrypted using a triple DES technique. The encryption process as described above involves using a symmetric key and the conventionally known DES-based algorithm to transform the data. The data can later be recovered by applying the key to a conventionally known DES-based decryption algorithm. Encrypted location 311 shows a second portion of the application unit 203 which has been encrypted using triple DES. Encrypted location 313 shows a third portion which is encrypted using single DES. Single DES requires less computation to decrypt and takes up less space as part of the KTU as described below. If the application unit were intercepted by a third party while it was being transmitted from the application loader to the IC card, the encrypted portions could not be read unless the third party had the correct keys and decryption algorithm. That information, therefore, is protected in the KTU.

The KTU is used to allow the IC card for which the application and associated data is intended to decrypt the encrypted portions of the Application Unit by describing which portions of the application unit are encrypted, which encryption algorithm was used and the key or keys to be used to decipher the text. This information is highly confidential between the application provider and the intended IC card and therefore is protected in a manner unique to the intended card. In order to encrypt the KTU which is part of the overall ALU being transmitted, an individual key set for the particular intended IC card is used. The key set and its generation will now be described.

In accordance with an embodiment of the present invention, one of the security operations performed at the CA is to generate an individualized key set for each IC card which is stored on the card. The keys are used for off-card verification (i.e., to verify that the card is an authentic card) and for secure data transportation. The key generation process is shown generally in Figure 4. The key set is made up of three different key data items: the card's secret key which is known only to the card, the card's public key which is stored on the card and the card's public key certificate which is the card's public key signed by the CA's secret key. The individual keys of the key set are described in more detail below.

Step 401 stores a card specific transport secret key for the individual IC card in the memory of the card. This secret key is generated by the CA from a standard asymmetric encryption technique such as RSA® and loaded onto the card via a card acceptance device. Once stored on the card, the CA deletes from its own memory any data relating to the secret key. Thus, only the card itself knows its secret key. The data element containing the secret key information in the card is called "mkd_sk" which stands for MULTOS key data secret key.

Step 403 stores a card specific transport public key for the individual IC card in the memory of the card. This public key is preferably generated by the CA from the asymmetric encryption technique used to produce the secret key in step 401. As with the secret key, once the public key is stored on the card, the CA (or other key provider) deletes from its systems the public key data so that the only copy of the public key is kept in the card. The data element containing the card's public key information is called "mkd_pk" which stands for MULTOS key data public key.

Step 405 stores a card specific transport public key certificate for the individual IC card in the memory of the card. The data element containing the card's public key certificate information is called "mkd_pk_c" which stands for MULTOS key data public key certificate. This public key certificate is preferably generated by signing the transport public key mkd_pk with the secret key of the

CA, indicated as follows:

$$mkd\_pk\_c = [mkd\_pk]_{CA\_sk}$$

which means the individual card's public key certificate is formed by applying the CA's secret key to the individual card's public key. The process is carried out at the CA. The public key certificate is retained by the CA so that it can regenerate the public key as needed.

A terminal can read the public key certificate from the IC cards to verify that the CA had signed and therefore approved the individual IC card. This is accomplished by verifying the public key certificate with the public component of the CA key set used to sign the mkd_pk.

Figure 5 is a graphic depiction of the contents of KTU 207, which contains Header portion 501 and KTU Ciphertext portion 503. As shown in Figure 5, header information 501 includes, for example, identifier or permissions information 505 such as the application_id_no (application identification number), mcd_no (IC card no) and/or msm_control_data_date (the date the IC card was issued). Additional identifiers could also be included. These identifiers allow the system to verify that the IC card which receives the ALU is the intended IC card. The permissions data is discussed in detail in the above referenced related application.

KTU Ciphertext 503 corresponds to KTU Plaintext (not encrypted) encrypted with the public key mkd_pk of the intended IC card as shown in box 507. The KTU Plaintext in further described in Figure 6. The public key mkd_pk is obtained from the intended IC card by the application provider. The public key

of an IC card is freely available to anyone and can be obtained directly from the card or from the CA. By encrypting the KTU Plaintext with the IC card public key, only the intended IC card can use its secret key of the public/secret key pair to decrypt the KTU Ciphertext. This means that only the intended IC card can determine the contents of the KTU plaint text, identify the encrypted portions of the application being loaded and use the keys to decrypt and recover the entire application and associate data. Because no other entity has the secret key of the IC card, the security and integrity of the program code and data being transmitted in ensured.

Figure 6 is a graphic representation of KTU Plaintext 601. KTU Plaintext 601 preferably includes identifier field 603, no_area_discriptors field 605, alg_id field 607, area_start field 609, area_length 611, key_length field 613, key_data field 615 and additional area and key fields depending upon the number of encrypted areas present in the Application Unit. Identifiers 603 contain identifying information of the Application Unit to which the KTU applies. No_area_descriptors 605 indicates how many different portions of the AU have been encrypted. In the example of Figure 3, the number or area descriptors would be three. Field 607 contains the algorithm identifier for the first area which has been encrypted. The algorithm could be DES or triple DES, for example. Field 609 indicates the start of the first encrypted area. This indication could be an offset from the start of the AU. For example, the offset could by 100 which means that the first area starts at the 100th byte of the Application Unit. Field 611 indicates the area length for the first encrypted portions. This field allows the microprocessor on the IC card to know how large an area has been encrypted and when coupled with the start of the area, allows the IC card microprocessor to decrypt the correct portion of the Application Unit. Filed 613 indicates the key length for the particular encrypted portion of the application unit. The length of the key will differ for different encryption techniques. The key length field allows the IC card to know the length of the key data. Field 615 indicates the key data for the particular encrypted portion. The key data is used with the algorithm identity and the location of the encoded portion to decode the encrypted portion. If more than one encrypted area is indicated, then additional data referring to the algorithm, start location, length, key length and key data will be present in the KTU Plaintext. While a number of fields have been described, not all the fields are necessary for the invention. The most important field, however, is the key data itself.

Figure 7 is a graphic representation of the Application Load Certificate (ALC) 209. ALC 209 includes a header 701 and the Application Provider Public Key 703. Header 701 and Application Provider Public Key 703 are then signed (encrypted) with the CA secret key. Thus, the ALC 209 must be provided by the CA to the application provider for each application loaded because only the CA knows the CA private key. Header 701 contains information regarding the application provider and the IC card for which the application is intended. The ALC 209 is placed in the correct ALU by the application provider which can use the identification information. Application Provider Public Key 703 is provided to the CA along with the identification data. The CA then signs this information after verifying its authenticity and returns the signed ALC to the application provider.

The IC card, when it receives the ALC 209 as part of the ALU 201, will verify the ALC 209 with the public key of the CA. This ensures that the CA signed the Application Load Certificate and that it is genuine. After verifying the information, the header identification information 701 is checked and the application provider public key is recovered. This public key will be used to verify that the application and code which is to be loaded onto the IC card originated with the proper application provider.

Figure 8 is a graphic representation of the use of the application provider's public key to verify the signature of the AU 205 in order to verify that AU 203 was signed by the application provider. AU signature 205 is verified with the Application Provider Public Key 801 and compared with AU 203. If the data blocks match, then the IC card has verified that the application provider signed (encrypted) the application unit and the application is genuine. This authentication is valid because only the application provider has its own secret key. The IC card can process this information efficiently because the application provider's public key is provided to it as part of the Application Load Certificate 209 which is signed by the CA. Therefore, it does not need to retrieve the public key from an external location to authenticate the application.

Figure 9 shows a flow chart of the steps for processing the Application Load Unit when it is received by the IC card. Prior to receiving the ALU, identity checks as to the identity of the IC card can be performed if desired. The ALU processing techniques provide a number of further verifications including verifying that the application being loaded is: (1) from the correct application

provider, (2) being loaded on the intended card and (3) certified by the CA. The ALU processing techniques also allow the transportation of transport decryption keys which enable the IC card to decrypt portions of the program code and associated data in a secure manner. In step 901, the IC card receives the ALU from the application provider. The ALU can be transmitted via a terminal connection, contactless connection, telephone, computer, intranet, Internet or any other communication means. The ALU is placed in an I/O buffer of the IC card along with header information indicating the starting addresses of AU 203, AU signed 205, the KTU 207 and ALC 209. Alternatively, the IC card could determine the relative address locations of these four units.

Step 903 verifies the ALC 209 with the CA public key. Each IC card preferably stores in its memory a copy of the CA public key because it is used in many transactions. Alternatively, the IC card could obtain the public key from a known storage location. If the CA public key verifies the ALC 209 properly, then the IC card has verified that the CA has signed the ALC 209 with its secret key and thus the Application Load Certificate is proper. If the IC card cannot verify the ALC properly, then the ALC was not signed by the CA and the certificate is not proper. The application loading process would then end.

Step 905 then checks the identity of IC card against the identification information sent in the Application Load Certificate to make sure the card is intended to receive the application. This permissions checking is described in the related patent application identified above. If there is no match of identification data, the application loading process ends. If the identification data does match, then the process continues.

Step 907 uses the application providers public key which was recovered from the verified ALC to verify AU signature 205. When the ALU was generated by the application provider, the application unit 203 was signed with the application provider's secret key to authenticate that the application was provided by the correct application provider. The application provider then provides its public key to IC card through the ALC. The IC card then verifies the AU signature 205. If the two data blocks match, then the ALU is verified as being generated by the application provider. Because the application provider's public key is part of the ALC which is signed by the CA, the CA can make sure that the proper public key has been provided to the IC card. This unique key interaction between the application provider, CA and the intended IC card ensures that no counterfeit or unapproved applications or data are loaded onto an IC card which is part of the secure system.

Step 911 then processes a KTU authentication check which further verifies that only the intended card has received the application. The KTU authentication check makes sure that if a third party does somehow intercept the ALU, the third party cannot read the enciphered portions of the AU and cannot retrieve the keys to decrypt the AU. This step is further explained in Figure 10.

Figure 10 shows the steps of the KTU Authentication process. Step 1001, which is shown in dashed lines because it is preferably optional, checks the identification of the IC card a second time. The identification information can be sent as part of the KTU data. However, this check is optional as it has already been performed once in step 905.

Step 1003 then decrypts KTU ciphertext 503 using the IC card's secret key (mkd_sk). The KTU Plaintext was previously encrypted using the intended card's public key (mkd_pk). This means that only the holder of the intended card's secret key could decrypt the encrypted message. The application provider obtains the intended IC card's public key either from the IC card itself (See Figure 4 and related text for a discussion of the mkd key set) or from a database holding the public keys. If the IC card cannot decrypt the KTU ciphertext properly then the KTU is not meant for that card and the application loading process halts. If the IC card does properly decipher the KTU ciphertext, then the process continues.

Step 1005 identifies an encrypted area of the application unit (AU). In the example of the KTU Plaintext described in connection with Figure 6, the IC card uses a relative starting address and area length field to determine the encrypted portion. Step 1005 also identifies which encryption technique was used to encrypt the identified portion so that the proper decryption technique can be used. For example, the technique could by single or triple DES. Alternatively, the technique could be a default technique used in the system and need not be identified.

Step 1007 then retrieves the key from KTU Plaintext and decrypts the identified portion with the identified decryption technique. This allows the IC card to have the decrypted portion of the AU which it will store in its EEPROM once all the encrypted portions have been decrypted.

Step 1009 checks if there are any other additional encrypted areas.

In the example described in Figure 3, there are three encrypted areas. The number of encrypted areas was a field in the example of Figure 6. However, the number of portions can be determined using other conventional means. If there are additional encrypted portions, the process jumps to step 1005. If there are no additional encrypted portions, then the process continues with step 1011.

Step 1011 then loads the decrypted AU into the memory of the IC card. The ALU has passed all of the authentication and decryption checks and the application can now properly reside on the IC card and be executed and used by the card user. While the different checks have been presented in a particular order in Figures 9 and 10, the checks can be performed in any order. While all of the described techniques used in conjunction with the ALU provide the best security, one or more of the individual techniques could be used for their individual purposes or combined with other conventional security techniques.

Figure 11 shows an example of a block diagram of an IC card chip upon which an ALU can be loaded and processed. An integrated circuit is located on an IC card for use. The IC card preferably includes a central processing unit 1101, a RAM 1103, an EEPROM 1105, a ROM 1107, a timer 1109, control logic 1111, an I/O port 1113 and security circuitry 1115, which are connected together by a conventional data bus.

Control logic 1111 in memory cards provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports. CPU 1101 with its control logic can perform calculations, access memory locations, modify memory contents, and manage input/output ports.

Some cards have a coprocessor for handling complex computations like cryptographic operations. Input/output ports 1113 are used under the control of a CPU and control logic, for communications between the card and a card interface device. Timer 1109 (which generates or provides a clock pulse) drives the control logic 1111 and CPU 1101 through the sequence of steps that accomplish memory access, memory reading or writing, processing, and data communication. A timer may be used to provide application features such as call duration. Security circuitry 1115 includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed ("blown") upon completion of testing to prevent later access. The AU data after the ALU has been authenticated and verified is stored in EEPROM 1105. The IC card private key will be stored in a secure memory location. The IC card public key and public key certificate is preferably stored in EEPROM 1105. The authentication process as described herein is performed by the CPU 1101.

Figure 11 also shows a possible configuration for the application provider, transmitting entity and for the CA. CPU 1101 present in the application provider encrypts the necessary information using encryption techniques described herein and performs the necessary data operations. CPU 1101 present in the certification authority is used to sign the Application Load Certificate and the public key certificate as described herein.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

For example, while loading an application is discussed herein, the same secure loading processes can apply to transmitting other types of data such as data blocks, database files, word processing documents or any other type of data need to be transmitted in a secure manner.

The scope of the present disclosure includes any novel feature or combination of features disclosed therein either explicitly or implicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or mitigates any or all of the problems addressed by the present invention. The application hereby gives notice that new claims may be formulated to such features during the prosecution of this application or of any such further application derived therefrom. In particular, with reference to the appended claims, features from dependant claims may be combined with those of the independent claims in any appropriate manner and not merely in the specific combinations enumerated in the claims.

ANNEX A

KEY TRANSFORMATION UNIT FOR AN IC CARD

ANNEX A TO THE DESCRIPTION

BACKGROUND OF INVENTION

Integrated circuit ("IC") cards are becoming increasingly used for many different purposes in the world today. An IC card (also called a smart card) typically is the size of a conventional credit card which contains a computer chip including a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain a single application or may contain multiple independent applications in its memory. MULTOS™ is a multiple application operating system which runs on IC cards. among other platforms, and allows multiple applications to be executed on the card itself. This allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an electronic cash card, is loaded with a single application when it is manufactured and before it is given to a card user. That application, however, cannot be modified or changed after the card is issued even if the modification is desired by the card user or card issuer. Moreover, if a card user wanted a variety of application functions to be performed by IC cards issued to him or her, such as both an electronic purse and a credit/debit function, the card user would be required to carry multiple physical cards on his or her person, which would be quite cumbersome and inconvenient. If an application developer or card user desired two

different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving terminal, making the transaction difficult, lengthy and inconvenient.

Therefore, it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card. Although multiple applications could be preselected and placed in the memory of the card during its production stage, it would also be beneficial to have the ability to load and delete applications for the card post-production as needed.

The increased flexibility and power of storing multiple applications on a single card create new challenges to be overcome concerning the integrity and security of the information (including application code and associated data) exchanged between the individual card and the application provider as well as within the entire system when loading and deleting applications. It would be beneficial to have the capability in the IC card system to exchange data among cards, card issuers, system operators and application providers securely and to load and delete applications securely at any time from a local terminal or remotely over a telephone line, Internet or intranet connection or other data conduit. Because these data transmission lines are not typically secure lines, a number of security and

ANNEX A TO THE DESCRIPTION

entity authentication techniques must be implemented to make sure that applications being sent over the transmission lines are not tampered with and are only loaded on the intended cards.

As mentioned, it is important -- particularly where there is a continuing wide availability of new applications to the cardholder -- that the system has the capability of adding applications onto the IC card subsequent to issuance. This is necessary to protect the longevity of the IC cards; otherwise, once an application becomes outdated, the card would be useless. It would be beneficial to allow the addition of applications from a remote location as well as from a direct connection to an application provider's terminal. For example, it would be beneficial for a card user to be able to plug his IC card into his home computer and download an application over the Internet. This type of remote loading of applications raises a number of security risks when transmitting the application code and related data over an unsecured communications line such as the Internet. At least three issues need to be addressed in a system which provides such a capability.

The first issue is to make sure that the IC card receiving the application is the intended IC card and not another IC card. The second issue is determining how the IC card can authenticate that the application came from the proper application provider and not an unknown third party. The third issue concerns preventing third parties from reading the application and making an unauthorized copy. If a portion of the application is encrypted to address the latter issue, the intended IC card needs to have access to the correct key to decrypt the application. In a system with many IC cards and additionally many application

ANNEX A TO THE DESCRIPTION

providers, a secure key transfer technique is required so that the intended IC card can use the correct key for the application which is received. These concerns are raised by both remote application loading as well as local terminal application loading.

Accordingly, it is an object of this invention to provide a key transfer and authentication technique and specifically to provide a secure IC-card system that allows for the secure transfer of smart card applications which may be loaded onto IC cards.

SUMMARY OF THE INVENTION

These and other objectives are achieved by the present invention which provides an IC card system and method for securely loading an application onto an IC card including providing a secret and public key pair for the IC card, encrypting at least a portion of the application using a transfer key, encrypting the transfer key using the IC card's public key to form a key transformation unit, transmitting the encrypted application and the key transformation unit to the IC card, decrypting the key transformation unit using the IC card's secret key to provide the transfer key, decrypting the encrypted application using the provided transfer key and storing the decrypted application on the IC card.

In a preferred embodiment, the secure loading system and method allows the application provider to encrypt two or more portions of the application to be transmitted with two or more different keys, encrypt the two or more keys with the public key of the IC card to form a key transformation unit including the

ANNEX A TO THE DESCRIPTION

locations of the encrypted portions. Both the encrypted application and the key transformation unit are sent to the IC card. Because the decryption keys are encrypted with the IC card's public key, only the IC card's secret key can decrypt the key transformation unit. The transfer keys and the locations of the encrypted portions are recovered from the decrypted key transformation unit and the application is decrypted using the recovered transfer keys. This ensures that only the intended IC card can decrypt and use the application which was transmitted to that IC card.

In a preferred embodiment. an application load certificate is also sent to the IC card which is receiving the application. The application load certificate contains the public key of the application provider encrypted by the secret key of the certificate authority ("CA"), or the entity that manages the overall security of the IC card system. The IC card then uses a certificate authority public key to make sure that the certificate was valid by attempting to verify the application load certificate with the CA's public key. The IC card then uses the recovered application provider's public key to verify that the application provider was in fact the originator of the application by verifying the sent application signature generated with the application provider's corresponding secret key.

BRIEF DESCRIPTION OF THE DRAWINGS

Further objects, features and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

ANNEX A TO THE DESCRIPTION

Fig. 1 is block diagram of the application loading system which loads an application from an application provider to an IC card;

Fig. 2 is a graphic representation of the contents of an Application Loading Unit;

Fig. 3 is a graphic representation of an Application Unit;

Fig. 4 is a flow chart of the steps for providing an individual key set for an IC card;

Fig. 5 is a graphic representation of a Key Transformation Unit;

Fig. 6 is a graphic representation of a Key Transformation Unit plaintext;

Fig. 7 is a graphic representation of the Application Load Certificate;

Fig. 8 is a graphic representation of the Application Unit being decrypted;

Fig. 9 is a flowchart illustrating the steps undertaken in processing the Application Load Unit;

Fig. 10 is a flowchart illustrating the steps undertaken in processing the KTU; and

Fig. 11 is a block diagram showing the components of an IC card which can receive and process and Application Load Unit.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection

with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

DETAILED DESCRIPTION OF THE INVENTION

It is beneficial to have the capability to load applications onto IC cards containing multiple application operating systems at any time during the lifetime of the IC card. This flexibility allows a user of a card to periodically add new applications to the IC card and also allows older applications to be updated with newer versions of the application when they are released. For example, a card user may start with an IC card that contains a purse, or electronic cash application (e.g., MONDEX™), being stored on his IC card. Some time after the user has the card, he or she may load an additional application onto the card such as a credit/debit application. Some time after loading the credit/debit application on the card, a new version of the credit/debit application may become available and the card user should be able to erase the old application on his IC card and replace it with the new version of the credit/debit application which may contain additional features.

The flexibility of loading applications at different times during the IC card's life cycle creates security issues with the process of loading applications onto the card. In a multiple application operating system environment, it is beneficial to be able to load applications both at terminals, such as a bank ATM machine, as well as over remote communication links, such as telephone lines, cable

ANNEX A TO THE DESCRIPTION

lines, the Internet, satellite or other communications means. When loading applications onto an IC card, the application provider and the card issuer (which could be the same entity) needs to provide security regarding the applications to be loaded. First, the application provider must make sure the application is only sent to the correct card user who is intended to receive the application. One solution to this problem is addressed in a related application entitled "Secure Multi-Application IC Card System Having Selective Loading and Deleting Capability" by Everett et al., filed February 12, 1998 and assigned to Mondex International, which is hereby incorporated by reference. Two additional security concerns also need to be addressed when loading an application from a remote source, or even from a local terminal, onto an IC card. First, the source of the application must be authenticated as the proper originator so that applications which may contain viruses or simply take up the limited storage memory in an IC card are not allowed to be loaded onto an IC card. Second, the application and associated data may contain private or trade secret information which needs to be encrypted so other people cannot view the contents of the encrypted application code and data. A portion of the application code and data may be secret while other portions are not. These concerns of authentication and protecting the contents of some or all of the application and associated data being loaded onto a card is addressed herein.

A number of encryption/decryption techniques are described herein. There are two basic types of encryption, symmetric encryption and asymmetric encryption. Symmetric encryption uses a secret key as part of a mathematical formula which encrypts data by transforming the data using the formula and key.

After the data is encrypted, another party can decrypt the encrypted data using the same secret key with a related decryption algorithm. Thus the same key is used for encryption and decryption so the technique is symmetric. A conventional example of a symmetric algorithm is DES.

Asymmetric encryption techniques use two different keys of a pair for encrypting and decrypting information. The two keys are normally referred to as a private or secret key and a public key. When data is encrypted with one key of the pair, the other key is used to decrypt the data. If a sender of data signs the data with his secret key, anyone with the public key can verify the message. Since public keys are typically known to the public, the contents of a data signed with a secret key cannot be protected but the origination of the data can be verified by determining if a particular secret key signed the data. This authentication process is termed a digital signature. If person A wanted to authenticate a message he was sending to person B, the person A would sign the document with his secret key. When person B received the message, he would use person A's public key to decipher the message. If the message was readable after the public key was applied to it, person B would know that the document was signed with secret key of person A. Thus, the origin of the message has been authenticated.

The asymmetric key set can also be used to protect the contents of a message. If person A wanted to send an encrypted message to person B that no one else could read, he would encrypt the data or message with person B's public key and send it to person B. Now only the holder of B's secret key could decrypt the data. If a combination of keys is used, a person could both authenticate and

ANNEX A TO THE DESCRIPTION

encrypt the message. The asymmetric pair of keys has some powerful applications with respect to card security and is more robust than symmetric encryption. However, asymmetric encryption is more processor costly that symmetric encryption. A example of an asymmetric encryption method is RSA.

A hybrid of symmetric encryption which makes the encryption method more powerful is to encrypt data using two symmetric keys. This technique is called triple DES which encodes data with symmetric key 1, decodes the data using symmetric key 2 (which in effect further encodes the data) and then further encodes the data using key 1 again. Once the data has arrived at its destination, key 1 is used to decode the data, key 2 is used to encode the data, and key 1 is used to decode the data. These extra steps of encoding and decoding make the technique more powerful and more difficult to properly decipher without both keys.

Figure 1 shows a block diagram of the entities used in a secure remote application loading process. The application provider 101 can be a card issuer, bank or other entity which provides application loading services. The application provider 101 initiates an application loading process onto IC card 103. Application Provider 101 is connected to data conduit 107 which is connected to interface device 105 (e.g., a terminal that communicates with an IC card). Data conduit 107 can be a telephone line, an intranet, the Internet, a satellite link or any other type of communications link. The application provider 101, which is remotely located from the IC card 103, desires to send and load an application to the IC card. However, because the data link is an open link and subject to third parties possibly intercepting or replacing applications being transmitted, security

ANNEX A TO THE DESCRIPTION

measures which authenticate the application itself, the application provider and the IC card must be used to ensure the integrity of the system. The Certificate Authority 109 may also be used to help authenticate that some data being transferred is part of an identified system.

In Figure 1, the application provider sends an application load unit 111 to the interface device 105 and finally to IC card 103. The ALU includes the application itself and security data required to authenticate and protect the application code and associated data. The ALU is discussed specifically in Figure 2 and in connection with the other figures herein. The ALU 111 also preferably contains Application Load Certificate (ALC) 113 data which is sent from the Certification Authority (CA) 109 to the application provider 101. The Certification Authority manages the overall security of the system by providing an Application Load Certificate for each application which is to be loaded onto an IC card. The application provider 101 and the IC card 103 both have individual public/secret keys sets provided to them. The authentication and security processes will now be described.

Figure 2 shows a diagram illustrating the components of an Application Load Unit which is sent from the application loader to the IC card during the application load process. The Application Load Unit (ALU) 201 contains an Application Unit (AU) 203, an Application Unit Signature (AU_S) 205, a Key Transformation Unit (KTU) 207 and an Application Load Certificate (ALC) 209. The ALU 201 is formatted in a conventional format used during data transmission. AU 203 contains the application code and data which are to be stored

ANNEX A TO THE DESCRIPTION

on the IC card, some or all of which is encrypted to protect a secret portion or portions of the code and/or data. AU 203 is described in further detail in connection with Figure 3.

AU_S 205 is the application code and data AU 203 digitally signed with the secret key of the application provider. The public key of the application provider is sent as part of the ALC 209 and is used to authenticate the application provider as the originator of the application. ALC 209 is made up of card identification information and the application provider's public key and is signed by the secret key of the certification authority. All these elements will be described in more detail below.

KTU 207 contains information relating to the encryption of the AU 203 (the code and data of the application) which allows the IC card to decrypt the designated portions so that the application and data can be accessed by the IC card but protects the data during transmission between the application provider and the IC card. KTU 207 is signed with a public key of the IC card for which the application is intended which ensures that only the intended IC card can decrypt the application code and data using the KTU information. This element will be described in connection with Figure 5.

Figure 3 shows a graphic representation of the Application Unit 203 which is part of the application load unit. The AU 203 contains both the program code and associated data which is to be loaded onto the IC card of the card user. The program code consists of a number of program instructions which will be executed by the microprocessor on the IC card. The program instructions can be

ANNEX A TO THE DESCRIPTION

written in any programming language which the operating system stored on the IC card can interpret.

For example, in the MULTOS system the program can be written in MEL™ (MULTOS Executable Language). Most applications have associated data which must be loaded onto the card. For instance, data which identifies the card user such as a person's name or account number may be loaded in a secure manner with the credit/debit application. An application provider may provide electronic cash represented by data as a promotion when installing an electronic purse application. Some or all of this data is desired to be kept secret from third parties. Additionally, the application code itself may be considered proprietary and portions may be desired to be kept secret from others. The use of a Key Transformation Unit (KTU) will allow an application provider to designate and encrypt selected portions of its application as confidential and protect it from third parties.

Application Unit portion 305 indicates the program code which is to be transferred from the application provider to the IC card. Application Unit portion 307 indicates the associated data which is to be transferred as part of the application to be loaded onto the IC card. In this example, three discrete areas of the application unit are shown to be encrypted using either single DES or triple DES. Any number of variations regarding the portions encrypted and the type of encryption can be employed using the techniques described herein.

In this example, encrypted location 309 shows the first portion of the Application Unit 203 which has been encrypted using a triple DES technique. The encryption process as described above involves using a symmetrical key and the

ANNEX A TO THE DESCRIPTION

conventionally known DES algorithm to transform the data. The data can later be recovered by applying the key to the known DES algorithm. Encrypted location 311 shows a second portion of the application unit 203 which has been encrypted using triple DES. Encrypted location 313 shows a third portion which is encrypted using single DES. Single DES requires less computation to decrypt and takes up less space as part of the KTU as described below. If the application unit were intercepted by a third party while it was being transmitted from the application loader to the IC card, the encrypted portions could not be read unless the third party had the correct keys. That information, therefore, is protected in the KTU.

The KTU is used to allow the IC card for which the application and associated data is intended to decrypt the encrypted portions of the Application Unit by describing which portions of the application unit are encrypted, which encryption algorithm was used and the key or keys to be used to decipher the text. This information is highly confidential between the application provider and the intended IC card and therefore is protected in a manner unique to the intended card. In order to encrypt the KTU which is part of the overall ALU being transmitted, an individual key set for the particular intended IC card is used. The key set and its generation will now be described.

One of the security operations performed at the CA is to generate an individualized key set for each IC card which is stored on the card. The keys are used for off-card verification (i.e., to verify that the card is an authentic card) and for secure data transportation. The key generation process is shown generally in Figure 4. The key set is made up of three different key data items: the card's

ANNEX A TO THE DESCRIPTION

secret key which is known only to the card, the card's public key which is stored on the card and the card's public key certificate which is the card's public key signed by one of the CA's secret keys. The individual keys of the key set are described in more detail below.

Step 401 stores a card specific transport secret key for the individual IC card in the memory of the card. This secret key is generated by the CA and loaded onto the card via a card acceptance device. Once stored on the card, the CA deletes from its own memory any data relating to the secret key. Thus, only the card itself knows its secret key. The data element containing the secret key information in the card is called "mkd_sk" which stands for MULTOS key data secret key.

Step 403 stores a card specific transport public key for the individual IC card in the memory of the card. This public key is preferably generated by the CA from the asymmetric encryption technique used to produce the secret key in step 401. The data element containing the card's public key information is called "mkd_pk" which stands for MULTOS key data public key.

Step 405 stores a card specific transport public key certificate for the individual IC card in the memory of the card. The data element containing the card's public key certificate information is called "mkd_pk_c" which stands for MULTOS key data public key certificate. This public key certificate is preferably generated by encrypting the transport public key mkd_pk with the secret key of the CA, indicated as follows:

$$mkd\_pk\_c = [mkd\_pk]_{CA\_sk}$$

ANNEX A TO THE DESCRIPTION

which means the individual card's public key certificate is formed by applying the CA's secret key to the individual card's public key. The process is carried out at the CA. The public key certificate is retained by the CA so that it can regenerate the public key as needed.

A terminal can read the public key certificate from the IC cards to verify that the CA had signed and therefore approved the individual IC card. This is accomplished by verifying the public key certificate with the public component of the CA key set used to sign the mkd_pk. The decrypted public key certificate can then be compared with the public key to verify that the key certificate was certified (signed) by the CA.

Figure 5 is a graphic depiction of the contents of KTU 207, which contains Header portion 501 and KTU Ciphertext portion 503. As shown in Figure 5, header information 501 includes, for example, identifier or permissions information 505 such as the application_id_no (application identification number), mcd_no (IC card no) and/or msm_control_data_date (the date the IC card was issued). Additional identifiers could also be included. These identifiers allow the system to verify that the IC card which receives the ALU is the intended IC card. The permissions data is discussed in detail in the above referenced related application.

KTU Ciphertext 503 corresponds to KTU Plaintext (not encrypted) encrypted with the public key mkd_pk of the intended IC card as shown in box 507. The KTU Plaintext in further described in Figure 6. The public key mkd_pk is obtained from the intended IC card by the application provider. The public key

ANNEX A TO THE DESCRIPTION

of an IC card is freely available to anyone and can be obtained directly from the card or from the CA. By signing the KTU Plaintext with the IC card public key, only the intended IC card can use its secret key of the public/secret key pair to decrypt the KTU Ciphertext. This means that only the intended IC card can determine the contents of the KTU plaint text, identify the encrypted portions of the application being loaded and use the keys provided to decrypt and recover the entire application and associate data. Because no other entity has the secret key of the IC card, the security and integrity of the program code and data being transmitted in ensured.

Figure 6 is a graphic representation of KTU Plaintext 601. KTU Plaintext 601 preferably includes identifier field 603, no_area_discriptors field 605, alg_id field 607, area_start field 609, area_length 611, key_length field 613, key_data field 615 and additional area and key fields depending upon the number of encrypted areas present in the Application Unit. Identifiers 603 contain identifying information of the Application Unit to which the KTU applies. No_area_descriptors 605 indicates how many different portions of the AU have been encrypted. In the example of Figure 3, the number or area descriptors would be three. Field 607 contains the algorithm identifier for the first area which has been encrypted. The algorithm could be DES or triple DES, for example. Field 609 indicates the start of the first encrypted area. This indication could be an offset from the start of the AU. For example, the offset could be 100 which means that the first area starts at the 100th byte of the Application Unit. Field 611 indicates the area length for the first encrypted portions. This field allows the microprocessor on

ANNEX A TO THE DESCRIPTION

the IC card to know how large an area has been encrypted and when coupled with the start of the area, allows the IC card microprocessor to decrypt the correct portion of the Application Unit. Filed 613 indicates the key length for the particular encrypted portion of the application unit. The length of the key will differ for different encryption techniques. The key length field allows the IC card to know the length of the key data. Field 615 indicates the key data for the particular encrypted portion. The key data is used with the algorithm identity and the location of the encoded portion to decode the encrypted portion. If more than one encrypted area is indicated, then additional data referring of the algorithm, start location, length, key length and key data will be present in the KTU Plaintext. While a number of fields have been described, not all the fields are necessary for the invention. The most important field, however, is the key data itself.

Figure 7 is a graphic representation of the Application Load Certificate (ALC) 209. ALC 209 includes a header 701 and the Application Provider Public Key 703. Header 701 and Application Provider Public Key 703 are then signed (encrypted) with the CA secret key. Thus, the ALC 209 must be provided by the CA to the application provider for each application loaded because only the CA knows the CA private key. Header 701 contains information regarding the application provider and the IC card for which the application is intended. The ALC 209 is placed in the correct ALU by the application provider which can use the identification information. Application Provider Public Key 703 is provided to the CA along with the identification data. The CA then signs this information after verifying its authenticity and returns the signed ALC to the application provider.

The IC card, when it receives the ALC 209 as part of the ALU 201, will open the ALC 209 with the public key of the CA. This ensures that the CA signed the application load certificate and that it is genuine. After decrypting the information, the header identification information 701 is checked and the application provider public key is recovered. This public key will be used to verify that the application and code which is to be loaded onto the IC card originated with the proper application provider.

Figure 8 is a graphic representation of the use of the application provider's public key to decrypt the signed AU 205 in order to verify that AU 203 was signed by the application provider. AU signed 205 is verified with the Application Provider Public Key 801. The recovered AU 803 is then compared with AU 203. If the data blocks match, then the IC card has verified that the application provider signed (encrypted) the application unit and the application is genuine. This authentication is valid because only the application provider has its own secret key. The IC card can process this information because the application provider's public key is provided to it as part of the application load certificate 209 which is signed by the CA. Therefore, it does not need to retrieve the public key from an external location to authenticate the application.

Figure 9 shows a flow chart of the steps for processing the Application Load Unit when it is received by the IC card. Prior to receiving the ALU, identity checks as to the identity of the IC card can be performed if desired. The ALU processing techniques provide a number of further verifications including verifying that the application being loaded is: (1) from the correct application

provider, (2) being loaded on the intended card and (3) certified by the CA. The ALU processing techniques also allow the transportation of transport decryption keys which enable the IC card to decrypt portions of the program code and associated data in a secure manner. In step 901, the IC card receives the ALU from the application provider. The ALU can be transmitted via a terminal connection, contactless connection, telephone, computer, intranet, Internet or any other communication means. The ALU is placed in the EEPROM of the IC card along with header information indicating the starting addresses of AU 203, AU signed 205, the KTU 207 and ALC 209. Alternatively, the IC card could determine the relative address locations of these four units.

Step 903 decrypts the ALC 209 with the CA public key. Each IC card preferably stores in its memory a copy of the CA public key because it is used in many transactions. Alternatively, the IC card could obtain the public key from a known storage location. If the CA public key successfully verifies the ALC 209, then the IC card has verified that the CA has signed the ALC 209 with its secret key and thus the Application Load Certificate is proper. If the IC card cannot verify the ALC successfully, then the ALC was not signed by the CA and the certificate is not proper. The application loading process would then end.

Step 905 then checks the identity of IC card against the identification information sent in the application load certificate to make sure the card is intended to receive the application. This permissions checking is described in the related patent application identified above. If there is no match of identification data, the application loading process ends. If the identification data does match, then the

ANNEX A TO THE DESCRIPTION

process continues.

Step 907 uses the application providers public key which was recovered from the verified ALC to verify the AU signature 205. When the ALU was generated by the application provider, the application unit 203 was signed with the application provider's secret key. The application provider then provides its public key to IC card through the ALC. The IC card then verifies the AU signed 205. If the ALU is successfully verified, then it is accepted as having been generated by the application provider. Because the application provider's public key is part of the ALC which is signed by the CA. the CA can make sure that the proper public key has been provided to the IC card. This unique key interaction between the application provider, CA and the intended IC card ensures that no counterfeit or unapproved applications or data are loaded onto an IC card which is part of the secure system.

Step 911 then processes a KTU authentication check which further verifies that only the intended card has received the application. The KTU authentication check makes sure that if a third party does somehow intercept the ALU, the third party cannot read the enciphered portions of the AU and cannot retrieve the keys to decrypt the AU. This step is further explained in Figure 10.

Figure 10 shows the steps of the KTU Authentication process. Step 1001, which is shown in dashed lines because it is preferably optional, checks the identification of the IC card a second time. The identification information can be sent as part of the KTU data. However, this check is optional as it has already been performed once in step 905.

ANNEX A TO THE DESCRIPTION

Step 1003 then decrypts KTU ciphertext 503 using the IC card's secret key (mkd_sk). The KTU Plaintext was previously encrypted using the intended card's public key (mkd_pk). This means that only the holder of the intended card's secret key could decrypt the encrypted message. The application provider obtains the intended IC card's public key either from the IC card itself (See Figure 4 and related text for a discussion of the mkd key set) or from a database holding the public keys. If the IC card cannot decrypt the KTU ciphertext properly then the KTU is not meant for that card and the application loading process halts. If the IC card does properly decipher the KTU ciphertext, then the process continues.

Step 1005 identifies an encrypted area of the application unit (AU). In the example of the KTU Plaintext described in connection with Figure 6, the IC card uses a relative starting address and area length field to determine the encrypted portion. Step 1005 also identifies which encryption technique was used to encrypt the identified portion so that the proper decryption technique can be used. For example, the technique could by single or triple DES. Alternatively, the technique could be a default technique used in the system and need not be identified.

Step 1007 then retrieves the key from KTU Plaintext and decrypts the identified portion with the identified decryption technique. This allows the IC card to have the decrypted portion of the AU which it will store in its static memory once all the encrypted portions have been decrypted.

Step 1009 checks if there are any other additional encrypted areas. In the example described in Figure 3, there are three encrypted areas. The number

of encrypted areas was a field in the example of Figure 6. However, the number of portions can be determined using other conventional means. If there are additional encrypted portions, the process jumps to step 1005. If there are no additional encrypted portions, then the process continues with step 1011.

Step 1011 then loads the decrypted AU into the memory of the IC card. The ALU has passed all of the authentication and decryption checks and the application can now properly reside on the IC card and be executed and used by the card user. While the different checks have been presented in a particular order in Figures 9 and 10, the checks can be performed in any order. While all of the described techniques used in conjunction with the ALU provide the best security, one or more of the individual techniques could be used for their individual purposes or combined with other conventional security techniques.

Figure 11 shows an example of a block diagram of an IC card chip upon which an ALU can be loaded and processed. An integrated circuit is located on an IC card for use. The IC card preferably includes a central processing unit 1101, a RAM 1103, an EEPROM 1105, a ROM 1107, a timer 1109, control logic unit 1111, an I/O port 1113 and security circuitry 1115, which are connected together by a conventional data bus.

Control logic 1111 in memory cards provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports. CPU 1101 with its control logic can perform calculations, access memory locations, modify memory contents, and manage input/output ports. Some cards have a coprocessor for handling complex computations like performing

ANNEX A TO THE DESCRIPTION

cryptographic operations. Input/output ports 1113 are used under the control of a CPU and control logic, for communications between the card and a card interface device. Timer 1109 (which generates or provides a clock pulse) drives the control logic 1111 and CPU 1101 through the sequence of steps that accomplish memory access, memory reading or writing, processing, and data communication. A timer may be used to provide application features such as call duration. Security circuitry 1115 includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed ("blown") upon completion of testing to prevent later access. The AU data after the ALU has been authenticated and verified is stored in EEPROM 1105. The authentication process as described herein is performed by the CPU 1101.

Figure 11 also shows a possible configuration for the integrated circuit chip for the application provider and for the certification authority. CPU 1101 present in the IC chip for the application provider encrypts the necessary information using encryption techniques described herein and performs the necessary data operations. CPU 1101 at the certification authority is used to sign the Application Load Certificate as described herein.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

For example, while loading an application is discussed herein, the

same secure loading process can apply to transmitting other types of data such as data blocks, database files, word processing documents or any other type of data need to be transmitted in a secure manner.

I CLAIM:

ANNEX A TO THE DESCRIPTION

1. A method for securely loading an application onto an IC card comprising the steps of:

providing a secret key and public key pair for said IC card;

encrypting at least a portion of said application using a transfer key;

encrypting said transfer key using said IC card's public key to form a key transformation unit;

transmitting said encrypted application and said key transformation unit to said IC card;

decrypting said key transformation unit using said IC card's secret key to recover said transfer key; and

decrypting said encrypted application using said recovered transfer key.

2. The method of claim 1, further including the step of storing said decrypted application on said IC card.

3. The method of claim 1, wherein said encryption technique using said transfer key transfer key is symmetric.

4. The method of claim 3, wherein said symmetric technique is DES.

5. The method of claim 1, wherein said IC card's public and private keys are provided using an asymmetric technique.

6. The method of claim 5, wherein said asymmetric technique is RSA.

7. The method of claim 1, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

8. The method of claim 1, further including the steps of enciphering a second portion of said application exclusive of said at least a portion of said application.

9. The method of claim 8, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

10. The method of claim 8, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

11. The method of claim 8, wherein said key transformation unit indicates the location of said second portion of said application.

ANNEX A TO THE DESCRIPTION

12. The method of claim 1, wherein said key transformation unit indicates the location of said at least a portion of said application.

13. The method of claim 1, wherein said key transformation unit indicates the number of encrypted portions of said application.

14. The method of claim 1, further including the steps of providing a public key and secret key set for an application provider; providing a public and secret key set for a certification authority; encrypting said application provider's public key using said certificate authorities' secret key to produce an application load certificate; further signing said encrypted application using said application provider's secret key to produce a signed application and transmitting said signed application and said application load certificate to said IC card.

15. The method of claim 14, further including the step of the IC card verifying said application load certificate with said certification authority's public key.

16. The method of claim 15, further including the steps of verifying the signed encrypted application using the application provider's public key from said decrypted application load certificate.

ANNEX A TO THE DESCRIPTION

17. The method of claim 16, wherein said verified application signature is compared to sent encrypted application to determine if they are equivalent.

18. An IC card system comprising:

at least one IC card;

an application provider for providing an application to said at least one IC card;

a communications link coupled to said at least one IC card and said application provider;

a public key and secret key set generated for said IC card;

a transport key generated for use by said applications provider; and

an application, wherein at least a portion of said application is encrypted by said application provider using said transport key; said transport key is encrypted using said IC card's public key to form a key transformation unit; wherein said encrypted application and said key transformation unit are then transmitted to said IC card over said communications link; said transmitted key transformation unit is decrypted using said IC card's private key to recover said transport key; and said transmitted application is decrypted using said recovered transport key to recover said application.

19. The system of claim 18, wherein said recovered application is stored on said card.

ANNEX A TO THE DESCRIPTION

20. The system of claim 18, wherein said encryption technique using said transfer key transfer key is symmetric.

21. The system of claim 20, wherein said symmetric technique is DES.

22. The system of claim 18, wherein said IC card's public and private keys are provided using an asymmetric technique.

23. The system of claim 22, wherein said asymmetric technique is RSA.

24. The system of claim 18, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

25. The system of claim 18. further including the steps of enciphering a second portion of said application independently of said at least a portion of said application.

26. The system of claim 25, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

27. The system of claim 25, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

ANNEX A TO THE DESCRIPTION

28. The system of claim 25, wherein said key transformation unit indicates the location of said second portion of said application.

29. The system of claim 18, wherein said key transformation unit indicates the location of at least a portion of said application.

30. The system of claim 18, wherein said key transformation unit indicates the number of encrypted portions of said application.

31. The system of claim 18, further including a certification authority, wherein a public key and secret key set is provided for an application provider; a public and secret key set is provided for said certification authority; said certificate authority's secret key is used to sign said application provider's public key to produce an application load certificate; said application provider's secret key is used to further sign said encrypted application to produce a signed encrypted application and said signed encrypted application and said application load certificate is transmitted to said IC card.

32. The system of claim 31, wherein the IC card verifies said application load certificate with said certification authority's public key.

ANNEX A TO THE DESCRIPTION

33. The system of claim 32, wherein said IC card verifies the signed encrypted application using the application provider's public key from said verified application load certificate.

34. The system of claim 33, wherein said verified application signature is compared to said encrypted application to determine if they are equivalent.

35. A method for transmitting data in a secure manner from a first microprocessor based device to a second microprocessor based device, comprising the steps of:

encrypting at least a portion of said data at said first device using a transfer key;

encrypting said transfer key with a second key at said first device to form a key transformation unit;

transmitting said encrypted data and said key transformation unit to said second device;

decrypting said key transformation unit at said second device to recover said transfer key; and

decrypting said encrypted data using said recovered transfer key.

36. The method of claim 35, further including the step of storing said decrypted data in said second device.

ANNEX A TO THE DESCRIPTION

37. The method of claim 35, wherein said second key is from a public key and private key set used in asymmetric encryption.

38. The method of claim 35, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

39. The method of claim 35, further including the steps of enciphering a second portion of said application independently of said at least a portion of said application.

40. The method of claim 39, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

41. The method of claim 39, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

42. The method of claim 39, wherein said key transformation unit indicates the location of said second portion of said application.

43. The method of claim 35, wherein said key transformation unit indicates the location of said at least a portion of said application.

ANNEX A TO THE DESCRIPTION

44. The method of claim 35, further including the steps of providing a public key and secret key set for an application provider; providing a public and secret key set for a certification authority; signing said application provider's public key using said certificate authority's secret key to produce an application load certificate; further signing said encrypted application using said application provider's secret key to produce a signed encrypted application and transmitting said signed application and said application load certificate to said IC card.

45. A method for processing a data transmission comprising the steps of:

receiving said data transmission comprising an application encrypted with a first key and a key transformation unit encrypted with a second key, wherein said key transformation unit comprises said first key;

decrypting said key transformation unit to recover said first key;

decrypting said encrypted application using said first key; and

storing said decrypted application.

46. The method of claim 45, wherein said second key is from a public key and private key set used in asymmetric encryption.

47. The method of claim 45, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

ANNEX A TO THE DESCRIPTION

48. The method of claim 45, further including the steps of enciphering a second portion of said application independently of said at least a portion of said application.

49. The method of claim 48, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

50. The method of claim 48, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

51. The method of claim 48, wherein said key transformation unit indicates the location of said second portion of said application.

52. The method of claim 45, wherein said key transformation unit indicates the location of said at least a portion of said application.

53. The method of claim 45, further including the steps of providing a public key and secret key set for an application provider; providing a public and secret key set for a certification authority; signing said application provider's public key using said certificate authorities' secret key to produce an application load certificate; further encrypting said encrypted application using said application provider's secret key to produce a signed encrypted application and transmitting

said signed application and said application load certificate to said IC card.

54. The method of claim 53, further including the step of the IC card verifying said application load certificate with said certification authority's public key.

55. The method of claim 54, further including the steps of verifying the signed encrypted application using the application provider's public key from said verified application load certificate.

56. The method of claim 55, wherein said verified application signature is compared to said encrypted application to determine if they are equivalent.

57. An apparatus for processing a data transmission comprising the steps of:

means for receiving said data transmission comprising an application encrypted with a first key and a key transformation unit encrypted with a second key, wherein said key transformation unit comprises said first key;

means for decrypting said key transformation unit to recover said first key;

means for decrypting said encrypted application using said first key; and

means for storing said decrypted application.

58. The apparatus of claim 57, wherein said second key is from a public key and private key set used in asymmetric encryption.

59. The apparatus of claim 57, wherein said key transformation unit further indicates the technique used to encrypt said at least a portion of said application.

60. The apparatus of claim 57, further including means for enciphering a second portion of said application exclusive of said at least a portion of said application.

61. The apparatus of claim 60, wherein said second portion is encrypted using a second encryption technique and said key transformation unit indicates said second encryption technique.

62. The apparatus of claim 60, wherein said second portion is encrypted using a second key and said key transformation unit indicates said second key.

63. The apparatus of claim 60, wherein said key transformation unit indicates the location of said second portion of said application.

64. The apparatus of claim 57, wherein said key transformation unit indicates the location of said at least a portion of said application.

ANNEX A TO THE DESCRIPTION

65. The apparatus of claim 60, further including means for verifying an application load certificate with said certification authority's public key.

66. The apparatus of claim 65, further including means for verifying the signed encrypted application using an application provider's public key located in said verified application load certificate.

67. The apparatus of claim 66, wherein said verified application signature is compared to the said encrypted application to determine if they are equivalent.

ANNEX A TO THE DESCRIPTION

ABSTRACT OF THE DISCLOSURE

A multi-application IC card system and method is disclosed providing a secure data transmission technique. The method is used, for example, to load an application from an application provider, which could be remote, to an IC card. At least a portion of the application is encrypted using a transfer key. The transfer key is then encrypted using the public key of a public/secret key pair of the intended IC card to form a key transformation unit. The encrypted application and key transformation unit are then sent to the IC card and the IC card decrypts the key transformation unit using its secret key. The transfer key is then recovered and used to decrypt the encrypted application. The application can then by stored on the IC card and accessed by the card user.

WE CLAIM:

1. A method for transporting data onto an integrated circuit card by using an individualized key set for said card, comprising the steps of:

storing a private key and public key pair unique to said integrated circuit card in said memory located on said integrated circuit card;

retrieving said stored public key from said integrated circuit card;

encrypting at least a portion of said data to be transported onto said card, using said retrieved public key;

transmitting said encrypted data to said integrated circuit card; and

decrypting said encrypted data using said integrated circuit card's private key to recover said transported data.

2. The method of claim 1, further including the step of storing said decrypted data on said integrated circuit card.

3. The method of claim 1 or claim 2, wherein a certification authority digitally signs said integrated circuit card's public key to produce a public key certificate unique to said card and stored thereon, and wherein said public key certificate is verified prior to said transmitting step.

4. The method of claim 3, wherein said public key certificate is verified with said certification authority's stored public key prior to said transmitting steps.

5. The method of claim 3 or 4, wherein said retrieved public key certificate is recovered and compared with said stored public key.

6. The method of any preceding claim, wherein said integrated circuit card's public and private keys are provided using an asymmetric technique.

7. The method of claim 6, wherein said asymmetric technique is RSA.

8. A method performed by an integrated circuit card for processing incoming data transmission to said integrated circuit card by using an individualized key set for the card, comprising the steps of:

receiving said data transmission comprising data encrypted with a public key stored on said integrated circuit card, said public key forming part of said individualized key set;

retrieving a unique private key for said integrated circuit card which is part of said individualized key set; and

decrypting said encrypted data with said unique private key to recover said data.

9. The method of claim 8, further including the step of storing said decrypted data on said integrated circuit card.

10. The method of claim 8 or 9, wherein said individualized key set is generated by asymmetric encryption.

11. The method of any of claims 8 to 10, wherein a certification authority digitally signs said integrated circuit card's public key to produce a public key certificate unique to said card and stored thereon, and wherein said public key certificate is verified prior to said transmitting step.

12. The method of claim 11, wherein said public key certificate is retrieved prior to said transmitting steps.

13. The method of claim 11 or 12, wherein said retrieved public key certificate is verified using said certification authority's stored public key.

14. An apparatus located on an integrated circuit card by using an individualized key set for said card for processing an incoming secure data transmission comprising:

means for receiving said data transmission comprising data encrypted with a public key stored on said integrated circuit card, said public key forming part of said individualized key set;

means for retrieving a unique public key for said integrated circuit card which is part of said individualized key set; and

means for decrypting said encrypted data with said unique private key to recover said data.

15. The apparatus of claim 14, further comprising means for storing said data on said integrated circuit card.

16. The apparatus of claim 14 or 15, further including means for retrieving a public key certificate which is generated by a certificate authority digitally signing said unique public key.

17. The apparatus of claim 16, further including means for transmitting said public key certificate prior to said receiving means receiving.

18. The apparatus of claim 16 or 17, wherein said transmitted public key certificate is verified using said certification authority's stored public key.

19. A method of transporting data onto an integrated circuit card by using an individualized key set for the card, comprising the steps of:

providing a first unique private and public key pair for a certification authority;

storing a second unique private and public key pair which

form said individualized key set for said integrated circuit card in a memory located on said integrated circuit card;

encrypting said second public key with said first certification authority's private key to form a public key certificate;

storing said public key certificate on said integrated circuit card;

retrieving said stored public key certificate from said integrated circuit card;

verifying said public key certificate with said first public key to ensure that said public key certificate is valid;

encrypting at least a portion of said data using said retrieved second public key;

transporting said encrypted data to said integrated circuit card; and

decrypting said encrypted data using said second private key to retrieve said data.

20. The method of claim 19, wherein said data comprises an application.

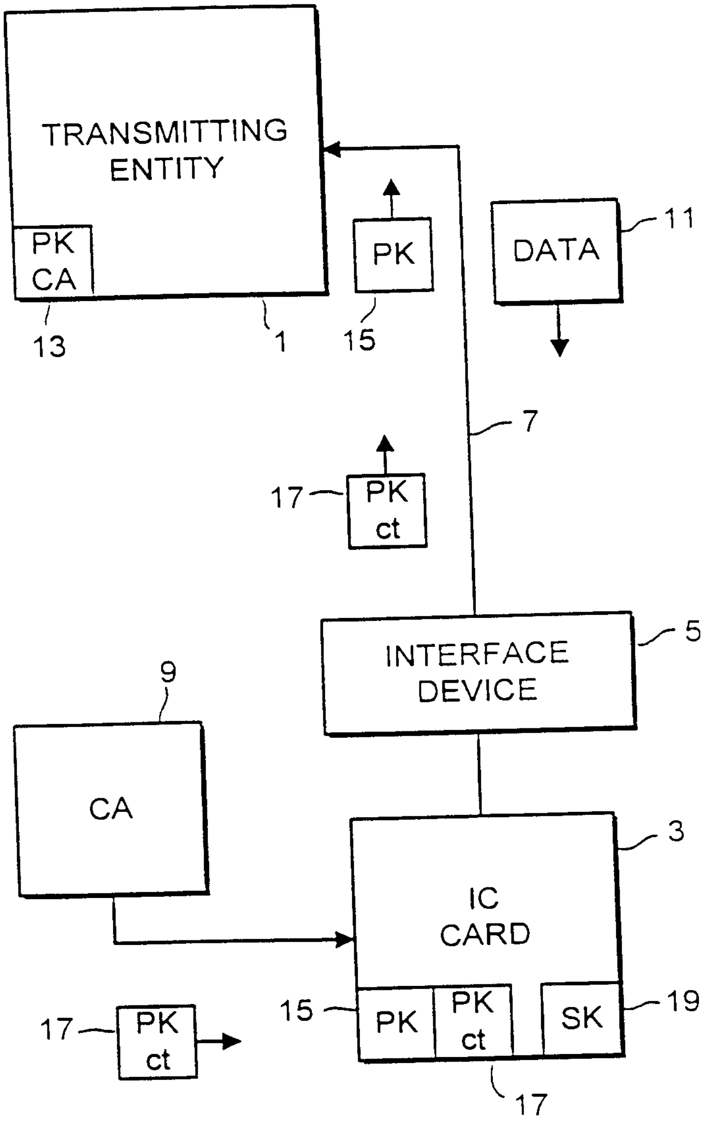

TRANSMITTING ENTITY
PK CA
13
1
PK
15
DATA
11
7
17
PK ct
INTERFACE DEVICE
5
9
CA
IC CARD
3
15
PK
PK ct
SK
19
17
17
PK ct

FIG. 1A

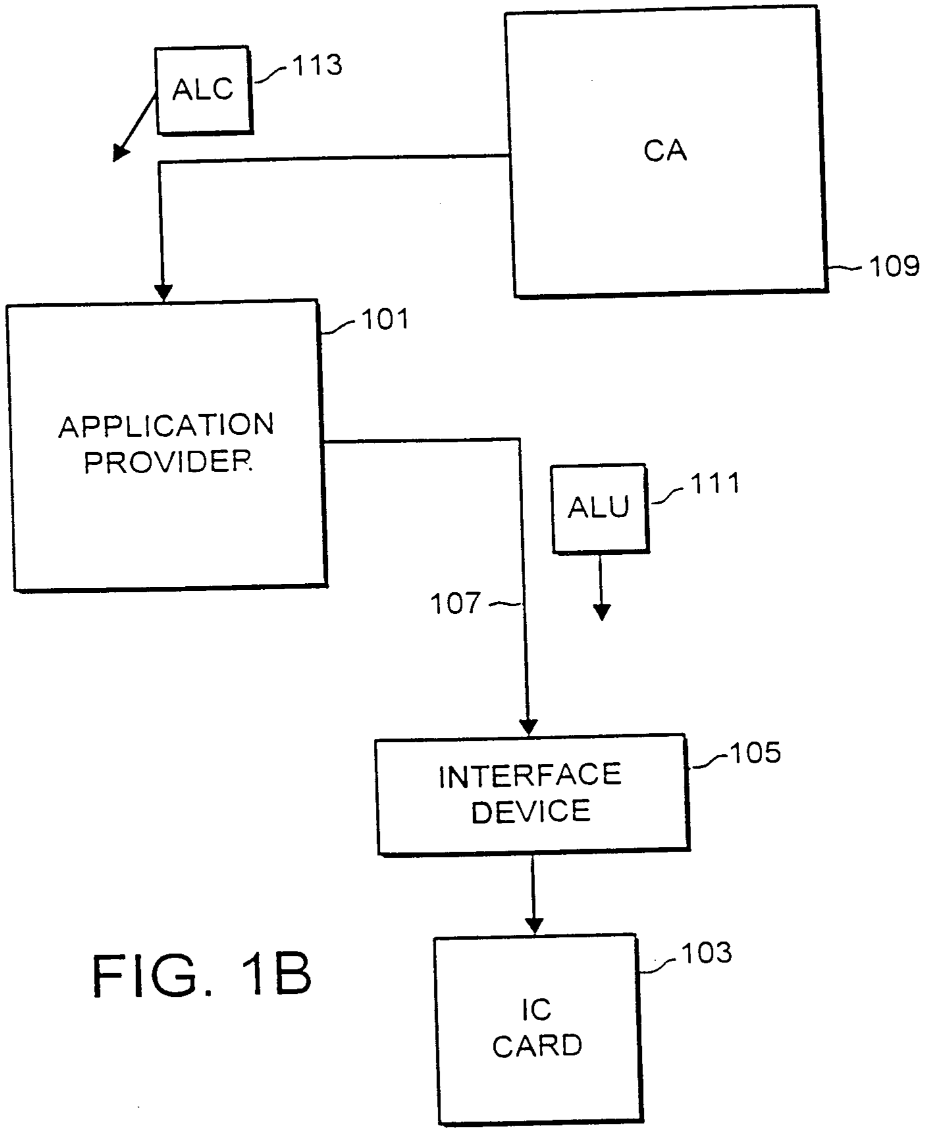

ALC
113
CA
109
101
APPLICATION PROVIDER
ALU
111
107
INTERFACE DEVICE
105
IC CARD
103

FIG. 1B

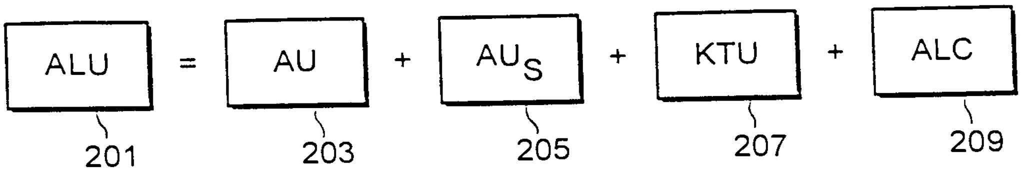

ALU
=
AU
+
AU$_S$
+
KTU
+
ALC
201
203
205
207
209

FIG. 2

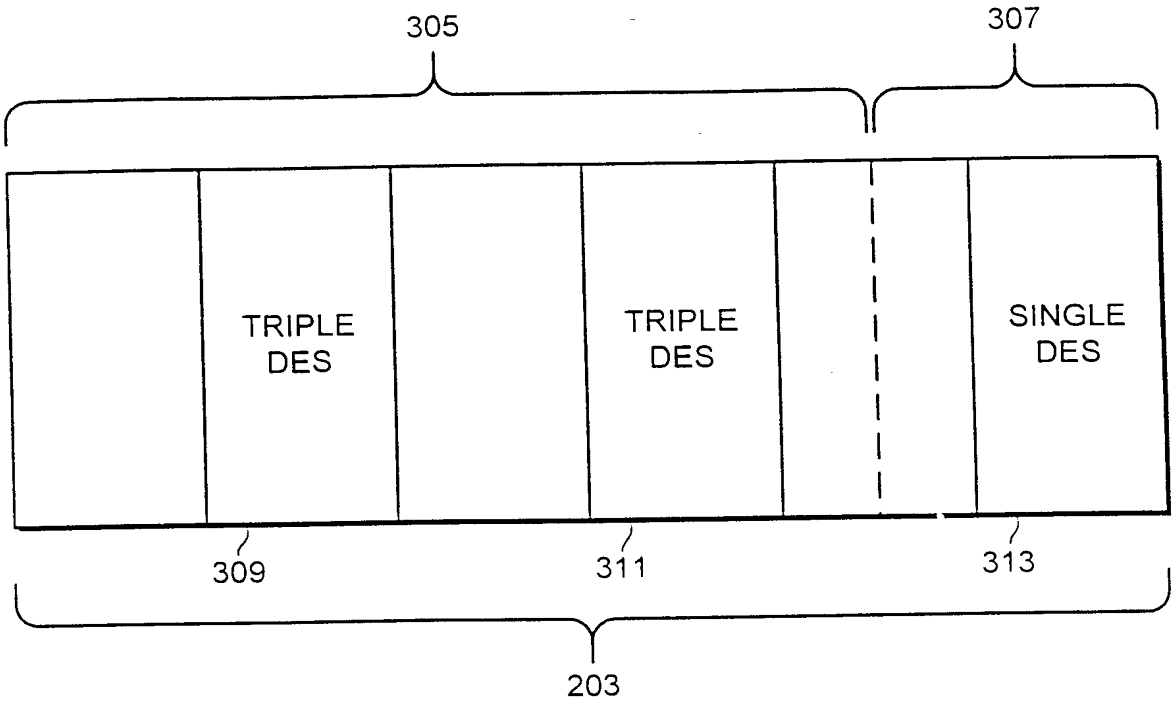

305
307
TRIPLE DES
TRIPLE DES
SINGLE DES
309
311
313
203

FIG. 3

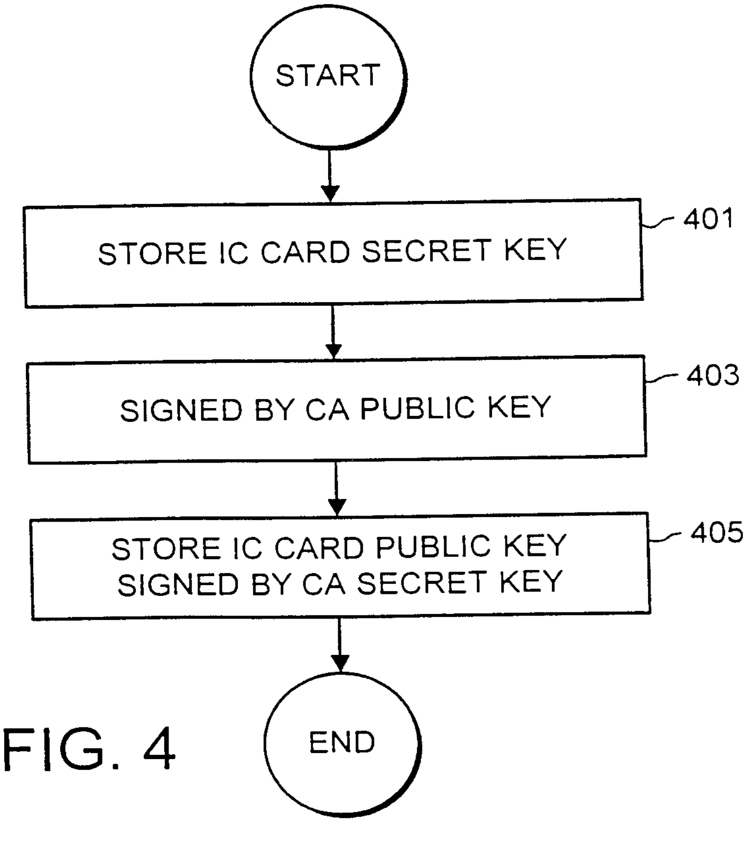

START
STORE IC CARD SECRET KEY
401
SIGNED BY CA PUBLIC KEY
403
STORE IC CARD PUBLIC KEY SIGNED BY CA SECRET KEY
405
END

FIG. 4

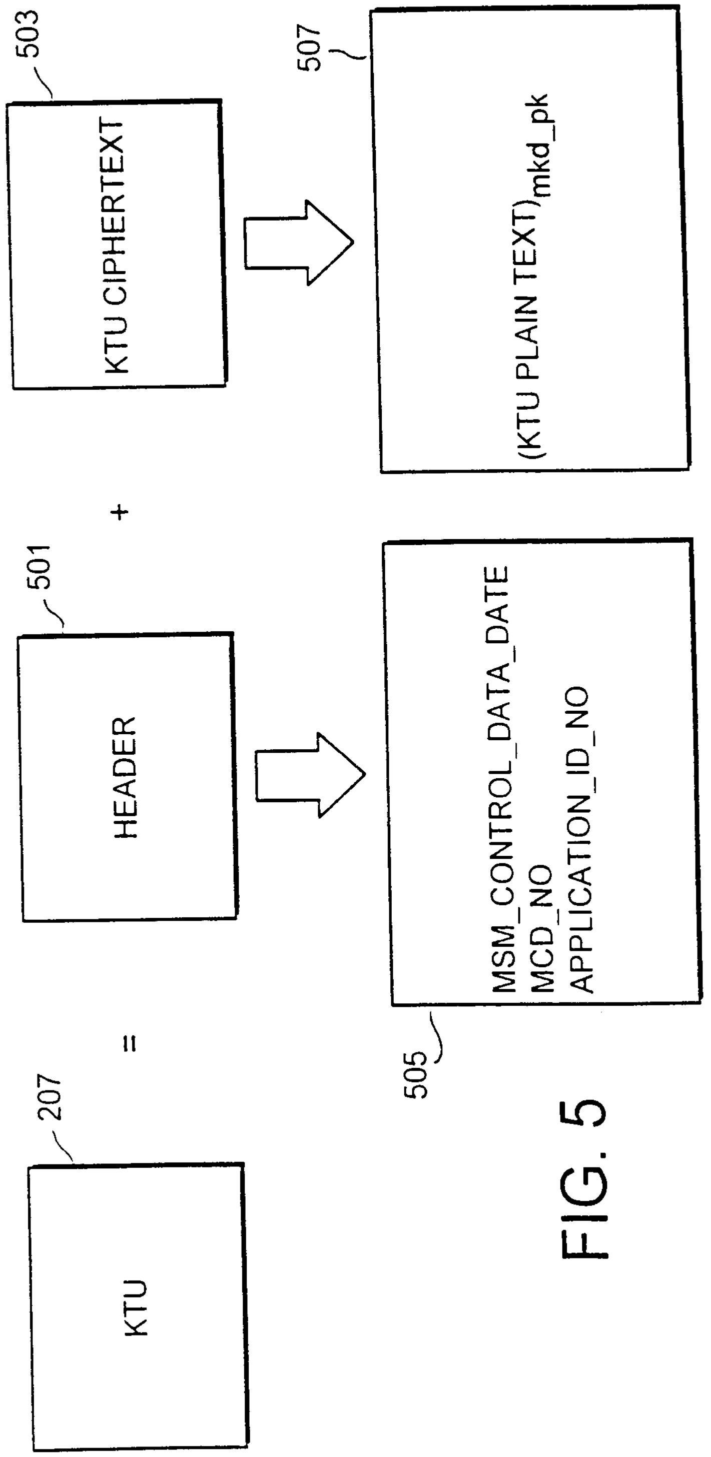

207
KTU
=
501
HEADER
+
503
KTU CIPHERTEXT
505
MSM_CONTROL_DATA_DATE
MCD_NO
APPLICATION_ID_NO
507
(KTU PLAIN TEXT)mkd_pk

FIG. 5

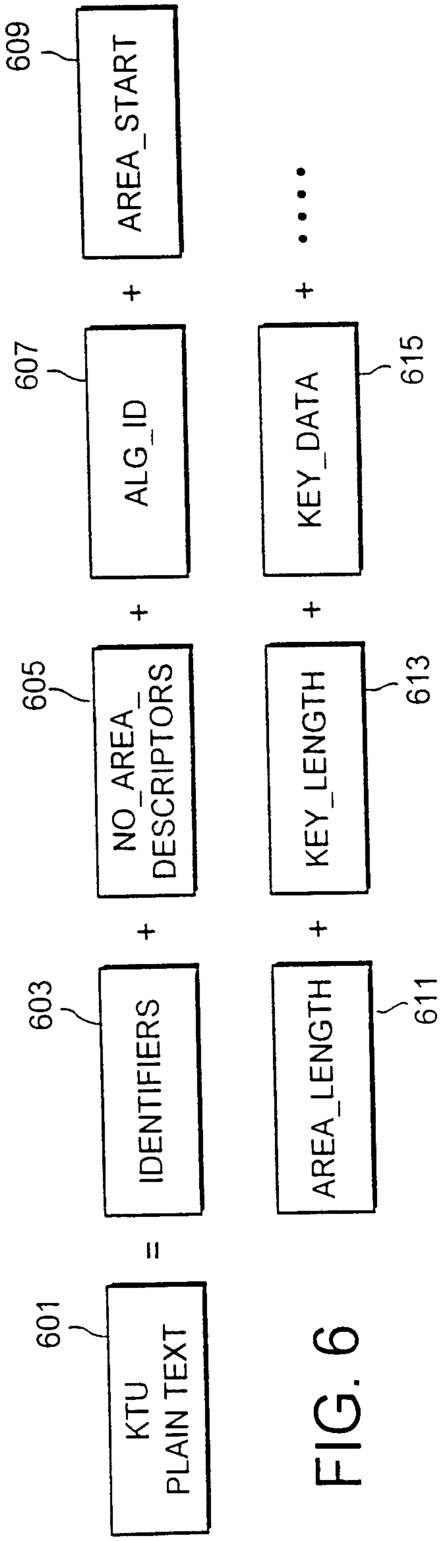

601
KTU
PLAIN TEXT
=
603
IDENTIFIERS
+
605
NO_AREA_
DESCRIPTORS
+
607
ALG_ID
+
609
AREA_START
+
611
AREA_LENGTH
+
613
KEY_LENGTH
+
615
KEY_DATA
+
....

FIG. 6

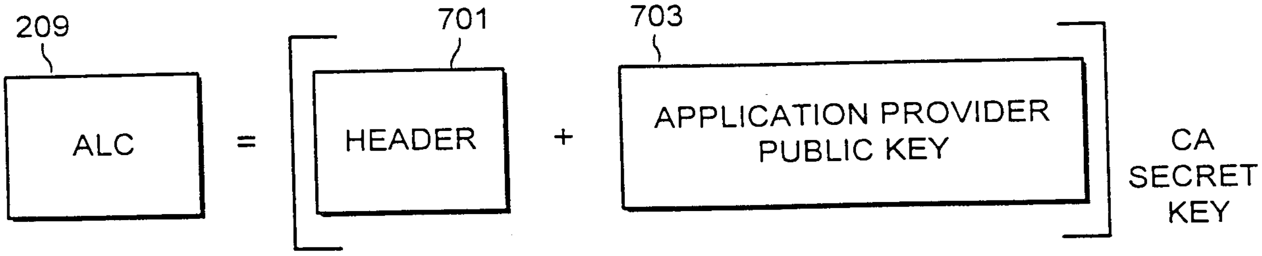


FIG. 7

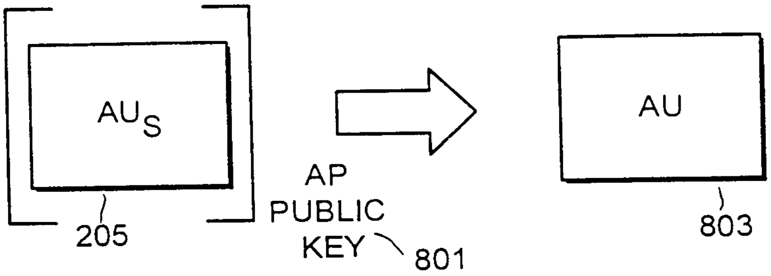


FIG. 8

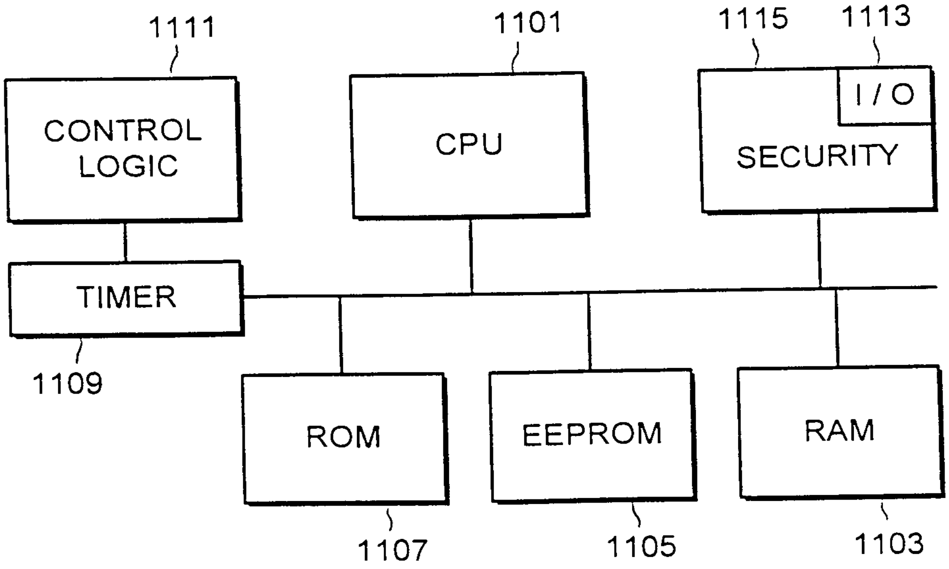


FIG. 11

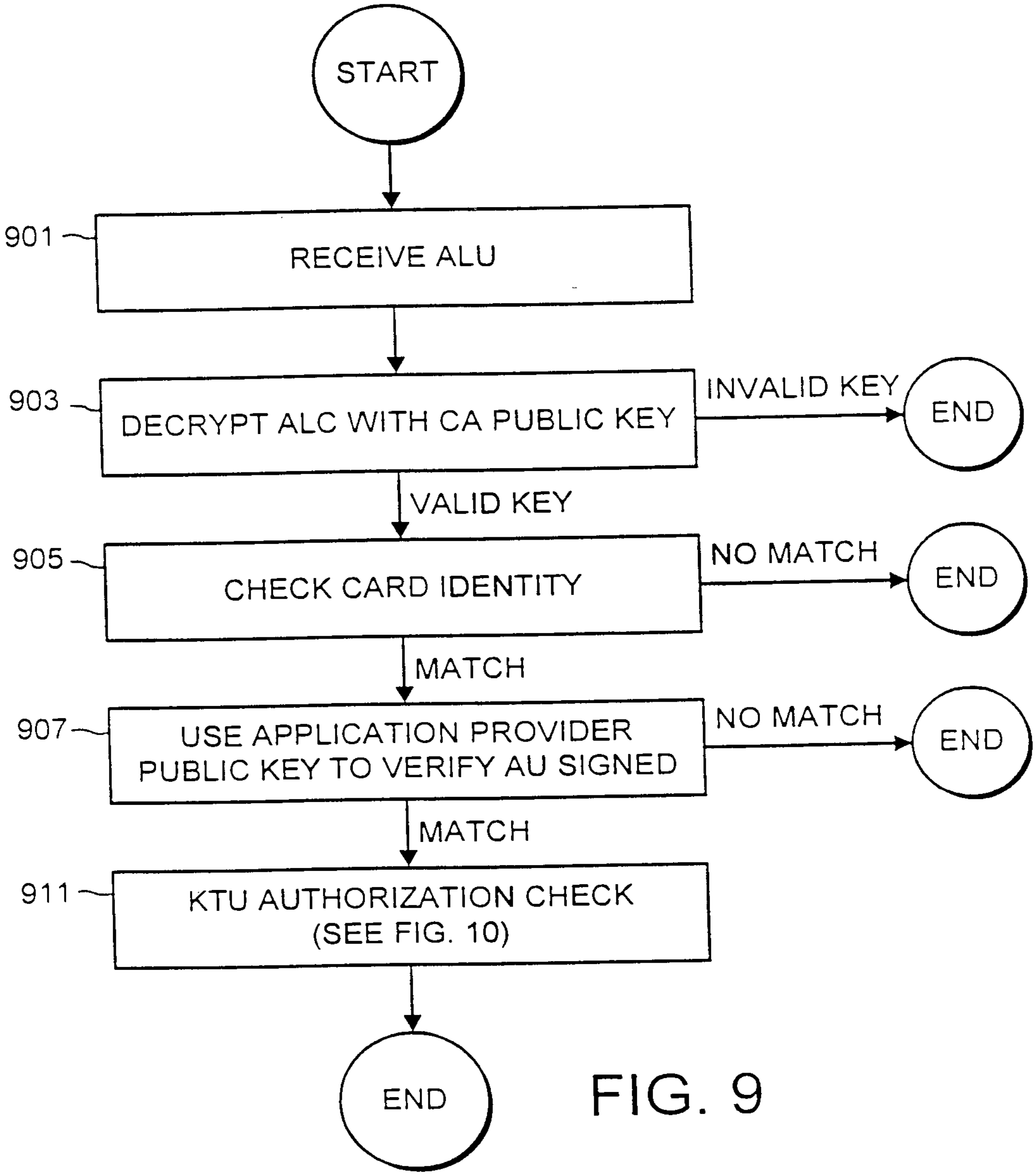

START
901
RECEIVE ALU
903
DECRYPT ALC WITH CA PUBLIC KEY
INVALID KEY
END
VALID KEY
905
CHECK CARD IDENTITY
NO MATCH
END
MATCH
907
USE APPLICATION PROVIDER PUBLIC KEY TO VERIFY AU SIGNED
NO MATCH
END
MATCH
911
KTU AUTHORIZATION CHECK (SEE FIG. 10)
END

FIG. 9

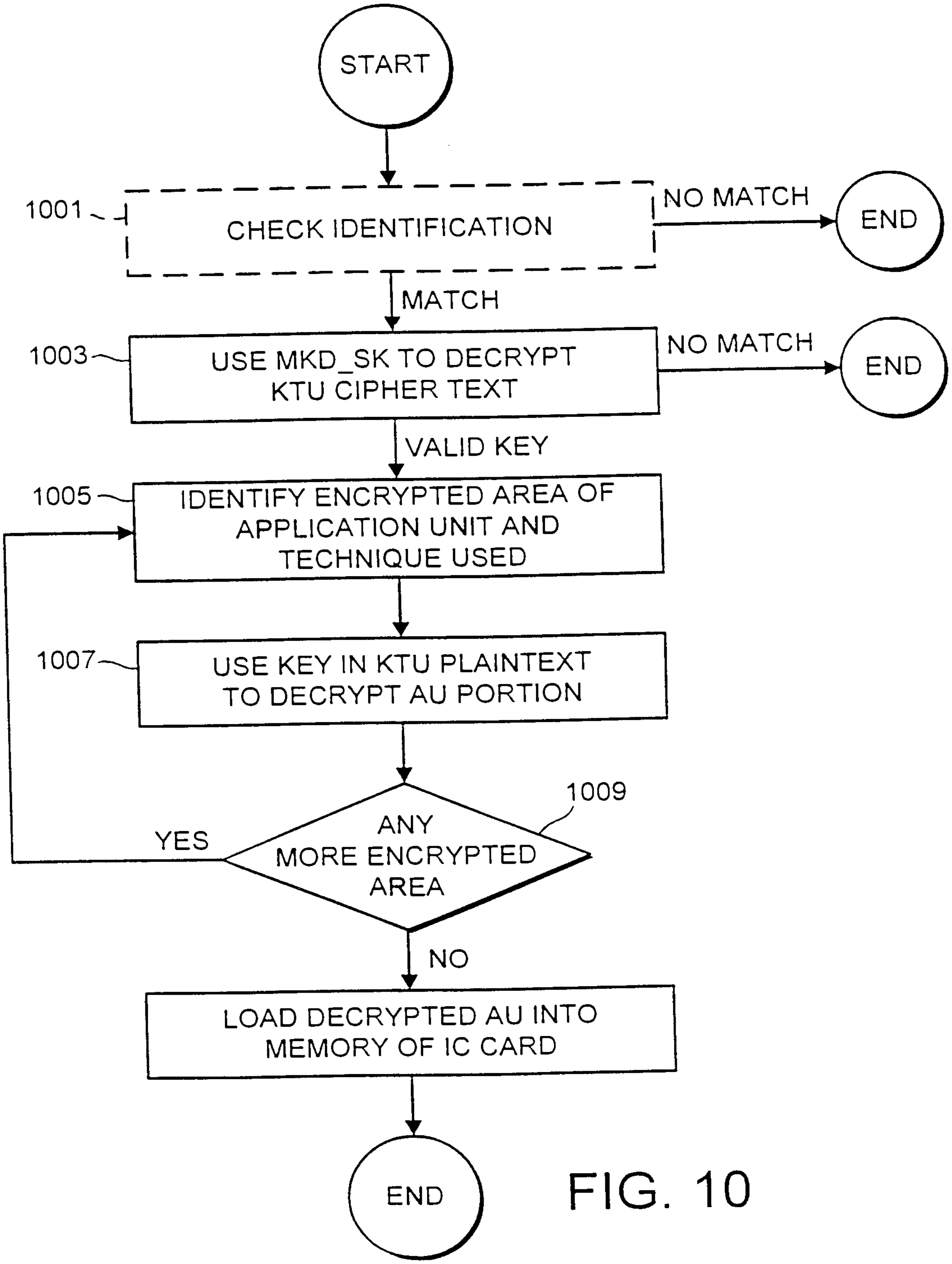

START
1001
CHECK IDENTIFICATION
NO MATCH
END
MATCH
1003
USE MKD_SK TO DECRYPT KTU CIPHER TEXT
NO MATCH
END
VALID KEY
1005
IDENTIFY ENCRYPTED AREA OF APPLICATION UNIT AND TECHNIQUE USED
1007
USE KEY IN KTU PLAINTEXT TO DECRYPT AU PORTION
1009
ANY MORE ENCRYPTED AREA
YES
NO
LOAD DECRYPTED AU INTO MEMORY OF IC CARD
END

FIG. 10

ANNEX A TO THE DRAWINGS

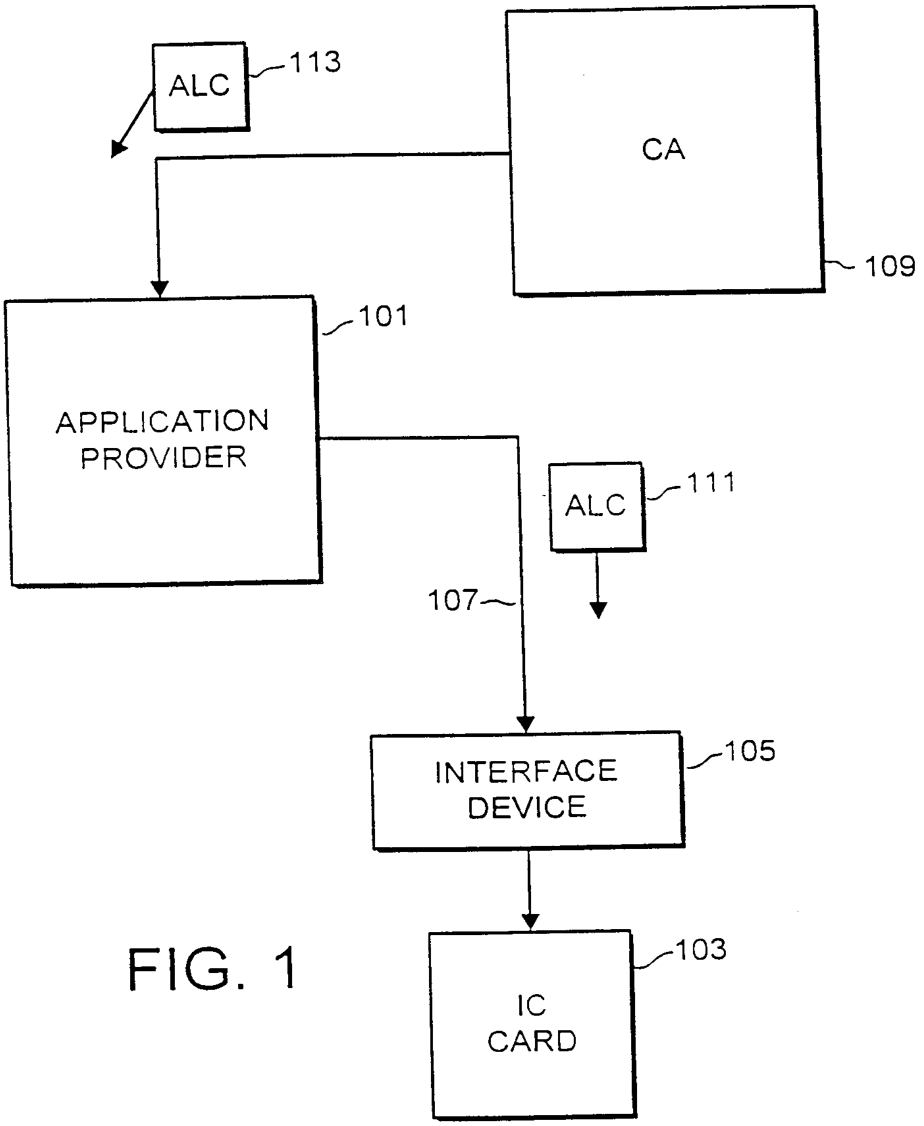


FIG. 1

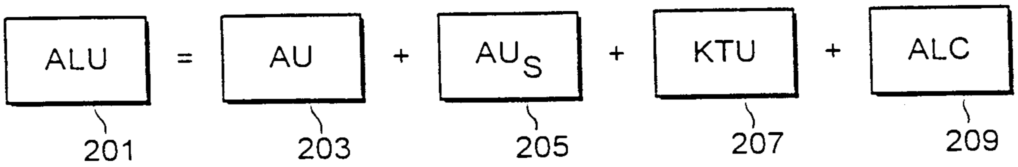


FIG. 2

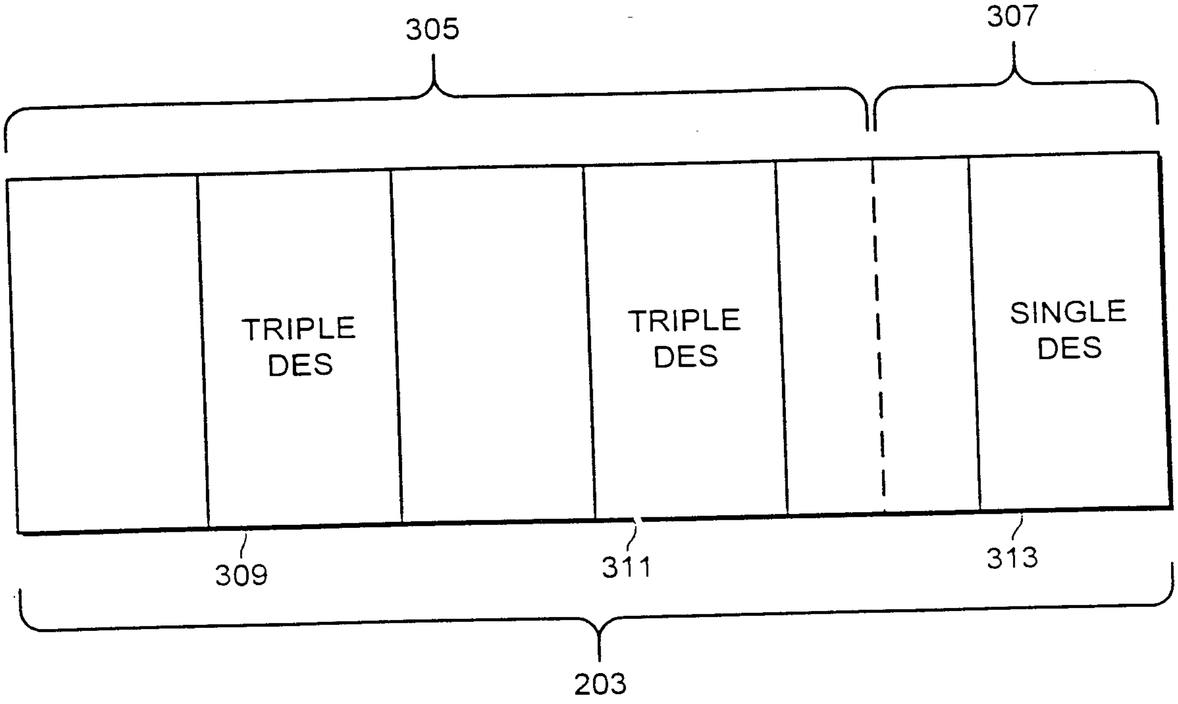

305
307
TRIPLE DES
TRIPLE DES
SINGLE DES
309
311
313
203

FIG. 3

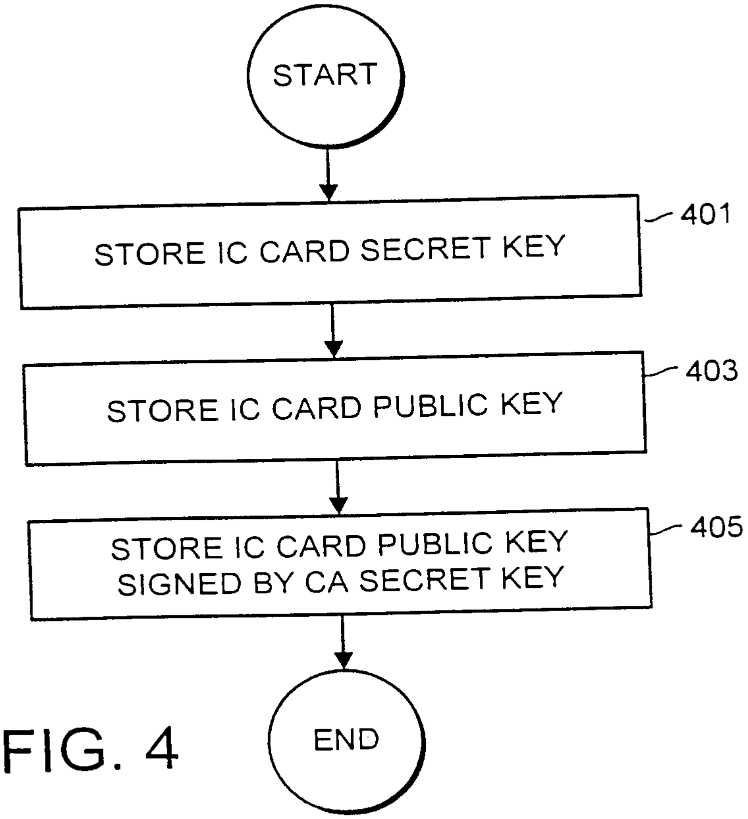

START
STORE IC CARD SECRET KEY
401
STORE IC CARD PUBLIC KEY
403
STORE IC CARD PUBLIC KEY SIGNED BY CA SECRET KEY
405
END

FIG. 4

ANNEX A TO THE DRAWINGS

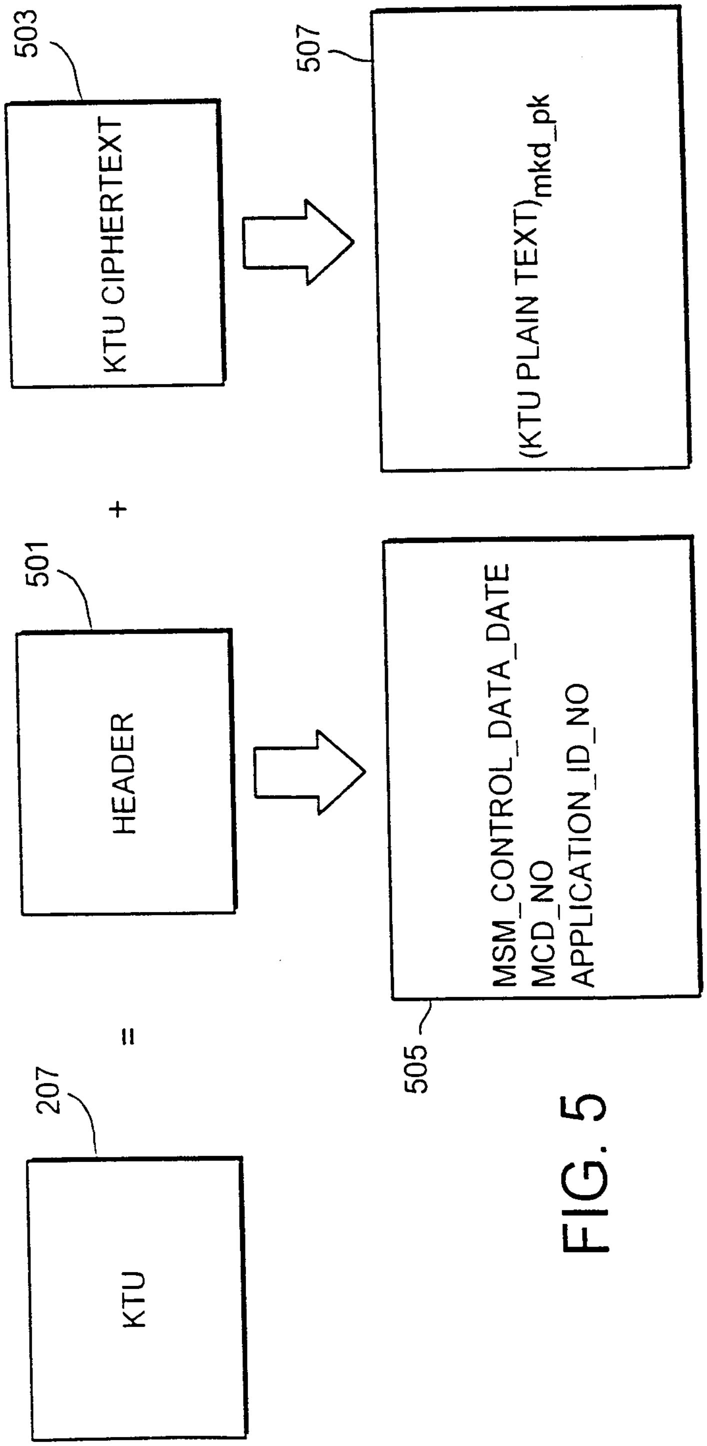


FIG. 5

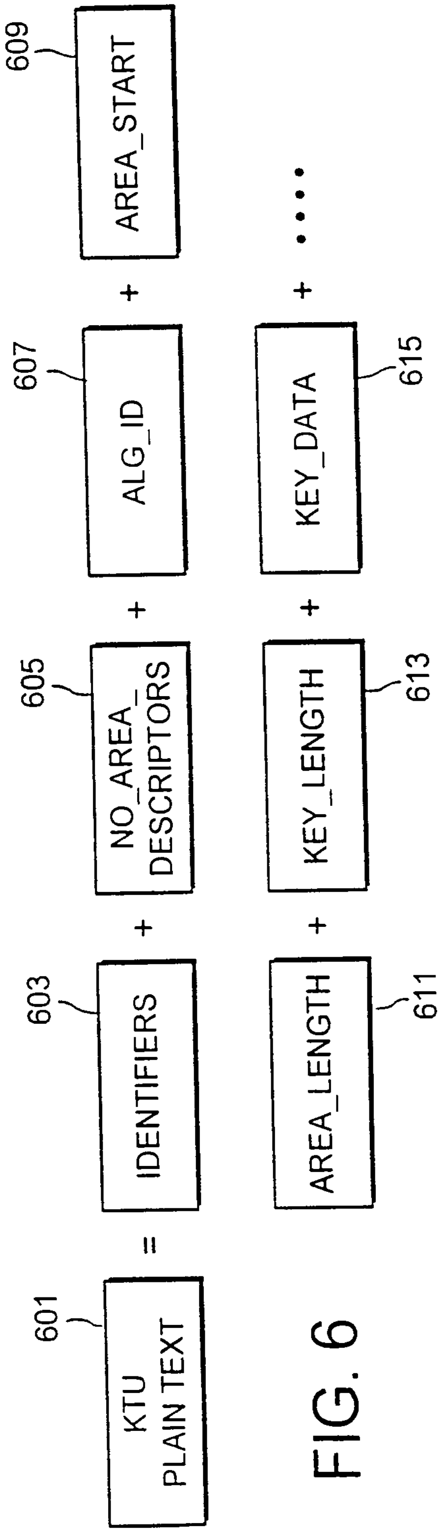


FIG. 6

ANNEX A TO THE DRAWINGS

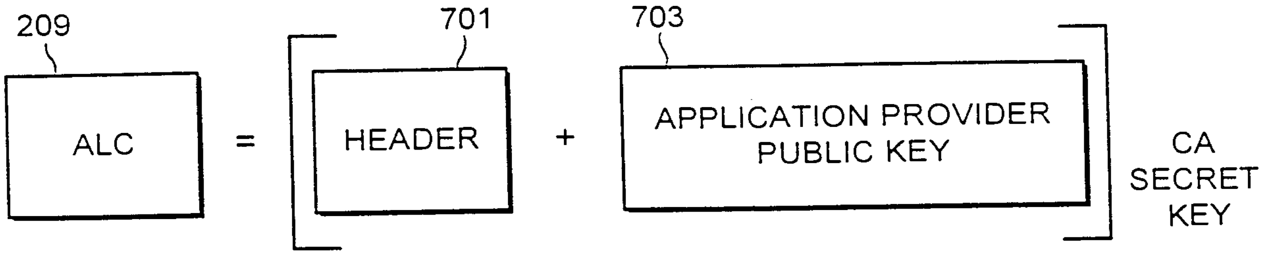


FIG. 7

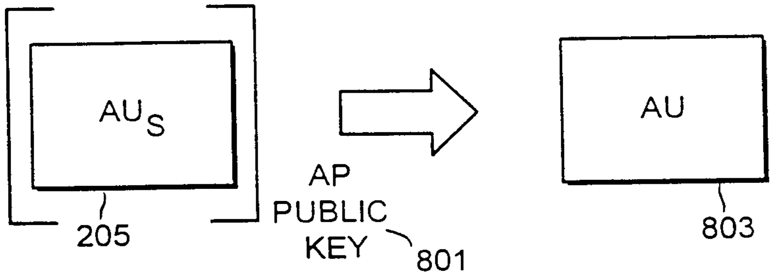


FIG. 8

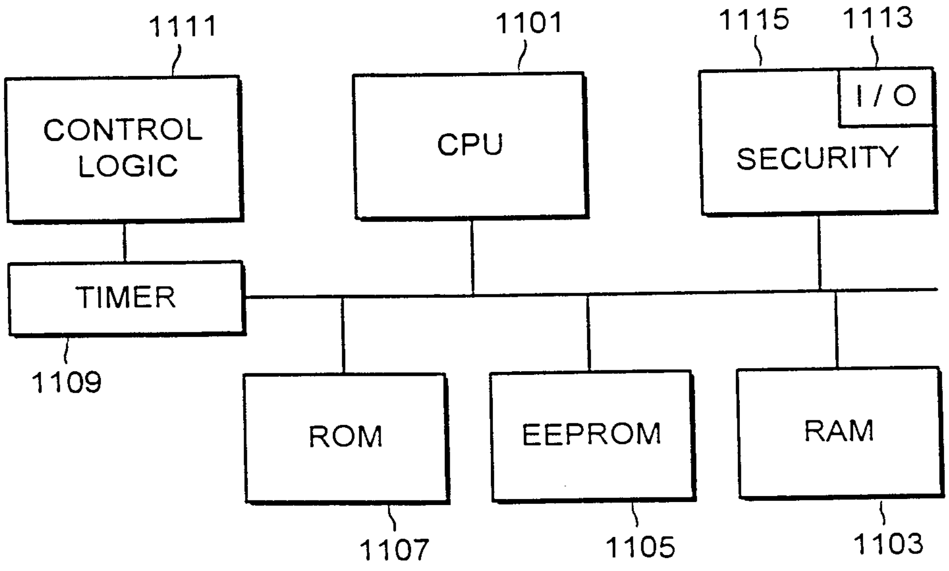


FIG. 11

ANNEX A TO THE DRAWINGS

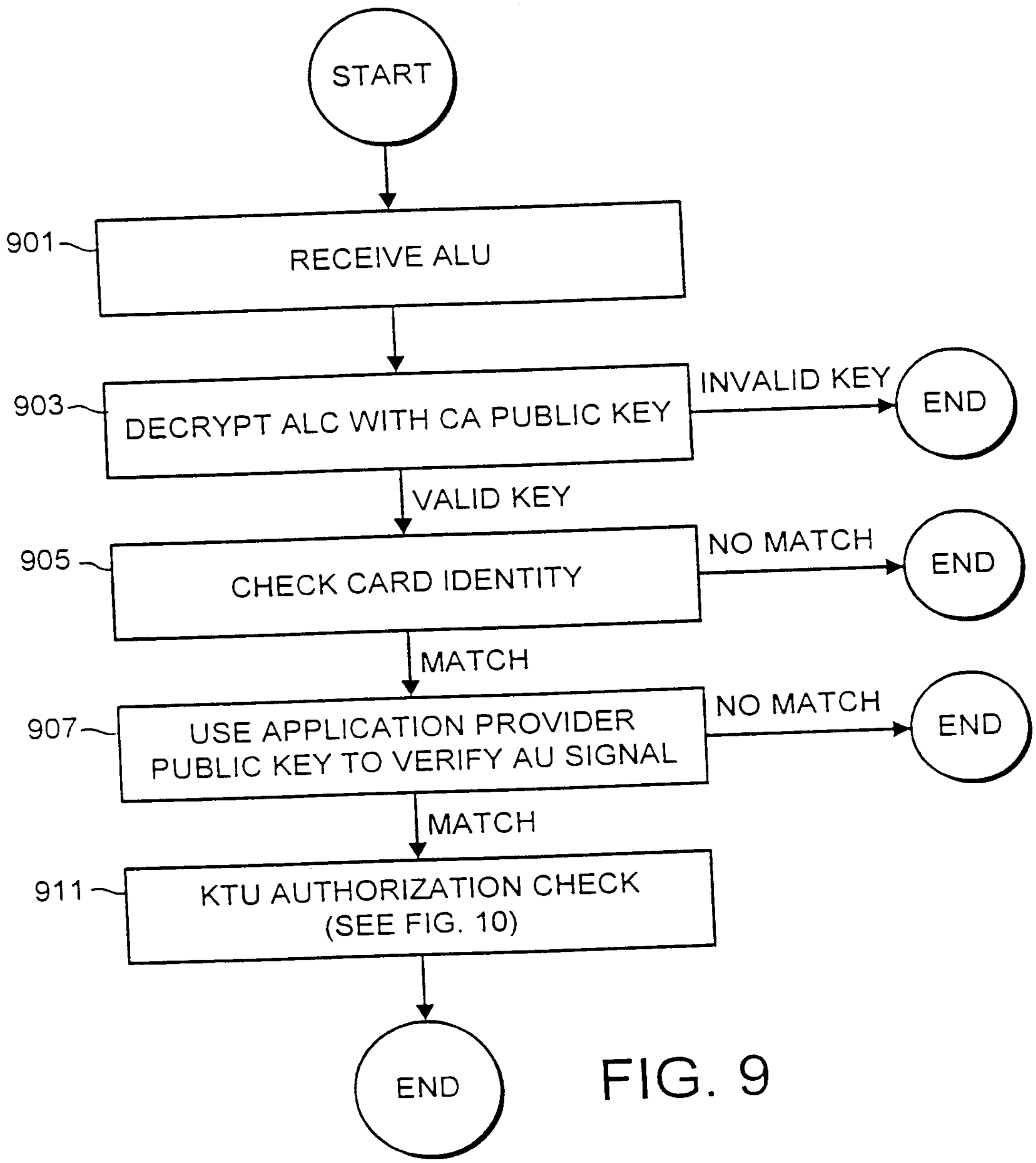

START
901
RECEIVE ALU
903
DECRYPT ALC WITH CA PUBLIC KEY
INVALID KEY
END
VALID KEY
905
CHECK CARD IDENTITY
NO MATCH
END
MATCH
907
USE APPLICATION PROVIDER PUBLIC KEY TO VERIFY AU SIGNAL
NO MATCH
END
MATCH
911
KTU AUTHORIZATION CHECK (SEE FIG. 10)
END

FIG. 9

ANNEX A TO THE DRAWINGS

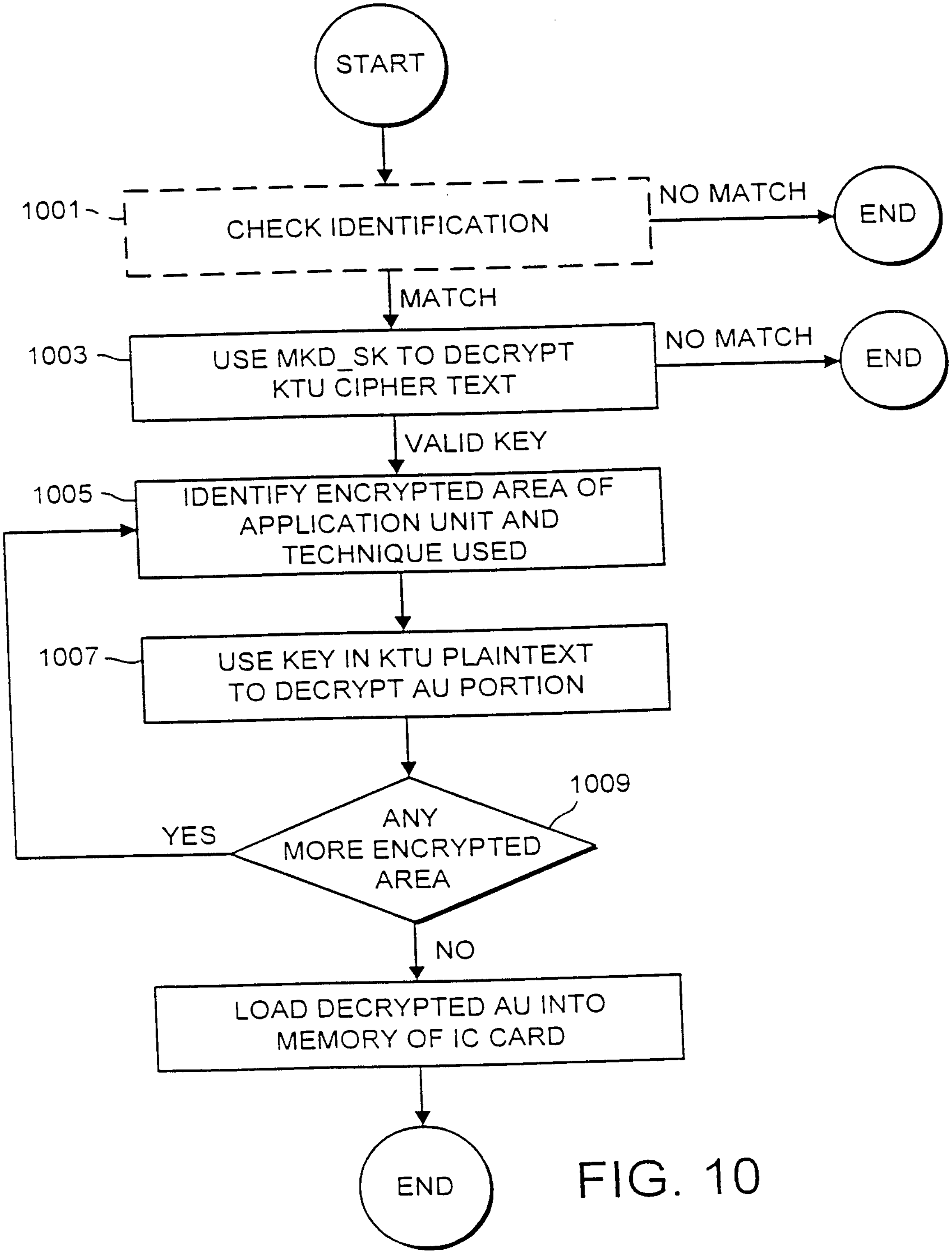

START
1001
CHECK IDENTIFICATION
NO MATCH
END
MATCH
1003
USE MKD_SK TO DECRYPT KTU CIPHER TEXT
NO MATCH
END
VALID KEY
1005
IDENTIFY ENCRYPTED AREA OF APPLICATION UNIT AND TECHNIQUE USED
1007
USE KEY IN KTU PLAINTEXT TO DECRYPT AU PORTION
1009
ANY MORE ENCRYPTED AREA
YES
NO
LOAD DECRYPTED AU INTO MEMORY OF IC CARD
END

FIG. 10

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04N 7/16, G06K 19/07	A1	(11) International Publication Number: WO 99/22516 (43) International Publication Date: 6 May 1999 (06.05.99)

(21) International Application Number: PCT/IB98/01766

(22) International Filing Date: 27 October 1998 (27.10.98)

(30) Priority Data:
97402561.1 28 October 1997 (28.10.97) EP

(71) Applicant *(for all designated States except US)*: CANAL+ SOCIETE ANONYME [FR/FR]; 85/89, quai André Citroën, F–75711 Paris Cedex 15 (FR).

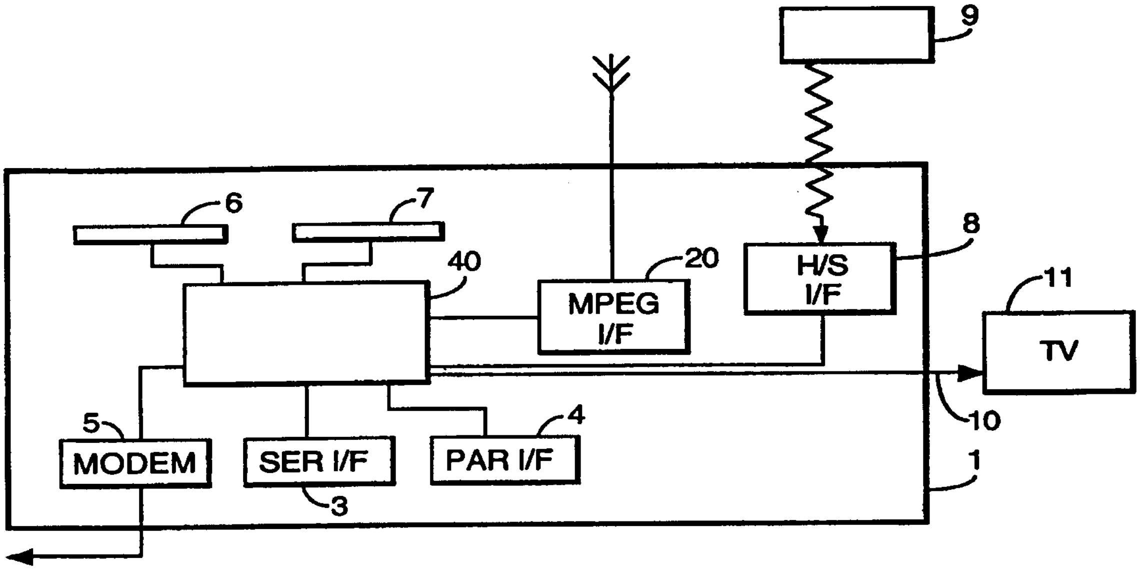
(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: SARFATI, Jean–Claude [FR/FR]; 2–4, place d'Oberursel, F–93800 Epinay sur Seine (FR).

(74) Agents: COZENS, Paul, Dennis et al.; Mathys & Squire, 100 Grays Inn Road, London WC1X 8AL (GB).

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

Published
With international search report.

(54) Title: DOWNLOADING OF APPLICATIONS IN A DIGITAL DECODER



(57) Abstract

A method for downloading an executable application into a decoder (1) for a digital broadcast system, characterised in that the application is stored on a portable memory card introduced into a card reader (6, 7) in the decoder, the decoder reading and downloading the application from the card. Data may preferably be stored in the smartcard in a broadcast data format such as the MPEG format to permit the processing of such data in the same manner as the control unit (40) of the decoder processes data downloading by a broadcast transmission. The invention extends equally to a decoder and a memory card for use in such a 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
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

DOWNLOADING OF APPLICATIONS IN A DIGITAL DECODER

The present application relates to a method and apparatus for downloading executable applications into a decoder used in a digital broadcast system, for example, as used in a digital television system.

Broadcast transmission of digital data is well-known in the field of pay TV systems, where scrambled audiovisual information is sent, usually by a satellite or satellite/cable link, to a number of subscribers, each possessing a decoder or receiver/decoder capable of descrambling the transmitted program for subsequent viewing. Terrestrial digital broadcast systems are also known. Recent systems have also used the broadcast link to transmit other data, in addition to or as well as audiovisual data, such as computer programs or interactive applications to the decoder or a to a connected PC.

The same decoder unit may be supplied by the system designer to a number of different service providers or broadcast companies in a number of different countries. In such circumstances, some degree of testing or customisation of the decoder unit by the service provider will usually be necessary. Typically, a testing application is used to check the correct operation of the hardware elements of the decoder, eg to confirm that the tuner within the decoder operates correctly etc.

This operation will typically be carried out by the service provider or distributor before the decoder is passed to the consumer, for example, using a dedicated PC and a parallel or series link to the decoder. An application supplied by the system designer and running on the PC is used to adjust the operating parameters of the decoder.

Depending the complexity of the operation and the skills of the operator employed to carry out this task the time necessary to test the decoder can be considerable and can increase the real cost of the finished item by a significant amount.

Furthermore, when installed in the field, a user may also wish to introduce at his own convenience a number of applications functioning with the decoder. Again, the user will be faced with the problem of configuring and running the decoder with an application loaded in a PC etc.

It is an object of the present invention to reduce the time and complexity of this type of operation and to provide a simple means for introducing applications in the decoder.

According to the present invention, there is provided a method for downloading an executable application into a decoder, characterised in that the application is stored on a portable memory card introduced into a card reader in the decoder, the decoder reading and downloading the application from the card.

Use of a portable memory card enables a predetermined application to be easily and simply introduced into the decoder without the necessity, for example, to connect the decoder to a PC, load a program into the PC etc. The time necessary to carry out, for example, a testing operation will be greatly reduced since an operator can load the application into the decoder by a simple insertion of the card into the decoder.

Whilst portable memory cards are known in the field of decoder technology, their use to date has either been restricted to the simple transfer of static data, for example, financial data from a credit card inserted in the decoder, or to hold decryption keys associated with broadcast transmissions. Up until now, such cards have not been used to download executable applications. This is in part due to the perceived slowness of the data link associated with the use of a card slot, which has acted to discourage system designers from this solution.

PCT WO93/07715 discloses a system in which static data corresponding to channel frequency information is held in the memory of a smart card, the smart card being inserted in the television to tune the television to the correct channels. A similar system is described in DE 4344317 in which a smart card is inserted in a slot in a television remote control to control the tuner of the television. Neither document discloses the downloading of an executable application into a decoder.

As will be understood, the present invention is not limited to the downloading of a testing type application. The card may equally be used to introduce an application used to initially configure the decoder. Alternative uses are also imaginable, for example, in which cards bearing a promotional application such as a video game or the like, are distributed directly to the end user of the decoder. Increasingly, decoder units are incorporating more and more functionalities associated with general multimedia products and using a portable memory card provides a relatively simple means for a non-technical consumer to introduce executable applications into the decoder.

The term "portable memory card" includes any portable cards that may be inserted within a corresponding card slot in the decoder. The card may include a microprocessor chip in addition to a simple memory element. The card may be powered via a connection to a power source located internally within the reader slot of the decoder or may include a battery power source.

In one embodiment, the card may conform to the standards necessary to permit reading in a PCMCIA reader in the decoder. Preferably, however, the card is adapted to be read in a smart card reader in the decoder. This solution possesses a number of advantages in comparison, for example, with a PCMCIA card, notably due to the simplicity of the contacts formed on the card which reduces the cost of production and the ubiquity of smart card readers in decoder units.

The characteristics of smartcards and smartcard readers are well known and are defined, for example, in the international standards ISO 7816_1 (physical characteristics), ISO 7816_2 (contact dimensions and placement) and ISO 7816_3 (electrical signals and transmission protocols).

Unlike, for example, bank cards, the smartcards associated with decoder units need not be fully inserted into the unit and may protrude some distance from the decoder. Consequently, whilst the card width and thickness for the inserted part of the card must correspond to the normalised values, the card may be longer than a standard credit card. This leads to the possibility to introduce more and larger components onto the card.

Advantageously, the executable application stored within the card and downloaded into the decoder is formatted according to a broadcast data format, such as an MPEG data format. In the case of application type data held in the payload of a transport packet, the MPEG standard describes the organisation of data into a series of tables, each table including a table ID etc.

In one embodiment, the application data may be subdivided into a number of modules in the memory of the card, the modules being assembled by the decoder to form the complete application.

The advantages associated with the use of MPEG format data are considerable, since the decoder can handle and process such applications in the same manner as it handles applications downloaded via the broadcast link. In the case, for example, where the decoder includes a virtual machine to process data, the application may be written in interpretative code, this code being interpreted and processed by the same logical units within the machine as used for broadcast MPEG applications.

As will be understood, where the decoder is adapted to download digital broadcast transmissions according to an alternative data format, the same advantages may be obtained by organising the data in the card in this format.

According to a further preferred embodiment, some or part of the application stored within the memory card is encrypted with one or more encryption keys. In particular, some or part of the data stored in the memory card may be encrypted and/or signed with a private key, the decoder having access to the equivalent public key so as to decrypt and/or authenticate the origin of the application. In the event of non–

authentication of the code, the decoder may refuse to download the code. Other arrangements, using two secret keys of a symmetric algorithm, or a combination hash/encryption technique, for example, are possible in addition to or instead of this signing process.

The advantage of a memory card lies in the simplicity in which an application may be introduced into the decoder. By the same token, the use of a memory card could potentially give rise to a problem of security by permitting the installation of pirate applications into a decoder. The use of signed code ensures the integrity of applications within the decoder and prevents, for example, the introduction of a "trojan horse" program or the like into the system.

Preferably, the decoder is provided with a plurality of smart card readers, to permit the reading of a smartcard carrying the executable application together with another smartcard, for example, a smartcard carrying a decryption key.

As mentioned above, a principal use of smart cards in the context of a decoder relates to the storage of decryption and encryption keys associated with that decoder. In the case where the executable code downloaded from the memory card is partially or wholly encrypted, decryption will most probably be carried out in relation to a public key stored on a subscription type smart card. A multislot decoder permits interaction between the two cards.

Other embodiments for a single-slot decoder are possible, for example, in which the application is downloaded from the first smartcard and stored in a buffer before the first card is removed and the second card inserted to verify the application, or in which an adapter is used to enable both cards to be inserted in parallel etc.

In one embodiment, the method may include the steps of downloading the application into the decoder, setting one or more parameters associated with the application and storing the parameters in the memory card for later use.

For example, in the case where the memory card is used as a vehicle for a testing application developed by the system designer, the application may include certain parameters, such as tuning frequency, which are to be set by the test operator.

The first time that the application is loaded into a decoder, the operator will have the option of selecting these parameters by, for example, using the remote controller of the decoder. Once fixed, the parameters can be stored on the card. Thereafter, testing of subsequent decoders will be carried out automatically in relation to these stored parameters.

For reasons of security, it is preferable that the application remain unchanged and only the newly set parameters reloaded back onto the card. The application may be, for example, stored in an access-restricted FLASH or ROM memory and the parameters loaded into an EEPROM memory unit on the memory card.

Advantageously, the memory card includes a physical switch means for selecting one of a plurality of applications stored on the card that will be downloaded upon insertion of the memory card in the decoder. For example, where the card is used as a vehicle for a number of configuration applications for a number of service providers, the card can include a DIL switch means which can be set by an operator to select the configuration application associated with that service provider.

The present invention extends to a decoder for use in a method as described above, in particular, a decoder adapted to read broadcast (eg MPEG) format data introduced via a card reader in the decoder. The present invention also extends to a memory card for use in such a method, in particular, including an application stored in a broadcast format in the card.

Whilst the description refers to " receiver/decoders " and " decoders " it will be understood that the present invention applies equally to embodiments having a receiver integrated with the decoder as to a decoder unit functioning in combination with a physically separate receiver. Such a decoder may be of the kind used in any satellite, terrestrial, cable etc digital broadcast system and may include other multimedia type capabilities or may be integrated with other devices such as a video recorder or television.

Similarly, the term “ executable application” covers applications written in any form of code (interpretative code, compiled code, native code etc) and capable of being executed by a microprocessor within the decoder.

The term MPEG refers to the data transmission standards developed by the International Standards Organisation working group “Motion Pictures Expert Group” and in particular but not exclusively the MPEG-2 standard developed for digital television applications and set out in the documents ISO 13818-1, ISO 13818-2, ISO 13818-3 and ISO 13818-4. In the context of the present patent application, the term includes all variants, modifications or developments of MPEG formats applicable to the field of digital data transmission.

There will now be described, by way of example only, a preferred embodiment of the present invention, with reference to the attached figures, in which:

Figure 1 shows an overview of the elements of a decoder;

Figure 2 shows a memory card, adapted to be read in a card reader slot in the decoder of Figure 1;

Figure 3 shows a circuit diagram of the components of the card of Figure 2; and

Figure 4 shows the software architecture of the decoder of Figure 1.

Referring to Figure 1, the elements of a receiver/decoder 1 or set-top box for use in a digital broadcast system and adapted to be used in the present invention will now be described. As will be understood, the hardware elements of this decoder are largely conventional and their implementation will be within the capabilities of one skilled in the art.

As shown, the decoder 1 is equipped with several interfaces for receiving and transmitting data, in particular an MPEG tuner and demultiplexer 2 for receiving broadcast MPEG transmissions, a serial interface 3, a parallel interface 4, and a modem back channel 5 for sending and receiving data via the telephone network. In this embodiment, the decoder also includes a first and second smart card reader 6 and 7, the first reader 6 for accepting a subscription smart card containing decryption keys associated with the system and the second reader 7 for accepting bank cards and, in this case, a smartcard containing an application to be downloaded.

The decoder also includes a receiver 8 for receiving infra-red control signals from a handset remote control 9 and a Peritel output 10 for sending audiovisual signals to a television 11 connected to the decoder.

Processing of digital signals received via the interfaces and generation of digital output signals is handled by a central control unit 40. The software architecture of the control unit within the decoder may take many forms. It may be based, for example, on a virtual machine interacting via an interface layer with a lower level operating system implemented in the hardware components of the decoder. In terms of the hardware architecture, the decoder will be equipped with a processor, memory elements such as ROM, RAM, FLASH memory etc. as in known decoders.

A particular implementation of a software architecture will now be described in relation to Figure 4. It will be seen that a layered architecture is used. The first layer 51 represents the operating system of the hardware of the receiver/decoder. This is a real-time operating system chosen by the manufacturer to control the hardware elements of the receiver/decoder. The real-time operating system has a relatively fast response time in order to be able to correctly synchronise hardware operations. A data processing system layer sits on top of the hardware operating system and comprises a middleware layer 52 and an application interface layer 53.

Event messages are passed between the operating system layer 51 and the middleware layer 52 immediately above. The middleware layer is written in a language such as C ANSI and comprises the elements of a virtual machine 54 and a number of interfaces 55 including a graphical interface 56, a FLASH/PROM memory interface 57, a protocol interface 58 and a device interface 59.

The use of a virtual machine 54 enables independence between upper level applications 66 which are usually provided by the system manager or one or more operators, and a lower level operating system 51, usually implemented by the hardware manufacturer of the decoder.

The interfaces 60 provide the link between operations of the virtual machine and the lower level operating system 51 and also include a number of intermediate level application modules more easily executed at this level.

The application interface (API) layer 53 comprises a number of high level packages 60–65, written in an object–oriented interpretative language, such as Java. These packages provide an interface between the high level applications generally created by the service provider (interactive program guide, teleshopping, internet browser etc) and the virtual machine of the system.

The lower level OS is normally embedded in the hardware components of the decoder, although in some realisations, the lower level OS can be downloaded. The middleware and application interface layer packages can be downloaded into the RAM or FLASH memory of the decoder from a broadcast transmission. Alternatively, some or all of the middleware or application interface layer elements can be stored in the ROM or (if present) FLASH memory of the decoder. As will be understood, the physical organisation of the memory elements of the decoder is quite distinct from the logical organisation of the memory.

Turning in detail to each layer, the interface layer 55 above the virtual machine 54 will now be described. As shown it comprises four modules, a graphics module

56, a memory file management module 57, a protocol module 58 and a device manager 59. Whilst the modules at this level are described as interface modules their function is to provide a "glue" layer for the implementation of the application interface packages and for the operation of the virtual machine generally.

The graphics module 56 provides the creation and management of graphical objects. It asks the low level OS to display basic graphic shapes such as single pixels, lines, rectangles etc. In a similar manner, the memory file management module 57 includes low level read/write file commands associated with the memory components of the system. The protocol management module 58 defines a library of communication protocols that may be called upon in communications via, for example, the TCP/IP layer of the decoder.

The device manager 59 is slightly different from the other modules in this layer in that it provides the link or interface between the hardware operating system and the layers above, including the other modules in the interface layer and the virtual machine. Commands or event messages that are received/sent to the hardware OS from the virtual machine, for example, are necessarily passed by the device manager for conversion according to the interface specifications between the two levels.

Referring now to the application interface layer 53, the packages in this layer are written in an object oriented language such as Java. Each package defines a set of class libraries called on during operation of the system. Their class behaviour will depend on the language chosen, a single inheritance class structure being adhered to in the case of Java. In the present system the following packages are installed.

Lang/Util Package 60. These packages define the classes necessary for the manipulation of objects by the virtual machine. These class libraries normally form part of a standard library associated with the object oriented language chosen.

MHEG-5 Package 61. This package defines the classes associated with the manipulation of graphical objects on the television display. Such objects are distinct from audio-visual data and can make up, for example, channel identifiers or text laid over displayed images. The definition of classes within this package should respect the MHEG-5 norms defined by the standards ETS 300777-3 and ISO/ISE 13522-5 (and the standard ISO/ISE 13522-6 in the case of a Java implemented system).

Toolbox Package 62. This package contains the classes used for downloading and decompression of information as well as the classes associated with the management of the file system and memory within the receiver/decoder and the classes associated with the connection to the internet etc.

Device Package 63. This package defines the classes necessary for management of peripherals attached to the receiver/decoder, as discussed above and including the modem, the smart card readers, the MPEG flow tuner etc

Service Package 64. This package defines the classes necessary for the implementation of developing higher level interactive applications, such as management of credit card data etc.

DSMCC-UU Package 65. This package implements the protocols necessary for communication between a client and a server for data file search and reading. Implementation of this package should respect the norm ISO/IEC 13818-6 and directives defined in DAVIC part 9.

Finally, a number of high level applications 66 sit on top of and communicate with lower levels in the system via the application interface layer 53. In the present embodiment, the use of a virtual machine type architecture means that applications will be written in an interpretative language, such as Java. Other software systems for handling executable applications written in alternative types of code are of course possible. As will be described below, applications may originate from a variety of sources and/or operators. In particular, in the present embodiment of the invention, executable applications are installed via a smart card interface.

An application introduced into the decoder corresponds to a section of code introduced into the machine that permits the control, for example, of higher level functions of the machine. These may include the generation of a graphic sequence on the screen of the television display in response to a command from the remote control, or the emission of a message via the modem 5 to the server associated with the digital broadcast system. The execution and maintenance of applications may be handled by an application manager 67, itself installed at the application layer.

Applications may be resident applications stored in the ROM or FLASH of the decoder or applications broadcast and downloaded via the MPEG interface 2 of the decoder. Applications can include program guide applications, games, interactive services, teleshopping applications, as well as initiating applications to enable the decoder to be immediately operational upon start–up and applications for configuring and testing the decoder. Applications are stored in memory locations in the decoder and represented as resource files comprising graphic object description files, unit files, variables block files, instruction sequence files, application files, data files etc.

In the case of a broadcast transmission, a number of types of data stream may be present, for example, a video data stream, an audio data stream, a text data stream etc. In accordance with MPEG standards each transport packet is preceded by a Packet Identifier (PID) of 13 bits, one PID for every packet transported in the MPEG stream. A programme map table (PMT) contains a list of the different streams for a particular service or "channel" and defines the content of each stream according to the respective PID. A PID may alert the device to the presence of applications in the data stream, the PID being identified by the PMT table.

Within an MPEG transport stream containing an application there may be three or more levels of packet structure. A first layer corresponds to the basic transport layer comprising a series of fixed size transport packets.

Furthermore, applications downloaded into the decoder via the broadcast link are divided into modules, each module corresponding to one or more MPEG tables encapsulated within the above mentioned transport packets. Each MPEG table may be divided into a number of sections. For data transfer via the serial and parallel ports, modules are also split into tables and sections, the size of the section depending on the channel used. A similar sectioning is applied to MPEG tables downloaded using the smartcard of the present embodiment.

Finally, this sectioning of an application into MPEG tables is independent of any structuring of the application data itself. For example, an application may be organised into a number of files arranged within a data carousel as per the DSM-CC protocol, for example.

Referring to Figures 2 and 3, the structure of a smartcard 12 adapted to charge an executable application in the decoder will now be described. Figure 2 shows a plan view of the smartcard, comprising an area of contacts 13, a FLASH ROM memory 14, an EEPROM memory 15, a microprocessor 16, a DIL switch unit 17 and a number of other discrete components Unlike standard smart cards, the presence of additional memory elements 14, 15 enables an executable application of a significant size to be stored on the smart card.

The memory card 2 possesses the width and thickness of a standard normalised smart card so as to enable its insertion in a smart card slot of the decoder. However, as will be seen from Figure 2, the card is longer than a standard card to enable the incorporation of all the components described on its surface. In the context of its use in the initial configuration of the decoder this increase in size may not be significant. In alternative situations, for example, where the card is intended to be supplied to the eventual user of the decoder, some components such as the DIL switch unit 17 and EEPROM 15 may be omitted. The remaining components may be miniaturised and the whole card designed to conform with smart card norms.

Referring now to Figure 3, the contacts 13 engaged in the smart card reader in the decoder may be divided by function into a power supply line 18 which supplies the card voltage Vcc, a reset line 19 connected to the corresponding reset terminal 20 of the microprocessor, a clock line 21 connected to clock terminal 22 of the microprocessor, and an I/O line 23 connected to corresponding input and output terminals 24, 25 of the microprocessor. As shown, connections are made via a series of op-amps 26. The power supply is regulated by means of a capacitor C4.

The EEPROM memory unit 15 is connected via lines 27, 28 to the microprocessor 16, these lines being biased by the power supply Vcc connected via the resistances R1 and R2. The function of the EEPROM memory will be discussed in more detail below in relation to the configuration application. The microprocessor 16 is connected by a series of lines 29 to corresponding terminals of the FLASH memory 14. The state of three of these lines 30, 31, 32 is determined by the switch unit 17 connected via a series of diodes D1, D2, D3 and biased by the power supply Vcc connected by resistances R3, R4, R5. By switching each of the switches ON or OFF, a binary control word 000, 001, 010, 011 etc can be defined. As will be discussed, this binary word is used to determine the first block in the FLASH memory that will be accessed upon insertion of the card and, hence, the application that will be charged into the decoder.

The card 12 is designed to engage in the credit card reader 7 of the decoder 1, the reader 6 being reserved for the subscription card associated with the broadcast system which contains the keys necessary for, inter alia, decoding scrambled transmissions and verifying downloaded code. Upon insertion, the reader checks the type of card inserted, by means of a simple handshake signal to the card. In the event that the reader identifies the card as being a card of the type containing application code for loading into the machine, the decoder will access the first block of code in the FLASH memory 15 at the hexadecimal address corresponding to the binary message indicated by the switch unit 17.

In the case, for example, where the card is intended to be used in the testing of decoders for a number of service providers, a different application may be loaded corresponding to the service provider in question or corresponding to the functions that need to be tested. In addition or alternatively, a first setting of the switches may be used to download the application supplied with the card and a second to download a different application and/or associated parameters set by the service provider (see below).

The application code is downloaded from the from the card in a series of modules, the modules then being assembled to form a series of MPEG-2 (short form) tables, as described above in connection with broadcast data. The advantage of formatting data according to the MPEG format is that the virtual machine within the central control unit of the decoder can directly process applications received in this format, in the same manner as it processes applications received via the broadcast link. As will be appreciated, this leads to considerable reductions in the time needed to process the application etc.

The format of the MPEG private sections in this case is as follows:

table_id	8 bits
section_syntax_indicator (=0)	1 bit
private_indicator (=1)	1 bit
reserved	2 bits
private_section_length	12 bits
table_id_extension	16 bits
reserved	2 bits
version_number	5 bits
current_next_indicator	1 bit
section_number	8 bits
last_section_number	8 bits
private_data_byte	undetermined

An application will be accessed by the decoder using the table_id and table_id_extension values.

Prior to storage in the card, the application code contained within the MPEG tables is encrypted to provide a digital signature. This signature is generated by the supplier of the card using a private key of a public/private key algorithm, such as RSA, and known only to himself. The decoder has access to a series of public keys on a subscription card inserted in the other card reader.

In the event that the decoder confirms that the code has originated from a known source, by verifying the digital signature, the application will be installed in the machine. Unverified code will be rejected by the decoder. In addition to verifying the code, the decoder may also use the public key to decrypt the code prior to operation.

Furthermore, encryption by a private/public algorithm may also be combined with a one–way hash type function, such as MD5. For example, a section of code may be processed to provide a hash value, this hash value then being encrypted by the private key to provide the digital signature.

Other encryption techniques used in broadcast digital systems may also be applied, for example, to encrypt the code according to one or more private keys known to the supplier of the application card to prevent a third party from decrypting and using the application stored on the card. The decoder possesses the key or keys necessary to decrypt the code as stored on a subscription card. This encryption can be carried out in addition to and after the signature of the code. This encryption/decryption may be carried out, for example, using a symmetric algorithm.

The use of a subscription card to hold the necessary decryption keys generally requires that the decoder is also provided with a second smartcard reader, since both cards will be addressed by the decoder during the downloading and verification steps. Alternative embodiments are conceivable, for example, in which data is first downloaded from the application card into a buffer, the application card removed and the card containing the decryption keys inserted etc. However, as will be appreciated, these are less convenient than the use of a decoder equipped with two or more smartcard readers, particularly since one or the other of the cards may need to be re-addressed at any moment.

The installation of a test application within the decoder will now be described. Typically, such a test application is used by a service provider to test the correct operation of the hardware layer. For example, the test application may control the tuner of the decoder to test that the decoder can correctly receive data transmitted on a given channel frequency.

The loaded application may be interactive so as to permit the operator to enter specific parameters into the decoder by means of, for example, the remote control handset. In the case of the tuning frequency the operator may manually adjust the set frequency until the clearest reception is obtained. Once these parameters are known for one decoder, they will be the same for the rest of the series. It is therefore desirable that this and other parameter values can be memorised in order to avoid repeating the operation for each decoder.

Accordingly, once defined by the operator in relation to a first decoder, these parameters are downloaded into the EEPROM memory 15 of the card. Upon removal of the card, the operator changes the setting of the switches in the switch unit 17 such that an application at a different address within the FLASH memory will be accessed upon its next insertion in a decoder. When the card is then reinserted in the next in the series of decoders, this new application will be loaded into the decoder. Upon execution, the application will signal the presence of pre-determined parameter values stored in the EEPROM and these values will be automatically loaded into and set in the decoder. In the case of the tuner, for example, the application will automatically set the tuner to the frequency selected by the operator for the first decoder and the operator can then immediately determine whether the tuner operates correctly or not.

In view of the relative difficulty in writing data to a FLASH unit (as compared to an EEPROM) it is preferable, though not essential, that the FLASH memory be used for applications that will not be modified in use and the EEPROM memory be reserved for data downloaded into the card.

Furthermore, in order to increase the security of the system, the FLASH memory may be locked into a read-only configuration by the microprocessor upon initial connection of the card, and/or upon receipt of an unknown instruction. Other memory combinations and configurations are of course possible, using ROM devices etc.

Whilst the above embodiment has been discussed in relation to a smartcard realisation, other portable memory cards, such as PCMCIA cards, may be used if the decoder is capable of reading such cards.

CLAIMS

1. A method for downloading an executable application into a decoder, characterised in that the application is stored on a portable memory card introduced into a card reader in the decoder, the decoder reading and downloading the application from the card.

2. A method as claimed in claim 1 in which the card is adapted to be read in a smart card reader in the decoder.

3. A method as claimed in claim 1 or 2 in which the executable application stored within the card and downloaded into the decoder is formatted according to a broadcast data format.

4. A method as claimed in claim 3 in which the executable application stored within the card and downloaded into the decoder is formatted according to an MPEG data format.

5. A method as claimed in claim 4, the application being subdivided into a plurality of modules in the memory of the card, the modules being downloaded and assembled by the decoder to form the complete application.

6. A method as claimed in any preceding claim, in which the application is written in interpretative code.

7. A method as claimed in any preceding claim, in which some or part of the application stored within the memory card is encrypted with one or more encryption keys.

8. A method as claimed in any preceding claim in which some or part of the data stored in the memory card has been encrypted and/or signed with a private key, the decoder having access to the equivalent public key so as to decrypt and/or authenticate the origin of the application.

9. A method as claimed in any preceding claim in which the decoder is provided with a plurality of smart card readers, to permit reading of a smartcard carrying the executable application and another smartcard.

10. A method as claimed in any preceding claim including the steps of downloading the application into the decoder, setting one or more parameters associated with the application and storing the parameters in the memory card for later use.

11. A method as claimed in any preceding claim in which the card includes a physical switch means for selecting one of a plurality of applications stored on the card that will be downloaded upon insertion of the memory card in the decoder.

12. A decoder for use in a method as claimed in any preceding claim.

13. A decoder as claimed in claim 12 adapted to read broadcast format data introduced via a card reader in the decoder.

14. A memory card for use in a method as claimed in any of claims 1 to 11.

15. A memory card as claimed in claim 14 including an application stored in a broadcast data format in the card.

16. A method for downloading an executable application into a decoder substantially as herein described.

17. A decoder for use in a method as claimed in any of claims 1 to 11 and substantially as herein described.

18. A memory card for use in a method as claimed in any of claims 1 to 11 and substantially as herein described.

Fig.1.

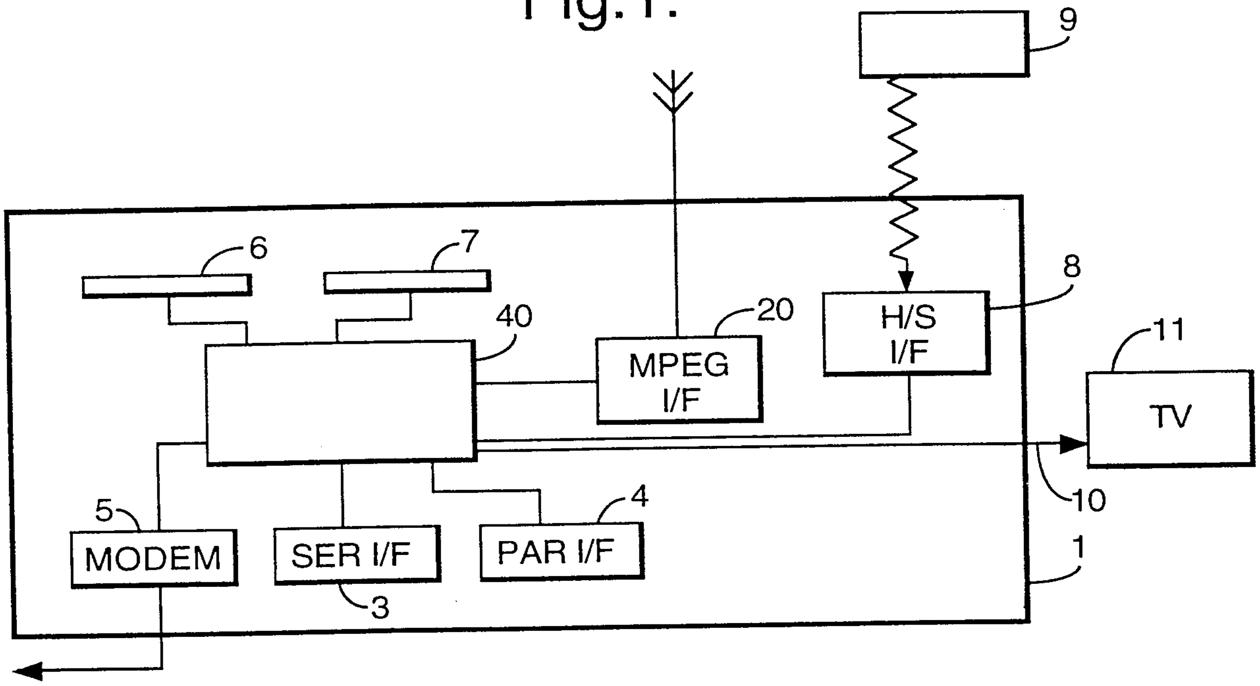

9
6
7
40
20
MPEG I/F
8
H/S I/F
11
TV
10
5
MODEM
SER I/F
3
4
PAR I/F
1

Fig.2.

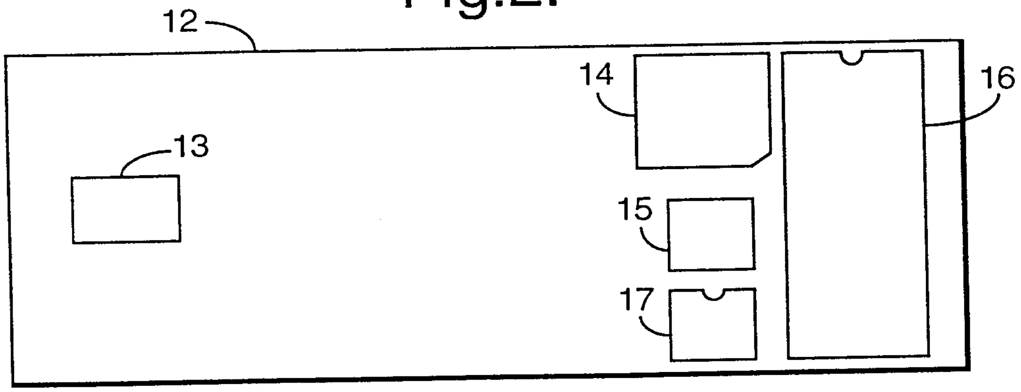

12
14
16
13
15
17

Fig.3.

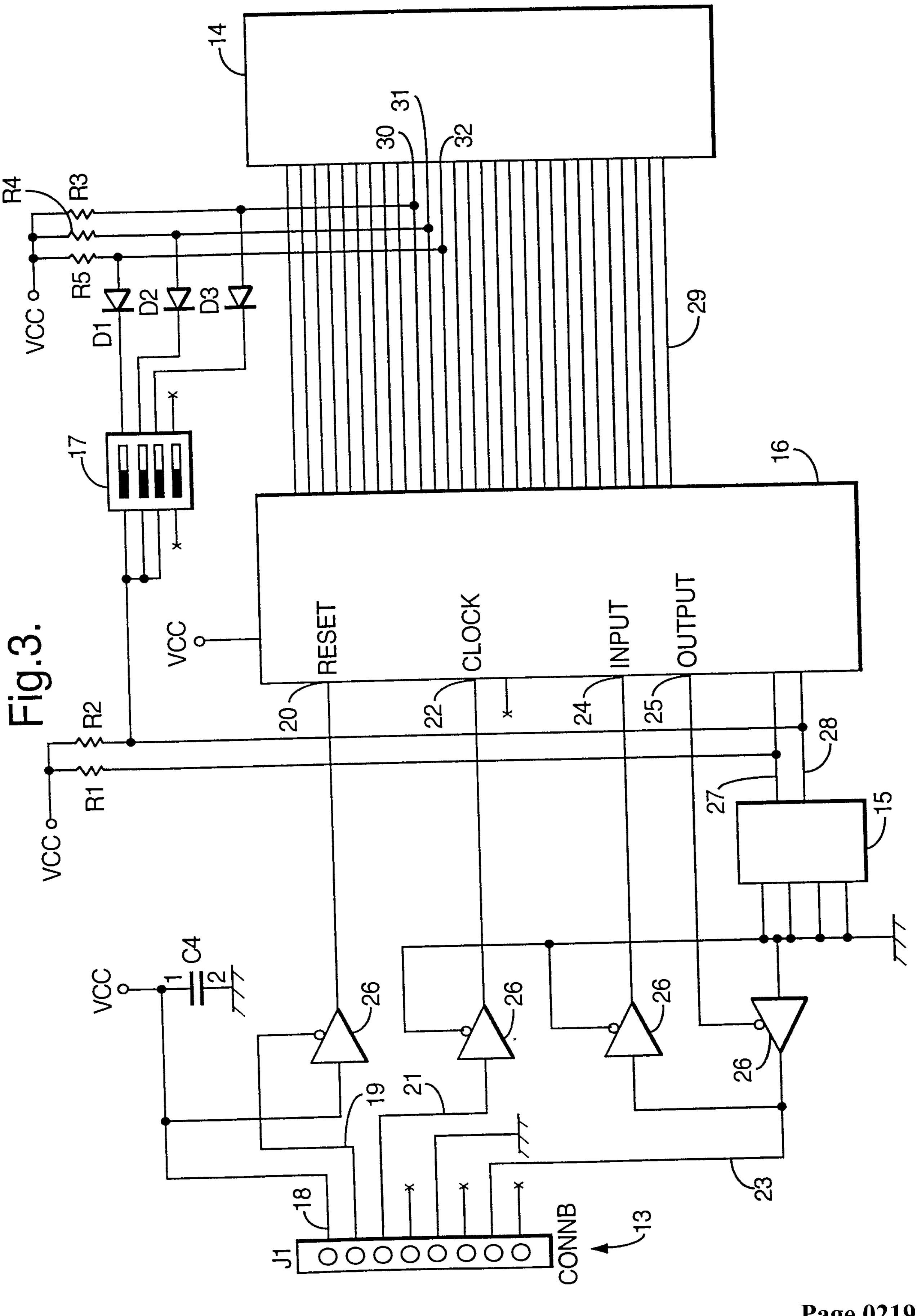
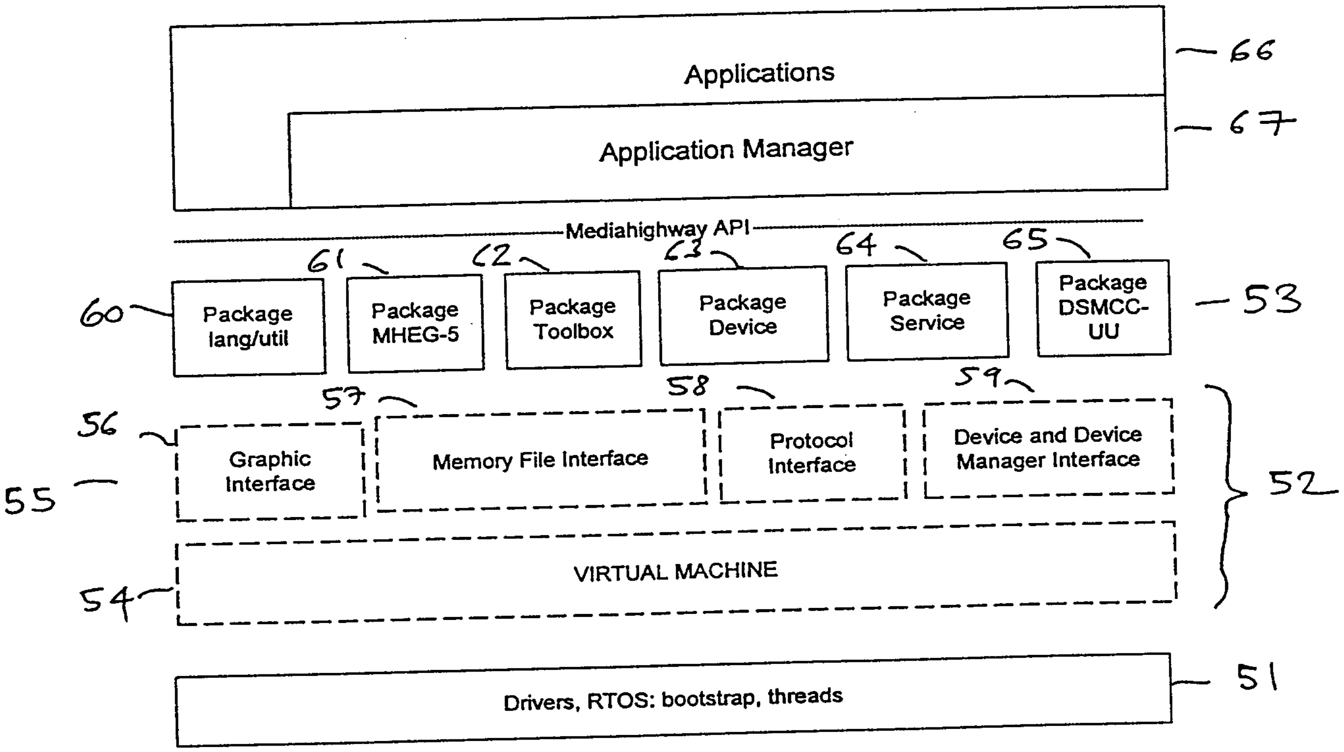

14
30
31
32
R4
R3
R5
VCC
D1
D2
D3
29
17
16
VCC
RESET
CLOCK
INPUT
OUTPUT
20
22
24
25
28
27
15
R2
R1
VCC
C4
1
2
VCC
26
26
26
26
19
21
23
18
J1
CONNB
13

Figure 4


Applications
66
Application Manager
67
Mediahighway API
60
Package lang/util
61
Package MHEG-5
62
Package Toolbox
63
Package Device
64
Package Service
65
Package DSMCC-UU
53
56
Graphic Interface
57
Memory File Interface
58
Protocol Interface
59
Device and Device Manager Interface
55
52
VIRTUAL MACHINE
54
Drivers, RTOS: bootstrap, threads
51

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ^{6} : **G06F 12/14, G06K 19/07, 19/073**	**A1**	**(11) International Publication Number:** **WO 99/46682** **(43) International Publication Date:** 16 September 1999 (16.09.99)

(21) International Application Number: PCT/AU99/00142

(22) International Filing Date: 10 March 1999 (10.03.99)

(30) Priority Data:
PP 2239 10 March 1998 (10.03.98) AU

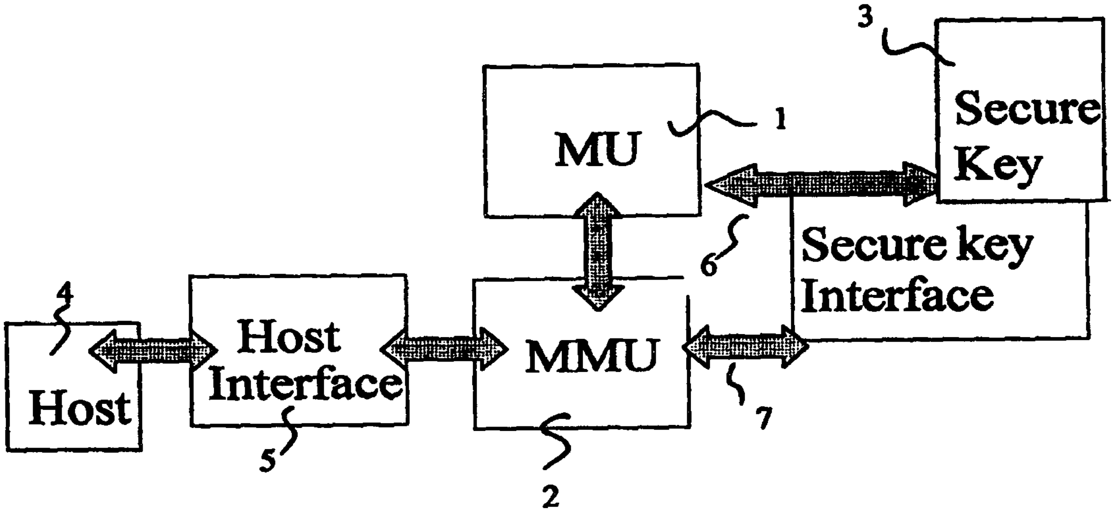
(71)(72) Applicant and Inventor: LINDLEY, Robyn, Alice [AU/AU]; 164–168 Derribong Drive, Cordeaux Heights, NSW 2526 (AU).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: CORDONNIER, Vincent, Maxine [FR/FR]; Université des Sciences et Technologies de Lille, Laboratoire d'Informatique Fondamentale de Lille, Lille (FR).

(81) Designated States: AU, CA, CN, IL, JP, US, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: MOBILE INTELLIGENT MEMORY (MIM) UNIT WITH REMOVABLE SECURITY KEY



(57) Abstract

A mobile intelligent memory (MIM) unit is a new small form factor device offering large capacity, portable, data storage, transmission, receipt and security management. The essential components of the device include at least one memory unit (MU) for data storage, at least one memory management unit (MMU), and at least one removable secure electronic key, and interfaces connecting the components to each other, and connecting the MMU(s) to an external device(s) responsible for initiating inquiries. The MMU(s), MU(s) and secure key(s) work together for security functions and access management of sensitive data. The device may be interfaced with a range of peripheral devices (e.g. a PC or GSM phone via a PCMCIA interface). The security functions can be performed by a range of small IC–security devices – such as an ISO smart card. The device is useful to securely receive, transmit, manage, store and archive data files.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

DESCRIPTION

TITLE

Mobile Intelligent Memory (MIM) unit with removable security key

TECHNICAL FIELD

The present invention relates generally to a device for the management and security of large data files in a hand held unit with a removable security key for additional physical security.

BACKGROUND ART

For many chip card applications, smart cards are the technology of choice for securing sensitive data and performing security functions such as validation, authentication, and non-repudiation. For many applications they are considered the most convenient technology to provide secure access to a range of service applications. They are also physically robust, relatively tamperproof, inexpensive, very secure and socially accepted. However, there are many limitations: First, the data storage capacity is, and will remain limited. Second, their data processing speed and the suitability for many multiapplications is limited. Third, there are limited opportunities for card holders to access a smart card reader.

Smart card technology is also limited because there are limitations on the operating systems and applications. Smart card functionality is not flexible: Smart cards can perform only operating system functions stored in the Read Only Memory (ROM) during manufacturing - and possibly some additional functions which are stored in the Non-Volatile Memory (NVM) during the initialisation stage. Smart cards are not able to accommodate executable codes which may be loaded by the different applications.

Smart card configuration is also inflexible: At present, multiapplication smart cards are managed and issued by a single organisation (*issuer*). An issuer generally creates access privilege control, divides NVM among the different uses, and loads data specific to the user. These functions are completed during initialisation that is required before the card can be used. Thus, the configuration and data storage functions cannot be altered during the life of the card. These are constrained by the small NVM, and the lack of memory management facilities for inter-application security.

On the other hand, there are a wide range of data storage devices that offer large memory capacity - but with little or no security. At present the portable data storage market is undergoing rapid expansion with a range of new innovative technologies capable of retaining data - and with some security. For example, CD ROMs, diskettes, PCMCIA memory cards, Zip and tape drives, optical disc technologies. These devices contain only passive memory with little or no internal memory management functions. Memory management is also inflexible. However, no one has conceived of, or manufactured a device which achieves the full set of operational objectives which can be met according to the set of claims of this invention. The said MIM device offers enhanced capacity, superior security, interoperability and management flexibility all within the confines of a small hand-held and physically robust unit.

Currently there are many possible applications that require more memory capacity and flexibility than is available in a smart card. In comparison to smart cards, the PC card (as defined by the Personal Computer Memory Card Industry Association - PCMCIA) has been developed to provide high memory storage capacity - but with less security. Other one card systems have also suffered from limitations that have continued to inhibit their ability to meet new demands. However, recent developments on a number of fronts mean that a new generation of more flexible, secure chip card technologies, such as the said MIM device, can now be conceived and developed.

DISCLOSURE OF INVENTION

For the purposes of understanding the specific claims embodying this invention, and operational requirements, the following definition has been adopted:

> *"A Mobile Intelligent Memory (MIM) device is a palm-size high security, large capacity data storage and management unit with access secured by a physically removable electronic key. A MIM card can be used to provide a flexible and small form factor unit used for the secure transmission, receipt, storage, management and mobility of large (or small) data files."*

The owner of a MIM device can store information on a single small palm-sized unit that is 'unlocked' with a physically separate electronic 'secure key'. A MIM device can also be interfaced with a range of peripheral devices such as a PC via a PCMCIA card interface, a floppy disc drive, a GSM handset via a SIM card interface, or by remote EM communications links. This means, that a MIM device can be manufactured in several forms. Additional hardware and software features might also be incorporated to improve utility for some applications.

The owner of a MIM device can also have the option to store unsecured data in the MIM memory with the use of an electronic secure key remaining optional. However for the protection of more sensitive files, the owner may select and configure the MIM device access rights so that the files can only be accessed with one or more nominated smart cards. Additional security for validation and authentication might also be added (eg. finger print, or PIN use). The owner is therefore able to have more flexible and personal control over the information storage and management. One MIM device may be configured to be accessed by one or more secure keys; and conversely, one secure key may be configured to have access and security management privileges for one or more MIM devices.

The said MIM device represents a significant new personal chip technology with the following *set* of operational advantages: portability (pocket´or palm size); physically robust; flexible and high level of security and tamperproof; large data storage capacity; compatibility with existing and emerging technologies; new and improved method of offline archiving of data; and offering an alternative way for individuals to manage, secure, store or transmit sensitive files and communications with one or more other compatible MIM device owners. The role of the MIM device is determined by the conditions of use and privileges afforded by the associated secure key. To now, no device has been able to combine all of these advantages in a small form factor device using known or emerging technologies.

BRIEF DESCRIPTION OF DRAWINGS

These and other objects, features and advantages will be understood from the following brief description of an embodiment given solely by way of example, illustrated by the accompanying drawings wherein:

Figure 1 is a block diagram of the unit structure according to the present invention;

Figure 2 shows an example of a block diagram of the architecture according to the invention;

Figure 3 illustrates one possible form of a MIM device and its operational use according to the present invention.

PREFERRED EMBODIMENTS

It will be understood that the specifications and examples used are illustrative but not limitative to the present invention and that other embodiments within the spirit and scope of the invention will suggest themselves to those skilled in the art.

A block diagram of the system according to the invention is shown in Figure 1.

Referring to Figure 1 it can be seen the architectural components of one form of the said device include: a CompactFlash (TM) memory storage unit (MU) (1), a Memory Management Unit (MMU) unit (2), smart card secure key (3), a PC host (4), a PCMCIA interface (5), and smart card interfaces (6,7). At least one authorised smart card (3) is needed to access the MU (1), and the MMU (2) is required to manage a unique directory to be shared only by the authorised cards (3). In one form of the invention a PCMCIA bus connection (5) links the MIM to the host PC (4). This interface (5) has also been designed and standardised for a wide range of peripheral devices which include most of the anticipated characteristics of the MIM: reduced size; fast data transfer rates; universal use; and low power consumption.

In one form, the said device is composed of three main operational layers: Memory; Logical Memory Manager; and the Supervisor. The memory can be flash memory which needs a few special features. The physical memory manager will take into account timing and format problems that will characterise the chosen technology. The logical memory manager will be responsible for controlling the relationship between the physical memory and the file memory unit. The MMU will contain a garbage collector and a directory which will describe each file according to its location, size and common attributes. A supervisor which will be responsible for the overall activity of the MMU and communications with the smart card (3) and the PC (4). In particular, the supervisor will be responsible for managing the security of the memory.

On Figure 2 can be seen one example of the unit architecture according to the invention. This architecture is given by way of example for the sole purpose of showing the flexibility of a system for implementation. In this form, the link between the PC (8) and the MMU can be a parallel 32 bit connection (9) and as fast as the memory can accept. The link between the MMU and the memory will depend upon the architecture and type of memory (10) used (eg. CompactFlash). Although it is necessary for the smart card (11) and the PC (8) to exchange commands, there will be no direct link for this architecture in this form of the invention. To simplify the design requirements, a supervisor will manage incoming messages according to the protocol suite and the attributes of the messages themselves. Thus, much of the security and the flexibility of the MIM will rely upon the protocols between the individual units.

The PC (8) will be used for the user to initiate card commands. The smart card (11) will verify the security conditions and then send a command to the supervisor for execution of access control. Access will only be granted by the supervisor if it recognises the presence of an authorised smart card (11).

To maintain and enhance the high level of security that can be offered by a smart card, the MMU will be required to manage a unique directory to be shared only with the use of an authorised smart card(s). A PCMCIA bus connection will form the only physical link between the host PC from which inquiries are initiated, and the MIM unit (12).

The MMU can be designed to accommodate many of the anticipated characteristics of the MIM unit including: large memory storage space, fast data transfer rates, ineroperability and low power consumption. This also means for example, that the MMU could be standardised to be interfaced with GSM handsets, as well as a range of field data collection or medical instruments. Other potential MIM hosts, such as digital TV reception sets and public information booths might also be considered at a future time.

The MMU therefore has three complimentary roles within the MIM unit: (1) to serve as an intermediary between the PC which provides commands to the MIM, and the MU which is able to serve these commands; (2) to manage and control the sequence of exchanges occurring between the MMU, the PC and the SC; and (3) to share the security of the MIM unit with the SC.

The memory unit (MU) will be responsible for storing data and will consist of two main parts: a physical manager; and the memory storage area. The memory unit (MU) will require a *physical manager* for the smart card and MMU interfaces, as well as the memory storage area.

The memory storage area in this example could consist of two main volumes: One area classified as *Public* that can be readily accessed and used without the need for a smart card. This unprotected area can be used for backup, storage and management of less sensitive information. The other is to be classified as *Private*. and secured using a smart card. The file configuration and access privileges can be selected, configured and dynamically managed during the lifetime of the MIM card and according to the needs of the MIM card owner.

The link between the smart card and the MU, and between the smart card and the MMU, will be a 2-way single channel interaction using the smart card's bidirectional serial I/O port. Use of a second smart card I/O port might be considered at a future stage. In this form of the said invention, the smart card will not be required to cipher data. However, the option to cipher data for transmission could be an added option to secure personal communications between partner MIM units. The main role of the smart card is to manage access to the MU private volume in co-operation with the MMU. In one form of the said device, this can be done by making available to the MU, the specific set of secure keys (*interfaces*) required to find and retrieve the data contents of a file. The specific set of operations required is known only to the smart card.

Figure 3 illustrates one example of the physical form and use of the said MIM device by an individual.

The owner(s) of the information stored in the MIM housing (13) is/are responsible for the portability, physical storage and access to the MIM containing the data, as well as the specific 'electronic key' (14) required to access the information. The owner(s) of the information is/are therefore able to have more personal control over the creation of higher levels of physical and personal trust in the security of the system. This invention also makes it possible for one 'electronic key' (14) to be configured so that access to information on one or many MIM units can be enabled. Conversely, it is also possible for one or many 'electronic keys' be enabled to provide access to information stored in a single MIM unit. A bio-identifier such as a thumb/finger imprint may also be added to the MIM housing (13), to increase the level of security by ensuring that the user of the unit is authorised (authentication). For very high security levels, the protocol may require a thumb/finger imprint after the secure key is inserted into the MIM unit to validate and authenticate the user and secure key prior to initiating requests from the PC (15). If the user and the secure key are accepted, then the user can initiate inquiries, read and write files to the MIM device in a secure environment. After use, the user may remove the MIM device (13) from the PC (15), then remove the secure key (14) from the MIM housing (13) for separate safe physical storage of both component parts.

This configuration could be used in an X-ray clinic where X-ray results can be moved, stored and securely accessed and protected with the device shown. This avoids the need to use expensive and environmentally harmful chemicals for X-ray image production. The X-ray file is secure from unauthorised access, is tamperproof - as well as reproducible. The storage media are also reusable and provide an alternative way of storing and retrieving files and X-rays which do not require a lot of physical space. Other patient data might also be added as patients are moved within the health system to seek other opinions or related services.

The MIM interface shown by way of example in Figure 3 meets PCMCIA/JEIDA standards and is designed to interface with an MS-DOS file structure within a Windows (TM) environment. The secure key is a smart card (SC). That is, this form of a MIM device will be of the same physical dimensions as a PC card (Type II or III) and designed to interface with a PC card reader housed in a personal computer (PC) - or other host. An International Standards Organisation (ISO) smart card will be able to be inserted/removed from the MIM card housing. CompactFlash (TM) could be used for storage in the MU. Time stamping modules, biometric identifier information and audit management functions might also be added to enhance the security of the MIM card. The MIM unit will also require the smart card to co-operate with the MMU to protect the MIM against anticipated hardware attacks.

Although it will be necessary for the smart card and the PC to exchange commands, there will be no direct logical or hardware link for this architecture. To meet the design requirements, the supervisor of the MMU will manage incoming messages according to the protocol suite and chosen attributes of the messages. Thus, much of the MIM unit's security and flexibility will rely upon the exchange protocols occurring between the individual units.

The need for complete flexibility between the organisation of the files and the security model leads to consideration of a "flat" distribution of files; all of them occupying the same level. They can be considered as a sequence or a list. Any subset of that sequence that has to be located under the same security reference will just be a subset of the main list. More precisely, the structure of the file management will be issued by the security model which will create the appropriate groups according to the security requirements of the application designer - rather than by an alternative model, such as a tree which does not necessarily match with the real needs of the application. The flat model is flexible and can accept almost any application requirement. However, other file models might also be used according to the invention.

The commands used are also an important consideration with several options which might be considered. In one example, any profile of operational privileges can be produced by a five bit word.

READ-FILE (RF) : bit 0
READ-ONCE (RO) : bit 1
UPDATE (UP): bit 2
WRITE-FILE (WF) : bit 3
DELETE (DEL): bit 4

We propose to add three bits for the domain manager: One for the permission to CREATE a file, the second for the permission to DISCARD a file, and the third one to act on domains.

CREATE-FILE (CF) : bit 5
DISCARD-FILE (DF): bit 6
CREATE-DOMAIN (CDD) and DISCARD-DOMAIN (CDD) : bit 7

All the possible codes of the corresponding eight bits words will not be used, especially for a given application but that choice does not restrict the design possibilities. Then the key is a byte. However, even with the same format, we shall probably observe that the designer works with two families of keys:
The managing keys: CDD, CF, DF, 0, 0, 0, 0, 0
The operational keys: 0, 0, 0, D, WF, UP, RO, RF

Some other commands can be useful but they are not necessarily compatible with the existing ones. We consider that the application manager may benefit from a combination of some of them in different ways rather than use an enlarged vocabulary of commands.

It will also be necessary to incorporate a set of security management commands. For example, the following set might be used:
G-GRANT Grant security access rights.
R-REVOKE Revoke security access rights.
H-HIDE To hide reference to a file in the MIM directory displayed on the PC's monitor; and
RH-REVEAL HIDDEN Reveal a previously hidden MIM file so that it will be seen in the MIM directory displayed on the PC's monitor.
CS-CHANGE SECURITY Change security access requirements. For example, the user can choose to add, remove or alter the access conditions for a certain file (or group of files). To alter the level of security the following might be possible security management options for a particular smart card: PIN, bio-identifier, electronic signature or a password.
A-AUDIT Manage and access audit functions.

The said device architecture makes it possible to bypass the computational bottleneck that occurs if confidential data and computations corresponding to data needs are all to be supported on a secure key card such as smart cards. As mobility and security becomes one of the major issues confronted by designers of new information and data communications devices, the tools which are being made available by industry do not readily meet the users requirements and do not suit all existing or future application and operational requirements - either for capacity or security reasons. This device seeks to overcome these problems by using a unique architecture and logical use of the component parts to combine the security features and functionality of technologies like smart card, with the large external memory capacity of technologies like CompactFlash memory. To create a MIM device, two different approaches are required: The application design and the technical arrangement; which converge to provide a basis for the design and manufacture of a new palm-size information system with advantages over existing portable data storage units.

To achieve the above stated physical and functional advantages over previously known portable data storage units, one significant architectural feature is the extension of 'electronic key' security to all components of the MIM unit. What is also different is the logical use of the component parts and protocols enabling the unit goals to be achieved. The proposed architecture is also flexible - thus making it possible for more than one model design capable of meeting the project requirements. The internal architecture of the chosen 'electronic key(s)' to secure the architecture, will also play an essential role in the MIM.

In this form of the invention it is also important to consider the application software: Considering the recent rapid development of object-oriented concepts and tools available for the construction of larger scale and more integrated card systems' for personal use and portability, object-oriented technologies will be useful to implement new generic distributed object-oriented information systems using the MIM architecture. Adopting an object-oriented approach to the said device also brings two added security features: The first is *modularity.* Since integrated card services designed as objects can be viewed as small and independently functioning modules with clearly defined inputs and outputs, the functions can meet testability and audibility design requirements. This can be done for each module - without effecting other modules. The second is *encapsulation* (or information hiding). Encapsulation and the independence of object interactions means that each object provides a way to isolate (or hide) information. Exchanges can be limited to messages sent to other objects which will execute operations and return the results - without the need to reveal how the operation was completed. In this form of the invention, protocol structure is also important and is usually defined in terms of the language used to express the protocols. Java can be used to bring chip card technology programming into mainstream computing and will enable this form of the said invention to be directly connected to standard applications and used on the Internet or GSM networks. Smart cards that support Java will help programmers to integrate the technologies.

The basic principles of adopting an object-orientation are based on the design and development of a new intelligent card system which combines dynamic downloading and a secure execution environment. Essentially chip card systems interface with heterogeneous environments consisting of a potential range of hardware, operating systems, interfaces, communication protocols, programming languages, and applications. In this form, we consider the smart card to provide the secure 'key' functions enabling application interoperability. An object-oriented model offers modularity and clearly defined interfaces for defining services to achieve the set operational objectives. A direct consequence is the ability to dynamically and securely download code that the owner can manage. The owner can add or remove services and configure the MIM to provide electronic and physical information protection and functionality. If data is uploaded to the MIM memory as objects, then the interfaces could be composed of the necessary set of object operations.

According to the said invention, application drivers may also be packaged and sold in a number of forms: For example, they may be sold with pre-personalised smart cards - with the secure key set already in the ROM mask. Standard MIM cards with application driver can also be packaged and sold with smart cards, and the key creation and management package for the user to manage. MIM ROMs can also be produced and personalised if required. They can also be produced 'blank' without a smart card to secure access. In this form, the owner may continue to use a pre-existing secure key for new MIMs or MIM ROMs purchased. The MIM ROM units produced without the need for a smart card can incorporate security features to ensure that the information has not been altered in any way and to be able to audit usage. For example, movies, software application files, educational multimedia files and a range of other information can be purchased by the user for later use. This can be useful in the corporate environment where software use needs to be managed or in the family where access to certain contents might need to be restricted to minors or siblings.

The following example illustrates the steps that might be involved in retrieving and reading one form of a MIM device:

1. A MIM owner looks at the MIM directory displayed on the PC screen and highlights the file (eg. Health Insurance) he/she wishes to open. The owner must ensure that the appropriate application which created the file (eg. MS Word) is already installed in the PC so that it can open the file - if file access is granted.
2. Once the file is highlighted, the R-Read file 'Health Insurance' command is sent to the MIM.
3. The MIM then performs several functions including confirmation that a smart card with the required 'key(s)' is present, that R-read access rights have been granted for that file, and that other security conditions are met.
4. If the required security conditions are not met, one of several messages is returned to the inquirer. For example, the MIM user may be sent the following message, "The smart card inserted cannot open the file < filename >. Please insert an authorised smart card and try again."

5. If the conditions are met, and no further information is required by the MIM (eg. password, PIN), then the file is retrieved and visually displayed on the host PC within the application that originally created it.
6. When the owner has completed whatever tasks they wish to perform with the file, then they may save it to the MIM - with the possibility of altering access privileges at that time.

The following provides a summary of some of the operational advantages which can be achieved though the said invention:
Portability: A MIM card is palm (or pocket) sized and can be easily carried by the owner.
Mobility: A MIM card can be carried in the pocket of the owner and potentially used in a range of hosts - including corporate network terminals, GSM handsets, public access booths or private laptop PC hosts.
Physically robust: A MIM unit is to be made using component parts that are physically very robust. Physically robust microchips for the MIM unit are now becoming available and these can be housed within the confines of a hard protective casing.
Large capacity: The MIM memory can be manufactured to store 4 G bytes of data - possibly up to 10G bytes within the next few years.
Interoperability: One preferred form of MIM interface will meet PCMCIA/JEIDA standards and designed to interface with an MS-DOS file structure. The secure key of choice will be a smart card (SC) and the MIM card will be designed to interface with a PC card reader housed in a personal computer (PC). The MIM card will therefore be compatible with existing and emerging technologies and applications.
User flexibility: The MIM file and security management scheme are designed to ensure that the user can configure and use the MIM card to meet individual or corporate goals.
Improved data storage and archiving: The MIM card offers users a new secure method of storing and archiving large amounts of sensitive compared to existing online distributed or centralised storage systems.
High level of security: The MIM card will be designed to have the highest possible level of security according to the selected options within the smart card and the additional security features that might be added to the MMU/MU and/or secure key unit(s). The security options can be managed by the individual or corporate card owner(s).
Greater individual freedom: The MIM card offers 'individuals' more degrees of freedom and control because the owner can

actively and dynamically manage the card to meet their own particular mode of behavior. This is important as there are few IC card applications with personal flexibility.

Application independent: The MIM card will return files to the owner in the format of the application that created it within a PC host (or other). This means that the MIM is application independent, even though the MIM requires its own software application to be created and some additional software/hardware features may be added.

Low Cost: If the design of the MIM card is based on new dedicated microchip technology, with few hardware components, then it can be mass produced at a fraction of the cost of competing portable memory storage technologies such as the Zip (TM) drive.

Ease-of-use: The MIM user interface will be designed to ensure that the MIM card is managed using an external command set and security management scheme that is analogous to that of many other PC-applications that currently exist. This is to ensure familiarity and 'ease-of-use' for the novice MIM user.

Security advantages: Another significant contribution can arise from the approach to be used for the development of the security of a MIM system. In the MIM card, the role of the smart card as a secure agent is fundamentally different to previous one-card systems. The software and hardware approach to be used enables a secure environment to be created which is suitable for the integration of multiple applications, as well as bypassing the computational bottleneck that occurs if all sensitive data and associated computations are to be fully supported by a single smart card. Much of the security of the MIM card will depend upon the range of protocols between the individual units that can be implemented according to the claims of the invention.

INDUSTRIAL APPLICABILITY

The following examples of applications are intended to be illustrative but not limitative of the present invention and that other embodiments or uses within the spirit and scope of the invention will suggest themselves to those skilled in the art.

MIM devices for individuals: For some individuals, a MIM card may be considered to be a convenient way of securely storing and managing personal data files - either at work or at home. For example, at home the MIM could be used to digitally store videos, games or journal subscriptions downloaded from the Internet - or to archive digital family 'snap shots'. Several family members could manage access to files such as 'snap shots' or games. In turn, these could be easily carried to another home for use. A compact and physically robust MIM card also offers a convenient method of storing a large range of multimedia/entertainment files. For the educational or entertainment field, a MIM ROM might also be produced as a convenient form of access and storage of video material. The cost of producing a MIM ROM would also be considerably less than the cost of producing a re-usable MIM card. Yet, other individuals may use a MIM as a secure and robust file backup system with little or no security required for many files. The MIM offers a more physically robust, cost effective, and lightweight alternative to the ZIP drive for example. However, the same individual may also want the option of securing access to more sensitive information on the 'private' section of the MIM memory unit. This option and additional flexibility is also available.

MIM devices for the healthcare industry: Currently there are many applications in the healthcare industry which require much more memory than is available in a smart card, and which must use other technologies - with a significant loss in security. There are numerous, examples to illustrate this point. For example, the healthcare industry often requires images to be digitally stored and secure (eg. X-rays and echography records). In the future, it is anticipated that continuous measurements such as realtime electrocardiogram data, or to realtime reactions to injections or electrical stimuli according to a complex mathematical protocol may need to be stored in a more accessible and flexible way. Some existing smart card applications in healthcare also pose problems which can be met by a MIM unit. Another example in healthcare where a MIM card could play an important role is in managing accumulated patient data for critically ill patients admitted to Casualty. It is here where unnecessary time delays can result in fatalities. The goal would be to improve the access to primary care diagnostic information that is necessary during the treatment of critical admissions. The MIM card can be used to dramatically improve the time and accuracy limitations of existing record and information systems now in use in some hospitals.

Scenario: A road accident victim Jean aged 10, is accompanied by his father and is admitted to the emergency department of the Mercy Hospital. On presentation at the emergency unit a number of actions are required in quick succession.

1] An admissions officer (AO) begins entering the following information into the MIM unit: patient contact information, next of kin details, patient profile (eg. sex, age, allergies, relevant medical history, previous admissions, current medication, health insurance, religion).
2] At the same time the resident medical officer (MO) will examine Jean and administer any necessary emergency care. The MO then reads Jean's medical profile entered by the AO from the MIM using the video zoom feature before making a decision about drug administration for pain relief and other services to be requested. The MO's emergency presentation report is then completed specifying medication and treatment administered, initial prognosis and diagnosis, pathology tests requested (blood taken and sent to laboratory), and the next service(s) requested.
3] Jean is then moved quickly to an operating theatre where the MIM card information can be viewed on a screen by the surgeon. The surgeon then notes that a radiologist is required to perform a series of cranial and pelvic x-rays before surgery commences. The radiologist then stores x-rays and a report in the MIM which are then available for immediate display on an overhead screen for the surgical team. While this is occurring an anaesthetist also checks Jean's medical profile to now on the MIM and prepares him for surgery. During this time the surgeon(s) are also able to simultaneously access and consider the necessary surgical and backup procedures required for the operation.
4] In the meantime, the pathology test results requested (eg. blood type) are also delivered and a pathology report file added to the MIM by an authorised person for use during and after surgery.
5] After surgery, the chief surgeon adds a report detailing procedures performed and requirements for post-operative care - including medication. This information is then transported with Jean from the operating theatre, to the post-operative care area and finally to a ward with the necessary updates added. Post-operative care staff are then able to manage Jean's post-operative care.
6] When the patient is to be discharged, full billing information, a discharge report, and prescriptions can be made available as required.

In this scenario, the MIM card is able to save time, provide more immediate, complete, integrated information which can be quickly shared among authorised medical staff. It is in scenarios like this where many patients die unnecessarily because of time delays - often due to paper-based, or x-ray film development delays - and sometimes combined with inaccurate and/or inadequate information provision. The most significant patient benefit is the security, completeness and integration of patient data files during the first few hours of emergency care. The MIM card also ensures that the data cannot be altered in any way during this critical time. For this reason, the MIM card offers a more convenient way of storing diagnostic information such as x-rays and CAT scans or dynamic ECG output graphs. Patient files may be backed up at any time on a centralised system if required. The MIM patient card may also be transported with ambulance officers if a critically ill patient is required to be moved to another hospital for more specialist care.

MIM devices for the telecommunications industry: One example is in the management of access and payment for Internet services - or future broadband ISDN services. This is an area of growing concern for carriers, service providers and the Internet users themselves. A MIM using an object-orientation will be an ideal interface between the user requesting a service and the large number of potential Internet service providers. Secure payment can also be an easily added feature by ensuring that the MIM smart card is SET enabled. Electronic articles, videos, games, music and images can all be downloaded onto a MIM, whether they attract a fee or not. The main benefit here is that the identity of the individual requesting the service can also be validated if required. For anonymity, there may be no need for the MIM card holder to be known, but the MIM is still able to store downloaded information.

Other MIM applications: There are also a number of other smart card applications which can benefit by using a MIM card. For example, many consumers accessing vending machines or services use smart cards. However, the daily or weekly collection of records describing details of every transaction cannot be stored on a smart card because of its limited capacity. The data transfers also need to be secure and portable. The management data storage unit will require at least the same level of security as that offered by the consumers' smart cards. A similar problem also exists for smart card applications designed for periodically collecting data from various sites such as gas and electricity meters, or automatic toll payment systems on freeways. The utility of military 'dog tags' based on smart card technology could also be expanded and improved if larger amounts of data could be secured and more flexibly managed using a MIM card.

The corporate office might also benefit. For example, in many corporations, certified software can pose a logistic problem. Distributing and updating the more sensitive applications is not always possible through a network and people often tend to use more and more diskettes - with little or no security.

The video services industry could benefit from a small, secure large data storage module. If for example, a person wishes to download a video to a MIM card, they could then manage the access and use of the video with a smart card. Bill payment, video piracy and customer service access rights could all be better 'managed by the video service provider because of the security features and flexibility of a MIM card.

It is also possible for the MIM card to be used as a medium for confidential file exchange. This can be done with or without the use of encryption. The following two scenarios illustrate how this might be achieved if files saved on the MIM are also encrypted.

Scenario I: Secret message exchange between two MIM holders A and B, A transmits a ciphertext produced by A's MIM. B's MIM is used to decrypt the files sent from A's MIM. That is, a secret exchange has occurred between 'partner' MIMs.

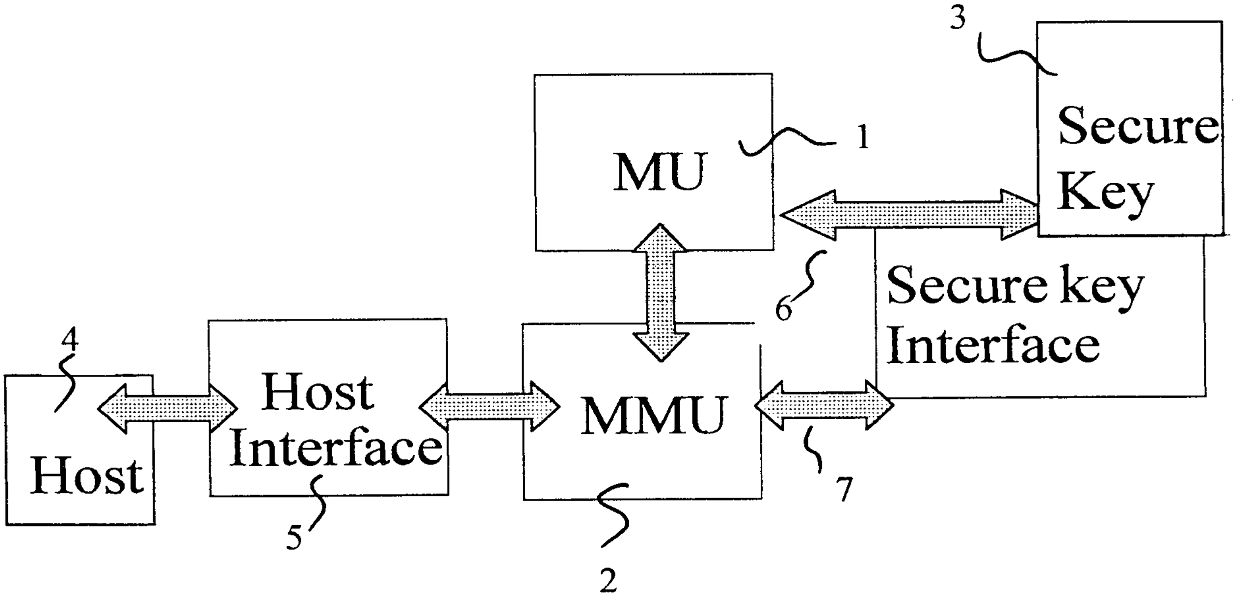
Scenario II: Secure transfer of contents of an individual MIM card between A and B. A saves a hidden files in a MIM and posts it to B without the secure key. B uses a 'partner' secure key to access and read the files sent by A.

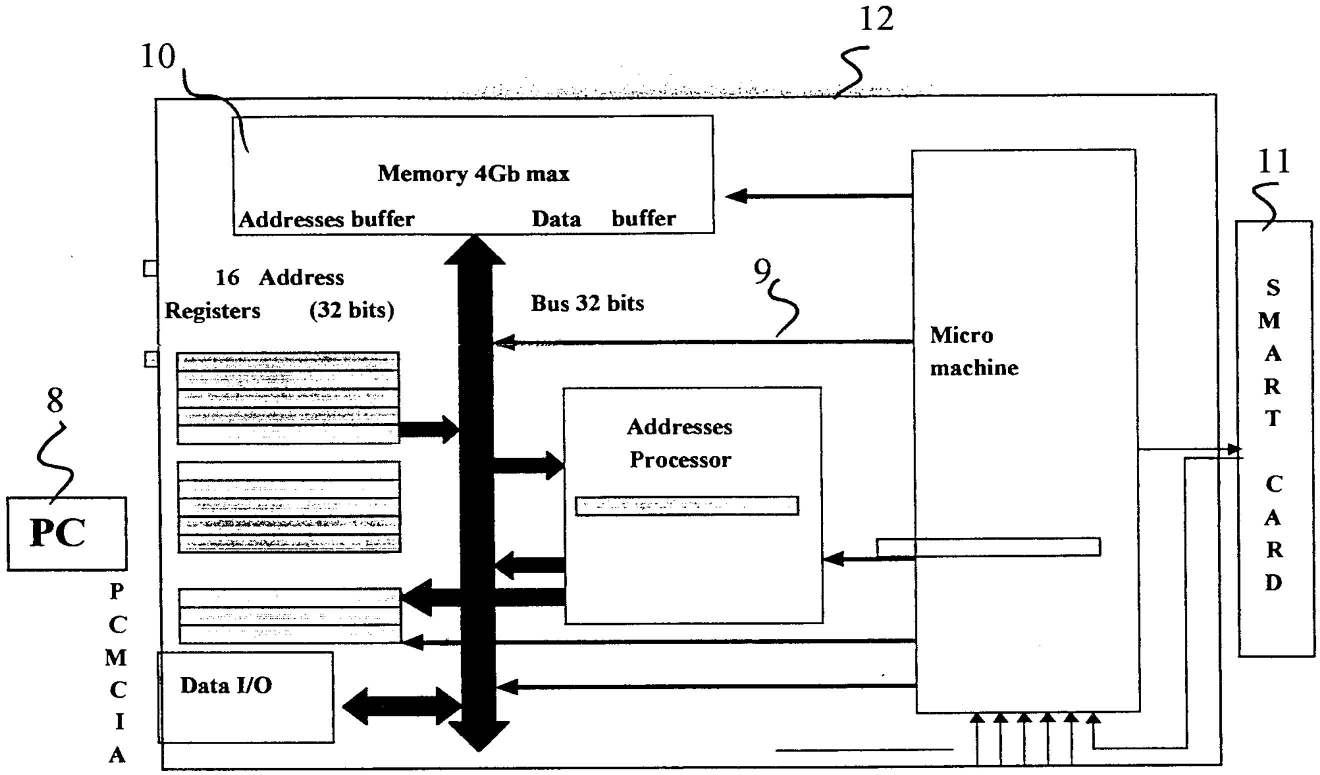
CLAIMS

What is claimed is:

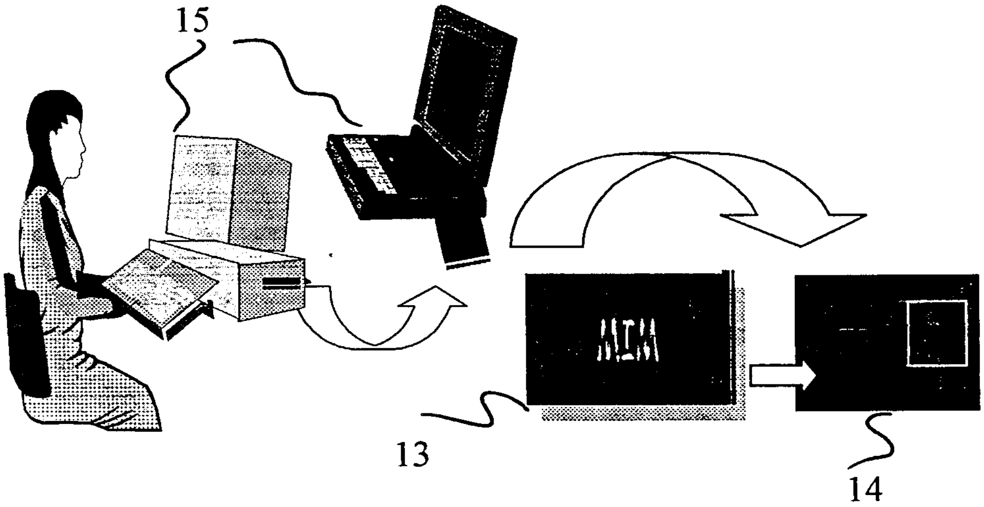
1. A mobile intelligent memory (MIM) unit with removable electronic security key and comprising:
   a) One or more memory units (MU) for data storage;
   b) One or more memory management units (MMU);
   c) At least one removable secure key;
   d) and interfaces connecting the MMU to a peripheral device(s) responsible for initiating inquiries; and
   e) interfaces linking the MMU, the MU and the removable secure key.
2. A MMU has at least three complimentary roles within the MIM unit:
   a) to serve as an intermediary between the host device from which a command is initiated to the MIM unit, and the MU which is able to serve these commands;
   b) to manage and control the sequence of exchanges occurring between the MMU, the host, and the secure key; and
   c) to share in the security management with the secure key.
3. One or more MUs will be responsible for storing data and will consist of at least two main parts: a physical manager; and a memory storage area.
4. The memory storage unit(s) can be configured to consist of one or more 'private' volumes which are secured using a secure key; and the possible option of incorporating one or more 'public' volumes which can be readily accessed without the need to use the secure key.
5. The secure key(s) will be responsible for managing access to the MU(s) in co-operation with the MMU(s). The MU(s) and the MMU(s) may be physically separate units within the said MIM device housing, or may be integrated to perform the required functions listed in Claims 1,3,4,7,8.
6. A range of additional security, software and hardware options can be incorporated into the MMU(s), MU(s) or secure key(s) to provide added levels of security or to enhance functionality.
7. The MIM device will also require the secure key(s) to co-operate with the MMU(s) and MU(s) to protect the said MIM device against possible hardware attacks.
8. Although it will be necessary for the secure key(s) and the host to exchange commands for secured applications, there will be no direct logical or hardware link(s) to this architecture. Thus much of the said MIM device's security and operational flexibility will rely upon the exchange protocols occurring between the above listed individual units.

1/3


3
Secure Key
MU
1
Secure key Interface
6
4
Host
Host Interface
MMU
7
5
2


10
12
Memory 4Gb max
Addresses buffer
Data buffer
11
16 Address
Registers (32 bits)
9
Bus 32 bits
Micro
machine
S
M
A
R
T
C
A
R
D
8
Addresses
Processor
PC
P
C
M
C
I
A
Data I/O

3/3



PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ^{6} : G06K 19/07	A1	(11) International Publication Number: WO 99/46727 (43) International Publication Date: 16 September 1999 (16.09.99)

(21) International Application Number: PCT/US99/05292

(22) International Filing Date: 11 March 1999 (11.03.99)

(30) Priority Data:
09/038,294 11 March 1998 (11.03.98) US

(71) Applicant: ERICSSON INC. [US/US]; 7001 Developmemt Drive, Research Triangle Park, NC 27709 (US).

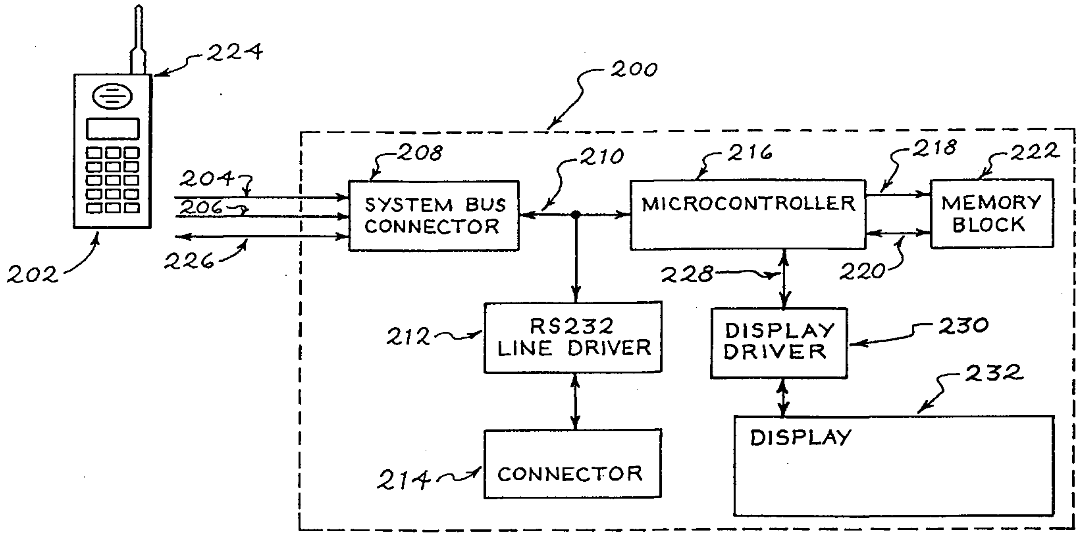
(72) Inventor: BRIGHT, Randall, G.; 104 Tripp Road, Pittsboro, NC 27312 (US).

(74) Agent: BENSON, Joel, W.; Brinks Hofer Gilson & Lione, P.O. Box 10087, Chicago, IL 60610 (US).

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

Published
With international search report.
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: PORTABLE TELEPHONE ACCESSORY FOR TEMPORARY STORAGE OF FAX AND DATA



(57) Abstract

A plug–in module for a portable telephone, capable of storing data, fax, and voice messages onto flash memory. The module is powered by the host telephone, and interfaced using serial lines. The serial signals are relayed to external supplemental devices using an RS–232 interface.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

PORTABLE TELEPHONE ACCESSORY FOR TEMPORARY STORAGE OF FAX AND DATA

Background of the Invention

The present disclosure relates to Digital Advanced Mobile Phone Service (DAMPS), and more particularly to the application of detachable nonvolatile memory to digital cellular telecommunication devices.

The first generation of cellular technology uses an analog modulation process to convey information from point to point. The system design known as Advanced Mobile Phone Service or AMPS employs analog frequency modulation for speech transmissions and frequency shift keying for signaling. The concept assigns each call a pair of unique frequencies within a limited geographic area. Unlike open platform protocols, each cellular call is serviced by a semi-private two channel line. The first channel is dedicated to broadcast transmissions and the second channel is dedicated to receiving transmissions.

To increase accessibility, interoperability, and functionality, the conventional analog infrastructure is gradually being replaced by digital technology. Digital telecommunication offers several advantages over conventional analog systems. Ease of processing, ease of multiplexing, ease of encryption processes, high noise immunity, improved spectral efficiency, improved data transmissions, enhanced speech quality, and an ability to support new functionality such as integrated paging, messaging services, and caller identification are a few of the advantages digital telecommunication offers over conventional analog architecture.

The first generation of digital protocol approved by the FCC in 1990 was IS 54 or Time-Division Multiple Access (TDMA). TDMA is a dual mode analog and digital platform that accommodates digital and analog protocols. TDMA triples the capacity of current analog channels by deriving three separate digital channels or time slots from each analog channel. With the advent of IS 136, the digital control channel offers enhanced services such as slotted paging, caller, number, and name identification, point-to-point text messaging services, and integrated paging which require application software residing in memory.

Code Division Multiple-Access (CDMA) is another standard of digital wireless communications. The concept behind CDMA is to digitally modulate data on a given common frequency assigned a complex pseudo random code. The process only deciphers transmissions by extracting data assigned to a given code, and hence, is an efficient use of available bandwidth. Because the key to performance of such system in detection of a signal is the signal coding, this type of cellular protocol requires complex processing supported by sufficient memory. According to conventional practice, cellular memory is burdened by the task of storing all the application software of the cellular system. Unfortunately, the underlying complexity of CDMA and TDMA protocols limit the functionality of cellular systems as cellular features are critically dependent on size and efficiency of cellular memory.

Progress in the cellular industry has been guided by the principal of better performance at a minimum cost. Given that additional memory is an effective way of increasing cellular performance, there exists a need to provide a reliable expansion card capable of providing real-time performance and fast read/write access at a low system cost. With continued reliance on permanent resident nonvolatile memory, current cellular technology is limited to the manufactured state of the art, and a point of diminishing return is continuously reached as innovation exceeds memory capacity.

The escalating requirements of digital cellular technology and services including fax, integrated paging, messaging, data transmission, caller alert, require a reliable inexpensive portable memory. The memory must be physically compatible with the decreasing size of hand-held portable units, easy to install, consume little energy, and offer long-term compatibility to ever changing digital standards.

SUMMARY OF THE INVENTION

A memory device for providing fast access, nonvolatility, and low power consumption memory in a TDMA or CDMA-based telecommunication system is disclosed. The memory device combines resident memory with the high performance of a dedicated detachable block of nonvolatile memory. The memory device is comprised of two interface buses capable of supporting a read/write architecture. The memory hierarchy connected to one interface bus is modeled as a non-interleaved functional unit having an internally managed nonvolatile memory operative to store a plurality of embedded algorithms and a modular memory cartridge. The modular memory cartridge includes a second interface bus linked to a microcontroller having at least one serial data driver built therein. The modular memory cartridge further includes a block of low-voltage nonvolatile memory. The memory device may also be virtual memory comprising a block of resident memory and a portable nonvolatile flash memory card.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 depicts a conventional digital cellular telephone having a permanent resident nonvolatile memory.

FIG. 2 is a block diagram of a first embodiment of a modular memory cartridge.

FIG. 3 depicts the interface of the modular memory cartridge to a digital cellular telephone in accordance with FIG. 2.

FIG. 4 depicts a second embodiment of the present invention.

FIG. 5 depicts the interface of the embodiment depicted in FIG. 4 with a digital cellular telephone.

FIG. 6 depicts a third embodiment of the present invention.

DETAILED DESCRIPTION OF THE PRESENTLY PREFERRED EMBODIMENTS

The present disclosure departs from conventional digital cellular technology by increasing the design functionality of digital cellular devices. The present disclosure enables the digital cellular user to adapt to a variety of cellular platforms that achieve improved reliability, higher operating speeds, and improved performance through a memory hierarchy modeled as a non-interleaved functional unit.

Digital cellular technology conventionally relied on a stand-alone fully integrated cellular telephones as illustrated in FIG. 1, while avoiding the use of modular assemblies. Cellular technology is static sensitive, and therefore, a pin or socket misalignment which may occur in modular assembly can adversely affect the performance of a digital cellular telephone or worse result in its catastrophic failure.

Moreover, due to the typical large sizes of external memories, smaller designs of digital cellular telephones, and focus on low-power electronic design, ancillary memory devices were considered incompatible with existing digital cellular design strategy.

In an embodiment of the present invention, a memory cartridge is fully enclosed within a nonconductive insulative sheath to protect against electronic static discharge (ESD). In a preferred aspect of this embodiment, a digital cellular circuit which includes a digital transmitter, a digital receiver, and a digital logic circuit interfaced to the memory cartridge share a common power and ground plane further reducing the possibility of static induced damage. In another preferred aspect of this embodiment, a system bus connector provides further isolation between the digital cellular circuit, the memory cartridge, and a plurality of peripheral electronic devices the memory cartridge may drive. Thus, the use of the system bus connector between the ancillary memory and the cellular circuitry also affords isolation between the digital cellular technology and a plurality of peripheral devices. In one embodiment of the invention, a nonvolatile flash memory card having between a two and eight megabyte density was enclosed in a thirty-eight by thirty-three millimeter package which is fully compatible with the smaller designs of digital cellular telephones.

An embodiment is illustrated in FIG. 2, wherein for simplicity, depicted elements are not necessarily drawn to scale and alike and similar elements may be designated by the same reference numerals through several views. As shown in FIG. 2, a memory cartridge 200 has a system bus connector 208, an RS232 line driver 212, an RJ-11 connector 214, a microcontroller 216, and a block of nonvolatile memory 222. Accordingly, the system bus connector 208 provides a serial interface 226 between a digital cellular telephone 224 serial port 202 and the micro-controller 216 serial port 210 to support a read/write architecture. While actual serial communication can occur in several ways, in this embodiment the microcontroller 216 drives and receives serial communication through a built in serial port chip or UART. Such serial communication may be based on an AT command structure. Besides providing a means for facilitating communication between the digital cellular telephone 224 and the microcontroller 216, the system bus connector 208 provides isolation between a power 204 and a ground 206 bus that derives power from a portable power source connected to the digital cellular telephone 224.

Operation of the memory cartridge 200 is controlled by the microcontroller 216. When the microcontroller 216 is booted up, it looks for memory. The nonvolatile block of memory 222 may be accessed by the microcontroller 216 by means of a parallel bus comprised of an address bus 218 and a data bus 220. The RS232 line driver 212 coupled to the RJ-11 connector 214 provides bipolar data and control signals of substantial drive capability to a plurality of peripheral devices. In this embodiment, a low voltage-flash memory block having less than one-hundred and twenty nanosecond access time, distributing data in a sequential file format, and compatible with a single power source was implemented. A dual driver positive receiver RS 232 integrated circuit may be used in this embodiment because it conveniently has an on-chip flying-capacitor voltage doubler and inverter and therefore is capable of running from a single positive supply.

As illustrated in FIG. 3, the memory module 200 is freely attachable and removable from the digital cellular telephone 224. The memory module 200 easily snaps onto the lower end of the digital cellular telephone 224 by the engagement of a plurality of locking tabs 306. The locking tabs 306 are flexibly connected to the memory module 200 which is enclosed within a nonconductive insulative sheath 304 hermetically sealed to repel contamination and cushion shock trauma. The performance characteristics of the memory module may be further improved by the use of elastomer connectors on the memory module 200 and the digital cellular telephone 224 to minimizes pin and socket misalignment that may occur in modular cellular assembly.

The use of ancillary memory expands the current functionality of digital cellular telephones by providing storage capacity that enables over the air reprogramming, point-to-point messaging, integrated answering functions, and data logging for later recovery and analysis.

The embodiment of FIGS. 4 and 5 is similar to that depicted in FIGS. 2 and 3 as it comprises a block of nonvolatile memory. However, the conditioning circuitry of the previous embodiment is integrated within the digital cellular telephone 224 having a portable power source and therefore is not needed in a nonvolatile miniature memory card 400. Thus, FIGS. 4 and 5 constitute a further improvement of the embodiment shown in FIG. 2, by decreasing the number of components of the ancillary memory thereby reducing its size without affecting its interchangability. In this embodiment, the nonvolatile miniature memory card 400 is a block of flash-memory seamlessly integrated with a block of resident cellular memory which in association is referred to as a virtual memory. In another embodiment, the miniature memory card is interfaced to a controller that monitors the integrity of the read/write cycles. By sequentially writing to a given memory address and then reading its content, the controller may detect a memory failure and notify the digital cellular circuitry to prevent further storage at that address.

As illustrated in FIGS. 4 and 5, the nonvolatile miniature memory card 400 is attachable and removable from the digital cellular telephone 224. The nonvolatile miniature memory card 400 easily snaps into the back of the digital cellular telephone 224 by the engagement of a plurality of locking tabs 404. The locking tabs 404 are flexibly connected to the nonvolatile miniature memory card 400 which is enclosed within a nonconductive insulative sheath 406.

The illustrated embodiments employ software that automatically configure the ancillary memory and limit the number of write cycles of each memory address. When a user attaches a memory expansion card or module, the memory is automatically operational without user support. The embodiments utilize a variety of Plug and Play technology wherein each module is uniquely identified, capable of stating the services it provides, capable of identifying the software driver that supports it, and allows the operating software to configure its use. A digital cellular telephone user simply attaches the ancillary memory device and it begins to play. Besides providing a common platform that enables digital cellular users to support new digital services, the digital cellular Plug and Play memory expansion device provides the user with greater mobility. A user may remove the portable memory device from the digital cellular telephone without interrupting a digital cellular transmission and dock the portable memory without losing memory content or having to configure the memory device to the docking station's operating software. A docking station could then retrieve the data for further processing or download additional data to be used or transmitted by the digital cellular telephone 224.

Accordingly, a docking station is any device that supports Plug and Play technology having a serial data communication port, like a computer.

The illustrated embodiments can also employ a visual display. FIG. 6 shows an LCD display 232 having a display driver 212 serially connected 228 to the microcontroller 216. The visual display may be mounted onto the memory cartridge 200 as shown in FIG. 6 or directly onto the nonvolatile miniature memory card 400. In the embodiment depicted in FIG. 6, when the memory module 200 is directly connected to the cellular telephone 224, the cellular telephone keypad 308 may function as a means for scanning the contents of the memory module 200 on the LCD display 232. In another embodiment, the visual display may be coupled directly to a portable scanning device and a portable power source so that the visual display is part of a fully functional portable memory module when it is detached from the digital cellular telephone 224. In a further embodiment, the visual display is an LED display. Various embodiments may also employ hybrid electronic displays.

The concepts and processes previously illustrated may be implemented through software and logic circuitry. The aforementioned embodiments were employed using conventional circuitry and software including an RS232 serial data driver, an Intel Series 100 Flash Memory Miniature Card, a FTL Flash File System, and software adapted from a Common Flash Interface Specification and a Plug and Play Design Specification. Although the disclosure is not limited to a block of flash memory as a battery backed SRAM or an EEPROM may also be used, the use of portable flash memory provides fast access times, high endurance cycles, low energy consumption, single power supply operation, direct executions meaning code and data may be read directly from memory, and a smaller size in comparison to conventional storage devices. The disclosed embodiments enjoy utility in any digital cellular telephone application.

Variations and modifications of the embodiments disclosed herein may be made without departing from scope and spirit of the invention. The aforementioned description is intended to be illustrative rather than limiting and it is understood that the scope of the invention is set forth by the following claims.

I CLAIM:

1. A digital cellular telephone system for storing incoming cellular voice, fax, and data transmissions, wherein said digital cellular system comprises:

a first interface bus capable of supporting a read/write architecture;

a memory hierarchy connected to said first interface bus, wherein said memory is modeled as a non-interleaved functional unit comprising:

an internally managed resident nonvolatile memory operative to store a plurality of embedded algorithms of said digital cellular telephone; and

a modular memory cartridge structured to include:

a second interface bus having address lines, data lines, and control lines;

a microcontroller coupled with said second interface bus and having at least one serial data driver built therein;

a block of low-voltage nonvolatile memory conductively coupled to second interface bus.

2. The digital cellular system as defined in claim 1, wherein said nonvolatile memory residing in said modular memory cartridge is a flash memory and wherein said memory module is enclosed in a nonconductive insulative sheath.

3. The digital cellular system as defined in claim 1, wherein said microcontroller is operative to distribute said data in a sequential file format.

4. The digital cellular system as defined in claim 1, wherein said embedded algorithms are reconfigurable operating software responsive to said digital cellular transmissions.

5. The digital cellular system as defined in claim 1 further comprising an expansion bus interface operatively coupled between said first interface bus and said microcontroller.

6. The digital cellular system as defined in claim 5 further comprising an RS232 interface operatively coupled to said expansion bus interface and said

microcontroller, said RS232 interface operative to distribute data to one or more electronic peripheral devices.

7. A smart digital cellular apparatus capable of temporarily storing fax and data transmissions, comprising:

a digital cellular circuit having a digital transmitter, a digital receiver, and a digital logic circuit, said digital cellular circuit digitally linked to one or more digital switching stations;

a portable power source operably coupled to provide power to said digital cellular circuit;

a virtual memory device seamlessly integrated with said digital cellular circuit and drawing power from said power source, comprising:

a resident nonvolatile memory device responsive to receiving and transmitting said digital cellular transmissions; and

a transportable nonvolatile miniature memory card responsive to user programmable features.

8. The smart digital cellular apparatus of claim 7, wherein said nonvolatile memory is a mask-programmed read only memory.

9. The smart digital cellular apparatus of claim 7, wherein said nonvolatile miniature memory card comprises a block of flash memory enclosed within a nonconductive insulative sheath.

10. The smart digital cellular apparatus of claim 7, wherein said miniature memory card is interfaced to a controller constructed to monitor the integrity of the programmable and erasure cycles of said memory, wherein said controller is associated with an authentication device to prevent the reprogramming of failed memory sectors and capable of communicating an address of said failed memory sectors to said digital cellular circuit.

11. The smart digital cellular apparatus of claim 7, wherein said nonvolatile memory device is embedded with algorithms responsive to over the air reprogramming in a digital cellular format including at least one of a plurality of digital cellular formats including a Time-Division Multiple Access format and a Code-Division Multiple Access format.

12. The smart digital cellular apparatus of claim 7 further including an answering device interfaced to said miniature memory card for recording an outgoing greeting and storing incoming messages from a digital cellular transmission in a compressed file format in said transportable nonvolatile miniature memory card.

13. The smart digital cellular apparatus of claim 7, wherein said miniature memory card is a flash miniature memory card easily removable from said digital cellular circuit and interchangeable with a plurality of pre-programmed miniature memory cards capable of reconfiguring the functionality of said digital circuit.

14. The smart digital cellular apparatus of claim 13, wherein said programmable features support a pre-programmed communication service configurable to the user of said smart cellular apparatus and reconfigurable by interchanging said pre-programmed miniature memory card with a plurality of said pre-programmed miniature memory card.

15. A method of interfacing a multi-level hierarchy of nonvolatile memory in a Digital Advanced Mobile Phone Service, comprising the steps of:

providing a digital cellular telephone having a digital logic circuit, a plurality of digital memory, an interface bus, and a portable power source;

electrically coupling said digital logic circuit to said power source;

operably coupling said interface bus to said digital logic circuit;

interfacing a fixed block of resident nonvolatile memory to said interface bus and said power source, said fixed block of resident nonvolatile memory operative to store the operating system of said digital cellular telephone; and interfacing a detachable block of nonvolatile memory to said interface bus and said power source.

16. The method of claim 15, wherein said detachable block of nonvolatile memory is a Plug and Play flash memory card configured to said Digital Advanced Mobile Phone Service for storing a plurality of user programmable features.

17. The method of claim 15, wherein said nonvolatile memory are flash memory that draw power from said power source only during read/write cycles.

18. The method of claim 15 further providing a display means, wherein said detachable block of nonvolatile memory is interfaced to a visual display that provides means for displaying the contents of said nonvolatile memory.

19. The method of claim 15 further providing a microcontroller having at least one serial data driver built therein operably coupled between said fixed block of nonvolatile memory and said detachable block of nonvolatile memory for managing data transfers between said fixed memory and said detachable memory.

20. The method of claim 19, wherein at least one of said serial data interfaces is an RS232 interface operative to distribute data to one or more electronic peripheral devices.

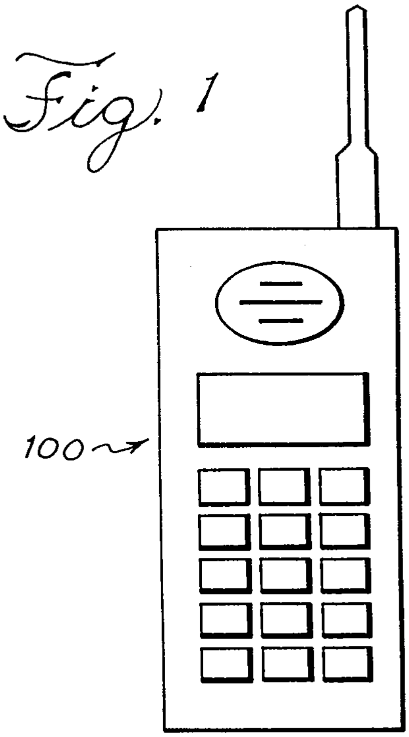


Fig. 1

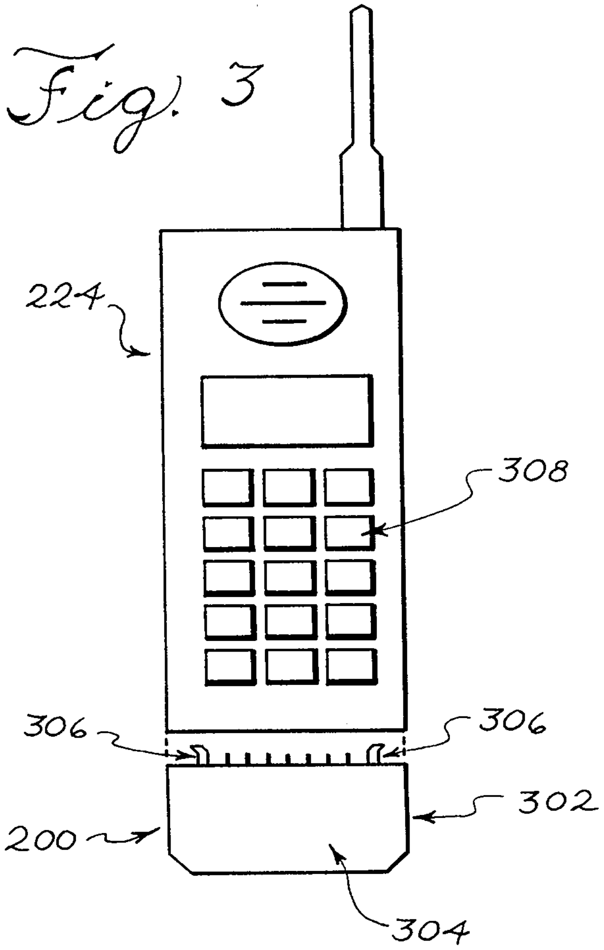


Fig. 3

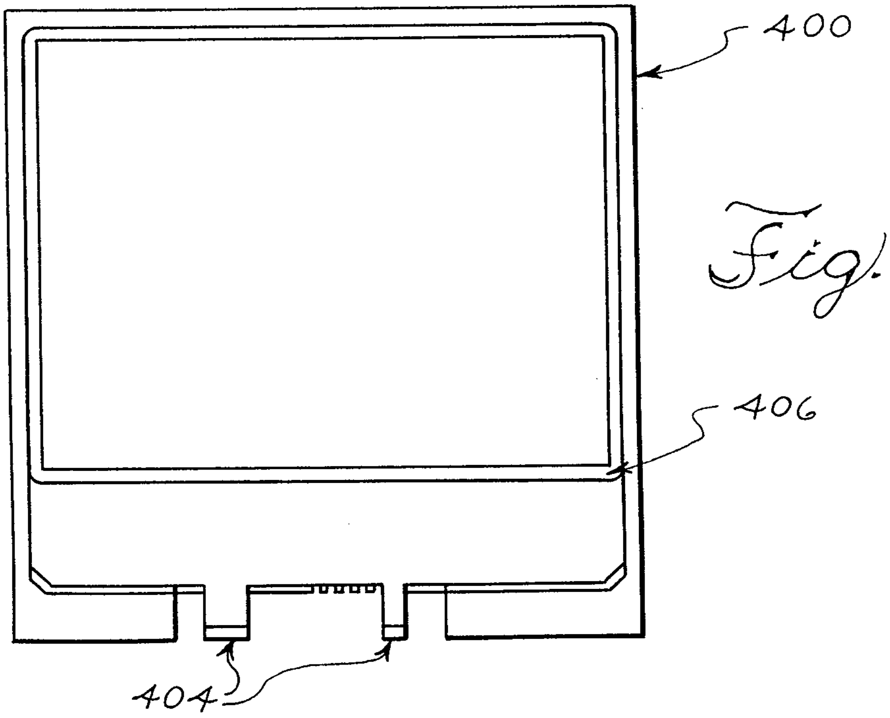


Fig. 4

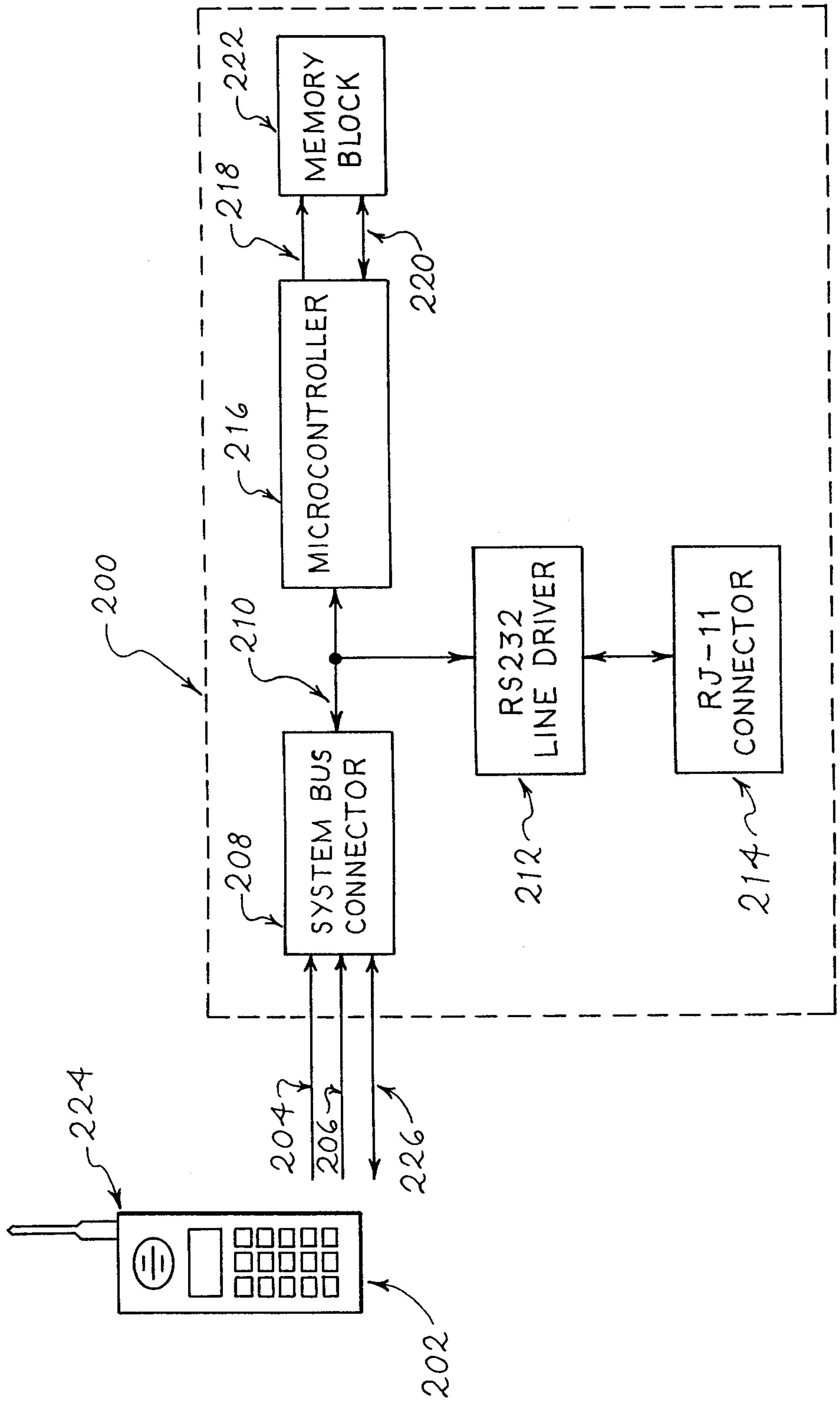

200
222
MEMORY BLOCK
218
220
216
MICROCONTROLLER
210
208
SYSTEM BUS CONNECTOR
RS232 LINE DRIVER
212
RJ-11 CONNECTOR
214
224
204
206
226
202

Fig. 2

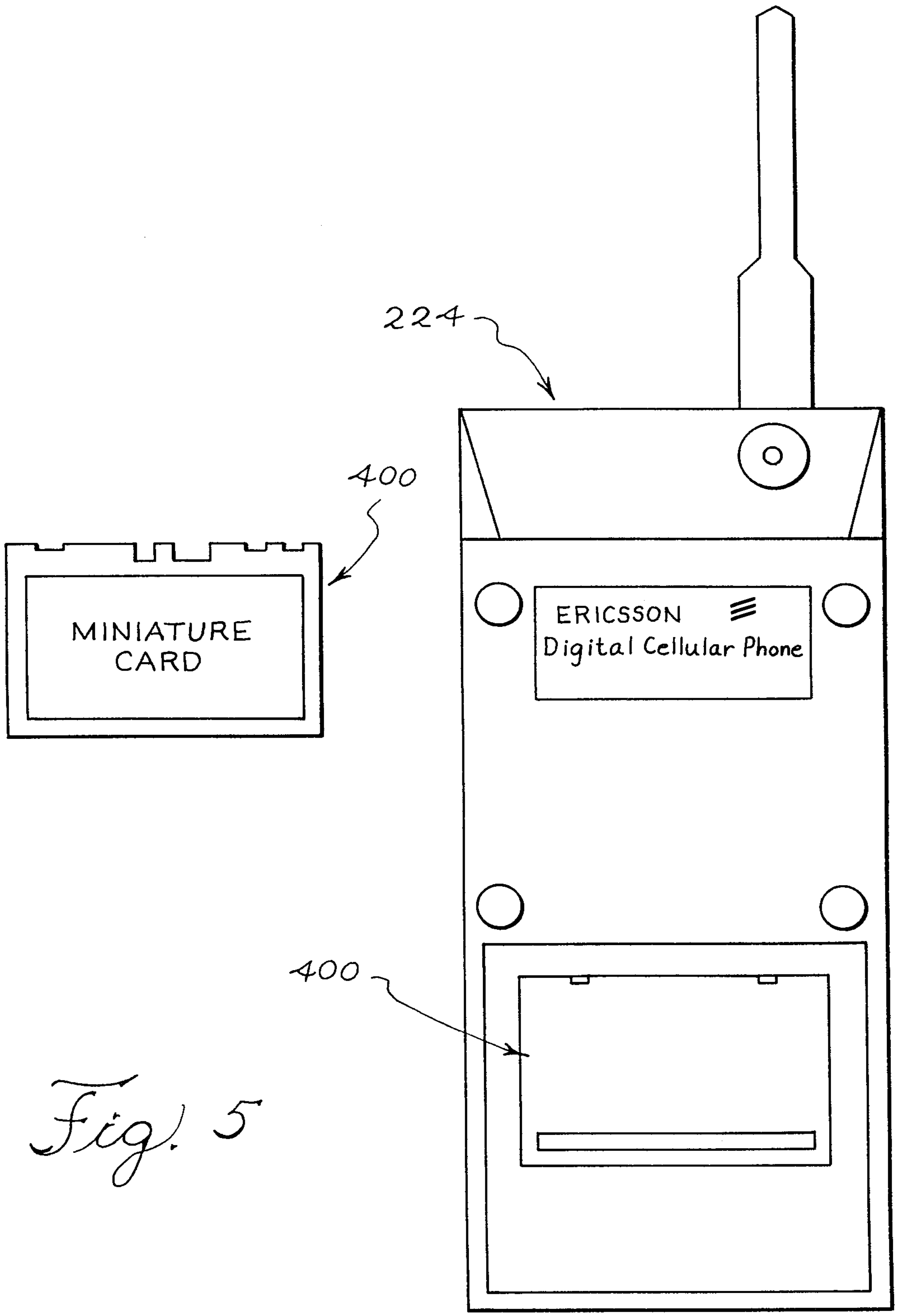

224
400
MINIATURE CARD
ERICSSON
Digital Cellular Phone
400

Fig. 5

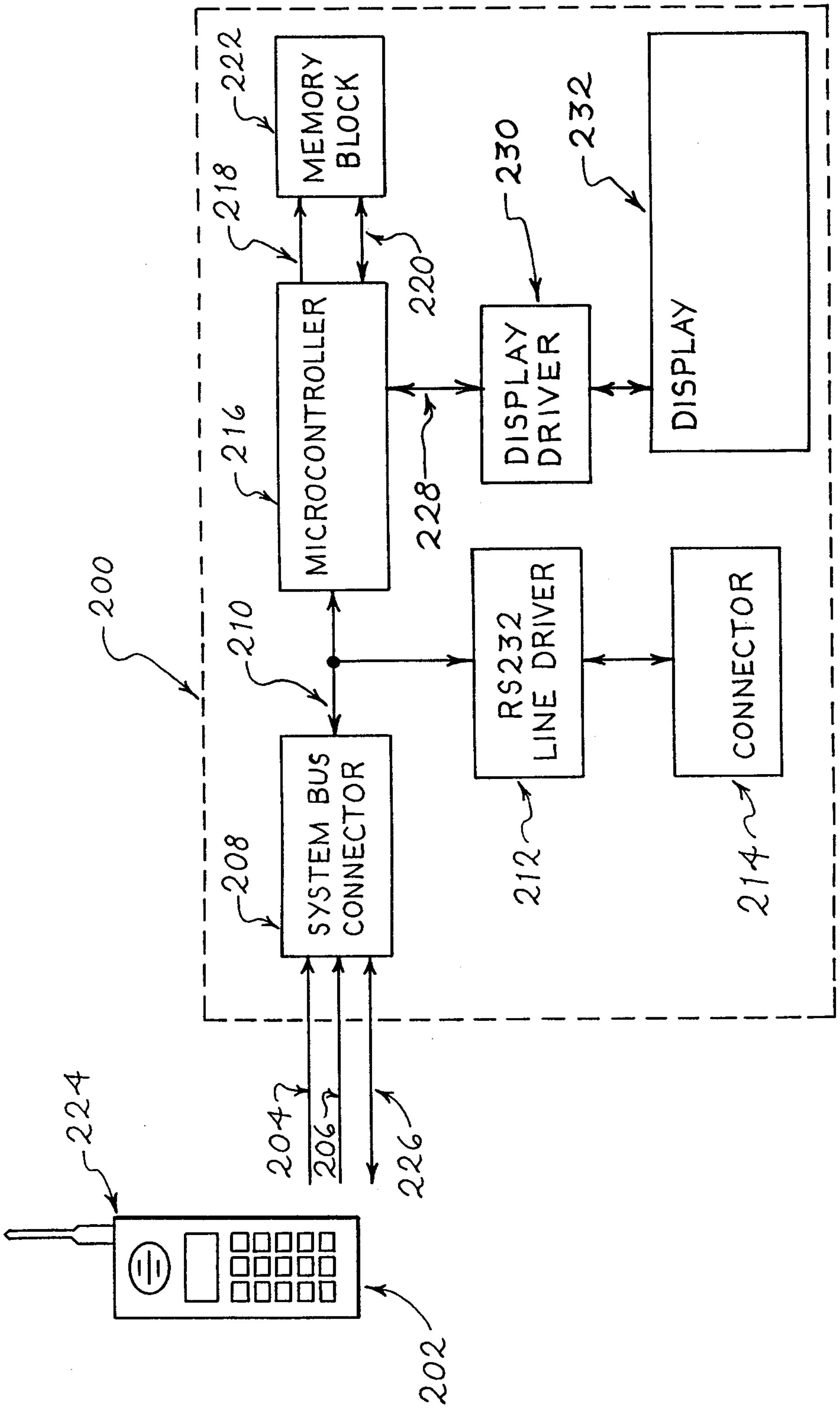

200
SYSTEM BUS CONNECTOR
208
MICROCONTROLLER
216
MEMORY BLOCK
222
218
220
210
RS232 LINE DRIVER
212
CONNECTOR
214
DISPLAY DRIVER
230
228
DISPLAY
232
204
206
226
224
202

Fig. 6

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ^{6} : **H04L 12/28, G06F 1/00**	**A1**	**(11) International Publication Number:** **WO 99/48250** **(43) International Publication Date:** 23 September 1999 (23.09.99)

(21) International Application Number: PCT/ZA99/00005

(22) International Filing Date: 19 March 1999 (19.03.99)

(30) Priority Data:

98/2302	19 March 1998 (19.03.98)	ZA
99/1811	8 March 1999 (08.03.99)	ZA

(71)(72) Applicant and Inventor: MOSTERT, Christiaan, Frederik, du Toit [ZA/ZA]; No. 5 Valley Road, Westcliff, Johannesburg 2193 (ZA).

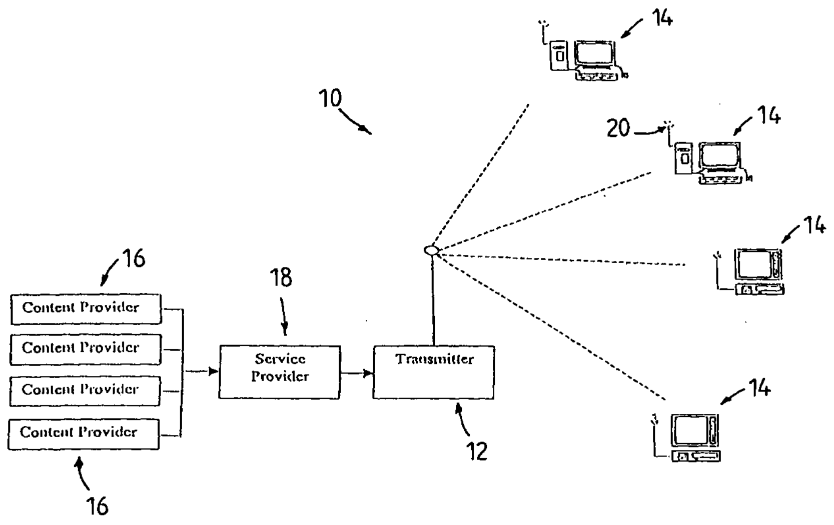
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: HIGGINSON, David, Charles [ZA/ZA]; 49 Joseph Avenue, Northcliff, Johannesburg 2115 (ZA). HIGGINSON, Martin, Roy [ZA/ZA]; 52 Bianca Avenue, Berario, Johannesburg 2195 (ZA). NEL, Pierre, Hercules [ZA/ZA]; 502 Tennessee Street, Faerie Glen, Pretoria 0043 (ZA).

(74) Agent: D.M. KISCH INC.; P.O. Box 781218, Sandton 2146 (ZA).

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

Published
With international search report.
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 DISTRIBUTING INTERNET TO MULTIPLE USERS



(57) Abstract

The invention provides a system and method for transmitting information to multiple users simultaneously, over a wireless communications network, and for receiving, demodulating, downloading and storing the information at user bases, for access at any future time. The system comprises: one or more content providers (16), such as a news company, stock brokerage firm, Internet service provider, publisher or university; one or more service providers (18) that manage the information into channels; existing wireless transmission infrastructure (12), for example, that used by radio companies, and; a plurality of PC–based receivers (14) at user bases. The user base typically comprises an antenna (20), in conjunction with a specialised radio card designed to implement modulation techniques such as GMSK, QPM and Galaxy Modulation; a modem for demodulating the broadcast signal, and; processing means, such as a personal computer.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

METHOD AND SYSTEM FOR DISTRIBUTTING INTERNET TO MULTIPLE USERS

TECHNICAL FIELD

This invention relates to a method and system for simultaneously distributing information to a plurality of user bases. More specifically, it relates to the wireless transmission of information from a broadcaster to multiple receivers for storage and access at the user's convenience.

BACKGROUND ART

With the introduction of the Internet and World Wide Web many users have access to information over the Internet. Although the availability of information has been advanced with so-called web sites, downloading and accessing of information is a time consuming process due to limited data throughput and transfer rates over standard telephone lines.

Modulation techniques that sacrifice signal data for a lower ambient noise threshold while maintaining a relatively high data transfer rate over a fixed bandwidth channel are conventionally used in the telecommunications industry. Examples of these advanced modulation techniques are Gaussian Minimum Shift Keying (GMSK), Quadrature Polyphase Modulation (QPM) and Galaxy Modulation (GM). However, technologic advancements in the field of electronics and the subsequent reduction in the size and cost of electronic components, has enabled these modulation techniques to be implemented in broadcasting systems.

In most instances, it is required that users of information networks purchase a modem and pay monthly subscription fees to an Internet Service Provider (ISP). Subscribers are entitled to dial into an ISP at a Point of Presence (POP), and are subsequently granted access to the information superhighway.

Research has shown that certain pre-registered web sites are favoured above others and are accessed more frequently. It has also revealed that certain favoured web sites are accessed more for the purpose of obtaining information, such as stock prices, news, weather, etc., rather than for purely for entertainment.

The process of logging into an ISP and subsequently being granted access to a web site for downloading information from the site, is tedious and in most cases time-consuming. Furthermore, most users access the Internet during business hours, which is the time when telephone costs are most expensive.

The Internet typically forwards information on a "pull" system which is facilitated by a dial-up connection. The current "Push" system does not allow large quantities of information to be provided to multiple users, due to the limitation of the telecommunications network.

A present dial-up connection to the Internet allows the user to request the service provider to transmit large amounts of information satellite, to avoid lengthly download times. However, the transmission has to be requested by the user and the information is independently sent by the service provider to a specific user for each request.

OBJECTIVES OF THE INVENTION

Accordingly, it is an object of the present invention to provide a system and method for simultaneously broadcasting large quantities of information over the airwaves to a plurality of receivers, as well as for downloading information at a user base with which the above disadvantages of known systems could at least be alleviated.

Furthermore, it is an object of the invention to provide users with a system that may enable and/or facilitate one or more of the following:

- product delivery systems – these are orders placed via the Internet or otherwise for data based products, such as software, which can be delivered effortlessly without time-consuming and costly Internet downloads. The sale of music CD's is a perfect application for the invention as a product delivery system;
- mail delivery notification – alerting the user to the presence of new mail. While

E-mail remains the fastest possible communication platform, most people have access to undedicated telephone systems with the result that the mȧilbox is checked on an intermittent basis. It is only those ISP subscribers with costly dedicated digital connectivity that are notified of new mail, seconds after it is sent. The invention enables delivery notifications to be broadcast as E-mail is sent, making it possible for the user to log into the Internet and access their mail as it arrives;

- downloads – transfer of information from a web page to a user base. While the Internet offers users an incessant source of free product downloads, the speed and associated costs remain a deterrent. The invention enables requested information to be transmitted inexpensively to a plurality of user bases without the tedious and costly exercise of Internet downloads;

- business information – receiving updates of business orientated information such as share-prices, exchange rates and the like, with the use of a subscription service. Similar systems are already in place in telecommunication systems, once again only effective for those who enjoy dedicated connectivity;

- community and crime prevention applications are also limitless - the timeous delivery of information such as stolen credit card lists, stolen vehicles, missing persons, etc. for the identification of fraud and combating of crime; and

- delivery of internet content to less privileged or rural areas without telecommunications systems and facilities.

DISCLOSURE OF INVENTION

According to a first aspect, the invention provides a system for facilitating the simultaneous transmission of information to multiple user bases over a wireless communications network and for receiving, demodulating, downloading, and storing the information at the user bases for subsequent retrieval, the system comprising at least one content provider; at least one service provider; a transmission infrastructure; multiple user bases, having receivers consisting of an antenna in conjunction with a receiving card; a modem for demodulating the broadcast signal; and processing means for storing and enabling subsequent access of the information.

The system may include means to manage one or more switchable channels, enabling them to be broadcast selectively to a certain subset of users by activating and/or deactivating a specific channel of information.

The transmitted information signal may incorporate means for encoding or encrypting, the corresponding receiver including means for decoding or decrypting the signal at the user base.

The means for encoding and encrypting may be provided with an encryption algorithm that is a function of the user-specific identification code inherent in the receiver card, and further may be provided with a key obtained on payment of the desired channel subscription, ensuring that only paying subscribers are able to decrypt the signal.

The receiver may have an antenna associated with it as part of a computer module, alternatively, the antenna may be a separate unit connectable with a display and processing device.

The system may include means for compressing the information signal and the user base may include means for decompressing the information after it has been downloaded.

The system may use existing transmission infrastructure such as that used by radio companies.

According to a second aspect of the invention there is provided a method for facilitating the simultaneous transmission of information to multiple user bases over a wireless communications network and for receiving, demodulating, downloading, and storing the information at the user bases for subsequent retrieval including the steps of collecting information from at least one content provider; classifying and grouping the information into channels; generating a modulated information signal for transmission; broadcasting the modulated information signal over a wireless transmission network; receiving the transmitted information signal at user bases via suitably tuned receivers; demodulating the received information signal; and storing the information for subsequent retrieval.

The method may include a step of automatically refreshing the stored information with an updated version.

The step of modulating the information signal may be achieved using any one or more of modulation techniques selected from the group consisting of Gaussian Minimum Shift Keying (GMSK), Quadrature Polyphase Modulation (QPM) and Galaxy Modulation.

The step of modulating the information signal may include implementing a redundancy check to ensure that the received signal is accurate and to enable a corrupted signal to be reconstructed at the receiver.

BRIEF DESCRIPTION OF DRAWINGS

Preferred embodiments of the invention will now be described by means of non-limiting examples only, with reference to the accompanying diagrams wherein:

Figure 1: is a block diagram of a first embodiment of the invention, which uses a RF transmission network to distribute information;

Figure 2: is a block diagram of a second embodiment of the invention, which uses a cellular transmission network to distribute information;

Figure 3: is a block diagram of a third embodiment of the invention, which uses a satellite transmission network to distribute information;

Figure 4: is a diagram of a fourth embodiment of the invention where the service provider manages the information into various channels before forwarding it to a transmission network to be broadcast; and

Figure 5: is a block diagram illustrating a system of switches for implementing the selective distribution of information to subscribers, according to the invention.

BEST MODES FOR CARRYING OUT THE INVENTION

Figure 1 shows an embodiment of a system 10 according to the invention, which uses a radio frequency transmitter 12 for distributing information to user bases 14, from one or more content providers16, which could be an Internet service provider, university or commercial institution such as a firm of stock brokers, magazine company, news network or software developer. Users subscribe or register themselves with a content provider 16 at a fee. The content providers prepare the information to distributed to their subscribers and forward it to a service provider 18 that manages and classifies the data to be transmitted. Existing transmission infrastructure 12, such as that used by radio companies, is used to broadcast the channels of information. At the user base 14, a PC-based receiving station, comprising an antenna 20, a specialised receiver card in the form of a radio card (not shown) and processing and storage means (not chown), is used to receive, demodulate, process and store the incoming information signal. Downloads are stored and automatically refreshed with up-to-date information. This information is retrieved by the user, and manipulated with appropriate software, such as conventional Internet browsers, customized software packages or applets.

Figure 2 relates, specifically, to the transmission of information to multiple users via a cellular network including one or more service providers 18, which are connected to a cellular network operator 30, and one or more content providers 16 . The cellular network operator 30 has multiple transmission areas serviced by base stations 32. At the user base 14, a remote terminal, such as a PC, is equipped with receiver means for receiving the cellular transmission. As in the case of RF broadcasting, the downloaded information is viewed on display means and manipulated with peripheral devices such as a keyboard and/or mouse.

Figure 3 relates, specifically, to the transmission of information to multiple users via a satellite network together with one or more service providers 18, which are connected to the satellite network operator 40, and one or more content providers 16. The satellite network is, in this example, the transmission medium for transmitting the information to a low earth-orbiting satellite, which relays the transmission to multiple users. At the user base, a satellite dish 42 is connected to a PC for display. In another example, it is envisaged that users could download the broadcast with a satellite dish connected to a set-top box.

Figure 4 shows an example of where a service provider 18 manages the content 50 to be broadcast over a wireless communication means 52 for reception by subscribers at their user bases 14.

Figure 5 is a block diagram illustrating the use of one or more switches as part of the managing device. Included in the information management system of the service provider is an electronic switching system 60 that ensures that only paying subscribers have decoding means to subscription channels. The encoding or encryption means is a function of the user access code inherent in the radio card, which enables the selective receiving of information by multiple users, i.e. only paying users are able to decode subscription channels.

While not being part of the receiver, clearly software in the PC is in overall control of the receiver unit. This software provides for various functions including issuing commands to tune the receiver, capture incoming data, decompress the information and decode or decrypt the data based on decryption keys provided to each user on payment of their subscription. Such decryption allows, for example, certain channels to be decoded by the intended recipient and not by other users of the system.

Likewise, prior to transmission by the service provider, suitable encryption and/or compression of data is required as well as directing the data to specific addresses or to general receivers.

It will be appreciated that certain embodiments of the invention have been described herein and that other embodiments, variations or modifications should therefore be understood to fall within the spirit and scope of the invention as claimed hereafter.

CLAIMS

1. A system for the simultaneous transmission of information to multiple users over a wireless communications network and for receiving, demodulating, downloading and storing the information at user bases, the system comprising at least one content provider; at least one service provider; a transmission infrastructure; multiple user bases, having receivers consisting of an antenna in conjunction with a receiver card; a modem for demodulating the broadcast signal; and processing means for storing and enabling subsequent retrieval of the information.
2. A system according to claim 1 including at least one switchable channel to be broadcast selectively to a subset of users and permitting the activation and or deactivation of a specific channel of information.
3. A system according to claim 1 or 2 including means for encoding the information signal prior to transmission.
4. A system according to claim 1 or 2 including means for encrypting the information signal prior to transmission.
5. A system according to any one of the preceding claims wherein the means for encrypting is a function of the user-specific identification code inherent in the receiver card and a key obtained by the user on payment of the channel subscription.
6. A system according to any one of the previous claims wherein the receiver has an antenna operatively associated therewith.
7. A system according to any one of the preceding claims including means for compressing the information signal prior to transmission and means for decompressing the information after it has been downloaded.
8. A system according to any one of the preceding claims where the transmission network is a radio network.
9. A method for facilitating the simultaneous transmission of information to multiple user bases over a wireless communications network and for receiving, demodulating, downloading, and storing the information at the user bases for subsequent retrieval, the method including the steps of collecting information from at least one content provider; classifying and grouping the information into channels; generating a modulated information signal for transmission; broadcasting the modulated

information signal over a wireless transmission network; receiving the transmitted information signal at user bases via suitably tuned receivers; demodulating the received information signal; and storing the information for subsequent retrieval.

10. A method according to claim 9 including the step of automatically refreshing the stored information with an updated version.

11. A method as claimed in claim 9 or 10 including the step of activating certain channels according to a subscriber's status using software switches at the transmitter.

12. A method as claimed in claim 9 or 10 including the step of activating certain channels according to a subscriber's status by encrypting information as a function of a user-specific identification code.

13. A method as claimed in any one of claims 9 to 12 wherein the step of modulating the information signal is achieved by using any one or more of modulation techniques selected from the group consisting of Gaussian Minimum Shift Keying (GMSK), Quadrature Polyphase Modulation (QPM) and Galaxy Modulation.

14. A method as claimed in claim 13 where the modulation technique includes a redundancy check.

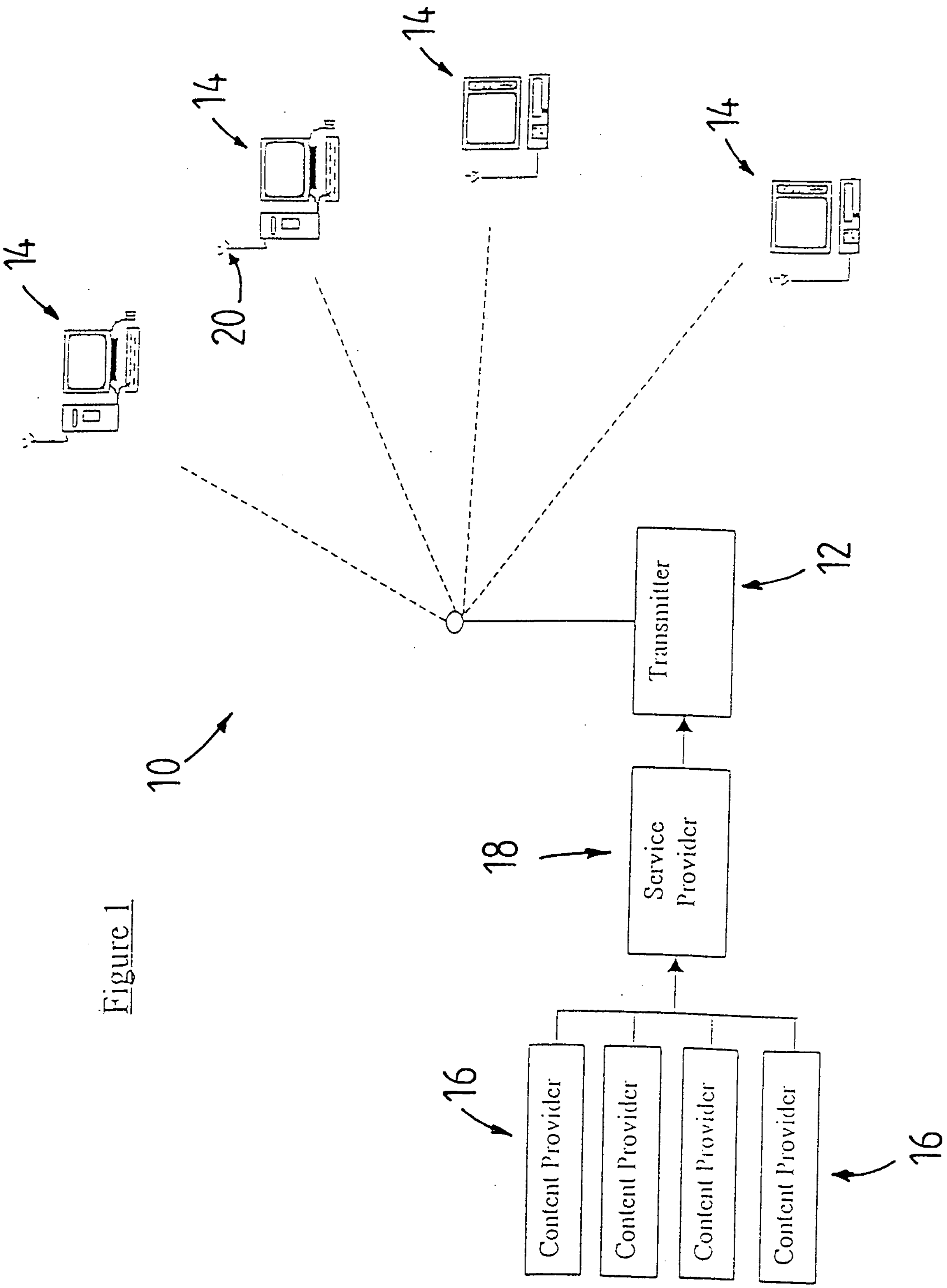

14
14
14
14
14
20
10
Transmitter
12
18
Service Provider
16
Content Provider
Content Provider
Content Provider
Content Provider
16

Figure 1

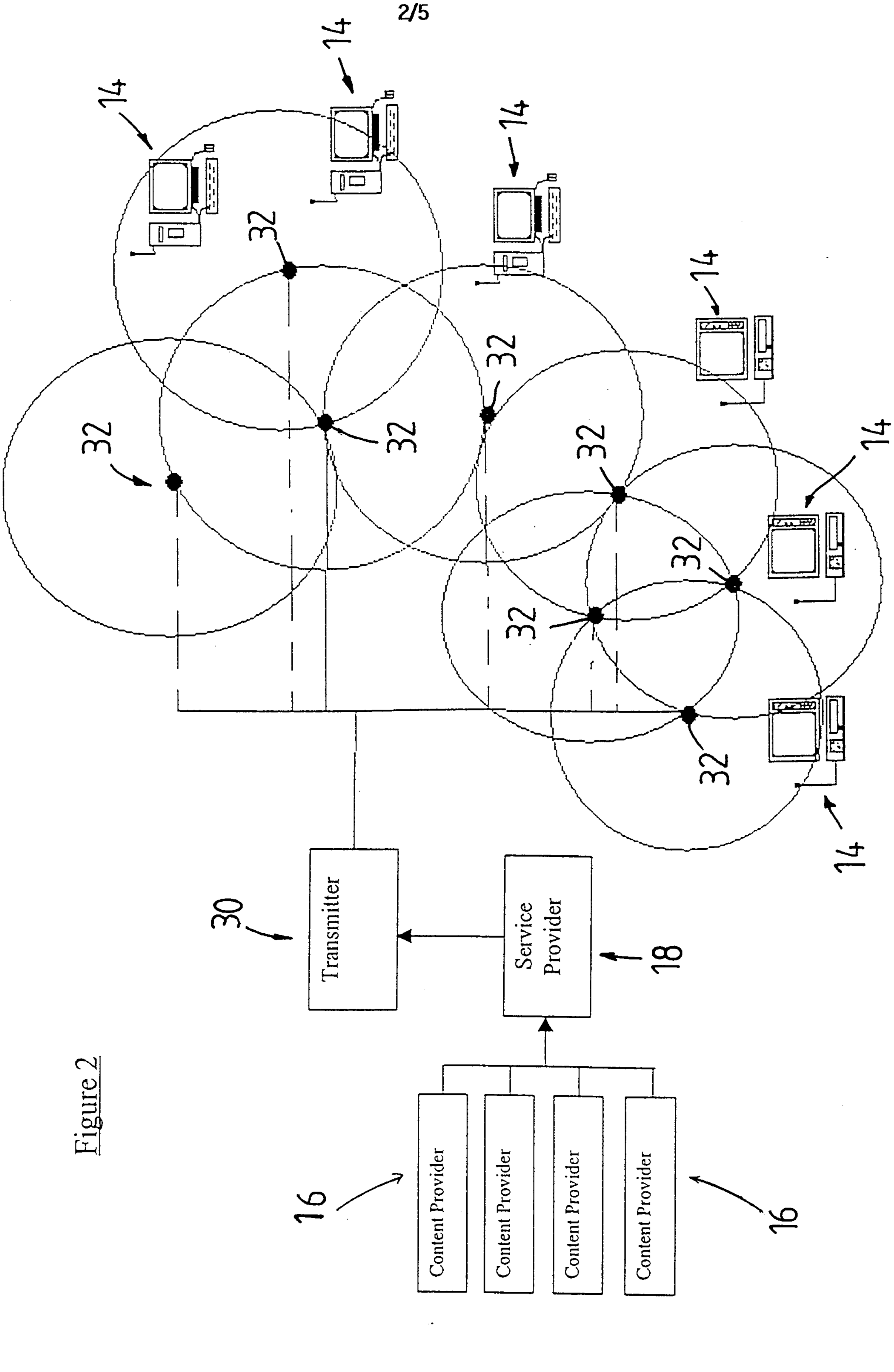

14
14
14
14
14
14
14
32
32
32
32
32
32
32
32
32
Transmitter
30
Service Provider
18
Content Provider
Content Provider
Content Provider
Content Provider
16
16

Figure 2

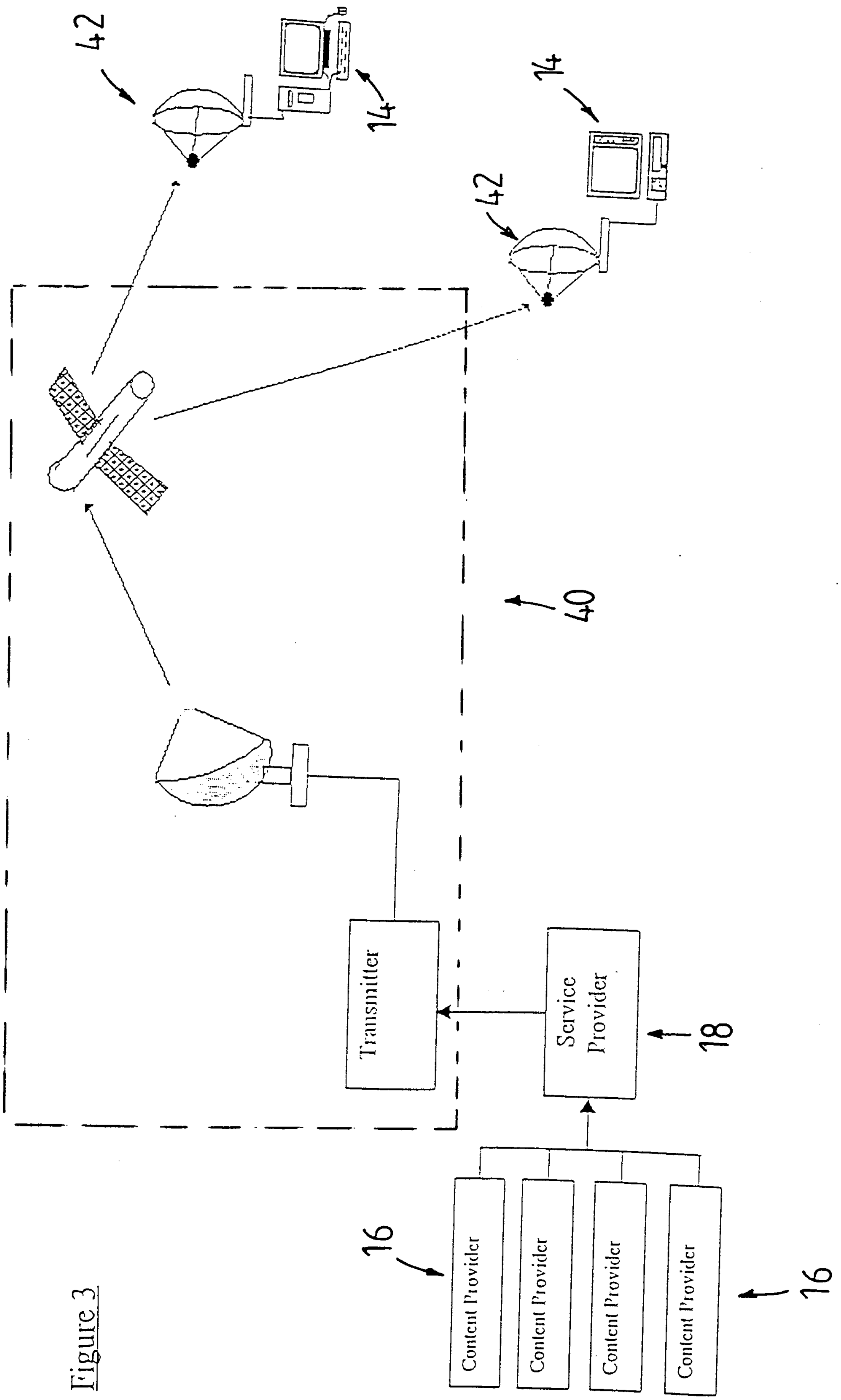

42
14
14
42
40
Transmitter
Service Provider
18
Content Provider
Content Provider
Content Provider
Content Provider
16
16

Figure 3

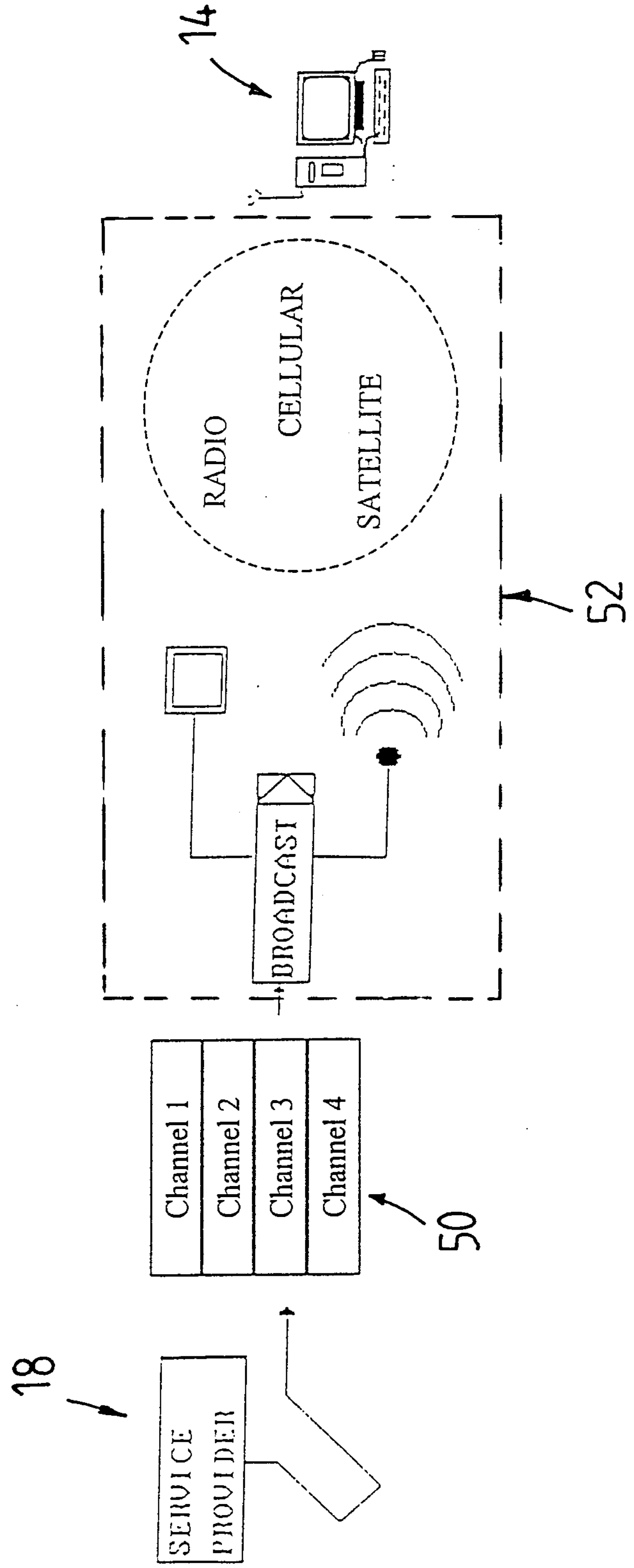
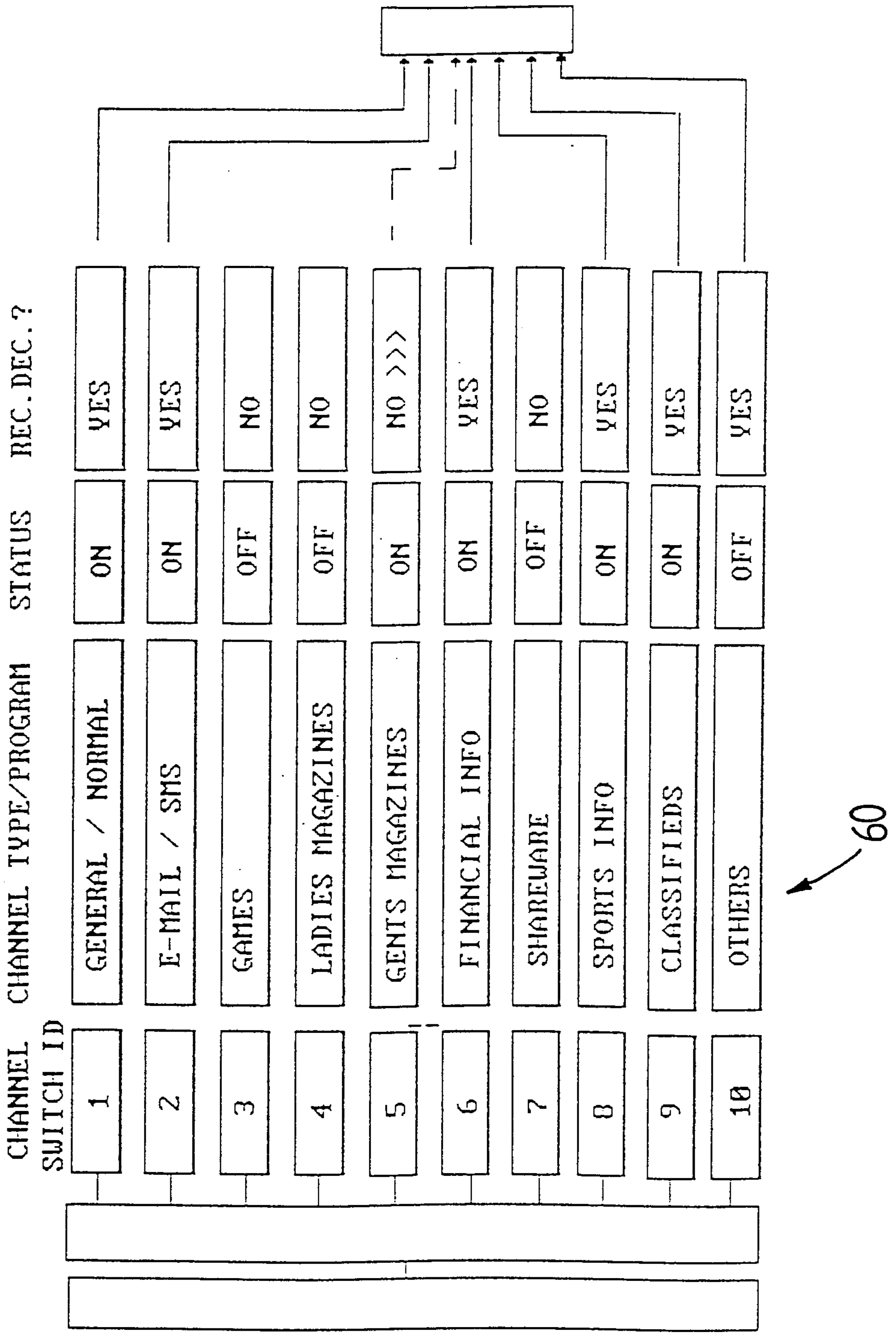

14
RADIO
CELLULAR
SATELLITE
52
BROADCAST
Channel 1
Channel 2
Channel 3
Channel 4
50
18
SERVICE
PROVIDER

Figure 4



CHANNEL SWITCH ID	CHANNEL TYPE/PROGRAM	STATUS	REC.DEC.?
1	GENERAL / NORMAL	ON	YES
2	E-MAIL / SMS	ON	YES
3	GAMES	OFF	NO
4	LADIES MAGAZINES	OFF	NO
5	GENTS MAGAZINES	ON	NO >>
6	FINANCIAL INFO	ON	YES
7	SHAREWARE	OFF	NO
8	SPORTS INFO	ON	YES
9	CLASSIFIEDS	ON	YES
10	OTHERS	OFF	YES

Figure 5

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
H04L

A2

(11) International Publication Number: WO 99/62210

(43) International Publication Date: 2 December 1999 (02.12.99)

(21) International Application Number: PCT/US99/11528

(22) International Filing Date: 25 May 1999 (25.05.99)

(30) Priority Data:
09/085,075 26 May 1998 (26.05.98) US

(71) Applicant: SUN MICROSYSTEMS, INC. [US/US]; 901 San Antonio Road, Palo Alto, CA 94303 (US).

(72) Inventors: DIGIORGIO, Rinaldo; 157 Christian Avenue, Stony Brook, NY 11790 (US). BENDER, Michael, S.; 155 Sunbeam Avenue, Boulder Creek, CA 95006 (US).

(74) Agents: CANNING, Kevin, J. et al.; Lahive & Cockfield, LLP, 28 State Street, Boston, MA 02109 (US).

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

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

(54) Title: SECURE TOKEN DEVICE ACCESS TO SERVICES PROVIDED BY AN INTERNET SERVICE PROVIDER (ISP)

(57) Abstract

A secure token device, such as a smart card or an ibutton, provides a user with a vehicle for accessing services that are provided by an Internet Service Provider (ISP). The user places the secure token device in communication with a reader that is coupled to a computer system. The computer system includes a web browser for accessing the services provided by the ISP. The secure token device may perform an authentication protocol to authenticate itself to the ISP. The ISP may also be required to authenticate itself. The secure token device may hold an electronic currency token for payment of services rendered by the ISP. The secure token device may contain stored personal information about the user. The user may stipulate what portions of this personal information are provided to the ISP upon request. Contextual information regarding sessions with the ISP may also be stored on the secure token device and used to restore a context of a previous session during a subsequent session.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

SECURE TOKEN DEVICE ACCESS TO SERVICES PROVIDED BY AN INTERNET SERVICE PROVIDER (ISP)

Technical Field of the Invention

The present invention relates generally to data processing systems and more particularly to secure token device access to services provided by an Internet Service Provider (ISP).

Background of the Invention

An ISP is a vendor who provides customers with access to the Internet. Examples of ISPs include America Online (AOL), CompuServe and the Microsoft Network (MSN). In addition to providing access to the Internet, ISPs may also provide additional services to their customers, including chat rooms, news services, electronic mail messaging and bulletin board services.

ISPs provide access to the Internet to customers by employing one or more Internet servers. These servers are directly connected to the Internet and act as conduits for customers to access web pages resident on other servers on the Internet. Typically, a customer uses a conventional modem to place a call to a designated ISP server. The modem need not be a conventional modem but may be instead, a cable modem or a wireless modem. The ISP server answers the call and a connection is established between the server and the customer's computer. After this connection is established, the customer is prompted to login. In particular, the customer is prompted usually to enter a user ID and a password. The information entered by the customer is compared to data stored in a database with the ISP to determine whether the user is who the user purports to be. If the customer provides the proper information and has sufficient privileges, the customer is granted access to the Internet.

There are a number of drawbacks associated with the above-described conventional approach to providing Internet access to customers. First, the Internet Protocol (IP) is used for messaging addressing on the Internet and the protocol is a connectionless protocol. As such, the protocol does not support the persistent storage of contextual information. Thus, any contextual information associated with one customer session on the Internet is not carried forward to the next customer session. Each session must start anew in creating a context. Second, the conventional approach to providing access to the Internet by ISPs is susceptible to fraud. If a party can obtain a user ID and password for a user, the party can gain access to the Internet via the user's account. Third, most ISPs currently provide only one variety of service such that all customers are offered this single variety of service. For example, all customers may be offered full access to a complete range of services provided by an ISP and all customers may be charged a flat fee for a designated time frame of service (e.g. for a month of service or a year of service). Customers who use the services more frequently than other customers are not charged additional amounts. Hence, there is a lack of flexibility in the pricing and service options available to customers from ISPs in conventional systems.

Summary of the Invention

The present invention addresses the limitations of the prior art by providing users with secure token device access to services offered by ISPs. "Secure token devices" are devices such as smart cards and ibuttons that hold currency tokens and other information in a secure fashion. Preferably, the secure token device is of a size, shape and weight that it is easily carried by a user. The secure token device may even be wearable by a user. When a user wishes to access services provided by an ISP, the user puts a secure token device in communication with a reader. The reader is a device that is configured to read and communicate with the secure token device. The reader is coupled to a computer system, such as a personal digital assistant (PDA), workstation or a personal computer (PC). When the user places the secure token device in or against the reader (depending on the type of reader), the reader recognizes the insertion of the secure token device and prompts the computer system to begin communicating with the secure token device. The computer system may seek to verify that the user is the proper owner of the secure token device. To that end, the computer system may request that the user enter a personal identification number (PIN). The user enters a PIN and the PIN is compared with a PIN value that is stored on the secure token device. If the PIN value entered by the user matches the PIN value on the secure token device, the computer system verifies that the user is the owner of the secure token device and the process of accessing the ISP services may be initiated.

The secure token device may hold identification information that is globally unique across geographic and political boundaries. This identification information is held securely on the secure token device. It is difficult for a party to physically access the identification information. The secure token device serves as a physical token of authenticity for the party. In order to fraudulently use the secure token device, a party must both physically take the secure token device and also be aware of the PIN associated with the user of the secure token device. Hence, the use of the secure token device helps to decrease the probability of fraud.

Contextual information (i.e., a context) may be stored on the secure token device of the user. The context may, for example, identify user preferences and configuration information. When a user seeks to access the services of the ISP, the context from a previous session may be restored by retrieving the context from the secure token device. This ability to preserve context enhances the services provided to the user and eliminates the need for the user to recreate a context each time the user accesses ISP services.

The secure token device may also support various electronic banking or electronic commerce mechanisms that facilitate the exchange of electronic currency. The secure token device may be used in realizing payment for services provided by ISPs. The user may download currency tokens from the secure token device to the ISP to cover expenses associated with the services provided during a given session. This ability to receive payment for services during a session with the user enhances the ability of ISPs to tailor pricing schemes on a per use basis. An ISP may charge a user for the services rendered during a given session as opposed to using a flat rate scheme over an extended time period, such as a month or a year. Thus, users are charged on the basis of the resources they consume rather than on a flat rate basis.

The secure token device of a user may contain personal information regarding a user, such as name, address, and credit card account information. The user has the ability to customize what portions of this personal information may be accessed by a service provider. Hence, the user may determine that an ISP should only be given access to the user's name and address and should not given access to the user's credit card account information. For another service provider, the user may grant the service provider full access to all of the personal information. This approach has the added benefit of storing personal information more securely than instances where the personal information is stored on database maintained by an ISP. It should be noted, however, that ISPs may store additional information on secure token devices that is not readily accessible to users. A further benefit of this approach is that it gives the user control over what personal information the user grants to respective parties. Still, further, the storage of personal information on the secure token device facilitates companies to develop loyalty marketing programs, such as frequent flier programs. The frequent flier miles of a user may be stored on the secure token device, added to the storage on the secure token device and redeemed from the secure token device.

Brief Description of the Drawings

An illustrative embodiment consistent with the principles of the present invention will be described below relative to the following drawings.

FIGURE 1 is a block diagram that illustrates hardware components used to practice the illustrative embodiment of the present invention.

FIGURES 2A and 2B illustrate the exemplary layout for a smart card to be used in the illustrative embodiment of the present invention.

FIGURE 2C illustrates the contacts on the smart card of FIGURE 2A in more detail.

FIGURE 3 illustrates an example of an ibutton ring to be used in the illustrative embodiment of the present invention.

FIGURE 4 is a block diagram illustrating computing components on the secure token device.

FIGURE 5 is a block diagram illustrating components of the computer system of FIGURE 1 in more detail.

FIGURE 6 illustrates the various Java packages that are found on the secure token device.

FIGURE 7A illustrates object classes that are supported by the computer system of FIGURE 1.

FIGURE 7B illustrates object-classes that are part of the CardTerminal component.

FIGURE 7C illustrates object-classes that are part of the CardAgent component.

FIGURE 7D illustrates object-classes that are part of the CardI0 component.

FIGURE 8A illustrates the logical format of a command APDU.

FIGURE 8B illustrates the logical format of a response APDU.

FIGURE 9 is a flow chart that illustrates the steps that are performed when a user logs in via a secure token device.

FIGURE 10 is a flow chart illustrating the steps that are performed when a user desires to access services provided by an ISP.

FIGURE 11 is a flow chart illustrating the steps that are performed when an ISP seeks context information from a user.

FIGURE 12 illustrates the logical organization of a user profile.

FIGURE 13 is a flow chart illustrating the steps that are performed to restore a context in the illustrative embodiment of the present invention.

FIGURE 14 is a flow chart illustrating the steps that are performed in billing a customer for services rendered by an ISP.

Detailed Description of the Invention

In the illustrative embodiment consistent with the present invention, a user gains access to services provided by an ISP by employing a secure token device, such as a smart card or an ibutton (such as produced by Dallas Semiconductor Corporation). The secure token device is a secure electronic device that holds globally unique identification information regarding the user. The user may be required to enter a password or PIN to verify that the user is the same party whose identification information is stored on the secure token device. The secure token device is programmed to support two-way verification between the user and the ISP. Specifically, the user must prove that the user is who the user purports to be, and the ISP must prove that the service is what it purports to be.

The secure token device may hold contextual information on behalf of the user. The contextual information may capture the context of a previous session with the ISP. When the user again gains access to the services of the ISP, the context from the previous session may be restored. For example, user preferences and other contextual information that were entered in a previous session may be carried forward into the new session.

The secure token device may run multiple programs. The programs may include code for facilitating access to the services of an ISP and code for electronic commerce transactions. These transactions may entail the exchange of electronic currency in the form of tokens. Thus, when the user accesses a web site or other service that requires payment for the tendering of goods or services, the user can pay for the goods or services using the tokens contained services based on the secure token devices. It should be appreciated that the ISPs may serve the role of distributor for distributing the secure token devices to customer.

The secure token device may hold information regarding the user that is potentially sensitive. The user has control over dissemination of this information. The user selects what portions of this information are available to respective requesters. Different requesters may be granted different permissions. For example, a first requester may receive a first set of personal information and a second requester may receive a second set of personal information that differs from the first set.

The use of the secure token device enables ISPs to tailor their service offerings and billing options to individual users. The users may be offered different service options. For example, a first user may be offered a service option where the user is only permitted to browse the Internet. A second user, in contrast, is offered the ability to browse the Internet and to send emails, visit chat rooms and visit news sites. The second user may be charged additional amounts for the expanded service. Other types of expanded service may include secure email and authenticated connections with other users.

Figure 1 is a block diagram that illustrates several of the hardware components employed in the illustrative embodiment consistent with the present invention. These components include a secure token device 10 that is provided for a user. The secure token device 10 may be any secure device that is capable of holding electronic currency tokens, identification information and context information. Preferably, the secure token device is of an appropriate size, weight and shape to be portable and easily carried by a user. Suitable secure token devices include smart cards and ibuttons. A secure token device is an integrated circuit card that preferably is sized to fit into a user's wallet or purse. Ideally, a smart card is the size of a credit card. The smart card has computer components such as a microprocessor and a storage embedded in it. A smart card that may be used to practice the present invention may comply with the ISO-7816 standard or the EMV integrated circuit card specification. For purposes of the discussion below, it is assumed that if a smart card is used as the secure token device, the smart card complies with the JavaCard 2.1 specification as defined by Sun Microsystems, Inc. The JavaCard 2.1 specification requires that the secure token device be capable of running programs written in the Java™ programming language. Java is a trademark of Sun

Microsystems, Inc. Those skilled in the art will appreciate that the programs used to practice the present invention may be written in programming language other than Java™, including C, C++ and Basic.

An ibutton is a computer chip that is housed in a cylindrical housing (such as a steel canister). The housing is designed to withstand the harsh conditions of outdoor environments. The ibutton may be incorporated into a ring or other wearable item. For instance, ibuttons may be affixed to badges, watches, rings key chains and the like. The chip within the housing includes a microprocessor and may also contain computer memory, a clock or sensors. Such ibuttons are used by contacting the ibuttons with readers (e.g. "blue dot receptors") that are cabled into the serial ports of associated computers. A suitable ibutton for practicing the illustrative embodiment consistent with the present invention is the Java™ Ring produced by Dallas Semiconductor Corporation.

The hardware components used in the illustrative embodiment consistent with the present invention also include a reader 12. The reader 12 is a device for facilitating communications between a computer system 14 and the secure token device 10. The reader 12 provides a path for application programs run on computer system 14 to communicate with the secure token device 10. Preferably, when the secure token device is a smart card, the reader 12 is compliant with the OpenCard standard. The OpenCard standard is a standard that provides for inter-operability of secure token device applications across devices, such as network computers, laptop computers, desktop boxes, desktop computers, cellular phones and personal digital assistants (PDAs). A number of different commercially available card terminals may be utilized as the reader 12 when the secure token device is a smart card. A suitable reader is the IBM 594A card terminal. When the secure token device 10 is an ibutton, a suitable reader is the DS1402 blue dot receptor from Dallas Semiconductor Corporation. The reader may also be a proximity detector.

The computer system 14 may be a PDA, a personal computer (PC) or a workstation. The configuration of the computer system 14 will be described in more detail below. The computer system 14 may communicate with a remote server computer system 16 via a communications link 15. The communications link 15 may be, for example, a telephone line connection. More generally, the communication link 15 may be a wireless connection, a cable modem connection, a satellite connection or a direct connection. The remote server 16 is controlled by the ISP and provides the user with access to the Internet.

Figures 2A and 2B illustrate an exemplary physical layout for a smart card to be used as the secure token device 10. The secure token device 10 is formed on a plastic substrate 20. The front of the card (as shown in Figure 2A) includes a number of electrical contacts 16 which facilitate communications with the smart card. Figure 2C shows these contacts 16 in more detail. Contact 24 is used to connect with the power source that is provided by the smart card reader. Contact 26 is to be coupled to a ground connection on the smart card reader. Contact 28 is used for input/output of data packets (described below). Contact 30 is used to reset the smart card, and contact 32 is used for a check procedure performed on the smart card to ensure that the smart card is operating properly. Optional contacts 34, 36 and 38 are also provided. The front of the smart card may also include an embossing area 18 where the user may sign the smart card. The back of the smart card (as shown in Figure 2B) may include a magnetic strip 22 for holding information that is magnetically encoded. In some applications, the smart card may be used as an ID badge that permits a user access to certain locales. The magnetic strip may hold information that permits the user to gain access to a secure area or other locales, for example.

Those skilled in the art will appreciate that the physical layout of the smart card shown in Figures 2A-2C is intended to be merely illustrative and not limiting of the present invention. The secure token device used to practice the present invention may have a different physical configuration with additional components or fewer components than shown in Figures 2A-2C.

Figure 3 depicts an example of the physical layout of a Java Ring 35 that is suitable for practicing the present invention. The Java™ Ring 35 includes a steel cylindrical housing 37 that houses an integrated circuit (IC) 41 that contains a microprocessor and a storage (i.e. a computer memory). The Java™ Ring 35 also includes a ring portion 39 that enable a user to wear the whole device like an ordinary ring. As will be described in more detail below, the processor and storage work in conjunction to runs programs that help facilitate the illustrative embodiment of the present invention.

Figure 4 shows a block diagram of the computer architecture of the secure token device 10. The computer architecture includes a microprocessor 40 and a storage 42. The storage 42 may be formed by different types of devices, including random access memory (RAM), read only memory (ROM), and electrically erasable programmable read only memory (EEPROM) devices. Those skilled in the art will appreciate that the storage 42 may also include other types of storage devices. The storage 42 holds a number of types of data and programs that may execute on the microprocessor 40. In the illustrative embodiment of the present invention, it is assumed that the processor 40 on the secure token device 10 is capable of running programs written in the Java™ programming language. An "applet" is a special type of program that runs inside an applet viewer, a web browser or a secure token device. The storage 42 holds a copy of an ISP applet 44. The ISP applet 44 enables the secure token device 10 to communicate with an ISP and to receive services from an ISP. Those skilled in the art will appreciate that the secure token device may instead run programs in programming languages other than Java™.

The storage 42 also holds a copy of a banking applet 46 that allows the secure token device 10 to be utilized in electronic commerce transactions. As will be described in more detail below, in the illustrative embodiment, the banking applet 46 allows the secure token device to be used with a MONDEX system or other type of electronic commerce system. The secure token device 10 may hold tokens representing units of electronic currency that may be used to pay for goods and services. The banking applet provides the intelligence for participating in such transactions. The storage 42 may also hold other applets 41.

The storage 42 holds a copy of a user profile 48. The user profile contains personal information regarding a user. Preferably, as will be described in more detail below, the user profile 48 complies with the Open Profiling Standard (OPS) and/or the Information & Content Exchange (ICE) protocol.

The storage 42 additionally holds the JavaCard API as defined in the JavaCard 2.1 specification. In instances where the secure token device is not a smart card, other similar API sets may be alternatively used. The JavaCard API is an application program interface that provides a broad range of functionality for the secure token device 10. The major components of the JavaCard API 50 will be described in more detail below. The applets stored on the secure token device 10 may instantiate object classes defined in the API to realize desired functionality. The storage 42 holds a copy of a JavaCard virtual machine (VM) 52. The JavaCard virtual machine is like a conventional Java virtual machine but is streamlined to operate with the memory and processing restrictions that are found with secure token device 10. The JavaCard VM provides platform independence for the Java programs that are run on the processor 40.

Those skilled in the art will appreciate that the secure token device 10 may hold additional programs and data that differ from that shown in Figure 4.

Figure 5 is a block diagram that shows the components of the computer system 14 in more detail. Computer system 14 includes a central processor unit (CPU) 54 for executing instructions. A number of peripheral devices, including a keyboard 56, a video display 58, and a mouse 60, may be provided as part of the computer system 14. A modem 62 may be provided to allow the computer system to communicate over analog telephone lines, and a network adapter 64 may be provided to facilitate the connection of the computer system 14 to a local area network (LAN). As has been discussed above, the computer system 14 may also include other components, such as a cable modem, for facilitating remote communications with the remote server 16.

The computer system 14 includes both primary storage 68 and secondary storage 66. The secondary storage 66 may include a number of types of persistent storage. For example, the secondary storage 66 may include CD-ROM drives, hard disk drives and other types of computer-readable mediums. The primary storage 68, likewise, may include a number of different types of storage, including DRAM, SRAM, and the like. The primary storage 68 holds a copy of an operating system 70. The Solaris® operating system is suitable for practicing the illustrative embodiment of the present invention. "Solaris" is a registered trademark of Sun Microsystems, Inc. A web browser 72 is provided in primary storage 68 to facilitate access to the Internet. Suitable web browsers include Netscape Navigator, Netscape Communicator and Microsoft Internet Explorer. It should be appreciated the web browser 72 may include intelligence for processing hypertext mark-up language (HTML) documents. A Java™ VM 74 is provided in primary storage 68 for interpreting Java programs. The OpenCard API 76 is also found within the primary storage 68. Additional applications 78, including Java applets, may also be stored in the primary storage 68. These applications may instantiate objects of the object classes defined in the OpenCard API to realize needed functionality.

Those skilled in the art will appreciate that various ones of the components depicted in Figure 5 as being stored in the primary storage 68 may alternatively be stored in the secondary storage 66. Those skilled in the art will also appreciate that the computer system 14 shown in Figure 5 is intended to be merely illustrative and not limiting of the present invention. Further, it should be appreciated that the reader 12 shown in Figure 1 may be integrated as part of the computer system 14.

The Java™ programming language is object-oriented. It generally supports the arrangement of sets of object classes into packages. The JavaCard API 50 is divided into a number of packages 78, as shown in Figure 6. The java.lang package 80 contains a number of object classes that are concerned with exceptions, such as run time exceptions and security exceptions. The javacard.framework package 82 contains object classes for APDUs (defined below), applets, PINs and various system constants. The javacardx.framework package 84 contains object classes relating to file system structures. The jacacardx.crypto package 86 holds objects that provide cryptography support on the secure token device 10. The javacardx.cryptoEnc package 88 contains object classes relating to the DES encryption scheme.

Programmatic support for use of the secure token device 10 is provided on the computer system 14. The OpenCard API 76 provides a number of interfaces that facilitate communications with the secure token device. Figure 7A depicts the major components of the OpenCard API 76. The CardTerminal component 90 abstracts the readers (also known as card terminals) that help to interface the secure token device 10 with the computer system 14. Each reader (see Figure 7B) is represented by an instance of the CardTerminal object class 85. A CardTerminal Factory 83 object class is defined to instantiate instances of the CardTerminal object class. The CardTerminalRegistry object class 81 (Figure 7B) is defined as part of the CardTerminal component 90. Only a single instance of this object class exists and this instance serves as the system-wide registry. Register() and unregister() methods are provided for this object class to dynamically add or remove card terminals from the registry. A slot object class is defined for each slot in a reader. Each instance 87A and 87B of this object class represents a physical card slot in a card terminal. A CardID object class 89 is defined in the CardTerminal component 90 to represent a secure token device.

The CardAgent component 92 abstracts an agent that operates on behalf of the secure token device 10. A CardAgent object class 91 (See Figure 7C) is defined in this package to abstract the functionality of the secure token device. Each agent has a separate instance of the object class. Communications between the secure token device 10 and the computer system 14 pass through the CardAgent. A CardAgent Factory object class 93 support instantiation of CardAgent objects 91 and a CardAgent Factory Registry object class 95 may be instantiated to hold a registry of all agents.

The CardIO component 94 contains object classes that are used to support input/output relative to the secure token device 10. All application interaction with the secure token device 10 takes place through objects of the object classes defined in this component 94. A SmartCard object class 97 (See Figure 7D) is defined to represent a physical secure token device. Access to the file system on the secure token device 10 is achieved by mounting a root master file, resulting in an instance of the CardFile object class 99A, which is defined as part of the CardIO component 94. An application can access other files on the secure token device 10 by instantiating appropriate CardFile objects 99B and 99C. Figure 7D show an example where three card file objects are instantiated. The CardRandomAccessFile object class 103 defines objects that allow programs to access contents of the associated files.

The secure token device 10 and the computer system 14 communicate by passing data packages back and forth. These data packages are known as application protocol data units (APDUs). The format for APDUs is defined in the ISO-7816 standard. Each APDU contains either a command or a response to a command. A master-slave model may be followed where the secure token device 10 plays the slave role and the computer system 14 plays the master role. The secure token device 10 always waits for a command APDU from the computer system 14 by way of the reader 12. The secure token device 10 then executes the command specified in the command APDU and replies to the terminal with a response APDU. A client/server model may also be followed wherein the computer system 14 serves as a security server and the secure token device 10 serves as a client.

Figure 8A depicts the logical format of a command APDU 100. The mandatory header 102 encodes the command that is to be encapsulated in the APDU. The header 102 includes four fields: the CLA field 106, the INS field 108, the P1 field 110, and the P2 field 112. The CLA field 106 is a class byte that identifies an object class, such as an application program. The INS field 108 is an instruction byte that identifies the instruction (i.e. the command). The P1 field 110 and the P2 field 112 are parameter bytes that provide further qualification of the APDU command. These fields 110 and 112 are used to pass parameters with the command.

The command APDU 100 also contains a conditional body 104. The conditional body 104 contains three fields: the Lc field 114, the data field 116, and the Le field 118. The Lc field 114 holds a value that identifies the number of bytes in the data field 116. The data field 116 is used to hold data, and the Le field 118 identifies a maximum number of bytes that are expected in the datafield in the response APDU that is to be received after the command APDU 100 is processed.

Figure 8B shows logical format of a response APDU 101. The response APDU may contain a conditional body 120 and a mandatory trailer 122. The conditional body 120 includes a data field 124 for holding data. The mandatory trailer 122 contains an SW1 field 126 and an SW2 field 128. These two fields each hold a respective status byte that reflects the status of the command for which the response is sent.

Figure 9 is a flow chart that illustrates the steps that are performed during initial login when a user using secure token device 10 attempts to gain access to computer system 14. The role played by the secure token device 10 during login may be encoded in one of the applets 41 stored in the storage. Initially, the user places the secure token device 10 in position for reading by reader 12 (step 130 in Figure 9). The reader 12 detects the presence of the secure token device 10 and then informs the computer system 14 (step 132 in Figure 9). A number of different login options may be followed but, in general, the computer system 14 begins the login process by sending appropriate command APDUs via the reader 12 to the secure token device 10. The commands prompt the user to enter a PIN value (step 134 in Figure 9). The PIN may be, for example, a code constituting between 4 to 8 digits that is uniquely assigned to the user. The reader 12 may include a keypad that is used to enter the PIN or, alternatively, the user may enter the PIN via the keyboard 56 that is part of the computer system 14 (step 136 in Figure 9).

The PIN entered by the user is then compared with the PIN value assigned to the user (step 138 in Figure 9). In particular, the secure token device 10 holds the proper PIN value for the user within its storage 42 (Figure 4). The PIN value may be stored as part of the user profile 48. The JavaCard API 50 defines a PIN object class for holding a PIN value, and this object class includes methods for accessing the PIN. These methods are used to obtain the proper PIN and to compare the stored PIN with that entered by the user. The use of the PIN helps to ensure that the proper party and not an unauthorized party is utilizing the secure token device. If the correct PIN has been entered (see step 140 in Figure 9), the user is granted access to the computer system 14 (step 142 in Figure 9). If the correct PIN is not entered, the user may be given an additional opportunity to enter the proper PIN. The information stored on the secure token device identifies the maximum number of tries that may be attempted before user is denied access. Hence, in step 144 of Figure 9, a determination is made whether the maximum number of tries has been reached or not. If the maximum number of tries has been reached, the user is denied access (step 146 in Figure 9). Otherwise, the process is repeated again, beginning with step 134 in Figure 9 where the user is prompted to enter a PIN.

After login, the user may desire to access services provided by the ISP (step 148 in Figure 10). For example, the user may double click on an icon associated with the ISP or the system may automatically attempt to grant the user access to the ISP services once login is completed. A two-way challenge response authentication is then initiated. First, the ISP (i.e. remote server 16) issues a challenge to the secure token device 10 to ensure that the user should be granted access to the ISP services (step 150 in Figure 10). The secure token device 10 receives the challenge and responds (step 152 in Figure 10).

A proper response reveals knowledge of a shared secret (such as an encryption key). The ISP applet 44 contains the appropriate intelligence for responding to such a challenge. The challenge may be issued by one of the applications 78 stored in the primary storage 68 of the computer system 14. If the response is not proper (step 154 in Figure 10), the services provided by the ISP are not accessed (step 164 in Figure 10). If the response, however, is proper, the user is authenticated, and a challenge is issued by the secure token device 10 to the ISP (step 156 in Figure 10). The ISP responds to the challenge by submitting a response (step 158 in Figure 10). If the response is proper (see step 160 in Figure 10), the services provided by the ISP are accessed (step 162 in Figure 10). In contrast, if the response is not proper, the services are not accessed (step 164 in Figure 10). Those skilled in the art that multiple two way authentications may be performed.

It should be appreciated that each user has a globally unique ID that is encoded on the secure token device. The user ID is unique across geographic and political boundaries. This user ID may be used in formulating the challenge that is issued by the ISP. Each ISP also has a globally unique ID. The ISP ID may be used in the challenge-response protocol.

Those skilled in the art will appreciate that a number of different challenge/response protocols may be utilized in performing this two-way authentication. For example, SHA-1, XOR and other protocols may be used. Moreover, those skilled in the art will appreciate that the ISP may be first presented with the challenge rather than the secure token device.

Before the ISP begins providing services or sometime during session where the ISP is providing services, the ISP may seek personal information from the user profile 48 stored on the secure token device 10. The format of the user profile 48 will be described in more detail below. The ISP begins the process by requesting information from the profile 48 (step 166 in Figure 11). Permissions are defined for each requester that may request personal information of the secure token device 10. These permissions identify what portion or subset of profile data may be accessed by the requester. In response to the request from the ISP, the secure token device 10 accesses the permissions that are provided for the ISP (step 168 in Figure 11). The request identifies what information is sought from the secure token device. The secure token device determines whether the ISP has the permissions needed to receive the requested information (step 170 in Figure 11). If the ISP lacks the appropriate permissions, the ISP is denied access (step 172 in Figure 11). If the ISP has the appropriate permissions, the ISP is granted access to the information, and the secure token device 10 forwards the information to the ISP (step 174 in Figure 11). This information may be forwarded from the secure token device 10 to the computer system 14 in encrypted form for security purposes. It should be appreciated that the secure token device may partially grant the request where an ISP requests information that it is permitted to receive as well as information it is not permitted to receive.

Figure 12 shows a logical organization of an illustrative user profile 178. In the illustrative embodiment consistent with the present invention, the user profile may conform with the Open Profiling Standard (OPS). In accordance with that standard, the information contained in the user profile is divided into sections and subsections. In the exemplary case shown in Figure 12, the profile 178 is divided into a first section 180 and a second section 182. Suppose that the first section 180 contains address information and section 182 contains credit card information. The first section 180 also contains a subsection 188. This subsection 188 may contain, for example, a phone number. Each statement is a name/value pair. The first section includes statements 184 and 186 that assign given values to properties. The subsection 188 also contains a statement 190 that assigns a data value to a property. Permissions are granted on a section or subsection basis.

The information contained in the user profile 48 may vary. The user profile may contain information such as name, address, and credit card information. In general, the information is personal to the user.

As was discussed above, the secure token device 10 may be used as a vehicle for preserving contextual information. In particular, the context of a given session with an ISP may be preserved for later restoration in a subsequent session. The context may hold a wide variety of different information. For instance, user preferences regarding settings and various web sites may be restored in the context. Where the ISP begins a session with the user, the ISP requests contextual information from the secure token device 10 (step 192 in Figure 13). The secure token device then provides the context to computer system 14, which forwards the information to the ISP at the remote server 16 (step 194 in Figure 13). The contextual information is used to restore the previous context (step 196 in Figure 13). Subsequently, the ISP seeks to store the new context of the current session with the secure token device 10 so that the new context may be subsequently restored in the next session (step 198 in Figure 13). The new context is sent to the secure token device 10 and the secure token device stores the new context for subsequent use (step 200 in Figure 13).

The secure token device 10 may provide the ability for the user to pay for services rendered by the ISP during a session with the ISP. As was discussed above, this also assists the ISP in tailoring services to a particular user and in charging the user based upon resource utilization. Initially, the user seeks an ISP service, such as web browsing or electronic mail (step 202 in Figure 14). The ISP then levies a charge for user to access the servers (step 204 in Figure 14). The secure token device returns an electronic token representing amount of currency to the ISP (step 206 in Figure 14). As was mentioned above, the secure token device 10 includes a banking applet 46 that supports the ability to respond to requests and to deliver tokens. The banking applet 46 may support transactions involving MONDEX tokens or other types of electronic currency tokens. MONDEX is an electronic transaction system that employs smart cards for person-to-person payments. MONDEX was developed by National Westminster Bank in conjunction with Midland Bank and British Telecom and has been in use since July 1995. MONDEX uses tokens of a specified format.

The ISP receives the tokens and deposits the tokens in an appropriate account (step 208 in Figure 14). After receiving payment, the ISP then grants the user access to the server (step 210 in Figure 14). The Java electronic commerce framework (defined by Sun Microsystems, Inc.) is an open platform for development of electronic commerce applicators in Java. This framework may be used by the banking applet 46.

Those skilled in the art will appreciate that a number of different electronic transaction systems may be utilized in the present invention. The present invention is not limited to using MONDEX currency. Moreover, the billing scheme may differ from that shown in Figure 14. The timing at which a party is charged for services may differ such that a party is charged after having finished using a service rather than before accessing the service. Furthermore, there may be instances where an ISP is required to provide change in the form of tokens that are returned to the secure token device 10.

While the present invention has been described with reference to an illustrative embodiment thereof, those skilled in the art will appreciate the various changes in form and detail may be made without departing from the intended scope of the present invention as defined in the appended claims. For example, different varieties of secure token devices may be used to practice the present invention.

What is claimed:

1. In a computer system where a user accesses services provided by a service provider during sessions via a connectionless protocol, a method comprising the steps of:

providing a secure token device for a user, said secure token device holding contextual information that captures a context of a last session the user had with the service provider;

on behalf of the service provider, receiving the contextual information from the secure token device; and

using the contextual information to restore the context of the last session the user had with the service provider during a current session where services are provided by the service provider to the user.

2. The method of claim 1 wherein the connectionless protocol is the Internet Protocol (IP).

3. The method of claim 1 wherein the service provider is an Internet Service Provider (ISP) that provides the user with access to the Internet.

4. The method of claim 1 wherein the contextual information identifies a uniform resource location.

5. The method of claim 1 wherein the secure token device is a smart card.

6. The method of claim 1 wherein the secure token device is an ibutton.

7. In a secure token device, for use by a user in accessing services of a service provider via a connectionless protocol during sessions, a method comprising the steps of:

providing contextual information that captures a context of a last session that the user had with the service provider on the secure token device;

receiving a request to read the contextual information on behalf of the service provider; and

in response to the requests, outputting the contextual information for use by the service provider in restoring the context of the last session in a current session where services are provided by the service provider to the user.

8. The method of claim 7 wherein the connectionless protocol is the Internet Protocol (IP).

9. The method of claim 7 wherein the service provider is an Internet Service Provider (ISP) that provides the user with access to the Internet.

10. The method of claim 7 wherein the contextual information identifies a web site.

11. The method of claim 7 wherein the secure token device is a smart card.

12. The method of claim 7 wherein the secure token device is an ibutton.

13. In a secure token device that interfaces with a computer system, wherein a user accesses services provided by a service provider on the computer system, a method comprising the steps of:

providing personal information about the user in the storage of the secure token device;

establishing what portion of the personal information is permitted to be given to the service provider upon request;

receiving a request from the service provider at the secure token device to obtain at least some of the personal information about the user; and

in response to the request, sending to the service provider only information from the portion of the personal information that is permitted to be sent to the service provider.

14. The method of claim 13 wherein the information that is sent to the provider includes less than all of the information requested.

15. The method of claim 13 wherein the user establishes the portion of the personal information that is permitted to be given to the service provider upon request.

16. The method of claim 13 wherein when the service provider requests only information that is not permitted to be given to the service provider, the request is rejected by the secure token device.

17. The method of claim 9 wherein the service provider is an Internet Service Provider (ISP).

18. The method of claim 13 wherein the secure token device is a smart card.

19. The method of claim 13 wherein the secure token device is an ibutton.

20. In a computer system wherein a service provider provides services to a customer and wherein the customer uses a secure token device holding tokens representing currency to access the services, a method of comprising the steps of:

with the service provider, providing services to the customer during a session;

assessing a charge to the customer for the services that were provided during the session; and

receiving some of the tokens from the secure token device at the service provider, wherein the received tokens constitute payment for covering the charges to the customer from the secure token device.

21. The method of claim 20 wherein the service provider is an Internet Service Provider (ISP) that provides the customer with access to the Internet.

22. In a secure token device that interfaces with a computer system, wherein a user of the secure token device receives services from an Internet service provider (ISP) on the computer system, a method comprising the steps of:

providing tokens representing currency on the secure token device;

receiving a request for payment for services from the ISP; and

forwarding at least one token from the secure token device to the ISP in response to the request.

23. In a network having a computer system where a user accesses services provided by a service provider during sessions via a connectionless protocol and a secure token device for enabling the user to access the services provided by the service provider, wherein the secure token device holds contextual information that captures a context of a last session the user had with the service provider, a computer-readable medium holding computer-executable instructions for performing a method, comprising the steps of:

receiving the contextual information from the secure token device on behalf of the service provider;

using the contextual information to restore the context of the last session the user had with the service provider during a current session where services are provided by the service provider to the user.

24. The computer-readable medium of claim 23 wherein connectionless protocol is the Internet Protocol (IP).

25. The computer-readable medium of claim 23 wherein the service provider is an Internet Service Provider (ISP) that provides the user with access to the Internet.

26. The computer-readable medium of claim 23 wherein the contextual information identifies a web site.

27. In a system where a secure token device that interfaces with a computer system, wherein a user accesses services provided by a service provider on the computer system and personal information about the user is provided in the storage of the secure token device, a computer-readable medium holding computer-executable instructions for performing a method comprising the steps of:

establishing what portion of the personal information is permitted to be given to the service provider upon request;

receiving a request from the service provider at the secure token device to obtain at least some of the personal information about the user; and

in response to the request, sending to the service provider only information from the portion of the personal information that is permitted to be sent to the service provider.

28. The computer-readable medium of claim 27 wherein the information that is sent to the provider includes less than all of the information requested.

29. The computer-readable medium of claim 27 wherein when the service provider requests only information that is not permitted to be given to the service provider, the request is rejected by the secure token device.

30. The computer-readable medium of claim 27 wherein when the service provider requests only information that is not permitted to be given to the service provider, the request is rejected by the secure token device.

31. The computer-readable medium of claim 27 wherein the service provider is an Internet Service Provider (ISP).

32. The computer-readable medium of claim 27 wherein the secure token device is a smart card.

33. The computer-readable medium of claim 27 wherein the secure token device is an ibutton.

34. In a computer system where a service provider provides services to a customer and wherein the customer uses a secure token device holding tokens representing currency to access the services, a computer-readable medium holding computer executable instructions for performing a method, comprising the steps of:

with the service provider, providing services to the customer during a session;

assessing a charge to the customer for the services that were provided during the session; and

receiving some of the tokens from the secure token device at the service provider, wherein the received tokens constitute payment for covering the charges to the customer.

35. The computer-readable medium of claim 34 wherein the service provider is an Internet Service Provider (ISP) that provides the customer with access to the Internet.

36. A secure token device, comprising:

a storage for storing at least one program that facilitates access to services provided by an Internet Services Provider (ISP), wherein the ISP provides a user with access to the Internet;

a processor for executing programs stored in the storage.

37. The secure token device of claim 36 wherein the storage holds personal information regarding the user and a set of permissions that identify what portions of the personal information may be sent to respective requesters when requested and a program that responds to requests by the ISP for the personal information to review the permissions and determine what portion of the personal information should be sent to the ISP in view of the permissions.

38. The secure token device of claim 36 wherein the storage holds contextual information that captures a context of a last session between the user and the ISP.

39. The secure token device of claim 36 wherein the storage holds tokens representing electronic currency.

40. The secure token device of claim 39 wherein the storage holds a program for forwarding a token from the storage to the ISP to cover payment of services by the ISP to the user.

41. The secure token device of claim 36 wherein the secure token device is a smart card.

42. The secure token device of claim 36 wherein the secure token device is an ibutton.

43. A system comprising:

a secure token device for a user, said secure token device holding a program for enabling the user to access services provided by an Internet Service Provider (ISP), wherein the ISP provides the user with access to the Internet;

a computer system for enabling a user to gain access to services provided by the ISP;

a reader interfaced with the computer system for interfacing the secure token device to gain access to the computer system and services provided by the ISP; and

a remote computer system that acts on behalf of the ISP to provide services to the user.

44. The system of claim 43 wherein the secure token device is a smart card.

45. The system of claim 43 wherein the secure token device is an ibutton.

46. A computer system wherein a user accesses services provided by a service provider during a during sessions via a connectionless protocol and a secure token device is provided for the user, said secure token device holding contextual information that captures a context of a last session that user had with the service provider, said computer system comprising:

interface means for interfacing with the secure token device to facilitate communications between the computer system and the secure token device;

means for receiving the contextual information from the secure token device; and

means for using the contextual information to restore the context of the last session the user had with the service provider during a current session where services are provided by the service provider to the user.

47. The computer system of claim 1 wherein the connectionless protocol is the Internet Protocol (IP).

48. The computer system of claim 46 wherein the secure token device is a smart card.

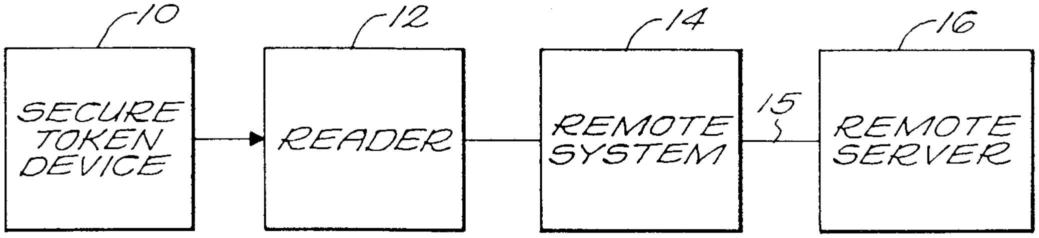


FIG. 1

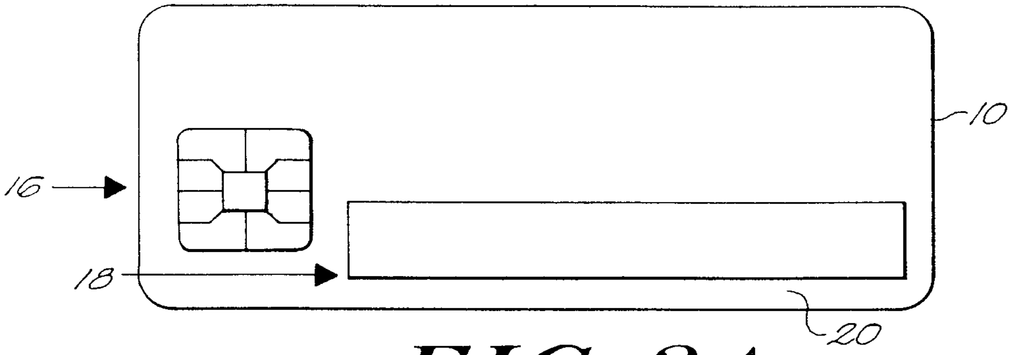


FIG. 2A

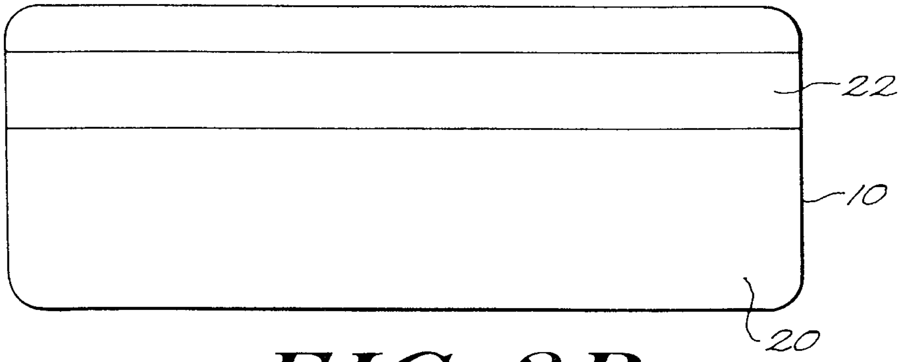


FIG. 2B

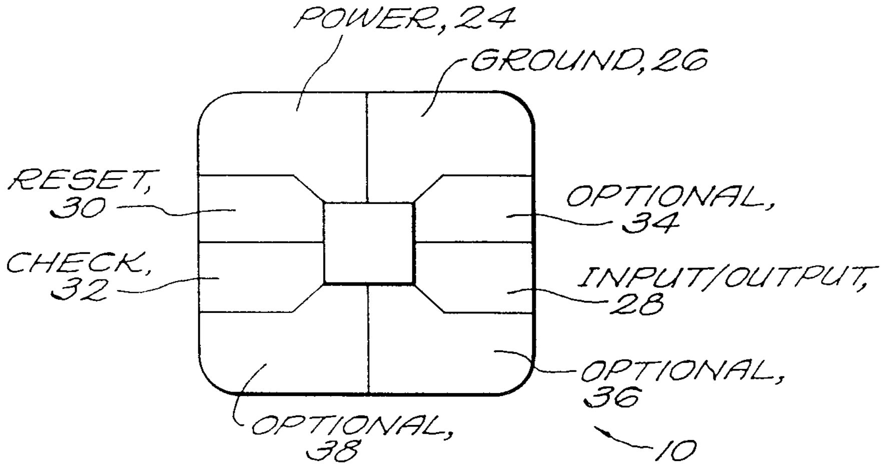


FIG. 2C

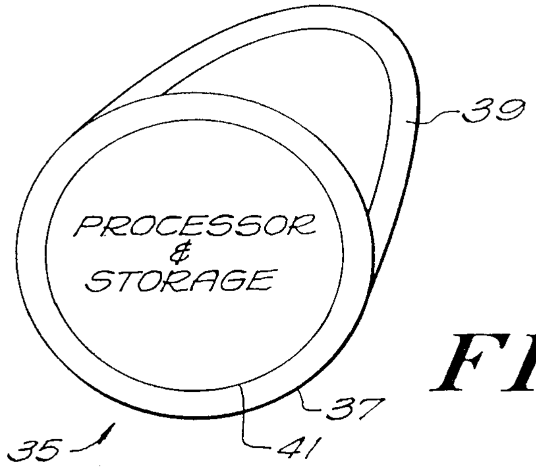


FIG. 3

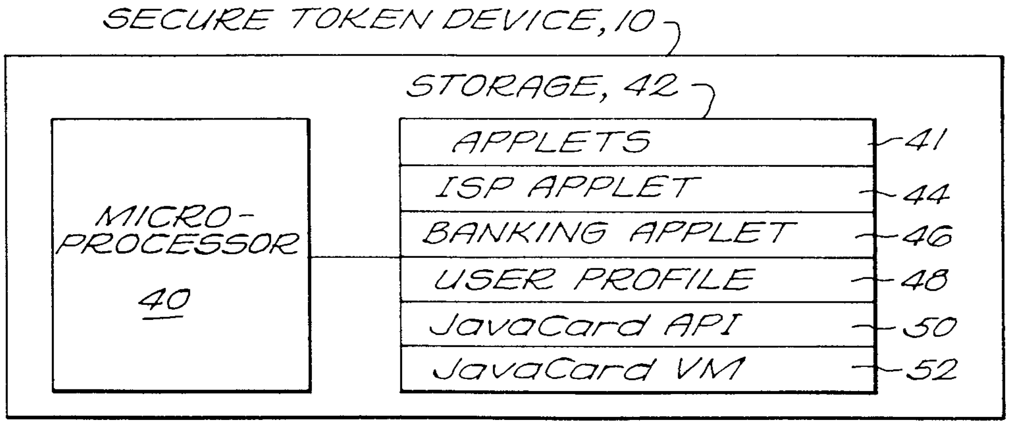


FIG. 4

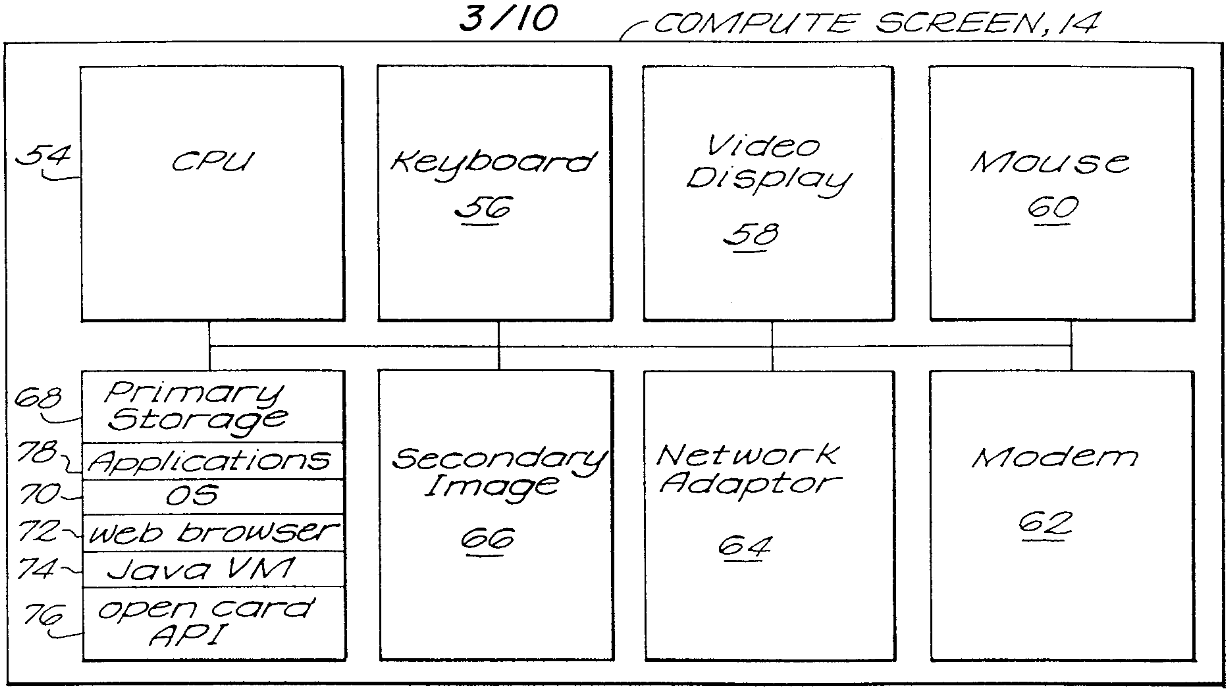


FIG. 5

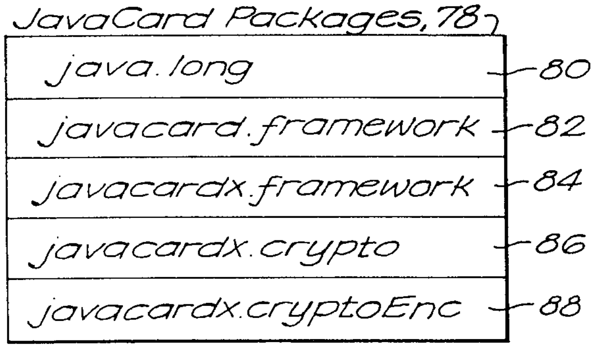


FIG. 6

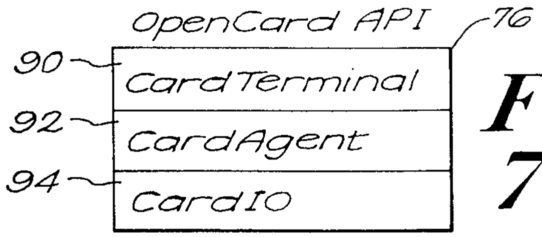


FIG. 7A

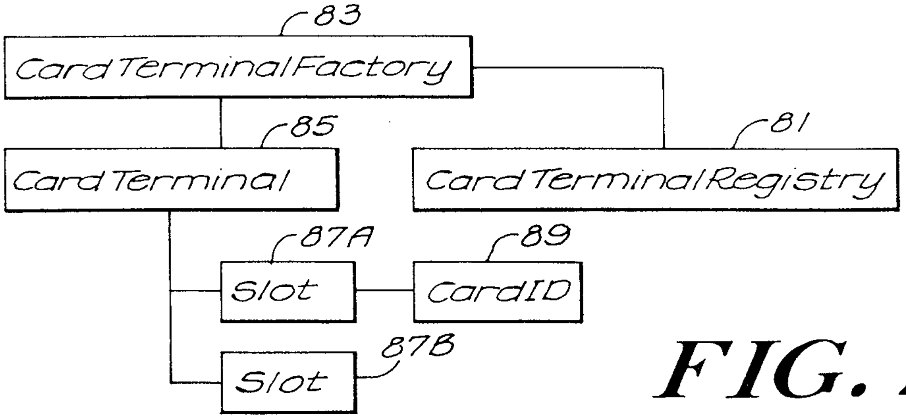


FIG. 7B

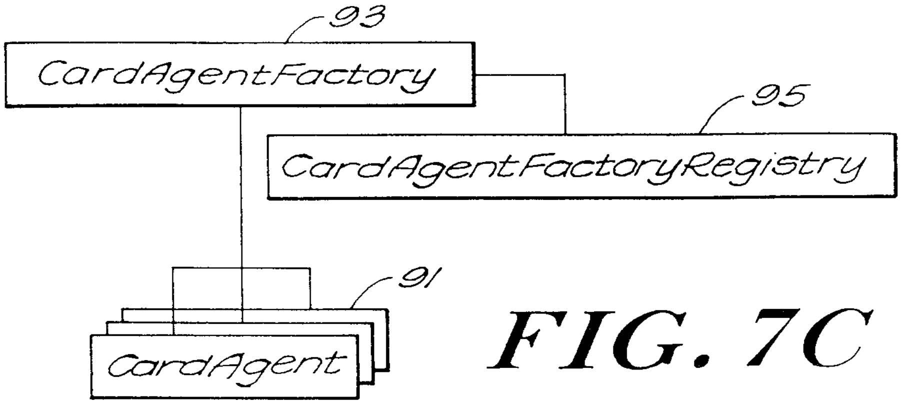


FIG. 7C

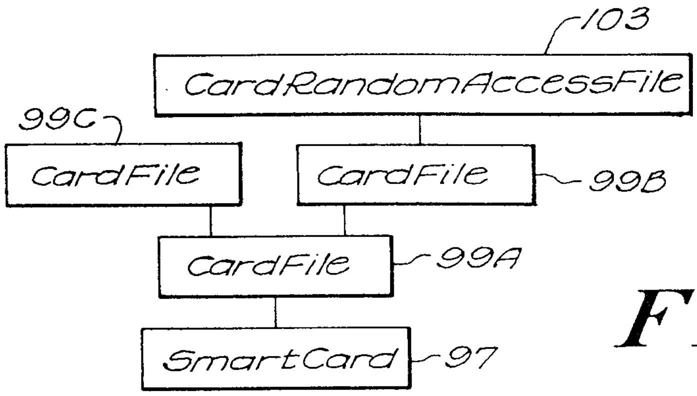


FIG. 7D

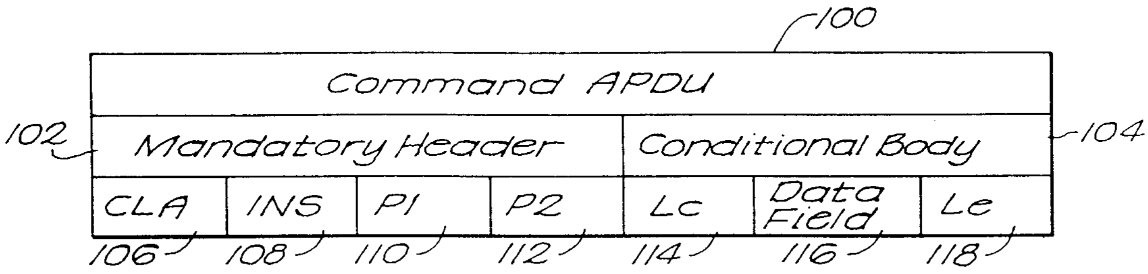


FIG. 8A

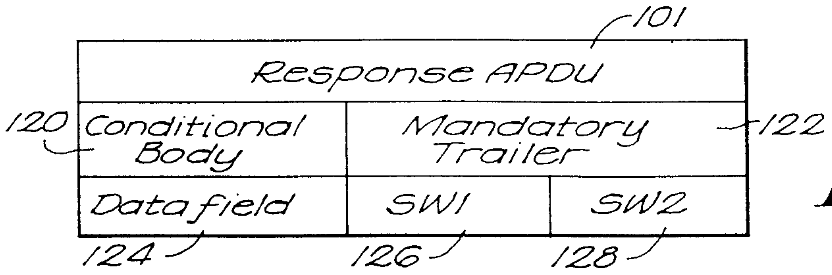


FIG. 8B

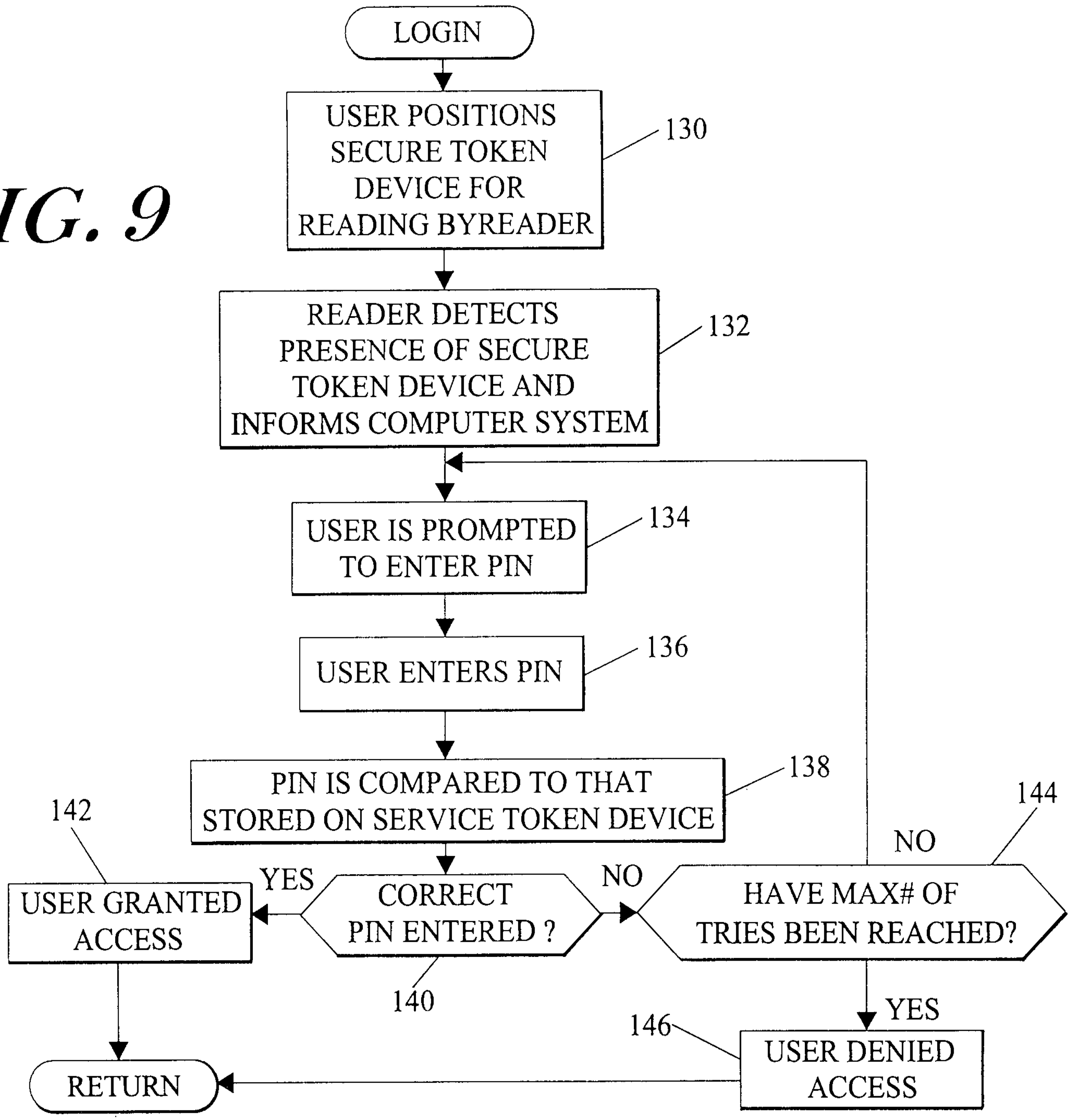

LOGIN
USER POSITIONS SECURE TOKEN DEVICE FOR READING BYREADER
130
READER DETECTS PRESENCE OF SECURE TOKEN DEVICE AND INFORMS COMPUTER SYSTEM
132
USER IS PROMPTED TO ENTER PIN
134
USER ENTERS PIN
136
PIN IS COMPARED TO THAT STORED ON SERVICE TOKEN DEVICE
138
CORRECT PIN ENTERED ?
140
YES
NO
USER GRANTED ACCESS
142
HAVE MAX# OF TRIES BEEN REACHED?
144
NO
YES
USER DENIED ACCESS
146
RETURN

FIG. 9

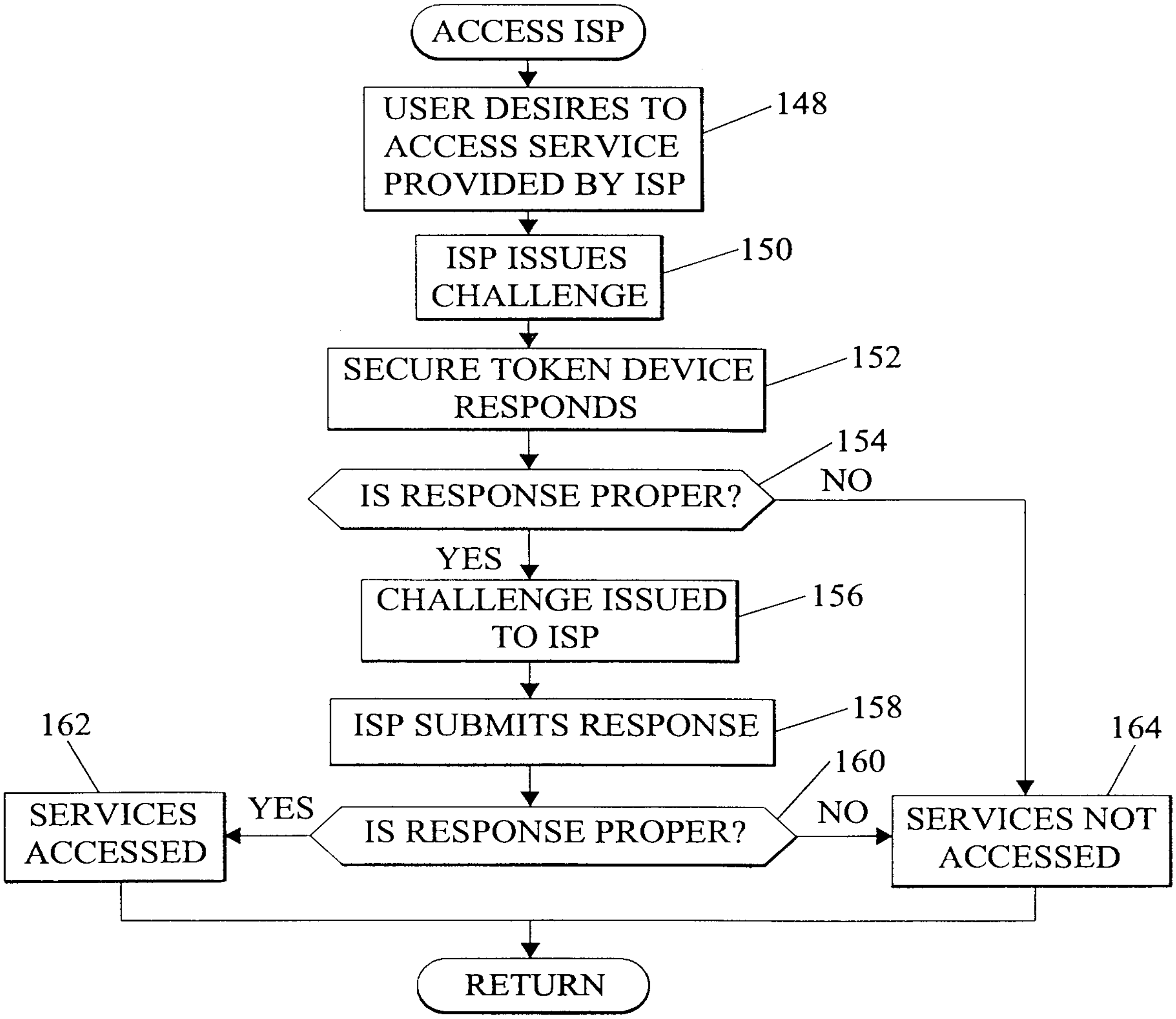

ACCESS ISP
USER DESIRES TO ACCESS SERVICE PROVIDED BY ISP
148
ISP ISSUES CHALLENGE
150
SECURE TOKEN DEVICE RESPONDS
152
154
IS RESPONSE PROPER?
NO
YES
CHALLENGE ISSUED TO ISP
156
ISP SUBMITS RESPONSE
158
162
160
164
SERVICES ACCESSED
YES
IS RESPONSE PROPER?
NO
SERVICES NOT ACCESSED
RETURN

FIG. 10

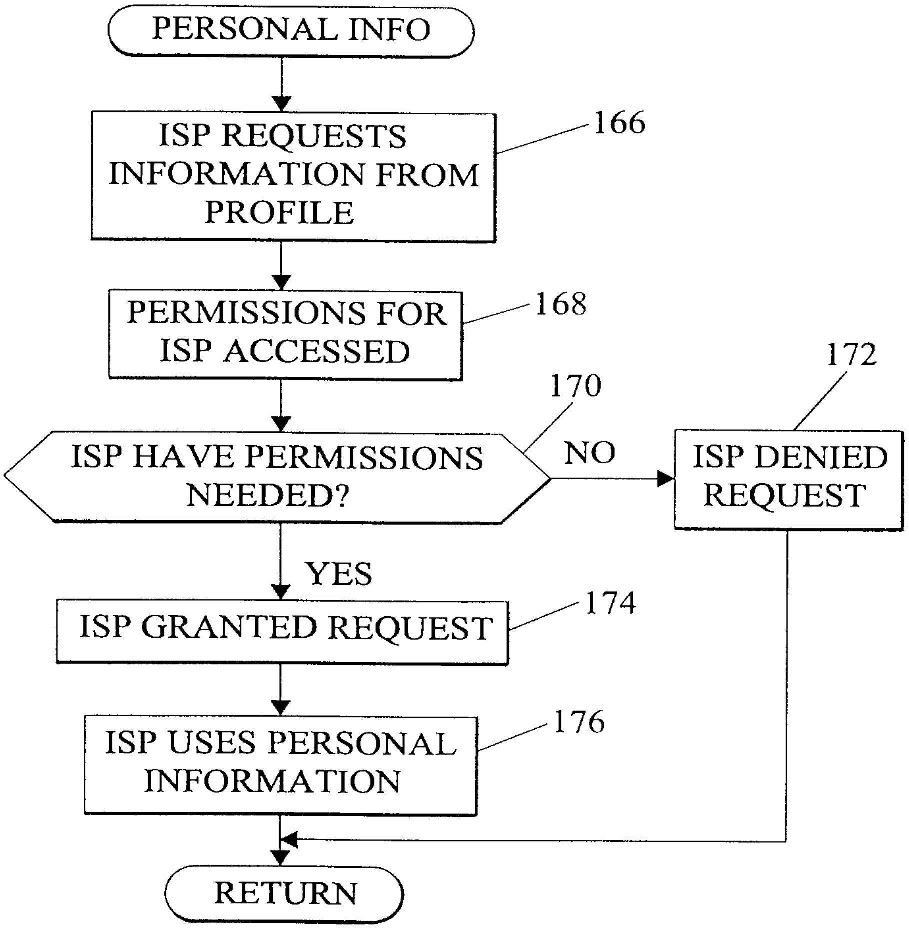

PERSONAL INFO
ISP REQUESTS INFORMATION FROM PROFILE
166
PERMISSIONS FOR ISP ACCESSED
168
ISP HAVE PERMISSIONS NEEDED?
170
NO
ISP DENIED REQUEST
172
YES
ISP GRANTED REQUEST
174
ISP USES PERSONAL INFORMATION
176
RETURN

FIG. 11

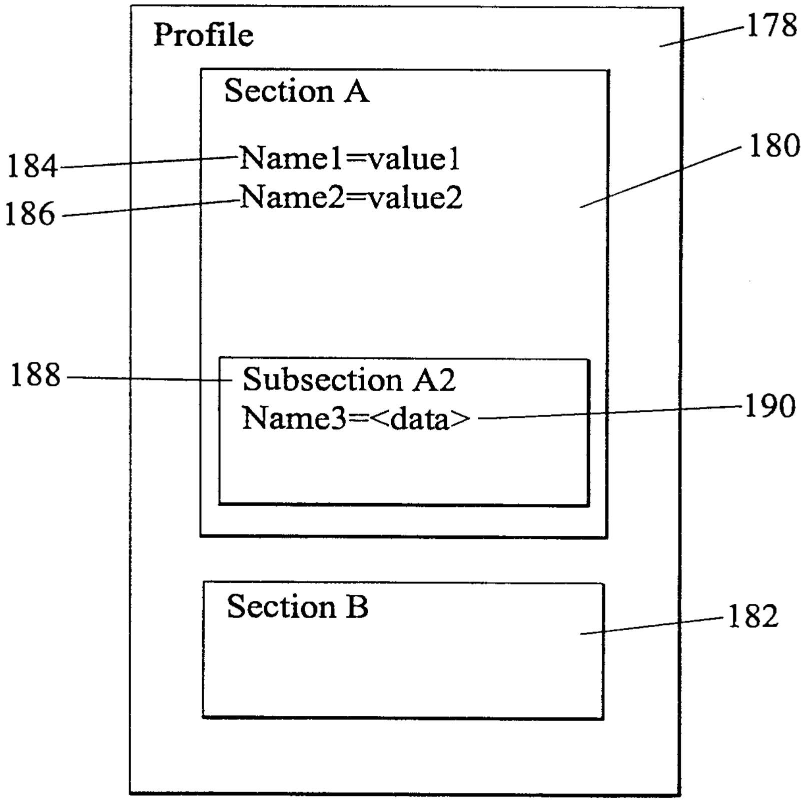

Profile
178
Section A
184
Name1=value1
180
186
Name2=value2
188
Subsection A2
Name3=<data>
190
Section B
182

FIG. 12

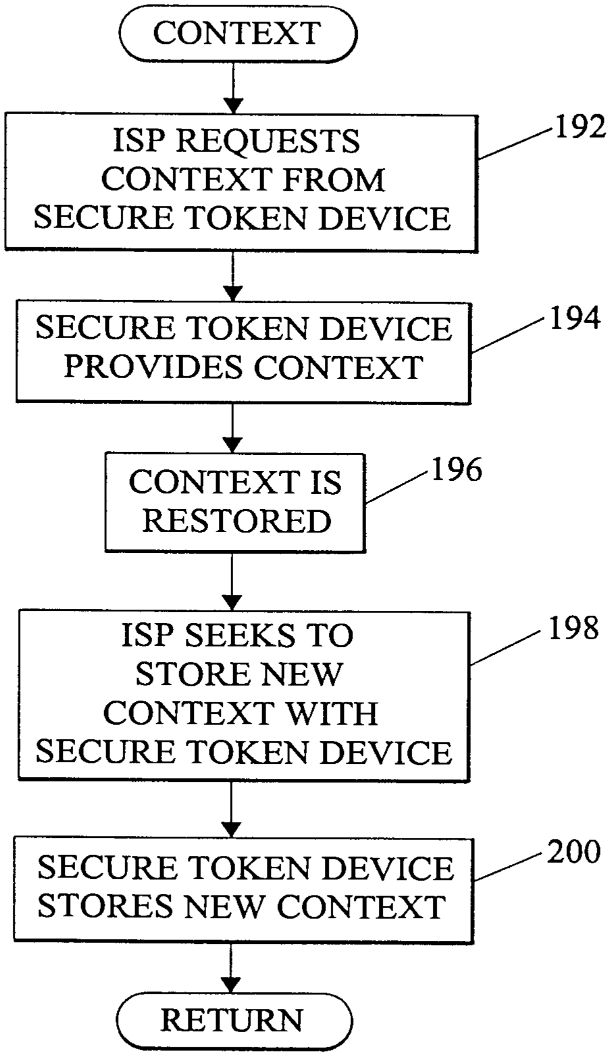

CONTEXT
ISP REQUESTS CONTEXT FROM SECURE TOKEN DEVICE
192
SECURE TOKEN DEVICE PROVIDES CONTEXT
194
CONTEXT IS RESTORED
196
ISP SEEKS TO STORE NEW CONTEXT WITH SECURE TOKEN DEVICE
198
SECURE TOKEN DEVICE STORES NEW CONTEXT
200
RETURN

FIG. 13

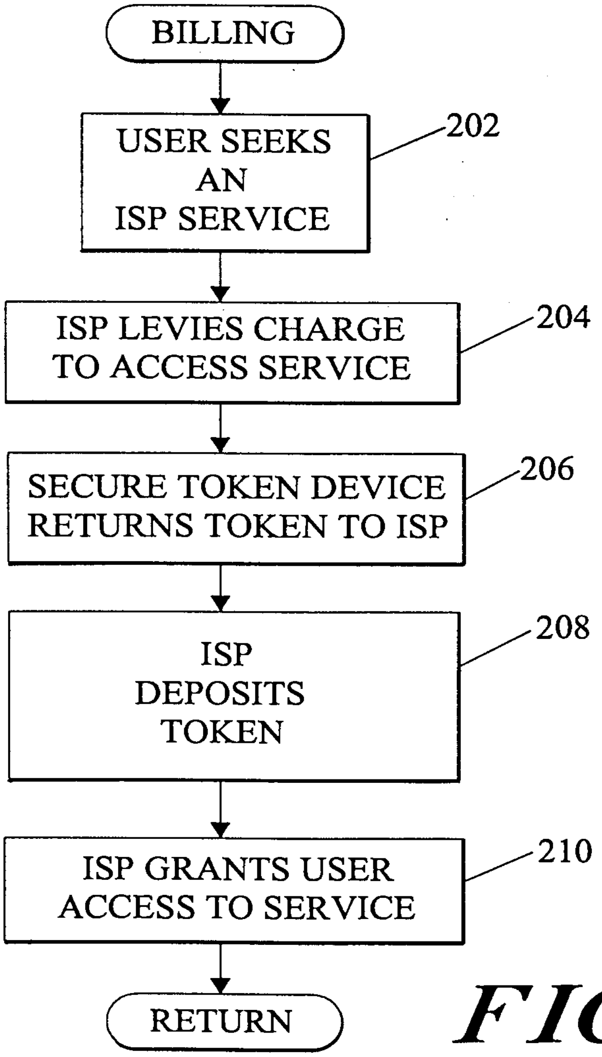

BILLING
USER SEEKS AN ISP SERVICE
202
ISP LEVIES CHARGE TO ACCESS SERVICE
204
SECURE TOKEN DEVICE RETURNS TOKEN TO ISP
206
ISP DEPOSITS TOKEN
208
ISP GRANTS USER ACCESS TO SERVICE
210
RETURN

FIG. 14

Electronic Patent Application Fee Transmittal

Application Number:	13438754
Filing Date:	03-Apr-2012
Title of Invention:	DATA STORAGE AND ACCESS SYSTEMS
First Named Inventor/Applicant Name:	Patrick Sandor Racz
Filer:	Michael R. Casey
Attorney Docket Number:	4037-0003

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
RCE - 2nd and Subsequent Request	1820	1	1700	1700
		Total in USD ($)		**1700**

Electronic Acknowledgement Receipt

EFS ID:	18546372
Application Number:	13438754
International Application Number:	
Confirmation Number:	3525
Title of Invention:	DATA STORAGE AND ACCESS SYSTEMS
First Named Inventor/Applicant Name:	Patrick Sandor Racz
Customer Number:	42624
Filer:	Michael R. Casey
Filer Authorized By:	
Attorney Docket Number:	4037-0003
Receipt Date:	21-MAR-2014
Filing Date:	03-APR-2012
Time Stamp:	12:17:42
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$1700
RAM confirmation Number	9890
Deposit Account	501860
Authorized User	

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 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Request for Continued Examination (RCE)	20140321_RCE_4037-0003.pdf	697795 881139e6d3d56301a2b7d4483ddab499c08fa9a9	no	3
Warnings:					
Information:					
2	Transmittal Letter	20140321_IDS_Transmittal.pdf	134797 d6a5c113ee67358066732a37ad186261ecd87433	no	2
Warnings:					
Information:					
3	Information Disclosure Statement (IDS) Form (SB08)	20140321_1449.pdf	127106 e152b8656007bf2c2049c2f7988f3380644fca34	no	28
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
4	Foreign Reference	F0000.pdf	913972 7d99fce297cbc0b8d3597efeeace4cefa666b41b	no	23
Warnings:					
Information:					
5	Foreign Reference	F0001.pdf	535689 4649e6f9562874db0276f7aca3b0bdecbdf2b0c1	no	9
Warnings:					
Information:					
6	Foreign Reference	F0002.pdf	1058714 4738b0ba66b70b35e44ba7842b4dfed52fb4a4cc	no	18
Warnings:					
Information:					
7	Foreign Reference	F0003.pdf	929820 6667423f1d16e87c27ebae321d3505bbf013b72d	no	10
Warnings:					
Information:					
8	Foreign Reference	F0004.pdf	1232807 37bb6140df577daaea63528d5d025303a65e66f0	no	19

Warnings:					
Information:					
9	Foreign Reference	F0005.pdf	713569 d54deab29ce4d6fdb6d17c58304ce9af6834973d	no	12
Warnings:					
Information:					
10	Foreign Reference	F0006.pdf	884208 30a32b3ecf589596c22e168777604ab5a18e6425	no	21
Warnings:					
Information:					
11	Foreign Reference	F0007.pdf	952516 63ada86d1787a87ae3d93405a7d9d258dc00aa87	no	20
Warnings:					
Information:					
12	Foreign Reference	F0008.pdf	640344 430e3f0e1e3cc69b1b59e293f66609125c39c375	no	10
Warnings:					
Information:					
13	Foreign Reference	F0009.pdf	171965 8ef6859aecedd18990f6cdc9ba69643fe6e78dd3	no	17
Warnings:					
Information:					
14	Foreign Reference	F0010.pdf	1475543 4804c8d630d8451ca07bacebe73649a833c84738	no	20
Warnings:					
Information:					
15	Foreign Reference	F0011.pdf	179712 01597fcd9a5f1437352d7a6c52d88f1feb87a654	no	5
Warnings:					
Information:					
16	Foreign Reference	F0012.pdf	1109841 2a25bf6933453c6feb4ecdf4462dd46d30c6d88c	no	26
Warnings:					
Information:					
17	Foreign Reference	F0013.pdf	700462 8971da745817f90b2be61c88bfddaa2e1aae564c	no	13

Warnings:					
Information:					
18	Foreign Reference	F0014.pdf	169594 a0bc807490e03432e07bed84422289a01dc bfb2e	no	3
Warnings:					
Information:					
19	Foreign Reference	F0015.pdf	1095266 08110b769c6136967db3164dcbdd0db977 bf5ce0	no	14
Warnings:					
Information:					
20	Foreign Reference	F0016.pdf	339507 e47fd41459a00ab6addcf421efdc0dc07c60 102c	no	12
Warnings:					
Information:					
21	Foreign Reference	F0017.pdf	741500 9dc11e8c7d4602f7a8b2f12ed1afa52845f1 1f0b	no	20
Warnings:					
Information:					
22	Foreign Reference	F0018.pdf	6750705 75e2ebbdd7c127aecf2c6c2522575c3a16a dac11	no	80
Warnings:					
Information:					
23	Foreign Reference	F0019.pdf	3187006 099c6eb38881358bb5657ec2db089c1e3a0 48c7a	no	80
Warnings:					
Information:					
24	Foreign Reference	F0020.pdf	383244 659af2ba1218a60b22431b0e3c6eb9bf993 b8168	no	13
Warnings:					
Information:					
25	Foreign Reference	F0021.pdf	674663 3a3a3f2110735c800b596c1d99f5dc0b027f 8271	no	17
Warnings:					
Information:					
26	Foreign Reference	F0022.pdf	766973 d7822afbf8b0b45292b923d04e39c3ef065 308d7	no	22

Warnings:					
Information:					
27	Foreign Reference	F0023.pdf	653192 626e8a7db8236dd917c0fc95f365a00db7a4ae6c	no	16
Warnings:					
Information:					
28	Foreign Reference	F0024.pdf	1136373 76e63410fc55cfd4f6981464fd417e9a51f7d211	no	35
Warnings:					
Information:					
29	Foreign Reference	F0025.pdf	4033244 a8fb867352fdba6d7302c5b8befac7aaaebbe491	no	138
Warnings:					
Information:					
30	Foreign Reference	F0026.pdf	450338 284d769ba1440943484bccbe196eafcb76fe5ee9	no	14
Warnings:					
Information:					
31	Foreign Reference	F0027.pdf	768688 33c33c31f206c835efc01b5759104c91b45bc971	no	19
Warnings:					
Information:					
32	Foreign Reference	F0028.pdf	1570062 93e3c8cca02db6b7f610bf8d70d8b4a3873c06b5	no	52
Warnings:					
Information:					
33	Foreign Reference	F0029.pdf	1593116 fdb5c2c2e3aac18fa69a13d80e6fe1d6873c07a4	no	49
Warnings:					
Information:					
34	Foreign Reference	F0030.pdf	3487691 fb276da6662077abd3c19da8b3da79161fe7b2b2	no	114
Warnings:					
Information:					
35	Foreign Reference	F0031.pdf	2690135 b574d63c4183893c43bf7de12ae70e3453c33af6	no	85

Warnings:					
Information:					
36	Foreign Reference	F0032.pdf	5336596 6f9d7df3b7a6fd4d941a44cf6e713b530165bc3e	no	171
Warnings:					
Information:					
37	Foreign Reference	F0033.pdf	1645366 eb610ed93b85a17a1f51c432f7e1a126df0c4e05	no	50
Warnings:					
Information:					
38	Foreign Reference	F0034.pdf	889290 1b3f89f3d928fc25d9d93ea1c90dbe914ccdbb9d	no	30
Warnings:					
Information:					
39	Foreign Reference	F0035.pdf	4380785 f7f8cdc76f6c1e4a05abe36e1397756a3c5de71f	no	139
Warnings:					
Information:					
40	Foreign Reference	F0036.pdf	3091932 e95d881a8f02bc1949fe44f88ede59ce83a564ae	no	97
Warnings:					
Information:					
41	Foreign Reference	F0037.pdf	2741100 7fd7cb342683b3319514980c7515a419bfbd2cf3	no	87
Warnings:					
Information:					
42	Foreign Reference	F0038.pdf	1034778 9fdc7160f52ecb32a327d6a50b14f63b363d0f9b	no	25
Warnings:					
Information:					
43	Foreign Reference	F0039.pdf	1216850 144aad508f4b33f4cb7653bc9cf4b4e1ccf8673c	no	23
Warnings:					
Information:					
44	Foreign Reference	F0040.pdf	682888 4c450788d70a77d8ece5cd645d0b5cf9203b3e7e	no	17

Warnings:					
Information:					
45	Foreign Reference	F0041.pdf	577484 9f0ff2a364473603ce4de20910c484a01979 50b9	no	17
Warnings:					
Information:					
46	Foreign Reference	F0042.pdf	1504636 5d0282ea7a2b13f694dec4d5d4b5e70a641 9524e	no	40
Warnings:					
Information:					
47	Fee Worksheet (SB06)	fee-info.pdf	29964 cdd4918a5eaaea122ad672d50075f246755 26032	no	2
Warnings:					
Information:					
			Total Files Size (in bytes):	66081826	

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.

Electronic Acknowledgement Receipt

EFS ID:	18546587
Application Number:	13438754
International Application Number:	
Confirmation Number:	3525
Title of Invention:	DATA STORAGE AND ACCESS SYSTEMS
First Named Inventor/Applicant Name:	Patrick Sandor Racz
Customer Number:	42624
Filer:	Michael R. Casey
Filer Authorized By:	
Attorney Docket Number:	4037-0003
Receipt Date:	21-MAR-2014
Filing Date:	03-APR-2012
Time Stamp:	12:29:26
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	Non Patent Literature	NP0000.pdf	323575 4dc3bd80f88804845042d48c8cbec2e2d6cd9d60	no	1

Warnings:

Information:

2	Non Patent Literature	NP0001.pdf	202094 aff195da91b80f4ec6d21230c119efcc46de172f	no	3
Warnings:					
Information:					
3	Non Patent Literature	NP0002.pdf	175425 d6a22e66cbd3e6031b49e71b4ca4bf3759147bdb	no	3
Warnings:					
Information:					
4	Non Patent Literature	NP0003.pdf	108176 a0cd12c640bc0eae52571ce88e508021130dbe09	no	2
Warnings:					
Information:					
5	Non Patent Literature	NP0004.pdf	249155 72c0da1dcfddd3e21a46c810fadbfeae2ffb36e5	no	2
Warnings:					
Information:					
6	Non Patent Literature	NP0005.pdf	660513 bee455d1c8d98cbb9306247538112403ade036a1	no	6
Warnings:					
Information:					
7	Non Patent Literature	NP0006.pdf	208570 9e8822855a7f08772d827faeaafef1330ff8c81b	no	3
Warnings:					
Information:					
8	Non Patent Literature	NP0007.pdf	157879 0dd0857b45c72eba5fb9d6583295d40e5d81fa37	no	2
Warnings:					
Information:					
9	Non Patent Literature	NP0008.pdf	154837 1eb8ec81174d8a143bff0d43cdd6cfdd32356ba0	no	2
Warnings:					
Information:					
10	Non Patent Literature	NP0009.pdf	181167 541be0e12c2340bfca6c3f4e4adf32f55f7d549e	no	3
Warnings:					
Information:					

11	Non Patent Literature	NP0010.pdf	67150 c780cfea7530fa83df0379011adae378bf5b57d3	no	1
Warnings:					
Information:					
12	Non Patent Literature	NP0011.pdf	117746 a6341a710704a512a8c9078d2b650a5a899f739a	no	1
Warnings:					
Information:					
13	Non Patent Literature	NP0012.pdf	227704 0bd6883a569f4ab3172f2a06d7981c2844f05430	no	3
Warnings:					
Information:					
14	Non Patent Literature	NP0013.pdf	141581 536effcd11b76d97a190e2fb84cc390ec4dd68e1	no	2
Warnings:					
Information:					
15	Non Patent Literature	NP0014.pdf	242720 6f236fbdc9f946a64efc0f6dbd7b285c5ce5c3bb	no	3
Warnings:					
Information:					
16	Non Patent Literature	NP0015.pdf	322274 12a295e54659a30d6401e63d23a6515595e8572d	no	5
Warnings:					
Information:					
17	Non Patent Literature	NP0016.pdf	177078 6e4daf39da72b1c14b3ff95293cda264fffaa54f	no	2
Warnings:					
Information:					
18	Non Patent Literature	NP0017.pdf	89517 aa386dac2d5e6c32b775f4ab98ae99d6a6365850	no	1
Warnings:					
Information:					
19	Non Patent Literature	NP0018.pdf	165283 2437324886e16789377936234dd79d1964bdc1c8	no	1
Warnings:					
Information:					

20	Non Patent Literature	NP0019.pdf	193271 70647c87abe64f563f3f9d178b7e95cd494f795a	no	2
Warnings:					
Information:					
21	Non Patent Literature	NP0020.pdf	150843 00a20645d0114be8ce6410bfcfaf2ed418158da7	no	3
Warnings:					
Information:					
22	Non Patent Literature	NP0021.pdf	156819 6982d800c10aad887d8071fdd7daf064668e7abf	no	2
Warnings:					
Information:					
23	Non Patent Literature	NP0022.pdf	409523 51339a33bf2370ce3019e897afa2a782b7fab0e7	no	4
Warnings:					
Information:					
24	Non Patent Literature	NP0023.pdf	120736 acc3798686dffc949477b0e58fc9ae1093dff408	no	2
Warnings:					
Information:					
25	Non Patent Literature	NP0024.pdf	706640 c99226366e44360e30c6f115c2c9c811914950f3	no	10
Warnings:					
Information:					
26	Non Patent Literature	NP0025.pdf	210950 f9d8ef7d01e8fcf33727c439eb78729625d1eae1	no	4
Warnings:					
Information:					
27	Non Patent Literature	NP0026.pdf	78677 2ad0cb17fb69e916a8b5706161cdf80b8170a38b	no	1
Warnings:					
Information:					
28	Non Patent Literature	NP0027.pdf	256875 55c28c80af9addadb840d04849517b8c39391654	no	4
Warnings:					
Information:					

29	Non Patent Literature	NP0028.pdf	163260 7a4e5c4bfa58166de4c15522440a3a49753e969e	no	2
Warnings:					
Information:					
30	Non Patent Literature	NP0029.pdf	84166 2cae4e2abc2dacaf4002337931fbb1be7759ac1e	no	1
Warnings:					
Information:					
31	Non Patent Literature	NP0030.pdf	166798 ced9e8bea67a0df2e95a7fd9ea3c52d8d891682b	no	2
Warnings:					
Information:					
32	Non Patent Literature	NP0031.pdf	289479 19dffdfd790eb0efa672bac565268f1ccab89ce8	no	5
Warnings:					
Information:					
33	Non Patent Literature	NP0032.pdf	756951 4918224f5cbd4cc48ccb62e17f009c88437e02bf	no	6
Warnings:					
Information:					
34	Non Patent Literature	NP0033.pdf	310222 26bfa3bb29769020ceebd33e163d7667c28a2d3d	no	2
Warnings:					
Information:					
35	Non Patent Literature	NP0034.pdf	180764 f3cc50b28fdafaaed53cfd796ed358954ab05131	no	2
Warnings:					
Information:					
36	Non Patent Literature	NP0035.pdf	152496 3933af8470086d232243cf326fc9045b643916f5	no	2
Warnings:					
Information:					
37	Non Patent Literature	NP0036.pdf	115367 37421dee0ee35520be975e42b37aa2b04d6bee3c	no	1
Warnings:					
Information:					

38	Non Patent Literature	NP0037.pdf	195166 f5a3ac9c31874438cf67c36e61bd5ef3c89fdfa0	no	3
Warnings:					
Information:					
39	Non Patent Literature	NP0038.pdf	112511 b5b118ed16d861149ca880d436fc60969c3a95ec	no	2
Warnings:					
Information:					
40	Non Patent Literature	NP0039.pdf	180705 c625c0cc9d059224649307bbb60a88cf75a2d7a5	no	3
Warnings:					
Information:					
41	Non Patent Literature	NP0040.pdf	129903 f727b689ccd1298fbaaeda80b55a9e22746c6d1e	no	2
Warnings:					
Information:					
42	Non Patent Literature	NP0041.pdf	333669 965f60da01584dfa522020df582acac8698af103	no	4
Warnings:					
Information:					
43	Non Patent Literature	NP0042.pdf	144060 196096b0637db7e60599187cff56566ad12a0dd5	no	2
Warnings:					
Information:					
44	Non Patent Literature	NP0043.pdf	515901 8b6ef1af06b2d9bea9a89c92e5ca4fe1998dfa88	no	6
Warnings:					
Information:					
45	Non Patent Literature	NP0044.pdf	245275 f5eaa698301f8c340b2f1afd587ef21d8f0f9a82	no	2
Warnings:					
Information:					
46	Non Patent Literature	NP0045.pdf	159319 80b26b7d5068eab3eea73dd5e54b7b717644c864	no	2
Warnings:					
Information:					

47	Non Patent Literature	NP0046.pdf	87837 ed737b39af858cec073c4abafe7295ddf2c3842e	no	1
Warnings:					
Information:					
48	Non Patent Literature	NP0047.pdf	98904 7d1fd023c27f33252407c59d4b34f15cb2633cc2	no	2
Warnings:					
Information:					
49	Non Patent Literature	NP0048.pdf	267613 63c9740b286a0efeb3b1755e48c7586d04755093	no	3
Warnings:					
Information:					
50	Non Patent Literature	NP0049.pdf	172076 7f26d7a17a5dcba23546d68920a68444b1551eb8	no	1
Warnings:					
Information:					
51	Non Patent Literature	NP0050.pdf	199281 583fc89cf3b9721e8404bfe97fce850130a8c602	no	2
Warnings:					
Information:					
52	Non Patent Literature	NP0051.pdf	162670 5659d4d789ac5771fb6b813d3360bab7c9d92499	no	3
Warnings:					
Information:					
53	Non Patent Literature	NP0052.pdf	288553 f8da6d42722d72399c551f9d383b6611fb9a9c84	no	4
Warnings:					
Information:					
54	Non Patent Literature	NP0053.pdf	129685 67f479510cc611eee083908eeb9c2caec6be6b06	no	2
Warnings:					
Information:					
55	Non Patent Literature	NP0054.pdf	149205 f13d2ad435f001b45c4b75ee9cb018169ee388e1	no	2
Warnings:					
Information:					

56	Non Patent Literature	NP0055.pdf	185938 f8f90cab4514e868c787ab5920eeb4619f6fcfd7	no	2
Warnings:					
Information:					
57	Non Patent Literature	NP0056.pdf	118646 52dd656f5a55ba031f9a6f83a22d217bd6d926db	no	2
Warnings:					
Information:					
58	Non Patent Literature	NP0057.pdf	106505 28f177689f4d50c9518fe491ced43aefad4d24c3	no	1
Warnings:					
Information:					
59	Non Patent Literature	NP0058.pdf	246556 bb2b6d514da8b965b60a041aa525c2436c1cbcb9	no	3
Warnings:					
Information:					
60	Non Patent Literature	NP0059.pdf	142735 03728d5d9312a60ce4a1f5a3907d3105319ee8ec	no	1
Warnings:					
Information:					
		Total Files Size (in bytes):	12848994		

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.

PATENT COOPERATION TREATY

From the INTERNATIONAL SEARCHING AUTHORITY

To:

LUCKHURST, Anthony, Henry, William
Marks & Clerk
57-60 Lincoln's Inn Fields
London WC2A 3LS
UNITED KINGDOM

PCT

NOTIFICATION OF TRANSMITTAL OF THE INTERNATIONAL SEARCH REPORT OR THE DECLARATION

(PCT Rule 44.1)

	Date of mailing *(day/month/year)* 13/02/2001
Applicant's or agent's file reference WPP81421	**FOR FURTHER ACTION** See paragraphs 1 and 4 below
International application No. PCT/GB 00/ 04110	International filing date *(day/month/year)* 25/10/2000
Applicant INTERNET LIMITED et al.	

1. ☒ The applicant is hereby notified that the International Search Report has been established and is transmitted herewith.

Filing of amendments and statement under Article 19:
The applicant is entitled, if he so wishes, to amend the claims of the International Application (see Rule 46):

When? The time limit for filing such amendments is normally 2 months from the date of transmittal of the International Search Report; however, for more details, see the notes on the accompanying sheet.

Where? Directly to the International Bureau of WIPO
34, chemin des Colombettes
1211 Geneva 20, Switzerland
Fascimile No.: (41-22) 740.14.35

For more detailed instructions, see the notes on the accompanying sheet.

2. □ The applicant is hereby notified that no International Search Report will be established and that the declaration under Article 17(2)(a) to that effect is transmitted herewith.

3. □ **With regard to the protest** against payment of (an) additional fee(s) under Rule 40.2, the applicant is notified that:

□ the protest together with the decision thereon has been transmitted to the International Bureau together with the applicant's request to forward the texts of both the protest and the decision thereon to the designated Offices.

□ no decision has been made yet on the protest; the applicant will be notified as soon as a decision is made.

4. **Further action(s):** The applicant is reminded of the following:

Shortly after **18 months** from the priority date, the international application will be published by the International Bureau. If the applicant wishes to avoid or postpone publication, a notice of withdrawal of the international application, or of the priority claim, must reach the International Bureau as provided in Rules 90*bis*.1 and 90*bis*.3, respectively, before the completion of the technical preparations for international publication.

Within **19 months** from the priority date, a demand for international preliminary examination must be filed if the applicant wishes to postpone the entry into the national phase until 30 months from the priority date (in some Offices even later).

Within **20 months** from the priority date, the applicant must perform the prescribed acts for entry into the national phase before all designated Offices which have not been elected in the demand or in a later election within 19 months from the priority date or could not be elected because they are not bound by Chapter II.

Name and mailing address of the International Searching Authority European Patent Office, P.B. 5818 Patentlaan 2 NL-2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Elisabeth Vonk

NOTES TO FORM PCT/ISA/220

These Notes are intended to give the basic instructions concerning the filing of amendments under article 19. The Notes are based on the requirements of the Patent Cooperation Treaty, the Regulations and the Administrative Instructions under that Treaty. In case of discrepancy between these Notes and those requirements, the latter are applicable. For more detailed information, see also the PCT Applicant's Guide, a publication of WIPO.

In these Notes, "Article", "Rule", and "Section" refer to theprovisions of the PCT, the PCT Regulations and the PCT Administrative Instructions respectively.

INSTRUCTIONS CONCERNING AMENDMENTS UNDER ARTICLE 19

The applicant has, after having received the international search report, one opportunity to amend the claims of the international application. It should however be emphasized that, since all parts of the international application (claims, description and drawings) may be amended during the international preliminary examination procedure, there is usually no need to file amendments of the claims under Article 19 except where, e.g. the applicant wants the latter to be published for the purposes of provisional protection or has another reason for amending the claims before international pbulication. Furthermore, it should be emphasized that provisional protection is available in some States only.

What parts of the international application may be amended?

Under Article 19, only the claims may be amended.

During the international phase, the claims may also be amended (or further amended) under Article 34 before the International Preliminary Examining Authority. The description and drawings may only be amended under Article 34 before the International Examining Authority.

Upon entry into the national phase, all parts of the international application may be amended under Article 28 or, where applicable, Article 41.

When? Within 2 months from the date of transmittal of the international search report or 16 months from the priority date, whichever time limit expires later. It should be noted, however, that the amendments will be considered as having been received on time if they are received by the International Bureau after the expiration of the applicable time limit but before the completion of the technical preparations for international publication (Rule 46.1).

Where not to file the amendments?

The amendments may only be filed with the International Bureau and not with the receiving Office or the International Searching Authority (Rule 46.2).

Where a demand for international preliminary examination has been/is filed, see below.

How? Either by cancelling one or more entire claims, by adding one or more new claims or by amending the text of one or more of the claims as filed.

A replacement sheet must be submitted for each sheet of the claims which, on account of an amendment or amendments, differs from the sheet originally filed.

All the claims appearing on a replacement sheet must be numbered in Arabic numerals. Where a claim is cancelled, no renumbering of the other claims is required. In all cases where claims are renumbered, they must be renumbered consecutively (Administrative Instructions, Section 205(b)).

The amendments must be made in the language in which the international application is to be published.

What documents must/may accompany the amendments?

Letter (Section 205(b)):

The amendments must be submitted with a letter.

The letter will not be published with the international application and the amended claims. It should not be confused with the "Statement under Article 19(1)" (see below, under "Statement under Article 19(1)").

The letter must be in English or French, at the choice of the applicant. However, if the language of the international application is English, the letter must be in English; if the language of the international application is French, the letter must be in French.

NOTES TO FORM PCT/ISA/220 (continued)

The letter must indicate the differences between the claims as filed and the claims as amended. It must, in particular, indicate, in connection with each claim appearing in the international application (it being understood that identical indications concerning several claims may be grouped),whether

(i) the claim is unchanged;

(ii) the claim is cancelled;

(iii) the claim is new;

(iv) the claim replaces one or more claims as filed;

(v) the claim is the result of the division of a claim as filed.

The following examples illustrate the manner in which amendments must be explained in the accompanying letter:

1. [Where originally there were 48 claims and after amendment of some claims there are 51]:
"Claims 1 to 29, 31, 32, 34, 35, 37 to 48 replaced by amended claims bearing the same numbers; claims 30, 33 and 36 unchanged; new claims 49 to 51 added."

2. [Where originally there were 15 claims and after amendment of all claims there are 11]:
"Claims 1 to 15 replaced by amended claims 1 to 11."

3. [Where originally there were 14 claims and the amendments consist in cancelling some claims and in adding new claims]:
"Claims 1 to 6 and 14 unchanged; claims 7 to 13 cancelled; new claims 15, 16 and 17 added." or
"Claims 7 to 13 cancelled; new claims 15, 16 and 17 added; all other claims unchanged."

4. [Where various kinds of amendments are made]:
"Claims 1-10 unchanged; claims 11 to 13, 18 and 19 cancelled; claims 14, 15 and 16 replaced by amended claim 14; claim 17 subdivided into amended claims 15, 16 and 17; new claims 20 and 21 added."

"Statement under article 19(1)" (Rule 46.4)

The amendments may be accompanied by a statement explaining the amendments and indicating any impact that such amendments might have on the description and the drawings (which cannot be amended under Article 19(1)).

The statement will be published with the international application and the amended claims.

It must be in the language in which the international appplication is to be published.

It must be brief, not exceeding 500 words if in English or if translated into English.

It should not be confused with and does not replace the letter indicating the differences between the claims as filed and as amended. It must be filed on a separate sheet and must be identified as such by a heading, preferably by using the words "Statement under Article 19(1)."

It may not contain any disparaging comments on the international search report or the relevance of citations contained in that report. Reference to citations, relevant to a given claim, contained in the international search report may be made only in connection with an amendment of that claim.

Consequence if a demand for international preliminary examination has already been filed

If, at the time of filing any amendments under Article 19, a demand for international preliminary examination has already been submitted, the applicant must preferably, at the same time of filing the amendments with the International Bureau, also file a copy of such amendments with the International Preliminary Examining Authority (see Rule 62.2(a), first sentence).

Consequence with regard to translation of the international application for entry into the national phase

The applicant's attention is drawn to the fact that, where upon entry into the national phase, a translation of the claims as amended under Article 19 may have to be furnished to the designated/elected Offices, instead of, or in addition to, the translation of the claims as filed.

For further details on the requirements of each designated/elected Office, see Volume II of the PCT Applicant's Guide.

PATENT COOPERATION TREATY

PCT

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference WPP81421	**FOR FURTHER ACTION** see Notification of Transmittal of International Search Report (Form PCT/ISA/220) as well as, where applicable, item 5 below.	
International application No. PCT/GB 00/04110	International filing date *(day/month/year)* 25/10/2000	(Earliest) Priority Date *(day/month/year)* 25/10/1999
Applicant INTERNET LIMITED et al.		

This International Search Report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This International Search Report consists of a total of 3 sheets.

☒ It is also accompanied by a copy of each prior art document cited in this report.

1. **Basis of the report**

a. With regard to the **language**, the international search was carried out on the basis of the international application in the language in which it was filed, unless otherwise indicated under this item.

□ the international search was carried out on the basis of a translation of the international application furnished to this Authority (Rule 23.1(b)).

b. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international search was carried out on the basis of the sequence listing :

□ contained in the international application in written form.

□ filed together with the international application in computer readable form.

□ furnished subsequently to this Authority in written form.

□ furnished subsequently to this Authority in computer readble form.

□ the statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.

□ the statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished

2. □ **Certain claims were found unsearchable** (See Box I).

3. □ **Unity of invention is lacking** (see Box II).

4. With regard to the **title,**

☒ the text is approved as submitted by the applicant.

□ the text has been established by this Authority to read as follows:

5. With regard to the **abstract,**

☒ the text is approved as submitted by the applicant.

□ the text has been established, according to Rule 38.2(b), by this Authority as it appears in Box III. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

6. The figure of the **drawings** to be published with the abstract is Figure No. 6

☒ as suggested by the applicant. □ None of the figures.

□ because the applicant failed to suggest a figure.

□ because this figure better characterizes the invention.

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 00/04110

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G07F17/16 G07F7/08

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 G07F

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 713 198 A (HITACHI LTD) 22 May 1996 (1996-05-22) claim 1; figure 1 ---	1-3,6,8, 10,11, 16, 20-27, 29-40, 43-58, 63-69, 71-74

-/--

☒ 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 5 February 2001	Date of mailing of the international search report 13/02/2001
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Kirsten, K

1

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 00/04110

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	WO 98 19237 A (SCHLUMBERGER TECHNOLOGIES INC) 7 May 1998 (1998-05-07) claim 1; figure 1 ---	1-6,8, 10,11, 16, 20-23, 35, 37-39, 43-50, 63,73,74
A	EP 0 843 449 A (SUNHAWK CORP INC) 20 May 1998 (1998-05-20) claim 1; figure 1 ---	1-74
A	EP 0 823 694 A (NEDERLAND PTT) 11 February 1998 (1998-02-11) claim 3; figure 3 ---	1-74
A	EP 0 914 001 A (CANAL PLUS SA) 6 May 1999 (1999-05-06) claim 1; figure 1 -----	1-74

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/GB 00/04110

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0713198 A	22-05-1996	JP 8147500 A	07-06-1996
		US 5754654 A	19-05-1998
WO 9819237 A	07-05-1998	AU 722463 B	03-08-2000
		AU 4911897 A	22-05-1998
		EP 0932865 A	04-08-1999
		JP 2000514584 T	31-10-2000
EP 0843449 A	20-05-1998	US 5889860 A	30-03-1999
		CA 2220457 A	08-05-1998
		JP 10301904 A	13-11-1998
EP 0823694 A	11-02-1998	AU 718123 B	06-04-2000
		AU 4118097 A	06-03-1998
		WO 9807120 A	19-02-1998
		EP 0920681 A	09-06-1999
		US 6119945 A	19-09-2000
EP 0914001 A	06-05-1999	AU 9639498 A	17-05-1999
		BR 9813309 A	22-08-2000
		CN 1277782 T	20-12-2000
		EP 1025698 A	09-08-2000
		HR 20000229 A	31-12-2000
		WO 9922516 A	06-05-1999
		NO 20002116 A	28-06-2000
		ZA 9809800 A	04-05-1999

Electronic Acknowledgement Receipt

EFS ID:	18546758
Application Number:	13438754
International Application Number:	
Confirmation Number:	3525
Title of Invention:	DATA STORAGE AND ACCESS SYSTEMS
First Named Inventor/Applicant Name:	Patrick Sandor Racz
Customer Number:	42624
Filer:	Michael R. Casey
Filer Authorized By:	
Attorney Docket Number:	4037-0003
Receipt Date:	21-MAR-2014
Filing Date:	03-APR-2012
Time Stamp:	12:40:13
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	Non Patent Literature	NP0060.pdf	88704 fc44ef1a79cee46ff37292df8fc7ff166e4a8cd5	no	2

Warnings:

Information:

2	Non Patent Literature	NP0061.pdf	167127 183d692949c3ab9934ed04483aaf1611d1d7c81a	no	3
Warnings:					
Information:					
3	Non Patent Literature	NP0062.pdf	299380 1fec1c7671de1bb54a43b61ae58f3933ba0e48f5	no	2
Warnings:					
Information:					
4	Non Patent Literature	NP0063.pdf	99817 7d159646c935147d417622028d15aa945a6537f9	no	2
Warnings:					
Information:					
5	Non Patent Literature	NP0064.pdf	149197 57fd6347462fb1a832641ce20d11c3c75b270b71	no	2
Warnings:					
Information:					
6	Non Patent Literature	NP0065.pdf	132488 91c7163a78e0fa8dd6fb81080ee0bd3fa3fffb3c	no	1
Warnings:					
Information:					
7	Non Patent Literature	NP0066.pdf	162322 6770fdaadaf14bbc5fcaf9ec901069db27f37ef2	no	2
Warnings:					
Information:					
8	Non Patent Literature	NP0067.pdf	341276 fc70ba4003156ff71c035900fb25088018083095	no	7
Warnings:					
Information:					
9	Non Patent Literature	NP0068.pdf	223826 95b69ebb346656edbe5b9c65176f80cb4483d712	no	3
Warnings:					
Information:					
10	Non Patent Literature	NP0069.pdf	84540 a4bb3b517a8615aff3bb6b3c4d6328b1eba15c2c	no	1
Warnings:					
Information:					

11	Non Patent Literature	NP0070.pdf	273976 d03f877fa13c2fee464e0dba33cd9109a4b256e7	no	2
Warnings:					
Information:					
12	Non Patent Literature	NP0071.pdf	120483 ebf6eb6558fbe3a6ce069711f2c65313dd208db1	no	2
Warnings:					
Information:					
13	Non Patent Literature	NP0072.pdf	126878 19162721473021844d7635a9d279e4bc646c7bca	no	3
Warnings:					
Information:					
14	Non Patent Literature	NP0073.pdf	495376 cd8684493c0e6c23ff4d1c2eede65e273205e5c6	no	4
Warnings:					
Information:					
15	Non Patent Literature	NP0074.pdf	119999 2c61ef7422719ae7a9078d60f747f00efea37972	no	2
Warnings:					
Information:					
16	Non Patent Literature	NP0075.pdf	189698 bbda9539fbfc583ef8d9a4feada9c7bfec2fd460	no	3
Warnings:					
Information:					
17	Non Patent Literature	NP0076.pdf	1139189 c57c90fe024b36c779b642e2b80f5f15cb91f94a	no	4
Warnings:					
Information:					
18	Non Patent Literature	NP0077.pdf	94792 3dfd3c47119f4ba2bdc52ea12e3b6a3a763ac3ba	no	1
Warnings:					
Information:					
19	Non Patent Literature	NP0078.pdf	87562 e065deefb436b1712536f5a5806989aa0f861b5c	no	1
Warnings:					
Information:					

20	Non Patent Literature	NP0079.pdf	165275 3c2c6c8108a92e9fa51a87bee6e8694d90e7a3c5	no	1
Warnings:					
Information:					
21	Non Patent Literature	NP0080.pdf	125591 f1d748f1044aaadb47abd36b33da816b43c500b3	no	2
Warnings:					
Information:					
22	Non Patent Literature	NP0081.pdf	128923 1cf3b5dd1dc0dd4d59cb7e7a07d58564ba4f9819	no	2
Warnings:					
Information:					
23	Non Patent Literature	NP0082.pdf	82236 8bc7013efa16664f814e5ff3415929832203fe42	no	1
Warnings:					
Information:					
24	Non Patent Literature	NP0083.pdf	94581 a1b7ddd764d425eb6e4e508206199c83a88e6277	no	1
Warnings:					
Information:					
25	Non Patent Literature	NP0084.pdf	301370 84e562b4a029d44e90c54452fcfc176864656052	no	3
Warnings:					
Information:					
26	Non Patent Literature	NP0085.pdf	220239 fd0982abd91ea92ad12c6d9dc0d8dcb3b9a16ddb	no	2
Warnings:					
Information:					
27	Non Patent Literature	NP0086.pdf	275270 c74512a988da24d27edf9c223e308a0c9f4975e8	no	2
Warnings:					
Information:					
28	Non Patent Literature	NP0087.pdf	83053 34662f162ae161828e6c764141952e2a65367b45	no	1
Warnings:					
Information:					

29	Non Patent Literature	NP0088.pdf	125711 9326dad05b090535e52d4e19cb885c365ced3ca1	no	3
Warnings:					
Information:					
30	Non Patent Literature	NP0089.pdf	109505 ea5d5d42f58df37192721ba479c8e97688e7f325	no	2
Warnings:					
Information:					
31	Non Patent Literature	NP0090.pdf	169313 f94417e17d0f4aa253d30cbf37a77cdda6dda057	no	3
Warnings:					
Information:					
32	Non Patent Literature	NP0091.pdf	113425 22b611351699656e26f23101af6ae6c3c2386d59	no	2
Warnings:					
Information:					
33	Non Patent Literature	NP0092.pdf	125456 389f1b5452d59fa1ef5f30111ea3f480ed78d269	no	2
Warnings:					
Information:					
34	Non Patent Literature	NP0093.pdf	72711 0f1674f3f4a39845055574596143f20db6540c47	no	2
Warnings:					
Information:					
35	Non Patent Literature	NP0094.pdf	285333 1a65a816f2ea3b4cc62e30645077539929be3a4a	no	5
Warnings:					
Information:					
36	Non Patent Literature	NP0095.pdf	223742 fea616541f560064021db0a54f010ee1be1262ab	no	2
Warnings:					
Information:					
37	Non Patent Literature	NP0096.pdf	538578 e5e60a87c23a4765d700bd1e901b4c333073f036	no	9
Warnings:					
Information:					

38	Non Patent Literature	NP0097.pdf	137749 25bc61422b3a4a51408658897f00d0e0896be190	no	3
Warnings:					
Information:					
39	Non Patent Literature	NP0098.pdf	1744569 7e9c4192bd50152b12e8d67499ff0d6c656326a1	no	35
Warnings:					
Information:					
40	Non Patent Literature	NP0099.pdf	214521 50ca21ece89acb3a3dc2c27873e815e4b13eab3d	no	4
Warnings:					
Information:					
41	Non Patent Literature	NP0100.pdf	88269 4283b4ff1eeee0948d08f532494eeea2895f6b69	no	1
Warnings:					
Information:					
42	Non Patent Literature	NP0101.pdf	510997 2445ef1caf2151bbeb0002820cf007cf869f65cb	no	6
Warnings:					
Information:					
43	Non Patent Literature	NP0102.pdf	185740 c0131ac93c8e3ed794f1515e6cd30d89c5350420	no	2
Warnings:					
Information:					
44	Non Patent Literature	NP0103.pdf	406151 51a199d8ac85d6dbd9f5c48f372ab9b508e71dc1	no	4
Warnings:					
Information:					
45	Non Patent Literature	NP0104.pdf	213951 0a610a45905fbc7459e27d82e8e8d5028f797c41	no	5
Warnings:					
Information:					
46	Non Patent Literature	NP0105.pdf	6914861 701f1b0470b68e062e63f556e350516eecdf0dbe	no	112
Warnings:					
Information:					

47	Non Patent Literature	NP0106.pdf	6161388 270f67539b7ab659a64e50bf95a2f9eefbad9b51	no	308
Warnings:					
Information:					
48	Non Patent Literature	NP0107.pdf	4772629 15fabae3ede41a5630b96470b89d718f6bfa7f20	no	237
Warnings:					
Information:					
49	Non Patent Literature	NP0108.pdf	6707477 459e1fb9f204f5a3db9a360225b3c6cbbbfbf239	no	316
Warnings:					
Information:					
50	Non Patent Literature	NP0109.pdf	152437 3a157c2dbc538aa585876d738f06592e13f37cb1	no	2
Warnings:					
Information:					
51	Non Patent Literature	NP0110.pdf	162374 8918b893fc118e3875e5376abb9ae202d194bbb8	no	2
Warnings:					
Information:					
			Total Files Size (in bytes):	36010055	

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.

Electronic Acknowledgement Receipt

EFS ID:	18547002
Application Number:	13438754
International Application Number:	
Confirmation Number:	3525
Title of Invention:	DATA STORAGE AND ACCESS SYSTEMS
First Named Inventor/Applicant Name:	Patrick Sandor Racz
Customer Number:	42624
Filer:	Michael R. Casey
Filer Authorized By:	
Attorney Docket Number:	4037-0003
Receipt Date:	21-MAR-2014
Filing Date:	03-APR-2012
Time Stamp:	13:15:11
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	Non Patent Literature	NP0111.pdf	268202 1b3f568e46f99c9707222a1da782b0c20331016a	no	2

Warnings:

Information:

2	Non Patent Literature	NP0112.pdf	471716 5161ef78f34a038d3d28fbe05b6ed38288cf4752	no	6
Warnings:					
Information:					
3	Non Patent Literature	NP0113.pdf	127416 7bf8534bfed88da94c4bf69af4b99919eaf6f033	no	2
Warnings:					
Information:					
4	Non Patent Literature	NP0114.pdf	21354 c0cc05a8dc6cd9c0113b59651a476dac362e0017	no	1
Warnings:					
Information:					
5	Non Patent Literature	NP0115.pdf	50285 49fcb88312e4a8143fcc015f792594e9345c0082	no	1
Warnings:					
Information:					
6	Non Patent Literature	NP0116.pdf	81768 2e60253e55a9618bb76209c2154990dfdb8e167d	no	1
Warnings:					
Information:					
7	Non Patent Literature	NP0117.pdf	94046 63aecfed2b104971e398a5c02fa6247131c8b3a5	no	2
Warnings:					
Information:					
8	Non Patent Literature	NP0118.pdf	283667 72c436de0ac929e5772298693ef7b19b967d49b7	no	4
Warnings:					
Information:					
9	Non Patent Literature	NP0119.pdf	203308 979e75b680ba49a845ec1cba3ccb7ebf616184ef	no	2
Warnings:					
Information:					
10	Non Patent Literature	NP0120.pdf	134417 4dab63071370c87b5aad96892ef5efee04994514	no	2
Warnings:					
Information:					

11	Non Patent Literature	NP0121.pdf	272317 511d6b6cd7798c54cb8ad9bb58598a1488cef366	no	2
Warnings:					
Information:					
12	Non Patent Literature	NP0122.pdf	155102 775c7fbf3456c69dec33fa0005b7f8cfcce5c360	no	2
Warnings:					
Information:					
13	Non Patent Literature	NP0123.pdf	85182 a777f6f4e827aae252359efce11cec9309ccf41e	no	1
Warnings:					
Information:					
14	Non Patent Literature	NP0124.pdf	132804 3f8ca84686a0289015d9ba857662fd9c34537eb4	no	1
Warnings:					
Information:					
15	Non Patent Literature	NP0125.pdf	533858 c7d45eaf3809991c78ee5d812485b2a5cb52a034	no	65
Warnings:					
Information:					
16	Non Patent Literature	NP0126.pdf	449275 9a3f1439bafe3032f7731c82db52298db021f7cc	no	8
Warnings:					
Information:					
17	Non Patent Literature	NP0127.pdf	5484632 487a2b282d0ebf8900d0402c5863d214d877dff7	no	6
Warnings:					
Information:					
18	Non Patent Literature	NP0128.pdf	126874 e69ca304eb35a2d6a7e9eb08126f95afd766ebbd	no	2
Warnings:					
Information:					
19	Non Patent Literature	NP0129.pdf	135293 1858c4e8f838c77ab7d82907846a51cb955aed61	no	2
Warnings:					
Information:					

20	Non Patent Literature	NP0130.pdf	160140 2eb74bfe9d95cb8169ace938489e5afa4836eede	no	2
Warnings:					
Information:					
21	Non Patent Literature	NP0131.pdf	206881 cc39dc45c16eee6267d04ddc0569f569b8958c67	no	3
Warnings:					
Information:					
22	Non Patent Literature	NP0132.pdf	115882 0d737a4a33309aa277b846f76a1fd8f9b3023047	no	2
Warnings:					
Information:					
23	Non Patent Literature	NP0133.pdf	151615 fac5be10f9d8c9e27f450f0eae63a99d79681f15	no	2
Warnings:					
Information:					
24	Non Patent Literature	NP0134.pdf	172719 73497b88641d5d1765d95361b88f22cf3bbf50e0	no	3
Warnings:					
Information:					
25	Non Patent Literature	NP0135.pdf	213526 81ef81e51641e600d164eac12257228c92571fb9	no	3
Warnings:					
Information:					
26	Non Patent Literature	NP0136.pdf	158303 eb7a12b9eec29cd9ae8051324594f2d58e7dcba7	no	2
Warnings:					
Information:					
27	Non Patent Literature	NP0137.pdf	132676 b51e5ab3bd0ea044336cb5cc0d31103681275ab6	no	2
Warnings:					
Information:					
28	Non Patent Literature	NP0138.pdf	137842 d41cb6139b33fb95147be061c9851aed15524e49	no	2
Warnings:					
Information:					

29	Non Patent Literature	NP0139.pdf	199160 5fa30982ac952e126df2a64eb1a72cbe4eb4a33e	no	3
Warnings:					
Information:					
30	Non Patent Literature	NP0140.pdf	134217 b378b4c32a1fc8c8831bf8b8826906ead9934529	no	2
Warnings:					
Information:					
31	Non Patent Literature	NP0141.pdf	178439 6fdd1ffe21dac92712d3cbc65e55e6f2601bbbbd	no	3
Warnings:					
Information:					
32	Non Patent Literature	NP0142.pdf	202684 804dbbaaa265a9ccc1703457604c6cf016c574cd	no	3
Warnings:					
Information:					
33	Non Patent Literature	NP0143.pdf	151380 0fdc24921a6909fa9e402c8712475c87bac7921f	no	3
Warnings:					
Information:					
34	Non Patent Literature	NP0144.pdf	295174 b4b4865f87c9259e36e7a0b08bd71ce633f7412d	no	4
Warnings:					
Information:					
35	Non Patent Literature	NP0145.pdf	182934 e0af11e808fe59b2f7ba02b9788980c834dedea9	no	3
Warnings:					
Information:					
36	Non Patent Literature	NP0146.pdf	143868 617f612e8d54eacf4782fa500fbb0f93a83b125e	no	2
Warnings:					
Information:					
37	Non Patent Literature	NP0147.pdf	137400 69529990823245d4b18079b13f92f088a110d706	no	2
Warnings:					
Information:					

38	Non Patent Literature	NP0148.pdf	351209 35f559b55b6f44a7ae8997af3eae014305f802fa	no	3
Warnings:					
Information:					
39	Non Patent Literature	NP0149.pdf	74334 5f04cea94b3b587f4d110a8f6ac39b1c07b94c0b	no	1
Warnings:					
Information:					
40	Non Patent Literature	NP0150.pdf	106819 23059a3fc2f34080b31912aa5bc4b5398249d6a3	no	2
Warnings:					
Information:					
41	Non Patent Literature	NP0151.pdf	226900 9ae26fe6f06a6aaafff504f419490e37de3cc0e8	no	3
Warnings:					
Information:					
42	Non Patent Literature	NP0152.pdf	331043 377d3a8d34f1e4b6373823d259b436d2e01df0cd	no	5
Warnings:					
Information:					
43	Non Patent Literature	NP0153.pdf	91398 243dd4db3aab7e4aca8bf918a30f253c905c7683	no	1
Warnings:					
Information:					
44	Non Patent Literature	NP0154.pdf	491626 8db258b373625d3a950ae79383a59e041678d984	no	2
Warnings:					
Information:					
45	Non Patent Literature	NP0155.pdf	571833 24681e64205abb275bed073907e8d3e67d6d2408	no	2
Warnings:					
Information:					
46	Non Patent Literature	NP0156.pdf	564589 9828e8b658fed0673f7e667d8b902099ca6df020	no	2
Warnings:					
Information:					

47	Non Patent Literature	NP0157.pdf	534030 85ac43f48f7d966d3a11d2887d632e47523cbba8	no	2
Warnings:					
Information:					
48	Non Patent Literature	NP0158.pdf	523677 2a682bc753da8ab7d46000d397a55e5ff53aef08	no	2
Warnings:					
Information:					
49	Non Patent Literature	NP0159.pdf	75527 5b6044ac2e3ba73bc0e34b33cc8dbd3964a85d77	no	1
Warnings:					
Information:					
50	Non Patent Literature	NP0160.pdf	196356 9a7eb0eede0abfd9e25b1063cdd2a1e834c69456	no	3
Warnings:					
Information:					
		Total Files Size (in bytes):	16325697		

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.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 1 of 4	Confirmation No.	3525

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.	Document No.	Publication/ Issue Date	Name of Patentee or Applicant of Cited Document
	1-1	US-4878245	1989/10/31	Bradley et al.
	1-2	US-4999806	1991/03/12	Chernow
	1-3			
	1-4			
	1-5			
	1-6			
	1-7			
	1-8			
	1-9			
	1-10			
	1-11			
	1-12			
	1-13			
	1-14			
	1-15			
	1-16			
	1-17			
	1-18			
	1-19			
	1-20			
	1-21			
	1-22			
	1-23			
	1-24			
	1-25			
	1-26			

Examiner Signature		Date Considered	

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
FORM PTO-1449 (modified)

Sheet 2 of 4

Application No.	13/438,754
Filing Date	April 3, 2012
First Named Inventor	Patrick Sandor Racz
Group Art Unit	2887
Examiner Name	Le, Thien Minh
Attorney Docket No.	4037-0003
Confirmation No.	3525

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.	Document No.	Publication Date	Name of Patentee or Applicant of Cited Document	Notes
	2-1	JP 10-269289	1998/10/09	Kouichi	
	2-2	JP11-164058	1999/06/18	Sato et al.	
	2-3	WO 99/43136	1999/08/26	Rydbeck et al.	
	2-4				
	2-5				
	2-6				
	2-7				
	2-8				
	2-9				
	2-10				
	2-11				
	2-12				
	2-13				
	2-14				
	2-15				
	2-16				
	2-17				
	2-18				
	2-19				
	2-20				
	2-21				
	2-22				
	2-23				
	2-24				
	2-25				
	2-26				

Examiner Signature		Date Considered	

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)		
	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 3 of 4	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	3-1	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00102 (U.S. Pat. No. 8,118,221), dated March 31, 2014 (including Declarations)	
	3-2	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00103 (U.S. Pat. No. 8,118,221), dated March 31, 2014 (including Declarations)	
	3-3	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00104 (U.S. Pat. No. 7,334,720), dated March 31, 2014 (including Declarations)	
	3-4	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00105 (U.S. Pat. No. 7,334,720), dated March 31, 2014 (including Declarations)	
	3-5	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00106 (U.S. Pat. No. 8,033,458), dated March 31, 2014 (including Declarations)	
	3-6	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00107 (U.S. Pat. No. 8,033,458), dated March 31, 2014 (including Declarations)	
	3-7	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00108 (U.S. Pat. No. 8,061,598), dated April 1, 2014 (including Declarations)	

Examiner Signature		Date Considered	

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 4 of 4	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	4-1	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00109 (U.S. Pat. No. 8,061,598), dated April 1, 2014 (including Declarations)	
	4-2	Eberhard von Faber, Robert Hammelrath, and Franz-Peter Heider, "The Secure Distribution of Digital Contents," IEEE (1997)	
	4-3		
	4-4		
	4-5		
	4-6		
	4-7		

Examiner Signature		Date Considered	

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

(19)日本国特許庁（ＪＰ）　　(12) 公開特許公報（Ａ）　　(11)特許出願公開番号

特開平10－269289

(43)公開日　平成10年(1998)10月９日

(51)Int.Cl.6		識別記号	F I		
Ｇ０６Ｆ	17/60		Ｇ０６Ｆ	15/21	３３０
	1/00	３７０		1/00	３７０Ｆ
	9/06	５５０		9/06	５５０Ｚ
	15/00	３３０		15/00	３３０Ｚ
Ｇ０９Ｃ	1/00	６６０	Ｇ０９Ｃ	1/00	６６０Ｆ

審査請求　未請求　請求項の数37　ＯＬ　（全 39 頁）　最終頁に続く

(21)出願番号　特願平9－74182

(22)出願日　平成９年(1997)３月26日

(71)出願人　000002185
ソニー株式会社
東京都品川区北品川６丁目７番35号

(72)発明者　真有　浩一
東京都品川区北品川６丁目７番35号　ソニー株式会社内

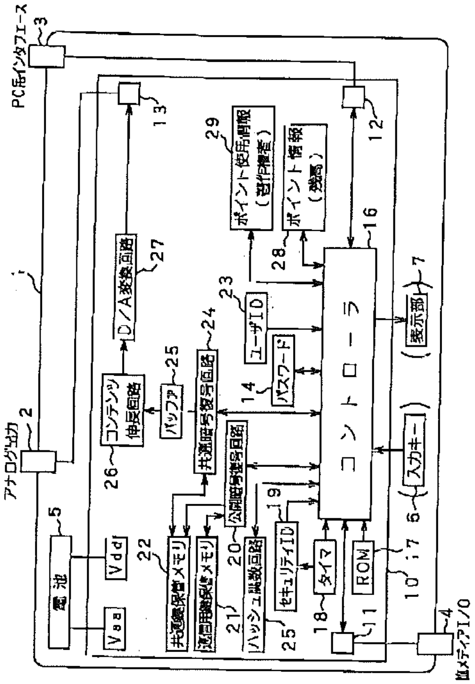
(74)代理人　弁理士　小池　晃　（外２名）

(54)【発明の名称】　ディジタルコンテンツ配付管理方法、ディジタルコンテンツ再生方法及び装置

(57)【要約】

【課題】　簡単に持ち運びができて何時でも何処でもディジタルコンテンツを楽しむことを可能とし、ディジタルコンテンツのコピー或いは不当な使用への防御として十分運用に耐え、且つ経済的なシステムを構築する。

【解決手段】　暗号化されたコンテンツ鍵を復号化し、セッション鍵を暗号化する公開暗号復号回路２０と、コンテンツ鍵やセッション鍵を保管する共通鍵保管メモリ２２と、公開暗号方式の鍵情報を保管する通信用鍵保管メモリ２１と、ポイント情報を格納するポイント情報格納メモリ２９と、ポイント使用情報を格納するポイント使用情報格納メモリ２８と、暗号化ディジタルコンテンツの復号化し、暗号化ポイント情報の復号化、ポイント使用情報の暗号化を行う共通暗号復号回路２４と、圧縮ディジタルコンテンツを伸長する伸長回路２６と、ディジタルコンテンツをＤ／Ａ変換するＤ／Ａ変換回路２７とを、１チップ化する。



【特許請求の範囲】

【請求項１】　ディジタルコンテンツを、当該ディジタルコンテンツ毎のコンテンツ鍵を用いて暗号化すると共に、圧縮するディジタルコンテンツ加工工程と、
上記加工したディジタルコンテンツを、通信相手側からのディジタルコンテンツ送信要求に応じて送信するコンテンツ送信工程と、
上記加工されたディジタルコンテンツの復号化に使用するコンテンツ鍵を暗号化し、通信相手側からのコンテンツ鍵送信要求に応じて送信するコンテン鍵送信工程と、
上記加工されたディジタルコンテンツを復号化する毎に減額される課金情報を暗号化し、通信相手側からの課金情報送信要求に応じて送信する課金情報送信工程と、
通信相手側から送信されてきた暗号化されたコンテンツ使用情報を受信して復号化するコンテンツ使用情報受信工程と、
上記コンテンツ使用情報に基づいて徴収した利用金を、上記ディジタルコンテンツの権利者に対して分配する利用金分配工程とを有してなることを特徴とするディジタルコンテンツ配付管理方法。

【請求項２】　上記コンテンツ鍵は共通鍵であることを特徴とする請求項１記載のディジタルコンテンツ配付管理方法。

【請求項３】　上記コンテンツ鍵を通信相手側の公開鍵を用いて暗号化することを特徴とする請求項１記載のディジタルコンテンツ配付管理方法。

【請求項４】　通信相手側から送信されてきた暗号化された共通鍵を受信して復号化する共通鍵復号化工程を有することを特徴とする請求項１記載のディジタルコンテンツ配付管理方法。

【請求項５】　上記共通鍵はセッション鍵であることを特徴とする請求項４記載のディジタルコンテンツ配付管理方法。

【請求項６】　上記課金情報送信工程では、課金情報を上記共通鍵を用いて暗号化することを特徴とする請求項４記載のディジタルコンテンツ配付管理方法。

【請求項７】　上記コンテンツ使用情報受信工程では、上記暗号化されたコンテンツ使用情報の復号化に上記共通鍵を用いることを特徴とする請求項４記載のディジタルコンテンツ配付管理方法。

【請求項８】　上記コンテンツ使用情報受信工程では、上記通信相手側からの上記課金情報の送信要求に伴って当該通信相手側から送信されてくる上記暗号化されたコンテンツ使用情報を受信することを特徴とする請求項１記載のディジタルコンテンツ配付管理方法。

【請求項９】　上記課金情報送信工程では、上記課金情報と共にコンテンツの使用条件を示す情報を送信することを特徴とする請求項１記載のディジタルコンテンツ配付管理方法。

【請求項１０】　暗号化及び圧縮処理によって加工されたディジタルコンテンツを受信して格納するコンテンツ受信工程と、
上記加工されたディジタルコンテンツの復号化に必要なコンテンツ鍵を要求するためのコンテンツ鍵要求情報を生成するコンテンツ鍵要求情報生成工程と、
上記コンテンツ鍵要求情報を暗号化して送信するコンテンツ鍵要求情報送信工程と、
上記コンテンツ鍵の要求に応じて送信されてきたコンテンツ鍵を受信するコンテンツ鍵受信工程と、
上記コンテンツ鍵に施されている暗号化を復号化するコンテンツ鍵復号化工程と、
上記暗号化されたコンテンツ鍵或いは上記復号化後のコンテンツ鍵を保管するコンテンツ鍵保管工程と、
上記加工されたディジタルコンテンツを上記コンテンツ鍵を用いて復号化するコンテンツ復号化工程と、
上記加工されたディジタルコンテンツを復号化する毎に減額される課金情報を要求するための課金情報要求情報を生成する課金情報要求情報生成工程と、
上記課金情報要求情報を暗号化して送信する課金情報要求情報送信工程と、
上記課金情報の要求に応じて送信されてきた課金情報を受信すると共に当該課金情報に施されている暗号化を復号化して格納する課金情報受信工程と、
上記加工されたディジタルコンテンツを伸長するコンテンツ伸長工程と、
上記加工されたディジタルコンテンツの復号化に応じたコンテンツ使用情報を生成して格納するコンテンツ使用情報格納工程と、
上記コンテンツ使用情報を暗号化して送信するコンテンツ使用情報送信工程とを有することを特徴とするディジタルコンテンツ再生方法。

【請求項１１】　コンテンツ使用情報格納工程では、上記格納されている課金情報の残高を確認し、上記加工されたディジタルコンテンツの復号化に応じて上記格納されている課金情報を減額し、少なくとも上記課金情報の減額量を含むコンテンツ使用情報を生成することを特徴とする請求項１０記載のディジタルコンテンツ再生方法。

【請求項１２】　上記復号化及び伸長がなされたディジタルコンテンツをディジタル／アナログ変換するディジタル／アナログ変換工程を有することを特徴とする請求項１０記載のディジタルコンテンツ再生方法。

【請求項１３】　上記コンテンツ受信工程では、上記加工されたディジタルコンテンツを外部記憶媒体に格納することを特徴とする請求項１０記載のディジタルコンテンツ再生方法。

【請求項１４】　上記コンテンツ鍵は共通鍵であることを特徴とする請求項１０記載のディジタルコンテンツ再生方法。

【請求項１５】　上記コンテンツ鍵復号化工程では、上

記コンテンツ鍵を固有の秘密鍵を用いて復号化することを特徴とする請求項１０記載のディジタルコンテンツ再生方法。

【請求項１６】　共通鍵を発生し、当該共通鍵を暗号化して送信する共通鍵送信工程を有することを特徴とする請求項１０記載のディジタルコンテンツ再生方法。

【請求項１７】　上記共通鍵送信工程では、上記共通鍵としてセッション鍵を生成することを特徴とする請求項１６記載のディジタルコンテンツ再生方法。

【請求項１８】　上記課金情報要求情報送信工程では、上記課金情報要求情報を上記共通鍵を用いて暗号化することを特徴とする請求項１６記載のディジタルコンテンツ再生方法。

【請求項１９】　上記コンテンツ使用情報送信工程では、上記コンテンツ使用情報の暗号化に上記共通鍵を用いることを特徴とする請求項１６記載のディジタルコンテンツ再生方法。

【請求項２０】　上記コンテンツ使用情報送信工程では、上記課金情報要求情報生成工程による上記課金情報の要求に伴って、上記暗号化したコンテンツ使用情報を送信することを特徴とする請求項１０記載のディジタルコンテンツ再生方法。

【請求項２１】　上記課金情報受信工程では、上記課金情報と共に暗号化されて送信されてくるコンテンツの使用条件を示す情報をも受信することを特徴とする請求項１０記載のディジタルコンテンツ再生方法。

【請求項２２】　データ通信を行うデータ通信手段と、
暗号化及び圧縮処理によって加工されたディジタルコンテンツを受信して記憶媒体に記憶させるコンテンツ記憶制御手段と、
暗号化されたコンテンツ鍵を復号化するコンテンツ鍵復号化手段と、
上記暗号化されたコンテンツ鍵或いは上記復号化後のコンテンツ鍵を保管するコンテンツ鍵保管手段と、
上記加工されたディジタルコンテンツを上記コンテンツ鍵を用いて復号化するコンテンツ復号化手段と、
上記加工されたディジタルコンテンツを復号化する毎に減額される課金情報に施されている暗号化を復号化する課金情報復号化手段と、
上記復号化された課金情報を格納する課金情報格納手段と、
上記加工されたディジタルコンテンツを伸長するコンテンツ伸長手段と、
上記加工されたディジタルコンテンツの復号化に応じたコンテンツ使用情報を生成するコンテンツ使用情報生成手段と、
上記コンテンツ使用情報を格納するコンテンツ使用情報格納手段と、
上記コンテンツ使用情報を暗号化するコンテンツ使用情報暗号化手段とを有することを特徴とするディジタルコンテンツ再生装置。

【請求項２３】　上記加工されたディジタルコンテンツの復号化に必要なコンテンツ鍵を要求するためのコンテンツ鍵要求情報を暗号化するコンテンツ鍵要求情報暗号化手段と、
上記加工されたディジタルコンテンツを復号化する毎に減額される課金情報を要求するための課金情報要求情報を暗号化する課金情報要求情報暗号化手段とを有することを特徴とする請求項２２記載のディジタルコンテンツ再生装置。

【請求項２４】　コンテンツ使用情報生成手段は、上記課金情報格納手段に格納されている課金情報の残高を確認し、上記加工されたディジタルコンテンツの復号化に応じて、上記格納されている課金情報を減額し、少なくとも上記課金情報の減額量を含むコンテンツ使用情報を生成することを特徴とする請求項２２記載のディジタルコンテンツ再生装置。

【請求項２５】　上記復号化及び伸長がなされたディジタルコンテンツをディジタル／アナログ変換するディジタル／アナログ変換手段を有することを特徴とする請求項２２記載のディジタルコンテンツ再生装置。

【請求項２６】　上記コンテンツ記憶制御手段は、上記加工されたディジタルコンテンツを外部記憶媒体に記憶させることを特徴とする請求項２２記載のディジタルコンテンツ再生装置。

【請求項２７】　上記コンテンツ鍵は共通鍵であることを特徴とする請求項２２記載のディジタルコンテンツ再生装置。

【請求項２８】　装置固有の鍵を保管する固有鍵格保管段を有し、
上記コンテンツ鍵復号化手段では、上記固有鍵保管手段に保管している装置固有の秘密鍵を用いて、上記暗号化されているコンテンツ鍵を復号化することを特徴とする請求項２２記載のディジタルコンテンツ再生装置。

【請求項２９】　共通鍵を発生する共通鍵発生手段と、
上記共通鍵を暗号化する共通鍵暗号化手段とを有することを特徴とする請求項２２記載のディジタルコンテンツ再生装置。

【請求項３０】　上記共通鍵発生手段は、上記共通鍵としてセッション鍵を生成することを特徴とする請求項２９記載のディジタルコンテンツ再生装置。

【請求項３１】　上記課金情報復号化手段は、上記課金情報を上記共通鍵を用いて復号化することを特徴とする請求項２９記載のディジタルコンテンツ再生装置。

【請求項３２】　上記コンテンツ使用情報暗号化手段は、上記コンテンツ使用情報を上記共通鍵を用いて暗号化することを特徴とする請求項２９記載のディジタルコンテンツ再生装置。

【請求項３３】　上記コンテンツ使用情報暗号化手段は、上記課金情報要求情報暗号化手段による上記課金情

報要求情報の暗号化に伴って、上記コンテンツ使用情報の暗号化を行うを有することを特徴とする請求項２２記載のディジタルコンテンツ再生装置。

【請求項３４】　上記課金情報復号化工程では、上記課金情報と共に暗号化されているコンテンツの使用条件を示す情報をも復号化することを特徴とする請求項２２記載のディジタルコンテンツ再生装置。

【請求項３５】　携帯可能に構成されてなることを特徴とする請求項２２記載のディジタルコンテンツ再生装置。

【請求項３６】　カード状の筐体を有することを特徴とする請求項２２記載のディジタルコンテンツ再生装置。

【請求項３７】　集積回路化してなることを特徴とする請求項２２記載のディジタルコンテンツ再生装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、例えばオーディオデータやビデオデータ等のディジタルコンテンツを配布し、それらディジタルコンテンツの利用量に応じて課金するシステムに好適なディジタルコンテンツ配付管理方法、並びにディジタルコンテンツ再生方法及び装置に関する。

【０００２】

【従来の技術】コンピュータプログラムやオーディオデータ、ビデオデータ等のディジタルコンテンツの流通を簡便化し、潜在需要を掘り下げ、市場拡大に有利な手法としては、例えば特公平６－１９７０７号公報に記載されるソフトウェア管理方式、特公平６－２８０３０号公報に記載されるソフトウェア利用管理方式、特公平６－９５３０２号公報に記載されるソフトウェア管理方式のような手法が存在する。上記特公平６－１９７０７号公報に記載されたソフトウェア管理方式は、無体財産であるコンピュータプログラムやビデオデータ等のソフトウェアの利用に際し、ソフトウェアの利用状況をソフトウェア権利者別などによって把握できるようにしたものである。また、特公平６－２８０３０号公報に記載されるソフトウェア利用管理方式は、無体財産であるコンピュータプログラムやビデオデータ等のソフトウェアの利用に際し、有償プログラムを買い取り（買い取った後は無料で使用できる）価格を付し、コンピュータシステム内には購入可能な金額を示すデータを設けておき、有償プログラム購入の際は、同システムにある利用可能なソフトウェアの名称としてテーブルに登録すると共に、当該購入可能金額を示すデータをソフトウェア価格分だけ減じ、また登録済みソフトウェアを該テーブルから抹消する際には状況に応じて該購入可能な金額を示すデータを増加更新するようにしたものである。また、上記特公平６－９５３０２号公報に記載されるソフトウェア管理方式は、無体財産であるコンピュータプログラムやビデオデータ等のソフトウェアの利用に際し、有償プログラムにつき実際の利用量（利用回数または利用時間など）に応じて利用料金を徴収するために、利用されたプログラムの識別と「利用者識別符号と料金とを記録」しておき、該記録を回収することでプログラム権利者が自分の所有するプログラムの利用料金を把握でき、プログラムの利用量に応じたプログラム利用料金を回収する場合のシステムで有効なものである。

【０００３】

【発明が解決しようとする課題】ところが、上述したディジタルコンテンツをネットワークを使って配信するシステムは、パーソナルコンピュータ上だけでの運用を考慮しており、したがって、簡単に持ち運びができ、何時でも、また何処でも上記ディジタルコンテンツを楽しむといったシステムは存在しない。

【０００４】一方、上述した各公報記載の手法は、潜在需要を掘り下げ、市場拡大に有利であるが、ディジタルコンテンツのコピー或いは不当な使用への防御として不十分であり、且つ経済的なシステムとは言い難い。

【０００５】そこで、本発明はこのような状況に鑑みてなされたものであり、簡単に持ち運びができて何時でも何処でもディジタルコンテンツを楽しむことを可能とし、また、ディジタルコンテンツのコピー或いは不当な使用への防御として十分運用に耐え、且つ経済的なシステムを構築することをも可能にするディジタルコンテンツ配付管理方法、並びにディジタルコンテンツ再生方法及び装置を提供することを目的とする。

【０００６】

【課題を解決するための手段】本発明によれば、ディジタルコンテンツの配付側では、ディジタルコンテンツを暗号化及び圧縮して加工し、この加工したディジタルコンテンツと暗号化したコンテンツ鍵、さらに暗号化した課金情報を通信相手側に送信し、通信相手側から送信されてきたコンテンツ使用情報に基づいて徴収した利用金を権利者に対して分配するようにしており、一方、ディジタルコンテンツの再生側では、その加工されたディジタルコンテンツをコンテンツ鍵にて復号すると共に伸長して再生し、同時にコンテンツの使用に応じて課金情報の減額とコンテンツ使用情報の生成を行い、このコンテンツ使用情報を配付側に送信するようにし、また本発明のディジタルコンテンツ再生装置は、携帯可能となされていることにより、上述した課題を解決する。

【０００７】

【発明の実施の形態】以下、本発明の好ましい実施の形態について、図面を参照しながら説明する。

【０００８】先ず、本発明のディジタルコンテンツ配付方法、ディジタルコンテンツ再生方法及び装置の具体的内容及び構成の説明を行う前に、これらの理解を容易にするために、本発明が適用されるシステム全体の概略構成及びシステムの運用方法について図１から図７までの各図を用いて簡単に説明する。

【０００９】図1にはシステム全体の概略的な構成を示す。

【００１０】この図1において、ユーザ側２００は、本発明のディジタルコンテンツ再生装置（以下、プレーヤ１と呼ぶことにする）及びいわゆるパーソナルコンピュータ（以下、ユーザ端末５０と呼ぶことにする）を保有しているものとする。

【００１１】ユーザ端末５０は、通常のパーソナルコンピュータであるが、本発明に使用する後述する各種ソフトウェアをアプリケーションソフトとして格納してなると共に、表示手段であるディスプレイ装置と放音手段であるスピーカ、及び情報入力手段であるキーボードやマウス等が接続されてなるものである。当該ユーザ端末５０は例えばネットワークを介してシステム管理会社２１０と接続可能であり、また、プレーヤ１との間のインターフェイス手段を有し、データ送受が可能である。

【００１２】プレーヤ1は例えば図2に示すような構成を有するものである。

【００１３】この図2の構成の詳細な説明については後述するが、当該プレーヤ1は、ディジタルコンテンツの処理経路の主要構成要素として、暗号化されているディジタルコンテンツをコンテンツ鍵を用いて復号化する共通鍵暗号復号回路２４と、圧縮されているディジタルコンテンツを伸長する伸長手段である伸長回路２６と、ディジタルデータをアナログ信号に変換するＤ／Ａ変換回路２７とを少なくとも有する。なお、以下に言う復号化とは、暗号化を解くことである。

【００１４】また、このプレーヤ1は、使用するディジタルコンテンツの権利情報及び使用状況を示す情報（以下、これら情報をポイント使用情報と呼ぶ）や、ディジタルコンテンツを使用する際に必要となる保有金額データ、すなわちディジタルコンテンツを使用する毎に減額される課金データ（以下、ポイント情報と呼ぶ）等を扱う主要構成要素として、上記ポイント使用情報を格納するポイント使用情報格納メモリ２９と、上記ポイント情報を格納するポイント情報格納メモリ２８とを少なくとも備えている。

【００１５】さらに、このプレーヤ1は、後述するような暗号化及び復号化に使用する各種鍵を格納するための構成として共通鍵保管メモリ２２及び通信用鍵保管メモリ２１と、これらに格納された鍵を用いて暗号化や復号化を行うための構成として共通暗号復号回路２４及び公開暗号復号回路２０を有している。また、このプレーヤ1は、上記暗号化及び復号化に関連する構成として、システム管理会社２１０のホストコンピュータと連動した乱数を発生してセキュリティＩＤを生成するセキュリティＩＤ発生回路１９及びタイマ１８や、後述するいわゆるハッシュ値を発生するハッシュ関数回路２５等をも有している。

【００１６】その他、当該プレーヤ1は、ディジタルコンテンツやその他各種のデータ及び各構成要素の制御をＲＯＭ１７に格納されたプログラムに基づいて行う制御手段であるコントローラ１６と、携帯時の動作電源としての電池5を備えている。

【００１７】ここで、図2のプレーヤ1の各主要構成要素は、セキュリティ上、ＩＣ（集積回路）或いはＬＳＩ（大規模集積回路）の１チップで構成されることが望ましい。この図2では、各主要構成要素が集積回路１０内に１チップ化されている。当該プレーヤ1には、外部とのインターフェイス用として３つの端子（アナログ出力端子2と、ＰＣ用インターフェイス端子3と、記録メディア用Ｉ／Ｏ端子4）を備え、これら各端子が集積回路１０のそれぞれ対応する端子１３、１２、１１に接続されている。なお、これら各端子は統合することも、また新たに別の端子を設けることも可能であり、特にこだわるものではない。

【００１８】システム管理会社２１０は、システム全体を管理する管理センタ２１１と、上記プレーヤ1を販売する販売店２１２とからなり、仮想店舗２３０を介してユーザ側２００のユーザ端末５０との間で、後述するようなディジタルコンテンツの供給に関する情報の送受、コンテンツプロバイダ２４０が保有するコンテンツを圧縮及び暗号化するディジタルコンテンツの加工、上記加工したディジタルコンテンツの供給、金融機関２２０との間の情報送受等を行う。なお、システム管理会社２１０と金融機関２２０との間では、ユーザ側２００の口座番号やクレジット番号，名前や連絡先等の確認や、ユーザ側２００との間で取引可能かどうかの情報等のやり取りなどが行われる。金融機関２２０とユーザ側２００との間では、実際の代金振込等の処理が行われる。また、販売店２１２は、必ずしもシステム管理会社２１０内に含まれる必要はなく、販売代理店であってもよい。

【００１９】上記システム管理会社２１０の管理センタ２１１は、例えば図3に示すような構成を有するものである。この図3の構成の詳細な説明については後述するが、主要構成要素として、ディジタルコンテンツを管理し、その展示、暗号化及び圧縮等の加工処理、ディジタルコンテンツの暗号化及び復号化に使用する鍵情報であるコンテンツ鍵やＩＤの発生等の各機能を有するコンテンツ管理機能ブロック１００と、ユーザ情報を管理し、通信文（メッセージやポイント情報等）の暗号化及び復号化、確認メッセージの発生、セキュリティＩＤの発生、金融機関２３０との間での決済申請、ポイントの発生等の各機能の他、ユーザ加入処理等を行うユーザ加入処理機能部１１８をも備えたユーザ管理機能ブロック１１０と、ポイント使用情報等を管理する使用情報管理機能ブロック１２０と、システム全体を管理し、通信機能を有する管理機能ブロック１３０とを、少なくとも有してなる。

【００２０】上述した図1のように構成されるシステム

の実際の運用方法の一例を、図４～図７を用いて説明する。なお、以下の運用方法は、ユーザ側２００やシステム管理会社２１０，金融機関２２０，コンテンツプロバイダ２４０等が実際に行う手順である。

【００２１】このシステムの運用方法の説明では、プレーヤ１の購入の手順、ディジタルコンテンツの検索からプレーヤ１用の記憶メディアに対するディジタルコンテンツのインストールまでの手順、当該ディジタルコンテンツを使用可能にするための課金用のポイント情報の購入と当該ディジタルコンテンツを使用した場合の精算の手順、ディジタルコンテンツの鑑賞に伴ってユーザから徴収した課金代金の分配の手順について順番に説明する。

【００２２】先ず、プレーヤ１の購入時の手順としては、図４の（１）及び（５）に示すように、ユーザ側２００が実際に店頭或いは通信販売等により、上記販売店２１２から上記プレーヤ１を購入する。

【００２３】このとき、上記販売店２１２は、図４の（２）に示すように、上記プレーヤ１の販売時に上記ユーザ側２００から提供された個人情報（名前や連絡先等）及び決済情報（銀行口座、クレジット番号等）と、上記販売したプレーヤ１固有の番号（プレーヤ固有鍵等を含む）とをシステム管理会社２１０の管理センタ２１１に登録する。

【００２４】管理センタ２１１は、図４の（３）に示すように、金融機関２２０に対して、上記ユーザ側２００から提供された口座番号やクレジット番号等の確認を行い、図４の（４）に示すように金融機関２２０から取引可能である旨の情報を得る。

【００２５】次に、ディジタルコンテンツの検索からプレーヤ１用の記憶メディアへのディジタルコンテンツのインストールまでの手順として、上記プレーヤ１を購入したユーザ側２００は、当該プレーヤ１とのインターフェイス手段を備えたユーザ端末５０を使って、図５の（１）に示すように、希望のディジタルコンテンツの検索，選択，編集，注文等を行う。このときの検索から注文までの処理は、ユーザ端末５０がアプリケーションソフトとして格納している検索ソフトを用い、例えばネットワークを介して接続された仮想店舗２３０に対して行う。

【００２６】仮想店舗２３０は、例えば管理センタ２１１がネットワーク上の仮想的に設けている店舗であり、この仮想店舗２３０には、例えば複数のコンテンツの内容を示す情報が展示されている。ユーザ側２００は、仮想店舗２３０にて提供されているこれらの情報に基づいて、所望のコンテンツの注文を行うことになる。なお、仮想店舗２３０に展示されるコンテンツの内容を示す情報としては、例えばコンテンツが映画等のビデオデータである場合には当該映画等のタイトルや広告、当該映画中の１シーン等の映像などが考えられ、また、コンテンツがオーディオデータである場合は曲名やアーティスト名、当該曲の最初の数フレーズ（いわゆるイントロ）等が考えられる。したがって、ユーザ側２００のユーザ端末５０にて上記仮想店舗２３０をアクセスした場合には、当該ユーザ端末５０上に上記仮想店舗２３０の複数のコンテンツの内容が仮想的に展示され、これら展示物の中から所望のものを選択することでコンテンツの注文が行われることになる。

【００２７】上記ユーザ側２００のユーザ端末５０からディジタルコンテンツの注文等があったとき、上記仮想店舗２３０は、図５の（２）に示すように管理センタ２１１に対してディジタルコンテンツの供給依頼を行う。

【００２８】当該ディジタルコンテンツの供給依頼を受け取った管理センタ２１１は、コンテンツプロバイダ２４０に対して上記供給依頼のあったディジタルコンテンツの配給依頼を行う。これにより、当該コンテンツプロバイダ２４０は、図５の（４）に示すように上記配給依頼のあったディジタルコンテンツを管理センタ２１１に配給する。

【００２９】管理センタ２１１は、上記コンテンツプロバイダ２４０から配給されたディジタルコンテンツに対して暗号化及び所定の圧縮方式を用いた圧縮を施すと共に、この圧縮及び暗号化されたディジタルコンテンツに対して、当該コンテンツのＩＤ（コンテンツＩＤ）とこのコンテンツの著作権者等の権利者情報と当該コンテンツを使用したときの課金額とコンテンツをユーザ側２００に供給する仮想店舗名等とを付加する。なお、コンテンツに対する課金額は、コンテンツプロバイダ２４０にて事前に決定される。

【００３０】上記管理センタ２１１にて加工されたコンテンツは、図５の（５）に示すように、仮想店舗２３０に送られ、さらにこの仮想店舗２３０を介して、図５の（６）のようにユーザ側２００のユーザ端末５０に供給される。これにより、プレーヤ１には、上記ユーザ端末５０からコンテンツが供給され、このコンテンツが当該プレーヤ１に格納されることになる。

【００３１】なお、この図５に（２）～（５）までの流れについては、事前に行っておくことも可能である。すなわち、仮想店舗２３０には、上記複数のコンテンツの内容を示す情報を展示するだけでなく、これら展示に対応した上記加工されたディジタルコンテンツを予め用意しておくようにしても良い。

【００３２】次に、上述のようにしてプレーヤ１にインストールされたディジタルコンテンツを使用可能にするための課金用のポイント情報の購入と当該ディジタルコンテンツを使用した場合の精算の手順では、先ず、ユーザ端末５０によってプレーヤ１に格納されているポイント情報の不足が確認されて、当該ユーザ端末５０からポイント情報の補充要求がなされる。

【００３３】このとき、図６の（１）のように、当該ユ

ーザ端末５０からは、プレーヤ１にて暗号化されたポイント情報の補充依頼が、管理センタ２１１に対し転送される。また同時に、既に使用したディジタルコンテンツに対応する著作権者等の権利者の情報すなわちポイント使用情報がプレーヤ１から読み出されて暗号化され、ユーザ端末５０を介して管理センタ２１１に送られる。このように、ポイント情報の補充依頼と同時にポイント使用情報の転送が行われるようにしたのは、当該ポイント使用情報の管理センタ２１１への送信のみのために、ユーザ側２００が管理センタ２１１にアクセスする手間を省くためである。勿論、このポイント使用情報の転送は、必ずしもポイント情報の購入と同時に行う必要はなく、独立に行っても良い。

【００３４】上記暗号化されたポイント情報の補充依頼及びポイント使用情報を受け取った管理センタ２１１は、当該暗号を解読することでユーザ側２００が要求しているポイント情報の補充量とポイント使用情報の内容を認識する。さらに、当該管理センタ２１１は、金融機関２２０に対して図6の（２）のように当該ポイント補充分の決済が可能かどうかの確認を行う。金融機関２２０にて、ユーザ側２００の口座を調べることによって、決済可能であることが確認されると、当該金融機関２２０から図6の（３）のように決済ＯＫの指示が管理センタ２１１に送られることになる。

【００３５】また、このときの管理センタ２１１は、図6の（４）に示すように、コンテンツプロバイダ２４０に対して著作権者等の権利者に支払われることになるポイント使用数、すなわち金額を連絡する。

【００３６】その後、管理センタ２１１では、ポイント補充情報の命令書を暗号化し、これをセキュリティＩＤと共にポイント補充指示情報として、図6の（５）に示すようにユーザ端末５０に送る。このユーザ端末５０からプレーヤ１に送られた上記ポイント補充指示情報は、当該プレーヤ１において復号化され、さらにセキュリティＩＤの確認後に、ポイント情報格納メモリ２８へのポイント情報の補充と、ポイント使用情報格納メモリ２９からの上記先に連絡した著作権情報等の権利者情報の削除とが行われる。

【００３７】次に、ディジタルコンテンツの鑑賞に伴ってユーザから徴収した課金代金、すなわちポイントの使用情報に応じてユーザの口座から引き落とされることになる代金の分配の手順では、先ず図7の（１）のようにユーザ側２００に対して代金振り込み依頼が金融機関２２０からなされる。このとき、ユーザ側２００の口座に十分な残高がある場合には、特に代金振り込み依頼はなされず、口座に十分な残高がない場合には、図7の（２）のようにユーザ側２００から金融機関２２０に対して代金の振り込みがなされる。

【００３８】金融機関２２０は、所定の手数料を差し引いて、図7の（３）のように、ユーザ側２００から受け取った代金を管理センタ２１１に対して送金する。すなわち管理センタ２１１では、金融機関２２０から送金された上記代金から、コンテンツ加工料と金融手数料とシステム管理費等を徴収する。また、当該管理センタ２１１は、先に使用されたポイントに応じた著作権料を、図7の（４）のようにコンテンツプロバイダ２４０に対して支払うと共に、仮想店舗２３０に対しては図7の（５）のように店舗手数料を支払う。上記著作権料を受け取ったコンテンツプロバイダ２４０は著作権料を各著作権者に支払い、上記店舗手数料を受け取った仮想店舗２３０は仮想店舗毎の手数料を各仮想店舗に対して支払う。

【００３９】このように、ユーザ側２００から支払われた代金は、前記ポイント使用情報に基づいて、著作権料と店舗手数料とコンテンツ加工手数料と決済手数料とシステム管理手数料とに分配され、上記著作権料はコンテンツプロバイダ２４０に、上記店舗手数料は上記仮想店舗２３０に、コンテンツ加工手数料はシステム管理会社２１０に、決済手数料はシステム管理会社と金融機関２２０に、システム管理手数料はシステム管理会社２１０に支払われる。

【００４０】ここで、本実施の形態のシステム間でのデータ送受、すなわち管理センタ２１１とプレーヤ１との間のデータ送受の際には、データ通信の安全性を確保するために、通信するデータの暗号化及び復号化が行われる。本発明実施の形態では、暗号化及び復号化の方式として共通鍵暗号方式及び公開鍵暗号方式の何れにも対応可能となっている。

【００４１】本発明の実施の形態では、上記ディジタルコンテンツ、上記ポイント使用情報、ポイント情報、メッセージやセキュリティＩＤ、その他の各種情報の伝送の際の暗号方式としては、処理速度の点から共通鍵暗号方式を採用している。これら各種情報の暗号化及び復号化に使用する共通鍵は、それぞれ各情報に対応して異なるものである。前記図２のプレーヤ１では、管理センタ２１１から伝送されてくる暗号化された情報の復号化に使用する共通鍵が前記共通鍵保管メモリ２２に保管され、この共通鍵保管メモリ２２に保管している共通鍵を用いて、前記共通暗号復号回路２４が、上記管理センタ２１１からの暗号化された情報の復号化を行う。

【００４２】一方、上記各種情報の暗号化や復号化に使用する上記共通鍵の伝送の際の暗号方式としては、前記プレーヤ１の固有の鍵であるプレーヤ固有鍵が何れの方式に対応しているかによって採用される暗号方式が変わるものである。すなわち、上記プレーヤ固有鍵が共通鍵暗号方式に対応している場合、上記共通鍵は当該プレーヤ固有鍵を用いて暗号化され、また当該暗号化された共通鍵は上記プレーヤ固有鍵を用いて復号化されることになる。これに対して、上記プレーヤ固有鍵が公開鍵暗号方式に対応している場合、上記共通鍵の暗号化には相手

先の公開鍵が用いられ、暗号化された上記共通鍵の復号化にはそれぞれ復号化を行う側の秘密鍵が用いられる。

【００４３】例えば上記プレーヤ１から管理センタ２１１に上記共通鍵（例えば後述するセッション鍵）が送られる場合において、上記プレーヤ固有鍵が共通鍵暗号方式に対応しているときには、上記プレーヤ１では通信用鍵保管メモリ２１が保管しているプレーヤ固有鍵を用いて上記共通鍵暗号復号回路２４が上記共通鍵を暗号化し、管理センタ２１１では当該管理センタ２１１が保管しているプレーヤ固有鍵を用いて、上記暗号化されてる共通鍵の復号化を行う。同じく、上記プレーヤ１から管理センタ２１１に上記共通鍵が送られる場合において、例えば上記プレーヤ固有鍵が公開鍵暗号方式に対応しているときには、上記プレーヤ１の通信用用鍵保管メモリ２１が保管している管理センタ２１１の公開鍵にて上記公開鍵暗号復号回路２０が上記共通鍵を暗号化し、管理センタ２１１では当該管理センタ２１１が保管している秘密鍵を用いて、上記暗号化されてる共通鍵の復号化を行う。

【００４４】逆に、例えば上記管理センタ２１１からプレーヤ１に上記共通鍵（例えばコンテンツ鍵）が送られる場合において、上記プレーヤ固有鍵が共通鍵暗号方式に対応しているときには、上記管理センタ２１１が保管しているプレーヤ固有鍵にて上記共通鍵が暗号化され、プレーヤ１では上記通信用鍵保管メモリ２１にて保管しているプレーヤ固有鍵を用いて、前記共通暗号復号回路２４が上記暗号化されてる共通鍵の復号化を行う。同じく、上記管理センタ２１１からプレーヤ１に上記共通鍵が送られる場合において、例えば上記プレーヤ固有鍵が公開鍵暗号方式に対応しているときには、上記管理センタ２１１が保管しているプレーヤ１の公開鍵にて上記共通鍵が暗号化され、プレーヤ１では上記通信用鍵保管メモリ２１にて保管しているプレーヤ固有鍵すなわち秘密鍵を用いて、前記公開暗号復号回路２０が上記暗号化されてる共通鍵の復号化を行う。

【００４５】上述したようなプレーヤ固有鍵自身の暗号方式は、当該プレーヤ固有鍵の配送（システム管理会社２１０からプレーヤ１への配送）が容易か否かによって決定されている。すなわち、コスト的には共通鍵暗号方式の方が有利であるので、プレーヤ固有鍵の配送が容易であれば共通鍵暗号方式を採用するが、当該プレーヤ固有鍵の配送が困難であるときにはコスト高であるが公開鍵暗号方式を採用する。プレーヤ固有鍵をハードウェアに実装する場合には共通鍵暗号方式を、ソフトウェアに実装する場合には公開鍵暗号方式を採用する。

【００４６】以下、本発明の実施の形態では、プレーヤ固有鍵自身の暗号方式としてソフトウェアに実装する場合の互換性を考慮して、上記公開鍵暗号方式を採用する例を挙げて説明することにする。すなわち、上記管理センタ２１１とプレーヤ１との間で前記共通鍵の伝送が行われる場合において、上記プレーヤ１側で共通鍵（セッション鍵）が暗号化されるときには管理センタ２１１の公開鍵を用いて暗号化がなされ、管理センタ２１１では上記プレーヤ固有鍵（すなわち秘密鍵）を用いて上記暗号化されてる共通鍵の復号化を行う。逆に、上記管理センタ２１１側で共通鍵（コンテンツ鍵）が暗号化されるときには、プレーヤの公開鍵にて暗号化がなされ、プレーヤ１では上記プレーヤ固有鍵（すなわち秘密鍵）を用いて上記暗号化されてる共通鍵の復号化を行う。

【００４７】前述したような各手順と暗号方式を用いて運用されるシステムを構成する上記プレーヤ１とユーザ端末５０と管理センタ２１１の実際の動作を、以下に順番に説明する。

【００４８】先ず、上述したポイント補充すなわちポイント購入時のプレーヤ１、ユーザ端末５０、管理センタ１０における処理の流れについて、図8から図１１を用い、前記図２及び図３を参照しながら説明する。

【００４９】図8には、ポイントを購入する際のプレーヤ１における処理の流れを示している。

【００５０】この図8において、ステップＳＴ１では、ユーザ端末５０すなわちパーソナルコンピュータに予めインストールされているポイント購入用のソフトウェアの立ち上げが行われ、この間のプレーヤ１のコントローラ１６は、当該ポイント購入用のソフトウェアが立ち上がるまで待っている。

【００５１】上記ポイント購入用のソフトウェアが立ち上がると、ステップＳＴ２にて、プレーヤ１のコントローラ１６は、上記ユーザ端末５０に入力された情報を、当該ユーザ端末５０から受信する。このときのユーザ端末５０に入力される情報とは、上記ポイント購入用のソフトウェアに従って、上記ユーザ端末５０を操作するユーザに対して当該ユーザ端末５０から入力要求がなされるものであり、例えばパスワードや購入したいポイント情報数等の情報である。

【００５２】これらユーザ端末５０からの情報は、プレーヤ１のＰＣ用インターフェース端子３及び当該プレーヤ１内に１チップ化された集積回路１０の端子１２を介して、コントローラ１６に受信される。当該ユーザ端末５０からの情報を受信したコントローラ１６は、ステップＳＴ３にて、当該プレーヤ１の集積回路１０内のパスワード格納メモリ１４が格納するパスワードと、上記受信した情報中のパスワードとの比較を行い、上記受信パスワードが正しいかどうかの確認を行う。

【００５３】上記パスワードが正しいと確認したコントローラ１６は、ステップＳＴ４にて、ポイントを購入したい旨の情報（ポイント購入の主旨）と購入したいポイント情報数その他の情報を生成すると同時に、セキュリティＩＤ発生回路１９からセキュリティＩＤを発生させ、次のステップＳＴ５にてこれらの情報を共通暗号復号回路２４にて暗号化させる。コントローラ１６は、次

にステップＳＴ６にて、ユーザＩＤ格納メモリ２３からユーザＩＤを読み出し、当該ユーザＩＤを上記暗号化した情報に付加し、さらに、ステップＳＴ７にて、当該ユーザＩＤを付加して作成したデータを上記端子１２及びＰＣ用インターフェース端子３を介してユーザ端末５０に転送する。このユーザ端末５０からは、上記作成データが管理センタ２１１に送られることになる。

【００５４】このとき、上記作成データの暗号化には前述したように共通鍵暗号方式が採用されているため、当該作成データの伝送に先立ち、共通鍵の生成が行われる。このため、上記コントローラ１６では、上記共通鍵として、例えば乱数発生手段であるセキュリティＩＤ発生回路１９からセッション鍵を発生させる。また、この共通鍵（セッション鍵）は、上記作成データの伝送に先だって、プレーヤ１から管理センタ２１１に対して送られることになる。ここで、当該共通鍵は前述のように公開鍵暗号方式にて暗号されるものであるため、上記コントローラ１６では、上記共通鍵であるセッション鍵を公開暗号復号回路２０に送ると同時に、通信用鍵保管メモリ２１に予め保管されている管理センタ２１１の公開鍵を取り出して上記公開暗号復号回路２０に送る。これにより当該公開暗号復号回路２０では、上記管理センタ２１１の公開鍵を用いて上記共通鍵（セッション鍵）の暗号化が行われる。このようにして暗号化されたセッション鍵はユーザＩＤと共に、上記作成データの伝送に先だって管理センタ２１１に送られている。

【００５５】なお、前述したように、ポイント情報の要求と共にポイント使用情報の転送も行う場合、コントローラ１６は、ポイント使用情報格納メモリ２９から前記権利者情報等を含むポイント使用情報を読み出し、これらも上記共通暗号復号回路２６に送って暗号化させる。この暗号化したポイント使用情報は、上記作成データと共に伝送される。また、ポイント使用情報の転送と同時に、ポイント情報の残高をも同様にして転送することも可能である。

【００５６】その後、コントローラ１６は、ステップＳＴ８にて、ユーザ端末５０を通して管理センタ２１１から送られてきた暗号化されているデータを受信する。この管理センタ２１１から送られてきたデータは、先に当該プレーヤ１から転送した上記購入したいポイント情報数に応じたポイント情報とセキュリティＩＤ等の情報が、上記セッション鍵と同じ共通鍵を用いて暗号化されたデータである。

【００５７】コントローラ１６は、上記管理センタ２１１からのデータを受信すると、ステップＳＴ９にて、当該データを上記共通暗号復号回路２４に送ると共に、先に発生して共通鍵保管メモリ２２に保管しておいた前記共通鍵を読み出して同じく共通暗号復号回路２４に送る。当該共通暗号復号回路２４では、上記共通鍵を用いて上記管理センタ２１１からの暗号化されたデータを復号化する。

【００５８】次に、上記コントローラ１６は、ステップＳＴ１０にて、上記復号化されたデータのセキュリティＩＤを、上記セキュリティＩＤ発生回路１９からのセキュリティＩＤとの比較によって確認し、その確認後、ステップＳＴ１１にて、上記ポイント情報格納メモリ２８に格納されていたポイント情報を、上記新たに送られてきたポイント情報にて修正する。

【００５９】上記ポイント情報の修正等の処理が終了すると、コントローラ１６は、ステップＳＴ１２にて、処理完了のサインを生成し、上記共通鍵保管メモリ２２から読み出した共通鍵と共に上記共通暗号復号回路２４に送り、当該共通暗号復号回路２４にて暗号化させる。その後、コントローラ１６は、ステップＳＴ１３にて当該暗号化された処理完了のサインを、端子１２及び３を介してユーザ端末５０に転送し、管理センタ２１１に送る。

【００６０】以上により、ポイント購入の際のプレーヤ１における処理の流れが終了する。

【００６１】次に、上記ポイント購入時のユーザ端末５０における処理の流れを、図９を用いて説明する。

【００６２】この図９において、ユーザ端末５０は、ステップＳＴ２１にて、ポイント購入用のソフトウェアの立ち上げを行う。当該ポイント購入用ソフトウェアが立ち上がると、このユーザ端末５０では、ステップＳＴ２２にて、上記ポイント購入用のソフトウェアに従い当該ユーザ端末５０を操作するユーザに対して上述したパスワードや購入したいポイント数等の情報の入力要求を行い、ユーザからこれらの情報が入力されると、当該入力された情報を前記図８のステップＳＴ２のように上記プレーヤ１に転送する。

【００６３】次に、ユーザ端末５０は、ステップＳＴ２３にて、上記プレーヤ１から前記図８のステップＳＴ７のように作成されたデータを受信すると、ステップＳＴ２４にて、当該プレーヤ１から転送されたデータを、予め登録されているアドレスすなわち管理センタ２１１へ転送する。

【００６４】上記データの転送を行った後のユーザ端末５０は、管理センタ２１１からの返送を待ち、管理センタ２１１からのデータ返送があると、ステップＳＴ２５にて当該管理センタ２１１からのデータをそのままプレーヤ１に転送する。

【００６５】当該ユーザ端末５０は、ステップＳＴ２６にて、上記プレーヤ１から前記図８のステップＳＴ１３のように処理完了のサインを受信すると、当該ポイント購入等の処理が終了したことをユーザに知らせるために、ステップＳＴ２７にて処理完了のサインをディスプレイに表示し、ユーザに確認させる。

【００６６】その後、当該ユーザ端末５０は、上記プレーヤ１から送られてきた処理完了のサインの暗号文を管

理センタ２１１に転送する。

【００６７】以上により、ポイント購入の際のユーザ端末５０における処理の流れが終了する。

【００６８】次に、ポイント購入時の管理センタ２１１における処理の流れを、図１０を用いて説明する。

【００６９】この図１０において、管理センタ２１１は、ステップＳＴ３１のように、コントロール機能部１３１にて全体が制御される管理機能ブロック１３０の通信機能部１３３によって、前記図8のステップＳＴ７及び図9のステップＳＴ２４のようにユーザ端末５０を介して転送されたプレーヤ1からの上記暗号化されたデータを受信する。このデータを受信すると、管理センタ２１１のユーザ管理機能ブロック１１０は、ステップＳＴ３２のように、コントロール機能部１１１の制御の元で、当該受信したデータに添付されたユーザＩＤに基づいて、データベース部１１２から共通鍵を入手すると共にセキュリティＩＤ発生機能部１１６からセキュリティＩＤを入手する。

【００７０】なお、この時の共通鍵は、前記プレーヤ1から予め送られてきている前記セッション鍵であり、このセッション鍵は前述のように公開鍵暗号方式にて暗号化されて送られてきたものである。したがって、この暗号化されているセッション鍵の復号時には、当該管理センタ２１１のユーザ管理機能ブロック１１０において、上記管理センタ２１１の公開鍵暗号方式の秘密鍵が取り出され、この秘密鍵と上記暗号化されているセッション鍵とが通信文暗号／復号機能部１１４に送られる。当該通信文暗号／復号機能部１１４では、上記管理センタ２１１の公開鍵を用いて上記暗号化されたセッション鍵の復号化が行われる。このようにして得られたセッション鍵（共通鍵）が上記データベース部１１２に格納されている。

【００７１】上記データベース部１１２から上記ユーザＩＤに対応する共通鍵を入手すると共にセキュリティＩＤ発生機能部１１６からセキュリティＩＤを入手すると、ステップＳＴ３３に示すように、管理センタ２１１のユーザ管理機能ブロック１１０の通信文暗号／復号機能部１１４において、上記共通鍵を用いて、上記プレーヤ1からの上記暗号化されたデータの復号化を行い、さらにコントロール機能部１１１において、当該復号化したデータ中のセキュリティＩＤと上記セキュリティＩＤ発生機能部１１６から読み出したセキュリティＩＤとの比較によって、アクセスしてきたユーザ側２００（プレーヤ１）が正当な使用者であるかどうかの内容確認を行う。

【００７２】上記アクセス元の正当性を確認した管理センタ２１１では、ステップＳＴ３４のように、ユーザ管理機能ブロック１１０のポイント発生機能部１１３にて、上記ユーザ端末５０から送られてきたデータの内容に応じたポイント情報の発行を行い、また、決済請求機能部１１７にて、ユーザの決済機関（金融機関２２０）への請求準備を行う。

【００７３】さらに、管理センタ２１１は、ステップＳＴ３５のように、例えばコントロール機能部１１１において、プレーヤ1からのポイント情報の残高とポイント使用情報に不正が無いことを確認し、後の処理のために情報のまとめを行う。すなわち、ポイント情報の残高と実際に使用したポイント情報の数とから不正な使用がないかどうかの確認とまとめとを行う。なお、この確認とまとめは、必ず行わなければならないものではないが、望ましくは行った方が良い。

【００７４】管理センタ２１１のユーザ管理機能ブロック１１０ではまた、上記ステップＳＴ３５の処理の後、ステップＳＴ３６のように、セキュリティＩＤ発生機能部１１５において上記プレーヤ１（ユーザ）への新たなセキュリティＩＤを例えば乱数発生に基づいて算出し、さらに、例えばコントロール機能部１１０にて、上記セキュリティＩＤを上記ポイント情報と共に暗号化する。このときの暗号化も前記プレーヤ1から予め送られてきている前記セッション鍵（共通鍵）を用いて行う。

【００７５】上記暗号化が終了すると、管理センタ２１１の管理機能ブロック１３０の通信機能部１３３では、コントロール機能部１３１の制御の元、上記暗号化したデータを前記図9のステップＳＴ２５及び図8のステップＳＴ8のようにユーザ端末５０を介してプレーヤ1に転送する。

【００７６】その後、管理センタ２１１の通信機能部１３３において、ステップＳＴ３８のように、前記図9のステップＳＴ２８に示したユーザ端末５０からの処理完了サインを受信して復号化すると、管理センタ２１１のユーザ管理機能ブロック１１０の決済請求機能部１１７では、ステップＳＴ３９のように、当該処理完了サインに基づいて金融機関２２０に決済を請求する。この金融機関２２０に対する決済請求は、管理機能ブロック１３０の通信機能部１３２から行われる。

【００７７】以上により、ポイント購入の際の管理センタ２１１における処理の流れが終了する。

【００７８】上述した図8から図１０の処理の流れにおけるプレーヤ1とユーザ端末５０と管理センタ２１１との間の情報送受のシーケンスは、図１１に示すように表すことができる。

【００７９】すなわちこの図１１において、入力情報転送Ｔ１では、前記図8のステップＳＴ２及び図9のステップＳＴ２２のように、ユーザ端末５０からプレーヤ１に対して、前記パスワードやポイント数等の入力情報が転送される。

【００８０】作成データ転送Ｔ２では、前記図8のステップＳＴ7及び図9のステップＳＴ２３のように、プレーヤ1からユーザ端末５０に対して、前記プレーヤ1にて作成したデータが転送される。また、データ転送Ｔ３

では、前記図９のステップＳＴ２４及び図１０のステップＳＴ３１のように、ユーザ端末５０から管理センタ２１１に対して、前記プレーヤ１が作成したデータが転送される。

【００８１】データ転送Ｔ４では、前記図１０のステップＳＴ３７及び図９のステップＳＴ２５のように、管理センタ２１１からユーザ端末５０に対して、管理センタ２１１にて暗号化したデータが転送される。また、転送Ｔ５では、前記図９のステップＳＴ２５及び図８のステップＳＴ８のように、管理センタ２１１からのデータをユーザ端末５０がそのままプレーヤ１に転送される。

【００８２】処理完了サイン転送Ｔ６では、前記図８のステップＳＴ１３及び図９のステップＳＴ２６のように、プレーヤ１からの処理完了サインがユーザ端末５０に転送される。さらに、処理完了サイン暗号文転送では、前記図９のステップＳＴ２８及び図１０のステップＳＴ３８のように、プレーヤ１からの暗号化された処理完了サインが管理センタ２１１に転送される。

【００８３】次に、上述したディジタルコンテンツの入手時のプレーヤ１、ユーザ端末５０、管理センタ２１１における処理の流れについて、図２及び図３を参照しながら、図１２から図１５を用いて説明する。

【００８４】図１２には、ディジタルコンテンツの入手時のプレーヤ１における処理の流れを示している。

【００８５】この図１２において、コントローラ１６は、ステップＳＴ４１のように、ユーザ端末５０すなわちパーソナルコンピュータに予めインストールされているディジタルコンテンツ入手用のソフトウェアの立ち上げが行われるまで待っている。

【００８６】上記ディジタルコンテンツ入手用のソフトウェアが立ち上がると、コントローラ１６は、ステップＳＴ４２のように、ユーザ端末５０を介して管理センタ２１１からディジタルコンテンツを含むデータを受信する。このときユーザ端末５０から端子３及び１２を介して受信するデータは、前述したようにコンテンツ鍵（コンテンツ毎に異なる共通鍵）で暗号化されたディジタルコンテンツと、当該ディジタルコンテンツに対応するコンテンツＩＤとを少なくとも有してなる。したがって、この暗号化されたディジタルコンテンツを使用するには、コンテンツ鍵を管理センタ２１１から入手しなけらばならない。このコンテンツ鍵の入手の方法については後述する。

【００８７】このユーザ端末５０からのデータを受信したコントローラ１６は、このデータすなわち暗号化されたディジタルコンテンツを、集積回路１０の端子１１を介し、記憶メディア用Ｉ／Ｏ端子４に接続されている記憶メディアに格納する。なお、この記憶メディアとしては、書き換え可能な光ディスクや半導体メモリ等の各種の記憶媒体が考えられるが、ランダムアクセス可能なものが望ましい。

【００８８】以上により、ディジタルコンテンツの入手時のプレーヤ１における処理の流れが終了する。

【００８９】次に、ディジタルコンテンツの入手時のユーザ端末５０における処理の流れを、図１３を用いて説明する。

【００９０】この図１３において、ユーザ端末５０は、ステップＳＴ５１にて、ディジタルコンテンツ入手用のソフトウェアの立ち上げを行う。当該ソフトウェアが立ち上がると、このユーザ端末５０では、ステップＳＴ５２にて、上記ディジタルコンテンツ入手用のソフトウェアに従い、予め登録されているアドレスの管理センタ２１１にアクセスする。

【００９１】このとき、当該管理センタ２１１は、前記仮想店舗２３０を用いて複数のディジタルコンテンツを展示している。ユーザ端末５０からは、ステップＳＴ５３にて、この仮想店舗２３０に展示されている複数のディジタルコンテンツのなかから、ユーザの選択操作に応じた所望のディジタルコンテンツが指定される。すなわち、ユーザ端末５０は、ステップＳＴ５４のように、仮想店舗２３０に展示されたディジタルコンテンツの中の所望のディジタルコンテンツを指定するためのコンテンツの指定情報を管理センタ２１１に送信する。

【００９２】ステップＳＴ５５のように、上記コンテンツ指定情報に応じて管理センタ２１１から返送されたデータ、すなわち前記暗号化されたディジタルコンテンツ及びコンテンツＩＤからなるデータを受信すると、当該ユーザ端末５０は、ステップＳＴ５６のように、内部の例えばハードディスクやメモリ等の格納手段に上記データを一旦格納する。

【００９３】その後、ユーザ端末５０は、当該格納したデータ（暗号化されたディジタルコンテンツ及びコンテンツＩＤ）を、前記図１２のステップＳＴ４２のようにプレーヤ１に転送する。

【００９４】以上により、ディジタルコンテンツの入手時のユーザ端末５０における処理の流れが終了する。

【００９５】次に、ディジタルコンテンツ入手時の管理センタ２１１における処理の流れを、図１４を用いて説明する。

【００９６】ここで、図３に示す管理センタ２１１は、前述した仮想店舗２３０に複数のコンテンツを展示させている。具体的には、管理センタ２１１ののコンテンツ管理機能ブロック１００において、前記仮想店舗２３０を生成しており、この仮想店舗２３０に上記複数のディジタルコンテンツの展示を行っている。

【００９７】このように仮想店舗２３０にディジタルコンテンツを展示している状態で、図１４のステップＳＴ６１のように、前記図１３のステップＳＴ５４にてユーザ端末５０からコンテンツ指定情報を受信する。

【００９８】当該ユーザ端末５０から上記コンテンツ指定情報を受信すると、コンテンツ管理機能ブロック１０

０のコントロール機能部１０１は、このコンテンツ指定情報を管理機能ブロック１３０に送る。管理機能ブロック１３０のコントロール機能部１３１は、上記コントロール管理機能ブロック１００から受け取ったコンテンツ指定情報を、権利者用の通信機能部１３４を通して、前記コンテンツプロバイダ２４０に転送する。これにより当該コンテンツプロバイダ２４０からは、上記コンテンツ指定情報にて要求されたディジタルコンテンツが転送されてくる。上記コンテンツプロバイダ２４０から入手したディジタルコンテンツは、管理機能ブロック１３０からコンテンツ管理機能ブロック１００に送られ、このコンテンツ暗号・圧縮化機能部１０４に入力される。このとき、コントロール機能部１０１は、コンテンツ鍵・ＩＤ発生機能部１０３にて発生されてデータベース１０２に格納されているコンテンツ鍵を、上記コンテンツ暗号・圧縮化機能部１０４に送る。このコンテンツ暗号・圧縮化機能部１０４では、上記ディジタルコンテンツに対して上記コンテンツ鍵を用いた暗号化を施し、さらに所定の圧縮処理を施す。コントロール機能部１０１は、上記暗号化及び圧縮処理されたディジタルコンテンツに対して、データベース１０２から取り出したコンテンツＩＤを付加し、管理機能ブロック１３０に送る。なお、ディジタルコンテンツがオーディオ信号である場合の所定の圧縮処理としては、例えば近年製品化されているいわゆるＭＤ（ミニディスク：商標）にて使用されている技術である、いわゆるＡＴＲＡＣ（Adaptive TRansform Acoustic Coding）のように、人間の聴覚特性を考慮してオーディオデータを高能率圧縮する処理を一例とした挙げることができる。

【００９９】その後、図１４のステップＳＴ６２に示すように、管理機能ブロック１３０のコントロール部１３１は、ユーザ端末との通信機能部１３３を通して、上記暗号化及び圧縮処理されてコンテンツＩＤが付加されたディジタルコンテンツを、上記ユーザ端末５０に送信する。

【０１００】ディジタルコンテンツ入手時の管理センタ２１１における処理の流れは以上である。

【０１０１】上述した図１２から図１４の処理の流れにおけるプレーヤ１とユーザ端末５０と管理センタ２１１との間の情報送受のシーケンスは、図１５に示すように表すことができる。

【０１０２】すなわちこの図１５において、入力情報転送Ｔ１１では、前記図１３のステップＳＴ５４のように、ユーザ端末５０から管理センタ２１１に対して、前記コンテンツ指定情報が転送される。コンテンツ転送Ｔ１２では、管理センタ２１１から、前記図１４のステップＳＴ６２のように、暗号化されたディジタルコンテンツとコンテンツＩＤがユーザ端末５０に転送される。

【０１０３】コンテンツ転送Ｔ１３では、前記図１３のステップＳＴ５７及び図１２のステップＳＴ４２のように、ユーザ端末５０に一旦格納された上記暗号化されたディジタルコンテンツとコンテンツＩＤがプレーヤ１に転送される。

【０１０４】次に、上述したディジタルコンテンツを使用する際に必要となるコンテンツ鍵とその使用条件の入手時のプレーヤ１、ユーザ端末５０、管理センタ２１１における処理の流れについて、図２及び図３を参照しながら、図１６から図１９を用いて説明する。

【０１０５】図１６には、コンテンツ鍵及び使用条件の入手時のプレーヤ１における処理の流れを示している。

【０１０６】この図１６のステップＳＴ７１では、プレーヤ１のコントローラ１６において、ユーザ端末５０に予めインストールされているコンテンツ鍵及び使用条件入手用のソフトウェアの立ち上げが行われるまで待っている。

【０１０７】上記ユーザ端末５０の上記コンテンツ鍵及び使用条件入手用のソフトウェアが立ち上がると、当該ソフトウェアに従ってユーザ端末５０に入力された情報が、ステップＳＴ７２のように、前記ＰＣ用インターフェース端子３及び集積回路１０の端子１２を介して受信される。このときの上記ユーザ端末５０から供給される入力情報は、鑑賞したいディジタルコンテンツの暗号化を解くのに必要なコンテンツ鍵を要求するための情報である。なお、この例では、上記コンテンツ鍵の要求情報として、このコンテンツ鍵を使用するディジタルコンテンツの指定情報を用いている。

【０１０８】このコンテンツ指定情報を上記ユーザ端末５０から受信したコントローラ１６は、ステップＳＴ７３にて、当該コンテンツ指定情報にて指定されたディジタルコンテンツのＩＤと、セキュリティＩＤ発生回路１９からのセキュリティＩＤとを作成し、この作成したデータを共通暗号復号回路２４にて暗号化させる。また、コントローラ１６は、当該作成したデータにユーザＩＤ格納メモリ２３から読み出したユーザＩＤを付加し、上記端子１２及びＰＣ用インターフェース端子３を介してユーザ端末５０に転送する。このユーザ端末５０からは、上記作成データが管理センタ２１１に送られることになる。

【０１０９】このときの作成データの暗号化にも、前述したように共通鍵暗号方式が採用されているため、当該作成データの伝送に先立ち、共通鍵の生成が行われる。このため、上記コントローラ１６では、上記共通鍵として、例えば乱数発生手段であるセキュリティＩＤ発生回路１９からセッション鍵を発生させる。また、この共通鍵（セッション鍵）は、上記作成データの伝送に先だって、プレーヤ１から管理センタ２１１に対して送られることになる。当該共通鍵は、前述のように公開鍵暗号方式にて暗号されるものであるため、上記コントローラ１６では、上記共通鍵であるセッション鍵を公開暗号復号回路２０に送ると同時に、通信用鍵保管メモリ２１に予

め保管されている管理センタ２１１の公開鍵を取り出して上記公開暗号復号回路２０に送る。これにより当該公開暗号復号回路２０では、上記管理センタ２１１の公開鍵を用いて上記共通鍵（セッション鍵）の暗号化が行われる。このようにして暗号化されたセッション鍵が、上記作成データの伝送に先だって管理センタ２１１に送られている。

【０１１０】その後、コントローラ１６は、ステップＳＴ７５にて、後述するようにユーザ端末５０を介して管理センタ２１１から送付されてきた暗号化されたデータを受信する。このときの管理センタ２１１から送られてきたデータは、後述するように上記コンテンツ鍵と使用条件とセキュリティＩＤ等が暗号化されたものである。

【０１１１】上記管理センタ２１１からの暗号化されたデータを受信すると、プレーヤ１では、ステップＳＴ７６のように、上記暗号化されたデータを復号化すると共にそのデータの正当性の確認を行う。すなわち、コントローラ１６は、上記復号化されたデータのセキュリティＩＤを、上記セキュリティＩＤ発生回路１９からのセキュリティＩＤとの比較によって確認することによる正当性の評価を行う。

【０１１２】ここで、コンテンツ鍵については後述するように公開鍵暗号方式にて暗号化がなされ、使用条件及びセキュリティＩＤについては共通鍵暗号方式にて暗号化がなされている。したがって、当該暗号化されているコンテンツ鍵を復号化するには、公開鍵暗号方式の秘密鍵が必要であり、本実施の形態のプレーヤ１では前述したようにプレーヤ固有鍵を秘密鍵として使用することにしているので、当該プレーヤ固有鍵が通信用鍵保管メモリ２１から取り出される。このプレーヤ固有鍵は、上記暗号化されたコンテンツ鍵と共に公開暗号復号回路２０に送られる。この公開暗号復号回路２０では、上記暗号化されているコンテンツ鍵を上記プレーヤ固有鍵を用いて復号化する。このように復号化されたコンテンツ鍵は、共通鍵保管メモリ２２に保管される。一方、上記共通鍵暗号方式にて暗号化されている使用条件とセキュリティＩＤを復号化する場合には、これらのデータを上記共通暗号復号回路２４に送ると共に、先に発生して共通鍵保管メモリ２２に保管しておいた前記共通鍵を読み出して同じく共通暗号復号回路２４に送る。当該共通暗号復号回路２４では、上記共通鍵を用いて上記使用条件とセキュリティＩＤを復号化する。このように復号化された使用条件は、ポイント使用情報格納メモリ２９に格納される。なお、ここで重要なのは、当該復号化されたコンテンツ鍵・使用条件は、当該プレーヤ１の外部、具体的には図２の集積回路１０内に設けられたコントローラ１６や共通鍵保管メモリ２２、ポイント使用情報格納メモリ２９から外部には取り出されないことである。

【０１１３】上記正当性の確認後、コントローラ１６は、ステップＳＴ７７のように、上記復号したコンテンツ鍵を上記コンテンツＩＤと共に上記共通鍵保管メモリ２２に格納させる。

【０１１４】その後、コントローラ１６は、ステップＳＴ７８にて、上記コンテンツ鍵を入手した旨を示すメッセージを作成し、このメッセージを前述同様に共通鍵暗号復号回路２４に送り、予め発生して共通鍵保管メモリ２２に保管しておいた前記共通鍵を読み出して同じく共通暗号復号回路２４に送る。当該共通暗号復号回路２４では、上記共通鍵を用いてメッセージを暗号化する。

【０１１５】当該メッセージの暗号化が終了すると、コントローラ１６は、ステップＳＴ７９のように、この暗号化されたメッセージを端子１２及び３を介してユーザ端末５０に送信する。この暗号化されたメッセージは、その後、管理センタ２１１に転送させる。

【０１１６】以上により、コンテンツ鍵・使用条件入手時のプレーヤ１における処理の流れが終了する。

【０１１７】次に、コンテンツ鍵・使用条件入手時のユーザ端末５０における処理の流れを、図１７を用いて説明する。

【０１１８】この図１７において、ユーザ端末５０は、ステップＳＴ８１にて、コンテンツ鍵・使用条件入手用のソフトウェアの立ち上げを行う。当該ソフトウェアが立ち上がると、このユーザ端末５０では、ステップＳＴ８２にて、上記ソフトウェアに従い当該ユーザ端末５０を操作するユーザに対して、希望のコンテンツの指定入力要求を行い、ユーザからコンテンツの指定がなされると、その指定情報を生成する。ユーザ端末５０は、上記ステップＳＴ８３にて、上記コンテンツの指定情報をプレーヤ１に対して送信する。

【０１１９】次に、ユーザ端末５０は、ステップＳＴ８４にて、前記図１６のステップＳＴ７４のように上記プレーヤ１にて作成されて転送されたデータを受信すると、ステップＳＴ８５にて、当該プレーヤ１から転送されたデータを、予めアドレスが登録されている管理センタ２１１へ転送する。

【０１２０】上記管理センタ２１１に対してデータの転送を行った後のユーザ端末５０は、管理センタ２１１からの返送を待ち、ステップＳＴ８６にて、管理センタ２１１から上記コンテンツＩＤで指定されたコンテンツ鍵・使用条件とセキュリティＩＤ等が暗号化されたデータの返送があると、ステップＳＴ８７にて当該管理センタ２１１からのデータをそのままプレーヤ１に転送する。

【０１２１】上記プレーヤ１に対してデータの転送を行った後のユーザ端末５０は、プレーヤ１からの返送を待ち、ステップＳＴ８８にて、プレーヤ１から前記図１６のステップＳＴ７９のように、上記コンテンツ鍵を入手した旨の暗号化されたメッセージの返送があると、ステップＳＴ８９にて当該ユーザ端末５０に接続されたディスプレイ装置に対して上記コンテンツ鍵入手が完了した旨の表示を行ってユーザに知らせる。

【０１２２】その後、上記プレーヤ１から返送された上記暗号化されたメッセージを、ステップＳＴ９０にて、管理センタ２１１に送付する。

【０１２３】以上により、コンテンツ鍵・使用条件入手時のユーザ端末５０における処理の流れが終了する。

【０１２４】次に、コンテンツ鍵・使用条件入手時の管理センタ２１１における処理の流れを、図１８を用いて説明する。

【０１２５】この図１８において、管理センタ２１１のユーザ端末との通信機能部１３３は、ステップＳＴ９１にて、前記図１６のステップＳＴ７４及び図１７のステップＳＴ８５のようにユーザ端末５０にてを介してプレーヤ１から送信されてきたコンテンツＩＤ，ユーザＩＤ、メッセージ、セキュリティＩＤの暗号化データを受信する。この受信したデータは、ユーザ管理機能ブロック１１０に送られる。

【０１２６】当該ユーザ管理機能ブロック１１０のコントロール機能部１１１は、上記受信した暗号化データに付加されたユーザＩＤに基づいて、当該暗号化を解くための共通鍵をデータベース部１１２から取り出し、通信文暗号・復号機能部１１４ではこの共通鍵を用いて上記暗号化データを復号する。また、コントロール機能部１１１は、データベース部１１２から読み出したユーザＩＤとセキュリティＩＤ発生機能部１１６からのセキュリティＩＤとを用いて、上記受信して復号化したデータの正当性を確認する。

【０１２７】なお、この時の共通鍵は、前記プレーヤ１から予め送られてきている前記セッション鍵であり、このセッション鍵は前述のように公開鍵暗号方式にて暗号化されて送られてきたものである。したがって、この暗号化されているセッション鍵の復号時には、前述同様に当該管理センタ２１１において、上記管理センタ２１１の公開鍵暗号方式の秘密鍵が取り出され、当該通信文暗号／復号機能部１１４にて上記暗号化されているセッション鍵が当該秘密鍵を用いて復号化される。このようにして得られたセッション鍵（共通鍵）が上記データベース部１１２に格納されている。

【０１２８】上記受信したデータの正当性を確認すると、コントロール機能部１１１は、コンテンツ管理機能ブロック１００に対して上記コンテンツＩＤにて指定されたコンテンツ鍵と使用条件を要求し、当該要求を受けたコンテンツ管理機能ブロック１００のコントロール機能部１０１は、上記コンテンツＩＤにて指定されたコンテンツ鍵と使用条件とをデータベース部１０２から読み出してユーザ管理機能ブロック１１０に転送する。コントロール機能部１１１は、ステップＳＴ９３に示すように、これらコンテンツ鍵と使用条件はセキュリティＩＤと共に通信文暗号／復号機能部１１４に送る。

【０１２９】ここで、コンテンツ鍵については前述した公開鍵暗号方式にて暗号化がなされ、使用条件及びセキュリティＩＤについては前述した共通鍵暗号方式にて暗号化がなされる。したがって、当該コンテンツ鍵を暗号化する時には、前記データベース部１１２からユーザ側２００の公開鍵（プレーヤ１に対応して予め格納されている公開鍵）が上記ユーザＩＤに基づいて取り出されて通信文暗号／復号機能部１１４に送られる。当該通信文暗号／復号機能部１１４では、上記公開鍵を用いて上記コンテンツ鍵を暗号化する。一方、上記使用条件及びセキュリティＩＤを暗号化する時には、上記データベース部１１２から上記ユーザＩＤで指定された共通鍵（セッション鍵）が取り出されて通信文暗号／復号機能部１１４に送られる。このときの通信文暗号／復号機能部１１４では、上記使用条件及びセキュリティＩＤを上記共通鍵を用いて暗号化する。

【０１３０】上記暗号化されたコンテンツ鍵と使用条件及びセキュリティＩＤは、管理機能ブロック１３０に送られ、ステップＳＴ９４のように、ユーザ端末との通信機能部１３３からユーザ端末５０に送信される。このユーザ端末５０に送信されたデータは、前記図１７のステップＳＴ８７及び図１６のステップＳＴ７５のようにユーザ端末５０を介してプレーヤ１に送付されることになる。

【０１３１】その後、管理センタ２１１は、前記図１６のステップＳＴ７９及び図１７のステップＳＴ９０のようにプレーヤ１にて生成されてユーザ端末５０を介して送信された暗号化メッセージの受信を待ち、ステップＳＴ９５のように上記通信機能部１３３が上記プレーヤ１が生成した暗号化メッセージを受信すると、当該管理センタ２１１は、ステップＳＴ９６のように、当該暗号化メッセージを共通鍵で復号化し、その復号メッセージから上記プレーヤ１がコンテンツ鍵と使用条件を入手したことを確認する。

【０１３２】以上により、コンテンツ鍵・使用条件入手時の管理センタ２１１における処理の流れが終了する。

【０１３３】上述した図１６から図１８の処理の流れにおけるプレーヤ１とユーザ端末５０と管理センタ２１１との間の情報送受のシーケンスは、図１９に示すように表すことができる。

【０１３４】すなわちこの図１９において、コンテンツ指定情報転送Ｔ２１では、前記図１７のステップＳＴ８３のように、ユーザ端末５０からプレーヤ１に対して、前記コンテンツ指定情報が転送される。作成データ転送Ｔ２２では、前記のステップＳＴ７４のように、プレーヤ１にて作成されたデータがユーザ端末５０に転送される。作成データ転送Ｔ２３では、当該ユーザ端末５０から上記プレーヤ１にて作成されたデータが管理センタ２１１に転送される。暗号化されたデータ送付Ｔ２４では、前記図１８のステップＳＴ９４のように、管理センタ２１１にて暗号化されたデータがユーザ端末５０に送付され、さらに、暗号化されたデータ送付Ｔ２５では、

当該暗号化されたデータがプレーヤ１に送付される。

【０１３５】メッセージ転送Ｔ２６では、前記図１６のステップＳＴ７９のように、コンテンツ鍵入手完了を示すメッセージを暗号化したデータがプレーヤ１からユーザ端末５０に転送され、さらに暗号化されたデータ送付Ｔ２７では、上記プレーヤ１からの暗号化されたメッセージが、ユーザ端末５０から管理センタ２１１に送付される。

【０１３６】次に、上述したようにしてポイント情報とディジタルコンテンツとコンテンツ鍵とを受け取ったプレーヤ１において、ユーザ端末５０を用いてディジタルコンテンツを実際に鑑賞する際の処理の流れについて、図２を参照しながら図２０を用いて説明する。

【０１３７】ここで、プレーヤ１の端子４には、前記ディジタルコンテンツが記憶された記憶メディアが接続されているとする。

【０１３８】この状態で、ステップＳＴ１０１のように、当該プレーヤ１に対して、ユーザ端末５０から鑑賞を希望するディジタルコンテンツが指定される。このとき、当該指定は、例えばユーザ端末５０をユーザが操作することによりなされる。

【０１３９】このとき、プレーヤ１のコントローラ１６は、ステップＳＴ１０２のように、上記ユーザ端末５０からのコンテンツ指定情報に応じて、上記記憶メディアに対するアクセスを行い、コンテンツのＩＤを読み取る。

【０１４０】上記コントローラ１６は、ステップＳＴ１０３のように、上記記憶メディアから読み取ったコンテンツＩＤに基づき、前記共通鍵保管メモリ２２に対してアクセスを行い、コンテンツ鍵が格納されているかどうかを確認すると共に、前記ポイント使用情報格納メモリ２９に対してアクセスを行い、使用条件が格納されているかどうかを確認する。

【０１４１】ここで、上記共通鍵保管メモリ２２やポイント使用情報格納メモリ２９内に、上記コンテンツ鍵と使用条件が格納されていないことを確認したとき、コントローラ１６は、ユーザ端末５０に対して当該コンテンツ鍵等が存在しない旨の情報を送り、これによりユーザ端末５０からは上記コンテンツ鍵等の入手を促すメッセージを前記ディスプレイ装置に表示する。この場合は、前述したコンテンツ鍵入手用のフローチャートのようにしてコンテンツ鍵等を入手する。このように、新たにコンテンツ鍵等を入手した場合には、ステップＳＴ１０４にて前述したように、その暗号化されているコンテンツ鍵等を復号化する。

【０１４２】次に、コントローラ１６は、ステップＳＴ１０５に示すように、上記復号化された使用条件を元に、ポイント情報格納メモリ２８に格納されているポイント情報の残高が足りているかどうかを確認する。上記ポイント情報格納メモリ２８に格納された上記ポイント情報の残高が足りないときには、コントローラ１６からユーザ端末５０に対して当該ポイント情報の残高が足りない旨の情報が送られ、これによりユーザ端末５０は、上記ポイント情報の入手を促すメッセージを前記ディスプレイ装置に表示する。この場合、前述したようなポイント情報入手用のフローチャートのようにしてポイント情報を入手する。

【０１４３】ここで、実際にディジタルコンテンツの鑑賞を行うとき、コントローラ１６は、ステップＳＴ１０６のように、当該鑑賞するディジタルコンテンツに応じて上記ポイント情報格納メモリ２８からポイント情報数を減額し、さらに当該ポイント情報の使用状態に応じた新たなポイント使用情報を、ポイント使用情報格納メモリ２９に格納する（ポイント使用情報の更新を行う）。このようにポイント使用情報格納メモリ２９に対して新たに格納されるポイント使用情報としては、上記鑑賞したディジタルコンテンツに対応する権利者情報（著作権者等）と減額されたポイント情報数の情報その他の情報などである。

【０１４４】その後、コントローラ１６は、ステップＳＴ１０７のように、これらポイント情報の減額やポイント使用情報の新たな格納等の課金用処理が完了したことを確認すると、記憶メディアからディジタルコンテンツを読み出す。

【０１４５】この記憶メディアから読み出されたディジタルコンテンツは暗号化されているため、コントローラ１６は、ステップＳＴ１０９のように、上記暗号化されたディジタルコンテンツを共通暗号復号回路２４に転送する。

【０１４６】この共通暗号復号回路２４では、ステップＳＴ１１０のように、コントローラ１６からの指示に基づいて、先に復号化して共通鍵保管メモリ２２に保管されているコンテンツ鍵を用いて、上記暗号化されているディジタルコンテンツの復号化を行う。

【０１４７】また、このディジタルコンテンツは前述したように所定の圧縮処理がなされているため、コントローラ１６は、ステップＳＴ１１１のように、上記暗号が復号化された上記圧縮処理されているディジタルコンテンツを、上記共通暗号復号回路２４から伸長回路２６に転送させ、ここで上記所定の圧縮処理に対応する伸長処理を行わせる。

【０１４８】その後、当該伸長されたディジタルコンテンツは、ステップＳＴ１１２のように、Ｄ／Ａ変換回路２７にてアナログ信号に変換され、ステップＳＴ１１３のように、集積回路１０の端子１３と当該プレーヤ１のアナログ出力端子２とを介して外部（例えばユーザ端末５０等）に出力される。

【０１４９】以上により、コンテンツ鑑賞時のプレーヤ１における処理の流れが終了し、ユーザはディジタルコンテンツの鑑賞が可能となる。

【０１５０】次に、上述したようなディジタルコンテンツの鑑賞に伴って前記プレーヤ１のポイント使用情報格納メディア２９に新たに格納されたポイント使用情報を、管理センタ２１１に返却する際の、プレーヤ１、ユーザ端末５０、管センタ３１０における処理の流れについて、図２と図３を参照しながら、図２１から図２４を用いて説明する。

【０１５１】図２１には、ポイント使用情報返却時のプレーヤ１における処理の流れを示している。

【０１５２】この図２１において、コントローラ１６は、ステップＳＴ１２１に示すように、ユーザ端末５０に予めインストールされているポイント使用情報返却用のソフトウェアの立ち上げが行われるまで待つ。

【０１５３】上記ユーザ端末５０の上記ポイント使用情報返却用のソフトウェアが立ち上がると、当該ソフトウェアに従ってユーザ端末５０に入力された情報が、ステップＳＴ１２２のように、前記ＰＣ用インターフェース端子３及び集積回路１０の端子１２を介して受信される。このときの上記ユーザ端末５０から供給される入力情報は、ユーザにより入力されるパスワード等である。

【０１５４】このコンテンツ指定情報を上記ユーザ端末５０から受信したコントローラ１６は、ステップＳＴ１２３にて、当該ユーザ端末５０から供給されたパスワードと、パスワード格納メモリ１４に格納されているパスワードとを比較して、当該パスワードが正しいどうかの確認をする。

【０１５５】上記パスワードの確認において正しいパスワードであると確認されたとき、コントローラ１６は、ステップＳＴ１２４のように、ポイント情報格納メモリ２８に格納されているポイント情報の残高と、ポイント使用情報格納メモリ２９に格納されているポイント使用情報とをそれぞれ読み出し、これら情報を暗号化する。

【０１５６】上記ポイント情報の残高とポイント使用情報の暗号化が終了すると、コントローラ１６は、ステップＳＴ１２５のように、ユーザＩＤ格納メモリ２３からユーザＩＤを読み出して上記暗号化したデータに添付する。

【０１５７】このユーザＩＤが添付されたデータは、ステップＳＴ１２６のように、コントローラ１６から端子１２及びＰＣ用インターフェース端子３を介してユーザ端末５０に転送される。このデータはその後管理センタ２１１に転送される。

【０１５８】なお、このときの暗号化にも前述したように共通鍵暗号方式が採用されている。すなわち、当該データの伝送に先立ち、前述同様に共通鍵の生成が行われ、この生成された共通鍵が前記公開鍵暗号方式にて暗号化（管理センタ２１１の公開鍵を用いた暗号化）され、ユーザＩＤと共に管理センタ２１１に送られている。

【０１５９】上述のようにしてユーザ端末５０にデータを転送した後、コントローラ１６は、上記管理センタ２１１から後述するデータがユーザ端末５０を介して転送されてくるのを待つ。

【０１６０】ここで、ステップＳＴ１２７のように上記管理センタ２１１からのデータを受信すると、プレーヤ１では、ステップＳＴ１２７のように、共通鍵暗号方式を使用して暗号化されている受信データを、前述同様に共通鍵を用いて復号化すると共にそのデータの正当性の確認を行う。すなわち、コントローラ１６は、上記復号化されたデータのセキュリティＩＤを、上記セキュリティＩＤ発生回路１９からのセキュリティＩＤとの比較によって確認することによる正当性の評価を行う。

【０１６１】また、上記管理センタ２１１から転送されてくるデータには、上記共通鍵を用いて暗号化された処理完了のメッセージも含まれている。したがって、上記セキュリティＩＤの確認が終了した後のコントローラ１６は、上記暗号化された処理完了メッセージを共通暗号復号回路２４に送り、ここで共通鍵を用いた復号化を行わせ、この復号化した処理完了メッセージを受け取ることで、上記管理センタ２１１での処理が完了したことを確認する。

【０１６２】以上により、ポイント使用情報返却時のプレーヤ１における処理の流れが終了する。

【０１６３】次に、ポイント使用情報返却時のユーザ端末５０における処理の流れを、図２２を用いて説明する。

【０１６４】この図２２において、ユーザ端末５０は、ステップＳＴ１３１にて、ポイント使用情報返却用のソフトウェアの立ち上げを行う。当該ソフトウェアが立ち上がると、このユーザ端末５０では、ステップＳＴ１３２にて、上記ソフトウェアに従い当該ユーザ端末５０を操作するユーザに対して、パスワード等の入力要求を行い、ユーザからパスワードの入力がなされると、そのパスワードをプレーヤ１に転送する。

【０１６５】次に、ユーザ端末５０は、ステップＳＴ１３３にて、前記図２１のステップＳＴ１２６のように上記プレーヤ１にて作成されて転送されたデータを受信すると、ステップＳＴ１３４にて、当該プレーヤ１から転送されたデータを、予めアドレスが登録されている管理センタ２１１へ転送する。

【０１６６】上記管理センタ２１１に対してデータの転送を行った後のユーザ端末５０は、管理センタ２１１からの返送を待ち、ステップＳＴ１３５にて、管理センタ２１１からプレーヤ１に対して送られるデータを受信すると、当該データをそのままプレーヤ１に転送する。

【０１６７】上記プレーヤ１に対してデータの転送を行った後のユーザ端末５０は、処理が完了した旨をユーザに知らしめるための表示をディスプレイ装置に行い、ユーザからの確認を受ける。

【０１６８】以上により、ポイント使用情報返却時のユ

ーザ端末５０における処理の流れが終了する。

【０１６９】次に、ポイント使用情報返却時の管理センタ２１１における処理の流れを、図２３を用いて説明する。

【０１７０】管理センタ２１１のユーザ端末との通信機能部１３３において、ステップＳＴ１４１のように、前記図２１のステップＳＴ１２６及び図２２のステップＳＴ１３４によって前記ユーザ端末５０を介してプレーヤ１から送信されてきたポイント使用情報等のデータを受信する。

【０１７１】このデータを受信すると、管理センタ２１１のユーザ管理機能ブロック１１０は、ステップＳＴ１４２のように、コントロール機能部１１１の制御の元で、当該受信したデータに添付されたユーザＩＤに基づいて、データベース部１１２から前述同様に予め受け取って格納している共通鍵を入手すると共にセキュリティＩＤを入手する。

【０１７２】上記データベース部１１２から上記ユーザＩＤに対応する共通鍵とセキュリティＩＤを入手すると、ステップＳＴ１４３に示すように、管理センタ２１１のユーザ管理機能ブロック１１０の通信文暗号／復号機能部１１４において、上記共通鍵を用いて、上記プレーヤ１からの上記暗号化されたポイント使用情報等のデータの復号化を行い、さらにコントロール機能部１１１において、当該復号化したデータ中のセキュリティＩＤと上記データベース部１１２から読み出したセキュリティＩＤとの比較によって、アクセスしてきたユーザ側２００（プレーヤ１）が正当な使用者であるかどうかの内容確認を行う。

【０１７３】上記正当性と内容の確認後のデータは、使用情報管理機能ブロック１２０に転送される。この使用情報管理機能ブロック１２０のコントロール機能部１２１は、ステップＳＴ１４４に示すように、上記プレーヤ１から送られてきたポイント情報の残高とポイント使用情報とを用い、データベース部１２２に格納されている情報を用いて上記ユーザ側２００の使用に不正がないかどうかの確認を行う。同時に、当該不正なきことを確認した場合には、使用情報演算機能部１２３においてポイント情報の残高とポイント使用情報をまとめる演算を行う。

【０１７４】その後、ステップＳＴ１４５に示すように、ユーザ管理機能ブロック１１０のコントロール機能部１１１は、セキュリティＩＤ発生機能部１１６を制御してセキュリティＩＤを算出させ、さらに確認メッセージ発生機能部１１５を制御して処理完了のメッセージを生成させる。これらセキュリティＩＤと処理完了メッセージは、ユーザ管理機能ブロック１１０の通信文暗号／復号機能部１１４にて前記共通鍵を用いて暗号化される。

【０１７５】上記暗号化されて生成されたデータは、ステップＳＴ１４６に示すように、ユーザ端末との通信機能部１３３からユーザ端末５０に送られ、前記図２２のステップＳＴ１３５と図２１のステップＳＴ１２７のように当該ユーザ端末５０からプレーヤ１に転送されることになる。

【０１７６】以上により、ポイント使用情報返却時の管理センタ２１１における処理の流れが終了する。

【０１７７】上述した図２１から図２３の処理の流れにおけるプレーヤ１とユーザ端末５０と管理センタ２１１との間の情報送受のシーケンスは、図２４に示すように表すことができる。

【０１７８】すなわちこの図２４において、入力情報転送Ｔ３１では、前記図２２のステップＳＴ１３２のように、ユーザ端末５０からプレーヤ１に対して、前記パスワード等の入力情報が転送される。作成データ転送Ｔ３２では、前記図２１のステップＳＴ１２６のように、プレーヤ１が作成したデータがユーザ端末５０に転送される。作成データ転送Ｔ３３では、前記図２２のステップＳＴ１３４のように、上記プレーヤ１にて作成されたデータが上記ユーザ端末５０から管理センタ２１１に転送される。データ転送Ｔ３４では、前記図２３のステップＳＴ１４６のように、管理センタ２１１にて作成されたデータが、ユーザ端末５０に転送される。データ転送Ｔ３５では、前記図２１のステップＳＴ１２７のように、管理センタ２１１にて作成されたデータがユーザ端末５０を介してプレーヤ１に転送される。

【０１７９】本実施の形態のシステムのプレーヤ１とユーザ端末５０と管理センタ２１１の実際の動作は、上述したような流れとなる。

【０１８０】ここまでは、本実施の形態のシステムにおける全体の処理の流れを説明してきたが、これ以降は、本実施の形態のシステムの主要部の個々の動作を詳細に説明する。

【０１８１】先ず、本発明実施の形態における暗号化及び圧縮と、伸長及び復号化の動作についての説明を行う。

【０１８２】上述した実施の形態のシステムのように、ネットワークを使ってディジタルコンテンツを配信する際には、そのデータ量を抑えるために圧縮／伸長技術を使用し、コピー防止或いは課金のために暗号化／圧縮技術が使われる。すなわち、配信側（上述の例では管理センタ２１１側）でディジタルコンテンツを圧縮し、さらに暗号化処理することが行われる。上述の例のように送信側（管理センタ２１１側）にて生成されたディジタルコンテンツ（暗号化／圧縮データ）をネットワークを使って配信するとき、受信側（上述の例ではプレーヤ１）では上記暗号化及び圧縮されたディジタルコンテンツを受信後に復号化し、さらに伸長してディジタルコンテンツを復元することが行われる。なお、上記暗号化と圧縮、復号化と伸長の処理の順番は入れ替わる場合もあ

る。

【０１８３】上記ディジタルコンテンツに著作権等が存在する場合、上記受信側は、上記ディジタルコンテンツを上記復号化と伸長する際に、上記著作権者等の意思に従い、課金されることになる。この課金は、主として復号化の鍵すなわちコンテンツ鍵を購入することにより行われるが、このコンテンツ鍵を購入する方法には種々ある。

【０１８４】ここで、上述したように、ディジタルコンテンツを圧縮して暗号化し、復号化して伸長するような処理手順に従った場合、例えば悪意を持ったユーザは上記復号化済みの圧縮データを比較的簡単に入手することができることになる。すなわちディジタルコンテンツの圧縮データは、一般に容量が大きく、したがって例えば受信側の一般的なコンテンツ再生装置の内部メモリではなく、安価が外部メモリに蓄積される場合が多いため、この外部メモリから直接、或いは外部メモリとの接続部分で上記圧縮されたディジタルコンテンツを不正に取り出すことが容易だからである。

【０１８５】また、圧縮に対する伸長方式のアルゴリズムは公開されている場合が多く、また伸長方式のアルゴリズムには一般的な暗号の鍵のようにそれぞれ隠しておけば処理できないようなものも存在していない。しかも、上記復号化された圧縮ディジタルコンテンツは、上記送信側から配信された暗号化と圧縮とがなされたディジタルコンテンツと比較して、データ量的に変わらず、したがって、上記復号化された圧縮ディジタルコンテンツを悪意を持って配信するののも容易である。すなわち、上記圧縮した後に暗号化されてディジタルコンテンツを配信する方式によると、誰でも容易に伸長できる圧縮ディジタルコンテンツが、悪意を持ったユーザに容易に盗難され、このため著作権者等の意思の届かないところでさらに配信されたり、伸長されたりする危険性が大きい。

【０１８６】そこで、本発明の実施の形態では、このような状況に鑑み、ネットワークを使って配信するディジタルコンテンツの安全性を向上させることを可能にするため、上記図２のプレーヤ１において、以下の図２５のフローチャートに示すような処理を行っている。

【０１８７】すなわち図２のプレーヤ１の共通暗号復号回路２４における復号化処理と上記伸長回路２６における伸長処理では、前記記憶メディアから読み出された暗号化と圧縮処理されたディジタルコンテンツのデータを、ステップＳＴ１５１のように、先ず、復号化処理のアルゴリズムの処理単位Ｘビットと、伸長処理のアルゴリズム処理単位Ｙビットとの最小公倍数１ｃｍ（Ｘ，Ｙ）の単位に分割する。

【０１８８】次に、上記最小公倍数１ｃｍ（Ｘ，Ｙ）の単位に分割された上記暗号化と圧縮処理がなされているディジタルコンテンツのデータは、ステップＳＴ１５２に示すように、当該最小公倍数１ｃｍ（Ｘ，Ｙ）の単位毎に、上記共通暗号復号回路２４にて復号化処理が行われる。

【０１８９】当該復号化処理により得られた最小公倍数１ｃｍ（Ｘ，Ｙ）の単位の圧縮されているディジタルコンテンツのデータは、ステップＳＴ１５４に示すように、当該単位分の全ての圧縮データに対して上記伸長回路２６にて伸長処理が行われる。

【０１９０】その後、この最小公倍数１ｃｍ（Ｘ，Ｙ）の単位毎の復号化及び伸長処理は、上記暗号化と圧縮処理されたディジタルコンテンツの全データについての処理が終了するまで続けられる。すなわち、ステップＳＴ１５５に示すように、最小公倍数１ｃｍ（Ｘ，Ｙ）の単位毎の復号化及び伸長処理がディジタルコンテンツの全データに対して完了したか否かの判断がなされ、完了していない時にはステップＳＴ１５２に戻り、完了したときに当該処理のフローチャートが終了する。

【０１９１】これにより全データの復号化及び伸長されたディジタルコンテンツが得られることになる。

【０１９２】なお、当該プレーヤ１における図２５のフローチャートの処理でも、上記最小公倍数１ｃｍ（Ｘ，Ｙ）単位の復号化データは存在することになるが、当該復号化データのデータ量は少ない。このため、比較的高価でも安全性の高い内部メモリに保存することができるようになり、したがって前述したような外部メモリに保存する場合のように盗まれる可能性は非常に低いものとなる。

【０１９３】また、本実施の形態における上記プレーヤ１では、上記安全性を確保するための内部メモリとして、図２のバッファメモリ２５が上記共通暗号復号回路２４と伸長回路２６との間に設けられている。すなわちこのバッファメモリ２５は、１チップの集積回路１０内に設けられており、外部からアクセスされ難く、したがってデータが外部に取り出されることはない。

【０１９４】上述のフローチャートでは、最小公倍数１ｃｍ（Ｘ，Ｙ）の単位分の全てのデータに対して復号化及び伸長処理を行うようにしており、このための具体的構成としては、例えば図２６に示す構成のように、最初に復号化処理のアルゴリズムの処理単位Ｘビットにディジタルコンテンツのデータを分割し、このＸビットのデータに復号化処理を施し、その後当該復号化処理されたＸビットの圧縮されているデータを、伸長処理のアルゴリズム処理単位Ｙビット分まとめ、当該Ｙビットの圧縮データを伸長することで、上述のように最小公倍数１ｃｍ（Ｘ，Ｙ）の単位での復号化及び伸長処理を実現するようにしている。

【０１９５】このことを実現するプレーヤ１の共通暗号復号回路２４は、入力部３０と暗号復号部３１とからなり、上記伸長回路２６は、伸長部３２と出力部３３とからなる。これら共通暗号復号回路２４と伸長回路２６の

間に前記バッファメモリ２５が設けられている。

【０１９６】ここで、より具体的な例として、上記ディジタルコンテンツに対する暗号化処理が例えばＤＥＳ（Data Encription Standard）暗号を用いて行われているのであれば、当該暗号化処理とそれに対応する復号化処理は、６４ビット単位で行われることになる。

【０１９７】また、圧縮されたディジタルコンテンツに対する伸長処理の場合、その圧縮率やサンプリング周波数によっても異なるが、現状では１Ｋ～２Ｋビット／チャンネル単位で処理される場合が多い。ここでは、便宜的に１．２８Ｋビット毎に処理されると仮定する。

【０１９８】したがって、上記ＤＥＳ暗号化方式と上記１．２８Ｋビット毎の圧縮伸長方式を用いたシステムの場合、上記最小公倍数１ｃｍは１．２８Ｋとなる。

【０１９９】このような条件のもと、図２６の共通暗号復号回路２４の入力部３０には、前記暗号化されて圧縮されたディジタルコンテンツが入力される。当該入力部３１では、上記暗号化されて圧縮されたディジタルコンテンツを、上記復号化処理のアルゴリズムの処理単位Ｘビット、すなわち６４ビットづつのデータに分割して暗号復号部３１に出力する。

【０２００】この暗号復号部３２では、上記Ｘビットすなわち６４ビットのデータを、当該６４ビット毎に復号化処理する。この６４ビット毎の復号化により得られた６４ビットの圧縮されているデータは、バッファメモリ２５に送られる。

【０２０１】当該バッファメモリ２５は、前記コントローラ１６からの指示に従い、伸長処理のアルゴリズム処理単位Ｙビット、すなわち１．２８Ｋビット分の圧縮データがたまった時点で、当該１．２８Ｋビット分の圧縮データを一括して出力し、この圧縮データが上記伸長回路２６の伸長部３２に送られる。

【０２０２】上記伸長部２６は、上記入力された１．２８Ｋビット分の圧縮データを伸長して出力部３３に出力する。

【０２０３】また、コントローラ１６は、バッファメモリ２５にたまったデータ量をモニタしながら、復号化部３１の処理と伸長部３２の処理をコントロールする。

【０２０４】なお、このケースであれば、復号化処理を２０個（＝１２８０／６４）並列で処理すれば、より高速な処理システムになる。

【０２０５】その他、前記図２や図２６のようなハードウェア構成ではなく、プログラマブルデバイスにて上述した処理を行う場合には、バッファメモリ２５の状況に応じて、例えばコントローラ１６が復号化プログラム或いは伸長プログラムに基づいて処理を行うことになる。

【０２０６】上述の説明では、圧縮した後に暗号化したディジタルコンテンツがプレーヤ１に供給され、プレーヤ１ではこの圧縮及び暗号化されたディジタルコンテンツを復号化した後に伸長する例を挙げたが、暗号化した後に圧縮されたディジタルコンテンツを伸長して復号化する場合であっても、上述同様の効果を得ることができる。

【０２０７】また、本発明は、圧縮／伸長並びに暗号化／復号化のアルゴリズムが限定されることはなく、いかなる方式に対しても有効である。

【０２０８】このように、本発明によれば、ネットワークを使って配信するディジタルコンテンツの安全性が向上する。

【０２０９】次に、前記セキュリティＩＤの発生動作についての説明を行う。

【０２１０】本実施の形態のように、ポイント情報を予め入手しておき、ディジタルコンテンツの鑑賞に応じて当該ポイント情報を減額するような方式の場合、前述したように、ネットワーク上の管理センタ２１１は、ユーザ側２００のユーザ端末５０からのポイント情報の購入依頼の通信を受けた後に、金融機関２２０その他と任意の確認を行った後、そのポイント情報を暗号化して、ユーザ側２００のプレーヤ１にネットワーク経由で送る。

【０２１１】本実施の形態のように、ポイント情報を予め入手しておき、ディジタルコンテンツの鑑賞に応じて当該ポイント情報を減額するような方式の場合、管理センタ２１１とプレーヤ１（ユーザ端末５０）との間で、ポイント情報の購入の度に、毎回同じようなデータのやり取りを行う（例えば暗号化された「３０００円分のポイント情報の補充要求」及びそれに対応した「３０００円分のポイント情報」といった情報のやりとりを行う）と、例えば悪意を持つ者による、金融機関２２０へのいわゆる「成り済まし」による金額補充が問題点となる。なお、ここに言う金融機関への「成り済まし」とは、上記悪意を持った者が本来のユーザ（本実施の形態ではユーザ側２００）に成り済まして、不正にポイント情報を入手するようなことを言う。

【０２１２】すなわち、ポイント情報の購入の度に毎回同じようなデータのやり取りを行っていると、例えば悪意を持った者が当該データを通信回線から盗み出して同じデータを生成し、管理センタ２１１に対して送り先を自分（悪意を持った者）にしてポイント情報の入手を依頼したような場合、当該悪意を持った者がポイント情報を入手できることになり、さらにこのポイント情報の購入代金の請求は本来のユーザ側２００になされることになるという問題が発生するおそれがある。

【０２１３】そこで、こういった不正を防止するために、本発明実施の形態のシステムでは、予め受信側（プレーヤ１側）と配信側（管理センタ２１１側）の両者で連動している乱数発生機能により発生させられた乱数を安全性向上のために使用している。本実施の形態では、上記乱数として前記セキュリティＩＤを発生している。なお、両者間で乱数発生を連動させるには、例えばユーザの登録手続きなどの際に、例えばタイマ１８を初期化

するなどして、両者間の動作を同期させれば良い。

【０２１４】すなわち、この乱数（セキュリティＩＤ）を用いた場合の管理センタ２１１からプレーヤ１への例えばポイント情報入手時の動作は、以下のような流れとなる。

【０２１５】ポイント情報の購入時、管理センタ２１１からプレーヤ１に対して送られるデータは、前述したように例えばプレーヤ１から予め入手した共通鍵（セッション鍵）を用いて暗号化されたポイント情報と上記発生されたセキュリティＩＤからなるデータとなされる。

【０２１６】プレーヤ１のコントローラ１６は、当該管理センタ２１１から受け取ったデータを前述したように共通暗号復号回路２４に送り、ここで前記共通鍵を用いて復号化処理を行う。これにより、管理センタ２１１から送られてきたポイント情報とセキュリティＩＤとが得られることになる。

【０２１７】その後、プレーヤ１のコントローラ１６は、上記管理センタ２１１から送られてきたセキュリティＩＤと、自身のセキュリティＩＤ発生回路１９にて発生したセキュリティＩＤとを比較する。この比較において、コントローラ１６は、管理センタ２１１からのセキュリティＩＤと、上記自身が発生したセキュリティＩＤとが一致したときのみ、上記管理センタ２１１から送られてきたポイント情報を、前記ポイント情報格納メモリ２８に格納する。

【０２１８】これにより、正当なユーザ側２００のプレーヤ１のみがポイント情報を入手できることになる。言い換えれば、正当なユーザ側２００のプレーヤ１と同じようなプレーヤを持っている悪意の者が、前記成り済ましによって不正にポイント情報を入手しようとしても、当該悪意の者が持っているプレーヤのセキュリティＩＤと上記管理センタ２１１から送られてきたセキュリティＩＤとは一致しないため、この悪意を持った者は前記成り済ましによる不正なポイント情報入手ができないことになる。

【０２１９】勿論、ユーザ側２００のプレーヤ１で発生するセキュリティＩＤは、当該プレーヤ１の集積回路１０内に設けられたセキュリティＩＤ発生回路１９によって発生されるものであり、外部には取り出せないものであるため、悪意を持った者が当該セキュリティＩＤを盗むことはできない。

【０２２０】上記セキュリティＩＤとしての乱数を発生する構成には種々のものがあるが、その一例を図２７に示す。この図２７の構成は、前記図２のセキュリティＩＤ発生回路１９の一具体例である。

【０２２１】この図２７において、一方向関数発生部４０は、いわゆる一方向性関数を発生する。なお、上記一方向性関数とは、比較的計算が簡単な関数で逆関数がはるかに計算が困難なものである。この一方向関数は、予め秘密通信等で受け取って当該一方向関数発生部４０に保存しておくことも可能である。なお、一方向関数発生部４０は、前記図２の集積回路１０内に設けられたタイマ１８からの時間情報を入力関数として上記一方向関数を発生するようにすることも可能である。上記一方向関数は、乱数決定部４３に送られる。

【０２２２】また、ユーザ定数発生部４１は、ユーザ毎に定められた所定のユーザ定数を発生する。このユーザ定数は、予め秘密通信等で送付されて当該ユーザ定数発生部４１に保存されるものである。なお、このユーザ定数は、例えば前記ユーザＩＤ格納メモリ２３が格納するユーザＩＤを用いることもできる。

【０２２３】乱数データベース４２は、乱数を格納するものであり、例えば９９個の乱数を格納している。

【０２２４】通信回数記憶部４４は、例えばコントローラ１６から送られてくる通信回数情報を記憶するものである。この通信回数情報とは、プレーヤ１と管理センタ２１１との間の通信回数を示す情報である。

【０２２５】これら一方向関数とユーザ定数と通信回数情報は、乱数決定部４３に送られる。当該乱数決定部４３は、例えば前記タイマ１８からの時間情報に基づき、上記一方向関数とユーザ定数から、予め乱数データベース部４２に記憶された範囲の乱数を発生させる（例えば９９個）。

【０２２６】すなわち、この乱数決定部４３では、上記通信回数情報が例えば１回目の通信であれば、９９個目の乱数を上記乱数データベース部４２から取り出し、また例えば通信回数情報がｎ回目の通信であれば１００－ｎ個目の乱数を上記乱数データベース４２から取り出し、この取り出した乱数を前記セキュリティＩＤとして出力する。

【０２２７】このセキュリティＩＤ発生の構成は、プレーヤ１と管理センタ２１１とで同じものを有している。

【０２２８】なお、乱数データベース部４２に格納している全ての乱数を使い終わったときには、上記乱数決定部４２において１００個～１９９個目の乱数を計算するか、或いは新たな乱数や１方向性関数を秘密通信するなどして、乱数データベース部４２に再格納したり、一方向性関数発生部４０に再構築する。

【０２２９】また、上述した説明では、乱数（セキュリティＩＤ）を発生させて通信毎の安全性を高めるようにしているが、本実施の形態では、前述のようにユーザ側２００と管理センタ２１１側との間で通信を行う毎に、毎回異なる共通鍵（セッション鍵）をプログラマブルに発生させるようにもしているので、さらに安全性が高められている。

【０２３０】ここで、実際に送信される送信文（例えばメッセージ等）について上記乱数が挿入されると共に、セッション鍵による暗号化がなされる様子と、受信文から乱数が取り出されて正当性の確認がなされる様子を図２８と図２９を用いて説明する。なお、これら図２８、

図２９の例では、送信文に署名（ディジタル署名）を付加するようにもしている。

【０２３１】この図２８において、先ず、前記共通鍵を公開鍵暗号方式にて暗号化して送信する流れとして、通信用共通鍵発生工程Ｐ７では前記セッション鍵を通信用に用いる共通鍵として発生し、この共通鍵は公開鍵暗号化工程Ｐ８にて受信側の公開鍵で暗号化される。この暗号化された共通鍵は、受信側に送られる。

【０２３２】一方、送信文としてのメッセージを共通鍵暗号方式にて暗号化して送信する場合の流れとして、例えばメッセージ生成行程Ｐ１ではメッセージＭが生成されると共に、乱数発生工程Ｐ５にて乱数（前記セキュリティＩＤ）が発生される。これらメッセージＭと乱数は、共通鍵暗号化工程Ｐ６に送られる。この共通鍵暗号化工程Ｐ６では、上記通信用共通鍵発生工程Ｐ７にて発生した共通鍵を用いて、上記メッセージＭと乱数を暗号化する。

【０２３３】さらに、上記ディジタル署名を付加する場合、上記メッセージＭはハッシュ値計算工程Ｐ２に送られる。当該ハッシュ値計算工程Ｐ２では、上記メッセージＭからいわゆるハッシュ値が計算される。なお、ハッシュ値とはハッシュ法にて求められるアドレス情報であり、ハッシュ法とはデータ（この場合はメッセージＭ）の内容の一部（キーワード）に所定の演算を施し、その結果をアドレスとして使用するものである。このメッセージから生成されたハッシュ値（Ｍ）はディジタル署名として、秘密鍵暗号化工程Ｐ４に送られる。この秘密鍵暗号化工程Ｐ４では、送信側の秘密鍵で上記ディジタル署名を暗号化する。この暗号化されたディジタル署名は、共通鍵暗号化工程Ｐ６に送られる。これにより共通鍵暗号化工程Ｐ６では、上記通信用共通鍵発生工程Ｐ７にて発生した共通鍵を用いて、上記ディジタル署名を暗号化する。

【０２３４】これらメッセージＭとディジタル署名と乱数が受信側に送信される。

【０２３５】次に、図２９を用いて、図２８に対応する受信側での処理の流れを説明する。

【０２３６】この図２９において、先ず、前記共通鍵を公開鍵暗号方式にて復号化する流れとして、秘密鍵復号化工程Ｐ１１では、上記送信側から送信されてきた共通鍵を当該受信側の秘密鍵で復号化する。

【０２３７】一方、前記共通鍵暗号方式にて暗号化されたメッセージＭを復号化する流れとして、共通鍵復号工程１３では、上記送信されてきたメッセージＭを上記秘密鍵復号化工程Ｐ１１にて復号化した共通鍵を用いて復号化する。この復号化されたメッセージＭは、他機能送信工程Ｐ２０にて他の工程に送られることになる。

【０２３８】また、ディジタル署名を復号する流れでは、上記共通鍵復号化工程Ｐ１３にて復号化されたハッシュ値が、公開鍵復号化工程Ｐ１４にて送信側の公開鍵を用いて復号化される。同時に、ハッシュ値計算工程Ｐ１７では、上記メッセージＭからハッシュ値を計算する。これら公開鍵復号化工程Ｐ１４により復号化されたハッシュ値と上記ハッシュ値計算工程Ｐ１７にて計算されたハッシュ値とは、比較工程Ｐ１９にて比較され、改竄されていないことの確認が行われる。

【０２３９】さらに、送信された乱数については、上記共通鍵復号化工程Ｐ１３にて復号化された乱数と、当該受信側の乱数発生工程Ｐ２１にて発生された乱数とが、正当正確認工程Ｐ２２にて比較され、正当性の確認が行われる。

【０２４０】ところで、前述した図１に示した本実施の形態のシステムでは、ユーザ側２００に対するシステム側として、システム管理会社２１０と仮想店舗２３０とコンテンツプロバイダ２４０とが設けられている。なお、図１の金融機関２２０は、例えば外部の銀行等である。

【０２４１】上記システム管理会社２１０の管理センタ２１０は、仮想店舗２３０におけるディジタルコンテンツの展示や配信の管理、金融機関２２０との間でユーザ側２００の課金情報や各種情報の収集，分配及びそれらの管理、コンテンツプロバイダ２４０からのディジタルコンテンツの暗号化、扱う情報のセキュリティ管理など、システム側の主要な作業のほぼ全てを行っている。

【０２４２】しかし、上述したようなネットワークを使ってディジタルコンテンツを配信するシステムにおいて、ユーザ側がシステム側からディジタルコンテンツを入手する際や、ディジタルコンテンツの使用に伴う課金の際には、システム側に通信が集中することになり、ユーザ側に対して満足のいくレスポンスが得られなくなるおそれがある。

【０２４３】そこで、本発明の他の実施の形態では、システム管理会社２１０の機能、より具体的には管理センタ２１１の機能を、以下のように分割することで、上述したような通信の集中を防ぎ、通信のレスポンスを向上させることを可能にしている。

【０２４４】すなわち、本発明の他の実施の形態では、図３０に示すように、ユーザ側２００に対するシステム側の構成を、ディジタルコンテンツを展示，配信する機能を有するコンテンツ展示配信機関３１０と、一定の地域のユーザの課金情報を管理する機能を有する課金情報管理機関３２０と、ディジタルコンテンツを暗号化する等のデータ生成と上記コンテンツ展示配信機関３１０への生成データの配信と上記課金情報管理機関３２０からの情報収集と収益分配とシステム全体のセキュリティ管理その他を行う機能を有するシステム管理機関３３０とに分割し、各機関３１０，３２０，３３０がそれぞれ独立にユーザ側２００と通信可能になされている。

【０２４５】この図３０のような構成において、コンテンツ展示配信機関３１０は、世界中のネットワーク上に

散らばって複数配置可能なものであり、ユーザ側２００は通信費さえ支払えばどの地域のコンテンツ展示配信機関３１０へでもアクセスできる。例えばユーザ側２００がディジタルコンテンツを入手したい場合には、ユーザ側２００から上記コンテンツ展示配信機関３１０にアクセスして、ディジタルコンテンツを入手する。このときのディジタルコンテンツは、システム管理機関３３０によって暗号化等されたディジタルコンテンツ、すなわちユーザ側２００にネットワークを使って直接送信可能な状態になされたものである。

【０２４６】また、課金情報管理機関３２０は、課金情報を扱うため、余り多くのユーザを抱え込むことは安全性管理上好ましくなく、したがって、適度な数のユーザ毎に設置する。但し、あまり多く設置すると、悪意を持った第３者からの攻撃ポイント（課金情報管理機関３２０）を増やすことになり、トレードオフになるので、最適化することが望ましい。例えばユーザ側２００が課金に関する通信を行う場合には、ユーザ側２００から上記課金情報管理機関３２０に対してアクセスする。

【０２４７】上記システム管理機関３３０は、ユーザのシステムへの加入や決済方法の登録、ユーザからの集金や前記権利者，コンテンツ展示配信機関３１０，課金情報管理機関３２０等の利益受益者への利益配付など、セキュリティ上重要な情報の管理をまとめて行うことで、セキュリティを向上させる。但し、当該システム管理機関３３０は世界に１箇所のみ設けるわけではなく、あるまとまった単位、例えば国などの単位で設置するのが望ましい。例えば、ユーザ側２００がこのシステムへの加入や決済方法の登録などセキュリティ上重要な通信を行う場合には、ユーザ側２００から上記システム管理機関３３０に対してアクセスして行う。当該ユーザからの集金と利益受益者への利益配付は上記課金情報管理機関３２０から情報を入手した当該システム管理機関３３０がまとめて行う。また、著作権者等が有するソースデータすなわちコンテンツは、当該システム管理機関３３０に供給され、ここで暗号化等がなされたディジタルコンテンツに変換され、上記コンテンツ展示配信機関３１０に配信される。

【０２４８】上述のように、システム側の機能を例えば３つの機関３１０，３２０，３３０に振り分け、ユーザ側２００と各機関３１０，３２０，３３０との間で直接アクセス可能とすることにより、通信の集中を防ぎ、通信のレスポンスを向上させることが可能となる。また、コンテンツ展示配信機関３１０によれば、既存のいわゆるバーチャルモールのようなものにも対応でき、販売促進にも有効であり、ユーザにとって魅力のあるものになる。課金情報管理機関３２０を別に分けることにより、コンテンツの展示や販売機能と結託した不正防止に役立つ。また、管理するユーザを一定の数に抑えられるため、不正に対する管理機能もより効果的である。

【０２４９】以下に、上述した図３０に示した本発明の他の実施の形態のシステムにおいて、ユーザのシステムへの加入、ポイント情報の購入や暗号化されたディジタルコンテンツの復号用のコンテンツ鍵等の入手時の情報の流れ、コンテンツとコンテンツ鑑賞用の情報の流通の際の流れ、コンテンツの使用に伴う課金情報の流れについて説明する。

【０２５０】先ず、図３１を用いて、ユーザのシステムへの加入時の流れの主要部を説明する。

【０２５１】ユーザのシステムへの加入登録の際には、システム管理機関３３０のユーザ加入サポート機能ブロック４０２による以下の手順の従って登録作業が行われる。

【０２５２】ユーザ側２００すなわち前記プレーヤ１及びユーザ端末５０からは、先ず加入意思送付Ｔ４１のように、システムへの加入の意思を示す情報が、システム管理機関３３０に対してネットワークを介して送付される。システム管理機関３３０の通信機能ブロック４０１に入力された上記加入意思の情報は、ユーザ加入サポート機能ブロック４０２に送られる。

【０２５３】当該ユーザ加入サポート機能ブロック４０２は、上記加入意思情報を受信すると、加入必要ファイル送付Ｔ４２のように、加入に必要なファイルの情報を通信機能ブロック４０１を介してユーザ側２００に送られる。

【０２５４】ユーザ側２００では、上記システム管理機関３３０から送られてきた加入必要ファイルに基づいて、所定のフォーマットに従った加入申請書の作成が行われる。当該作成された加入申請書は、加入申請書送付Ｔ４３のように、システム管理機関３３０に送付される。

【０２５５】上記加入申請書を受け取ったユーザ加入サポート機能ブロック４０２は、クライアント機能送付Ｔ４４のように、クライアントの機能を解説する情報を、ユーザ側２００に送付する。

【０２５６】当該クライアント機能の情報を受け取ったユーザ側２００からは、ユーザ情報送付Ｔ４５のように、ユーザ側の情報、例えば前述したような口座番号やクレジット番号，名前や連絡先等のユーザ情報を、システム管理機関３３０に送付する。

【０２５７】当該ユーザ情報の送付を受けたユーザ加入サポート機能ブロック４０２は、登録手続き完了通知Ｔ４６のように、加入の登録手続きが完了した旨の情報を、ユーザ側２００に通知する。

【０２５８】また、このユーザ加入登録の手続き完了後、システム管理機関３３０のユーザ加入サポート機能ブロック４０２は、ユーザ情報送付Ｔ４７のように、通信機能ブロック４０１を介して、課金情報管理機関３２０に対してユーザ情報を転送する。このユーザ情報を受け取った課金情報管理機関３２０は、当該ユーザ情報を

データベース機能ブロック３６７に保存する。

【０２５９】以上により、ユーザのシステムへの加入時の主な流れが終了する。なお、この図３１に挙げられている他の構成についての説明は後述する。

【０２６０】次に、図３２を用いて、ポイント情報の購入や暗号化されたディジタルコンテンツの復号用の鍵等の入手時の情報の流れの主要部を説明する。なお、上記ポイント情報の購入や暗号化されたディジタルコンテンツの復号用のコンテンツ鍵の情報は、コンテンツを使用するための情報であるので、以下の説明では、これらを簡略化して使用権情報と呼ぶことにする。

【０２６１】ユーザがシステムで使用する重要な情報（ここでは、コンテンツの使用権）を入手する際は、予めユーザ側２００毎に担当割当がなされている課金情報管理機関３２０に対し、ユーザ側２００からアクセスがなされる。上記ユーザ側２００から送られてくるコンテンツ使用権情報の入手要求のアクセスに対しては、課金情報管理機関３２０の使用権発行機能ブロック３６２が対応し、以下の手順に従って使用権の発行が行われる。

【０２６２】先ず、ユーザ側２００からは、購入依頼書送付Ｔ５１のように、使用権を購入したい旨の情報が課金情報管理機関３２０に対して送付される。使用権を購入したい旨の情報は、ユーザ側２００によって所定のフォーマットに従った購入依頼書の情報である。このようにネットワークを介し、この課金情報管理機関３２０の通信機能ブロック３６１に入力された上記購入依頼書の情報は、使用権発行機能ブロック３６２に送られる。

【０２６３】当該使用権発行機能ブロック３６２では、上記購入依頼書の情報を受け取ると、データベース機能ブロック３６７に保存されたユーザ情報を元にして、新しい使用権の情報を生成し、新規使用権送付Ｔ５２のように、当該使用権の情報をユーザ側２００に対して送付する。

【０２６４】ユーザ側２００は、上記新規使用権の情報の受取を確認すると、所定のフォーマットに従った受取確認書を作成し、受取確認書送付Ｔ５３のように、課金情報管理機関３２０の使用権発行機能ブロック３６２に送付する。

【０２６５】以上により、使用権の購入時の主な流れが終了する。なお、この図３２に挙げられている他の構成についての説明は後述する。

【０２６６】次に、図３３を用いて、コンテンツとコンテンツ鑑賞用の情報（ここでは使用条件とコンテンツ鍵）の流通の際の流れの主要部を説明する。

【０２６７】先ず、コンテンツ展示配信機関３１０のコンテンツ入手機能ブロック３４２は、コンテンツ請求書送付Ｔ６２のように、システム管理機関３３０に対して、ディジタルコンテンツを請求する。

【０２６８】当該コンテンツ請求書を受け取ったシステム管理機関３３０は、コンテンツ配布機能ブロック４０４において、要求されたコンテンツを流通できるように加工する。すなわち、このコンテンツ配布機能ブロック４０４では、ユーザ側２００に送付可能な状態のディジタルコンテンツ（暗号化されたディジタルコンテンツ）を生成する。この加工されたディジタルコンテンツは、コンテンツ送付６３のように、コンテンツ展示配信機関３１０に送られる。

【０２６９】当該コンテンツ展示配信機関３１０では、上記加工されたディジタルコンテンツを、コンテンツデータベース機能ブロック３４５に保存する。

【０２７０】また、システム管理機関３３０のコンテンツ配布機能ブロック４０４では、コンテンツ鑑賞用の情報として、コンテンツＩＤと使用条件と暗号化されたコンテンツを復号するためのコンテンツ鍵とを、コンテンツ鑑賞用情報送付Ｔ６４のように、課金情報管理機関３２０に送付する。

【０２７１】課金情報管理機関３２０では、上記コンテンツ鑑賞用の情報を、コンテンツ鍵・使用条件受取機能ブロック３６３にて受理し、データベース機能ブロック３６７に保存する。

【０２７２】次に、ユーザ側２００は、コンテンツ入手依頼Ｔ６１のように、コンテンツ展示配信機関３１０に対してアクセスし、コンテンツを入手する。すなわち、コンテンツ展示配信機関３１０は、通信機能ブロック３４１を介して上記ユーザ側２００からコンテンツの入手の要求がなされると、コンテンツデータベース機能ブロック３５４に保存している暗号化されたディジタルコンテンツを読み出し、当該読み出したディジタルコンテンツをユーザ側２００の送付する。

【０２７３】その後、ユーザ側２００は、コンテンツ鑑賞用情報請求Ｔ６５にて課金情報管理機関３２０に対してアクセスし、コンテンツ鑑賞用情報送付Ｔ６６のようにコンテンツ鑑賞用の情報を入手する。すなわち、課金情報管理機関３２０では、通信機能ブロック３６１を介して、上記ユーザ側２００からコンテンツ鑑賞用の情報として使用条件とコンテンツ鍵の請求がなされると、コンテンツ鍵・使用条件発行機能ブロック３６４からコンテンツ鍵と使用条件とを発行し、これらを通信機能ブロック３６１を介してユーザ側２００に送付する。

【０２７４】以上により、コンテンツとコンテンツ鑑賞用の情報の流通の際の流れが終了する。なお、この図３３に挙げられている他の構成についての説明は後述する。

【０２７５】次に、図３４を用いて、コンテンツが実際に鑑賞されたときの精算、すなわちコンテンツ使用料金の精算の流れの主要部を説明する。

【０２７６】先ず、ユーザ側２００にてコンテンツの鑑賞が行われた後、当該ユーザ側２００からは、精算書送付Ｔ７１のように、例えば前述のようにしてポイント使用情報すなわちコンテンツの使用記録が課金情報管理機

関３２０に対して送付される。このように通信機能ブロック３６１を介して上記ユーザ側２００から上記コンテンツ使用記録の送付を受けると、課金情報管理機関３２０の精算手続き受付機能ブロック３６５にて当該コンテンツ使用記録を受け取り、これに対応する精算確認書を発行する。当該精算確認書は、精算確認書送付Ｔ７３のように、同じく通信機能ブロック３６１を介してユーザ側２００に送付される。これにより、ユーザ側２００は精算が行われたことを知ることができる。

【０２７７】次に、課金情報管理機関３２０の精算手続き受付機能ブロック３６５は、使用権発行機能ブロック３６２から使用権発行情報を発行させる。この使用権発行情報は、上記ユーザ側２００から送られてきたコンテンツ使用記録と共に、通信機能ブロック３６１を介し、ユーザ決済・コンテンツ使用記録送付Ｔ７４としてシステム管理機関３３０に送付される。

【０２７８】システム管理機関３３０は、集金及び分配機能ブロック４０５にて、各地に分散している課金情報管理機関３２０から送付されてきた情報をまとめ、集金額と集金先とお金の分配先を集計し、実際の金融機関を通して決済する。

【０２７９】以上により、コンテンツ使用料金の精算の流れが終了する。なお、この図３４に挙げられている他の構成についての説明は後述する。

【０２８０】上述の図３０から図３４までの説明において、コンテンツ展示配信機関３１０、課金情報管理機関３２０、システム管理機関３３０とユーザ側２００との間のデータ送受や、コンテンツ展示配信機関３１０、課金情報管理機関３２０とシステム管理機関３３０との間のデータ送受においても、前述同様にデータの暗号化と復号化が行われていることは言うまでもない。またこの暗号化と復号化においても、公開鍵暗号方式と共通鍵暗号方式の何れを用いても良いし、前述したようにコンテンツ鍵や共通鍵の暗号化方式としては公開鍵暗号方式を使用し、メッセージや各種の書類等の暗号化方式としては共通鍵暗号方式を使用することができる。また、これら暗号化と共に前記乱数を用いたセキュリティ向上の手法や、コンテンツを扱う際の暗号化と圧縮の処理単位の最小公倍数化を使用することも可能である。

【０２８１】次に、上述した各機関３１０、３２０、３３０の具体的な構成について簡単に説明する。

【０２８２】先ず、図３５を用いてコンテンツ展示配信機関３１０の構成の説明を行う。

【０２８３】この図３５において、当該コンテンツ展示配信機関３１０は、大別して、ユーザ側２００とシステム管理機関３３０との間の通信機能を担当する通信機能ブロック３４１と、コンテンツの入手機能を担当するコンテンツ入手機能ブロック３４２と、コンテンツの展示機能を担当するコンテンツ展示機能ブロック３４３と、精算を担当する精算機能ブロック３４４と、コンテンツを保存するコンテンツデータベース機能ブロック３４５とからなる。

【０２８４】上記コンテンツ入手機能ブロック３４２は、システム管理機関３３０に対してコンテンツを請求するときの請求書の作成を担当するコンテンツ請求書作成機能部３５１と、システム管理機関３３０からコンテンツを受け取ったときの受領書の作成を担当するコンテンツ受領書作成機能部３５２と、これらあつかったコンテンツとコンテンツデータベース機能ブロック３４５に保存しているコンテンツとの対応を担当するコンテンツデータベース対応機能部３５３とからなる。

【０２８５】上記コンテンツ展示機能ブロック３４３は、実際に仮想店舗にコンテンツを展示する機能を担当するコンテンツ展示機能部３５４と、これら展示しているコンテンツと上記コンテンツデータベース機能ブロック３４５に保存しているコンテンツとの対応を担当するコンテンツデータベース対応機能部３５５とからなる。

【０２８６】上記精算機能ブロック３４４は、領収書を発行する機能を担当する領収書発行機能部３５６と、金融機関２２０との間の対応を担当する金融機関対応機能部３５７とからなる。

【０２８７】次に、図３６を用いて、課金情報管理機関３２０の構成の説明を行う。

【０２８８】この図３６において、当該課金情報管理機関３２０は、大別して、ユーザ側２００とシステム管理機関３３０との間の通信機能を担当する通信機能ブロック３６１と、使用権を発行する機能を担当する使用権発行機能ブロック３６２と、コンテンツ鍵と使用条件の受け取りを担当するコンテンツ鍵・使用条件受取機能ブロック３６３と、コンテンツ鍵と使用条件の発行を担当するコンテンツ鍵・使用条件発行機能ブロック３６４と、精算手続きの受け付け機能を担当する精算手続き受付機能ブロック３６５と、分配と受け取りの機能を担当する分配受取機能ブロック３６６と、データベース機能ブロック３７６とからなる。

【０２８９】上記使用権発行機能ブロック３６２は、購入依頼書の確認機能を担当する購入依頼書確認機能部３７１と、クライアントすなわちユーザ側２００の使用権の残高（ポイント情報の残高）や使用記録（ポイント使用情報）等のデータの確認を担当するポイントデータ確認機能部３７２と、使用権を発生する機能を担当する使用権発生機能部３７３と、使用権の送付書を作成する機能を担当する使用権送付書作成機能部３７４と、使用権と使用権送付書を実際に送付する機能を担当する送付機能部３７５と、使用権の受け取り書の確認を担当する使用権受取確認機能部３７６と、発行した使用権の情報を保存する機能を担当する使用権発行情報保存機能部３７７とからなる。

【０２９０】上記コンテンツ鍵・使用条件受取機能ブロック３６３は、コンテンツ鍵と使用条件の受取を担当す

る受取機能部３７８と、コンテンツ鍵と使用条件を保存する保存機能部３７９とからなる。

【０２９１】上記コンテンツ鍵・使用条件発行機能ブロック３６４は、コンテンツ鍵と使用条件の入手依頼を受信する機能を担当する受信機能部３８０と、コンテンツ鍵と使用条件をデータベース機能ブロック３６７から検索して探し出す機能を担当する検索機能部３８１と、コンテンツ鍵と使用条件を暗号化して送付する機能を担当する送信機能部３８２と、コンテンツ鍵と使用条件の受取書の確認機能を担当する確認機能部３８３とからなる。

【０２９２】上記精算手続き受付機能ブロック３６５は、暗号化されているコンテンツ使用記録（ポイント使用情報）を受信して復号化する機能を担当するコンテンツ使用記録受信機能部３８４と、コンテンツ使用記録の確認を担当するコンテンツ使用記録確認機能部３８５と、コンテンツ使用記録をデータベース機能ブロック３６７の保存する機能を担当するコンテンツ使用記録保存機能部３８６と、精算手続きの完了書を作成する機能を担当する完了書作成機能部３８７と、コンテンツ使用記録をまとめて編集する機能を担当するまとめ機能部３８９とからなる。

【０２９３】上記分配受取機能ブロック３６６は、集金を行う際の資料を請求する資料請求書の確認機能を担当する請求書確認機能部３９０と、システム管理機関３３０に対して提出するコンテンツ使用記録の報告書を作成する機能を担当する使用記録報告書作成機能部３９１と、システム管理機関３３０に対して提出する使用権発行情報の報告書を作成する機能を担当する使用権発行報告書作成機能部３９２と、報告書の受信確認書の確認機能を担当する確認書確認機能部３９３とからなる。

【０２９４】データベース機能ブロック３６７は、使用権のデータを保存する機能を担当する使用権データベース機能部３９４と、コンテンツ鍵と使用条件のデータを保存する機能を担当するコンテンツ鍵・使用権データベース機能部３９５と、コンテンツ使用記録を保存するコンテンツ使用記録データベース機能部３９６と、ユーザに関する情報を保存するユーザ管理データベース機能部３９７とからなる。

【０２９５】次に、図３７を用いて、システム管理機関３３０の構成の説明を行う。

【０２９６】この図３７において、当該システム管理機関３３０は、大別して、ユーザ側２００、コンテンツ展示配信機関３１０、及び課金情報管理機関３２０との間の通信機能を担当する通信機能ブロック４０１と、ユーザ加入の際のサポートを行うユーザ加入サポート機能ブロック４０２と、コンテンツの配布を担当するコンテンツ配布機能ブロック４０４と、データベース機能ブロック４０３と、集金と分配の機能を担当する集金及ぶ分配機能ブロック４０５とからなる。

【０２９７】上記ユーザ加入サポート機能ブロック４０２は、加入申請書の作成と送信を担当する加入申請書作成送信機能部４１１と、暗号化された共通鍵を受信して復号化する機能を担当する共通鍵受信機能部４１２と、ユーザ側２００から送信されてきた加入申請書の確認機能を担当する加入申請書確認機能部４１３と、クライアントＩＤすなわちユーザＩＤを発生する機能を担当するＩＤ発生機能部４１４と、加入申請書をデータベース機能ブロック４０３に保存する機能を担当する加入申請書保存機能部４１５と、クライアント機能を生成するクライアント機能生成機能部４１６と、登録情報をデータベース機能ブロック４０３に保存する機能を担当する登録情報保存機能部４１７とからなる。

【０２９８】データベース機能ブロック４０３は、ユーザの情報を保存管理するユーザ管理データベース機能部４１８と、コンテンツを保存するコンテンツデータベース機能部４１９と、課金情報管理機関３２０の情報を保存管理する課金情報管理機関データベース機能部４２０と、コンテンツ展示配信機関３１０の情報を保存管理するコンテンツ展示配信機関データベース機能部４２１とからなる。

【０２９９】コンテンツ配信機能ブロック４０４は、コンテンツの請求書の確認機能を担当する請求書確認機能部４２２と、生コンテンツすなわち加工前のコンテンツ（ソースデータ）をデータベース機能ブロック４０３のコンテンツデータベース機能部４１９から検索する機能を担当するコンテンツ検索機能部４２３と、コンテンツＩＤを生成するコンテンツＩＤ生成機能部４２４と、コンテンツ鍵を生成するコンテンツ鍵生成機能部４２５と、コンテンツ使用条件を生成するコンテンツ使用条件生成機能部４２６と、生コンテンツすなわち加工前のコンテンツを圧縮するコンテンツ圧縮機能部４２７と、コンテンツの暗号化を行うコンテンツ加工機能部４２８と、コンテンツＩＤとコンテンツ鍵と使用条件とをデータベース機能ブロック４０３のコンテンツデータベース機能部４１９に保存する機能を担当する保存機能部４２９と、コンテンツを通信機能ブロック４０１を介して送付する機能を担当するコンテンツ送付機能部４３０と、コンテンツの受領書を確認する機能を担当するコンテンツ受領書確認機能部４３１と、コンテンツＩＤとコンテンツ鍵と使用条件を通信機能ブロック４０１を介して送付する機能を担当するＩＤ・鍵・使用条件送付機能部４３２と、コンテンツＩＤとコンテンツ鍵と使用条件の受領書を確認する機能を担当するＩＤ・鍵・使用条件受領書確認機能部４３３とからなる。

【０３００】集金及び分配機能ブロック４０５は、集金に使用する資料の請求書を作成する資料請求書作成機能部４３４と、コンテンツ使用権を通信機能ブロック４０１を介して受信する機能を担当するコンテンツ使用権受信機能部４３５と、コンテンツ使用記録を通信機能ブロ

ック４０１を介して受信する機能を担当するコンテンツ使用記録受信機能部４３６と、受信の確認書を作成する機能を担当する受信確認書作成機能部４３７と、ユーザへ請求する請求額の計算と請求書の作成を行う請求書の作成を行う計算・請求書作成機能部４３８と、使用により集金した使用金を権利者に分配する際の分配金の計算と納付書の作成を行う計算・納付書作成機能部４３９とからなる。

【０３０１】次に、当該他の実施の形態のシステムに対応するユーザ側２００の構成を、図３８を用いて説明する。なお、この図３８は、前記プレーヤ１とユーザ端末５０の各機能をまとめて表している。

【０３０２】この図３８において、当該ユーザ側２００の構成は、大別すると、システム管理機関３３０、コンテンツ展示配信機関３１０、及び課金情報管理機関３２０との間の通信機能を担当する通信機能ブロック４５１と、コンテンツの入手を担当するコンテンツ入手機能ブロック４５２と、ポイント情報やコンテンツ鍵、使用条件等の使用権の購入を担当する使用権購入機能ブロック４５３と、コンテンツ鍵と使用条件の入手を担当するコンテンツ鍵・使用条件入手機能ブロック４５４と、精算手続きを担当する精算手続き機能ブロック４５５と、システムへの加入をサポートする機能を担当するユーザ加入サポート機能ブロック４５６と、コンテンツの鑑賞と課金の機能を担当するコンテンツ鑑賞課金機能ブロック４５７と、データベース機能ブロック４５８とからなる。

【０３０３】上記コンテンツ入手機能ブロック４５２は、実際にコンテンツを入手する機能を担当するコンテンツ入手機能部４６１と、コンテンツを記憶メディアに保存させる機能を担当するコンテンツ保存機能部４６２とからなる。

【０３０４】使用権購入機能ブロック４５３は、使用権の購入依頼書を作成する購入依頼書作成機能部４６３と、クライアント（ユーザ）の使用権の残高（ポイント残高）や使用記録（ポイント使用情報）等のデータのまとめを担当するまとめ機能部４６４と、使用権としての各情報をインストールする機能を担当する使用権インストール機能部４６５と、使用権受取書を作成する使用権受取書作成機能部４６７とからなる。

【０３０５】コンテンツ鍵・使用条件入手機能ブロック４５４は、コンテンツ鍵と使用条件の入手依頼書を作成する入手依頼書作成機能部４６８と、コンテンツ鍵と使用条件の受信を担当する受信機能部４６９と、コンテンツ鍵と使用条件の受取書を作成する受取書作成機能部４７０とからなる。

【０３０６】精算手続き機能ブロック４５５は、コンテンツ使用記録（ポイント使用情報）のまとめを行うまとめ機能部４７１と、精算手続きの完了書の受信を担当する完了書受信機能部４７２とからなる。

【０３０７】上記ユーザ加入サポート機能ブロック４５６は、加入申請書の作成を担当する加入申請書作成機能部４７３と、クライアント機能のインストールすなわちユーザのプレーヤ１の初期化を担当するクライアント機能インストール機能部４７４、登録情報を作成する機能を担当する登録情報作成機能部４７５とからなる。

【０３０８】コンテンツ鑑賞課金機能ブロック４５７は、記憶メディアに保存されたコンテンツの検索を担当するコンテンツ検索機能部４７６と、使用権の確認を担当する使用権確認機能部４７７と、例えばコンテンツの選択を行うときに簡易的にコンテンツを再生する簡易コンテンツ鑑賞機能部４７８と、課金情報（ポイント情報）の管理を行う課金機能部４７９と、暗号化されているコンテンツを復号化するコンテンツ復号機能部４８０と、圧縮されているコンテンツを伸長するコンテンツ伸長機能部４８１と、例えば記憶メディアに保存されているコンテンツの内容を認識可能にするためのコンテンツビューア機能部４８２とからなる。

【０３０９】データベース機能ブロック４５８は、使用権のデータを保存する使用権データベース機能部４８３と、コンテンツ鍵と使用条件を保存するコンテンツ鍵・使用条件データベース機能部４８４と、コンテンツ使用記録を保存するコンテンツ使用記録データベース機能部４８５と、ユーザ情報を保存するユーザ情報データベース機能部４８６とからなる。

【０３１０】次に、上述したような各実施の形態のプレーヤ１とユーザ端末５０の具体的な使用形態について、図３９と図４０を用いて説明する。

【０３１１】図３９に示すように、プレーヤ１は、前記アナログ出力端子２とＰＣ用インターフェース端子３と記憶メディア用Ｉ／Ｏ端子４がプレーヤ１の筐体外に突き出た状態で配置されており、上記記憶メディア用Ｉ／Ｏ端子４には、記憶メディア６１が接続されるようになっている。また、これらプレーヤ１と記憶メディア６１は、例えばケース６０内に収納可能に形成されており、このケース６０の例えば一端側に上記プレーヤ１のアナログ出力端子２とＰＣ用インターフェース端子３が配置されるようになされている。

【０３１２】このプレーヤ１及び記憶メディア６１が収納されたケース６０は、上記プレーヤ１のアナログ出力端子２とＰＣ用インターフェース端子３が配置される側から、上記ユーザ端末５０としてのパーソナルコンピュータ５０の入出力ポート５３に挿入接続可能なように形成されている。

【０３１３】当該パーソナルコンピュータ５０は、コンピュータ本体に、ディスプレイ装置５２とキーボード５４とマウス５５とを備えた一般的な構成を有するものであるが、上記入出力ポート５３内には上記プレーヤ１のアナログ出力端子２及びＰＣ用インターフェース端子３と対応したインターフェースが形成されている。したが

って、上記プレーヤ１及び記憶メディア６１が収納されたケース６０を上記パーソナルコンピュータ５０の入出力ポート５３に挿入するだけで、上記プレーヤ１のアナログ出力端子２とＰＣ用インターフェース端子３が上記パーソナルコンピュータ５０と接続されるようになる。

【０３１４】上記図３９の例では、パーソナルコンピュータ５０の入出力ポート５３内に、上記プレーヤ１のアナログ出力端子２及びＰＣ用インターフェース端子３と対応したインターフェースを形成するようにしているが、例えば図４０に示すように、パーソナルコンピュータ５０の汎用入出力ポートのインターフェースに対応できるアダプタ６２を、上記プレーヤ１のアナログ出力端子２及びＰＣ用インターフェース端子３の間に配置することも可能である。

【０３１５】以上述べてきたことから、本発明の実施の形態のシステムにおいては、ディジタルコンテンツはシステムの共通鍵であるコンテンツ鍵にて暗号化されているので、本実施の形態のシステムに登録したユーザ（プレーヤ１）であれば、この暗号化されたコンテンツを自由にコピーでき、コンテンツ鍵を入手しさえすればこのコンテンツの鑑賞も可能である。したがって、このコンテンツ（暗号化されたコンテンツの）記憶メディアへのインストールも簡単に行える。一方、本実施の形態システムに準拠していない端末装置では、暗号化されたディジタルコンテンツを復号できないので、コンテンツの著作権や当該コンテンツの権利者の権利は保護される。

【０３１６】また、本発明の実施の形態システムによれば、ポイント情報をプリペイド方式（料金前払い方式）により補充することにし、コンテンツ鑑賞時にポイント情報が減額されるようにするとともに、そのポイントの使用情報を収集するようにしているので、使用済みのポイントに関する権利をもつ権利者（著作権者等）及びコンテンツ販売店舗等は、鑑賞代金の回収が可能である。

【０３１７】さらに、ポイント情報やポイント使用情報のデータのやりとりの際には、前述したように暗号化が施されているので、セキュリティ性が向上している。例えば全く前回のデータと同じものを偽造して課金用のポイント情報を盗もうとしても、前述したように、システム側とプレーヤ側とで連動した乱数（セキュリティＩＤ）を使用し、両者が一致していることを確認してから取引を行うものとしているので、安全である。

【０３１８】またさらに、プレーヤの主要構成要素は１チップ化されており、鍵情報や復号化されたディジタルコンテンツを外部に取り出すことが困難となっている。このプレーヤ１は、当該プレーヤ１の破壊によるデータ横取りを防ぐためにプレーヤ１自体にタンパーレジスタンス機能を備えている。

【０３１９】上述したように、本発明の実施の形態によれば、セキュリティ上強度の高いディジタルコンテンツ配信システムが構築されている。

【０３２０】なお、上述のディジタルコンテンツとしては、ディジタルオーディオデータの他に、ディジタルビデオデータ等の各種のものを挙げることができる。上記ディジタルビデオデータとして動画像データ（オーディオデータも含む）使用した場合、前記圧縮の手法としては、例えばＭＰＥＧ（Moving Picture Image CodingExperts Group）等の圧縮手法を使用できる。なお、上記ＭＰＥＧは、ＩＳＯ（国際標準化機構）とＩＥＣ（国際電気標準会議）のＪＴＣ（Joint Technical Committee）１のＳＣ（Sub Committee）２９のＷＧ（Working Group）１１においてまとめられた動画像符号化方式の通称であり、ＭＰＥＧ１，ＭＰＥＧ２，ＭＰＥＧ４等がある。

【０３２１】さらに、上記暗号化の手法としては、前述したように、例えばいわゆるＤＥＳ（Data Encryption Standard）と呼ばれている暗号化手法を使用することができる。なお、ＤＥＳとは、米国のＮＩＳＴ（National Institute of Standards and Technology）が１９７６年に発表した標準暗号方式（暗号アルゴリズム）である。具体的には、６４ビットのデータブロック毎にデータ変換を行うものであり、関数を使った変換を１６回繰り返す。上記ディジタルコンテンツやポイント情報等は、当該ＤＥＳを用い、いわゆる共通鍵方式にて暗号化されている。なお、上記共通鍵方式とは、暗号化するための鍵データ（暗号鍵データ）と復号化するための鍵（復号鍵データ）が同一となる方式である。

【０３２２】また、前記図１のプレーヤ１の共通鍵保管メモリ２２や通信用鍵保管メモリ２１、ポイント使用情報格納メモリ２９、ポイント情報格納メモリ２８等には、例えばいわゆるＥＥＰＲＯＭ（電気的に消去可能なＲＯＭ）を使用できる。

【０３２３】他に記憶メディアとしては、例えばハードディスクやフロッピィディスク、光磁気ディスク，相変化型光ディスク等の記録媒体、或いは半導体メモリ（ＩＣカード等）の記憶メディアを使用できる。

【０３２４】その他、上述の実施の形態では、コンテンツの選択や仮想店舗２３０に展示されたコンテンツの内容確認等の際には、ユーザ端末５０のキーボード５４やマウス５５、ディスプレイ装置５２を使用して選択、確認等を行っていたが、これらキーボードやマウス、ディスプレイ装置に機能を簡略化して、プレーヤ１に持たせることも可能である。すなわち。図２のように、入力キー部６や表示部７をプレーヤ１に設けることも可能である。

【０３２５】

【発明の効果】以上の説明で明らかなように、本発明によれば、簡単に持ち運びができて何時でも何処でもディジタルコンテンツを楽しむことが可能であり、また、ディジタルコンテンツのコピー或いは不当な使用への防御として十分運用に耐え、且つ経済的なシステムを構築す

ることをも可能である。

【図面の簡単な説明】

【図１】本発明の実施の形態のディジタルコンテンツ配布システムの全体構成を示すシステム構成図である。

【図２】本発明の実施の形態のシステムに対応するプレーヤの具体的構成を示すブロック回路図である。

【図３】本発明の実施の形態のシステムに対応する管理センタの具体的構成を示すブロック回路図である。

【図４】本実施の形態のシステムにおいてプレーヤの購入時の手順の説明に用いる図である。

【図５】本実施の形態のシステムにおいてディジタルコンテンツの検索からプレーヤ用の記憶メディアへのディジタルコンテンツのインストールまでの手順の説明に用いる図である。

【図６】実施の形態のシステムにおいて課金用のポイント情報の購入と当該ディジタルコンテンツを使用した場合の精算の手順の説明に用いる図である。

【図７】実施の形態のシステムにおいて課金代金の分配の手順の説明に用いる図である。

【図８】実施の形態のシステムにおいてポイント購入時のプレーヤにおける処理の流れを示すフローチャートである。

【図９】実施の形態のシステムにおいてポイント購入時のユーザ端末における処理の流れを示すフローチャートである。

【図１０】実施の形態のシステムにおいてポイント購入時の管理センタにおける処理の流れを示すフローチャートである。

【図１１】実施の形態のシステムにおいてポイント購入時の情報送受のシーケンスを示す図である。

【図１２】実施の形態のシステムにおいてディジタルコンテンツの入手時のプレーヤにおける処理の流れを示すフローチャートである。

【図１３】実施の形態のシステムにおいてディジタルコンテンツの入手時のユーザ端末における処理の流れを示すフローチャートである。

【図１４】実施の形態のシステムにおいてディジタルコンテンツの入手時の管理センタにおける処理の流れを示すフローチャートである。

【図１５】実施の形態のシステムにおいてディジタルコンテンツの入手時の情報送受のシーケンスを示す図である。

【図１６】実施の形態のシステムにおいてコンテンツ鍵及び使用条件の入手時のプレーヤにおける処理の流れを示すフローチャートである。

【図１７】実施の形態のシステムにおいてコンテンツ鍵及び使用条件の入手時のユーザ端末における処理の流れを示すフローチャートである。

【図１８】実施の形態のシステムにおいてコンテンツ鍵及び使用条件の入手時の管理センタにおける処理の流れを示すフローチャートである。

【図１９】実施の形態のシステムにおいてコンテンツ鍵及び使用条件の入手時の情報送受のシーケンスを示す図である。

【図２０】実施の形態のシステムにおいてプレーヤとユーザ端末を用いてディジタルコンテンツを実際に鑑賞する際の処理の流れを示すフローチャートである。

【図２１】実施の形態のシステムにおいてポイント使用情報返却時のプレーヤにおける処理の流れを示すフローチャートである。

【図２２】実施の形態のシステムにおいてポイント使用情報返却時のユーザ端末における処理の流れを示すフローチャートである。

【図２３】実施の形態のシステムにおいてポイント使用情報返却時の管理センタにおける処理の流れを示すフローチャートである。

【図２４】実施の形態のシステムにおいてポイント使用情報返却時の情報送受のシーケンスを示す図である。

【図２５】暗号化と圧縮の処理単位の最小公倍数にて復号化と伸長を行う際の処理の流れを示すフローチャートである。

【図２６】暗号化と圧縮の処理単位の最小公倍数の単位毎の復号化及び伸長処理を行う構成を示すブロック回路図である。

【図２７】セキュリティＩＤとしての乱数を発生する具体的構成を示すブロック回路図である。

【図２８】共通鍵を公開鍵暗号方式にて暗号化して送信する際に乱数が挿入される様子を説明するための図である。

【図２９】受信文から乱数が取り出されて正当性の確認がなされる様子を説明するための図である。

【図３０】システム側の機能を分割したときの各機関の説明に用いる図である。

【図３１】システム側の機能を分割した実施の形態において、ユーザのシステムへの加入時の流れの主要部を説明するための図である。

【図３２】システム側の機能を分割した実施の形態において、ポイント情報の購入や暗号化されたディジタルコンテンツの復号用の鍵等の入手時の情報の流れの主要部を説明するための図である。

【図３３】システム側の機能を分割した実施の形態において、コンテンツとコンテンツ鑑賞用の情報の流通の際の流れの主要部を説明するための図である。

【図３４】システム側の機能を分割した実施の形態において、コンテンツが実際に鑑賞されたときの精算の流れの主要部を説明するための図である。

【図３５】システム側の機能を分割した実施の形態において、コンテンツ展示配信機関の構成を示すブロック図である。

【図３６】システム側の機能を分割した実施の形態にお

いて、課金情報管理機関の構成を示すブロック図である。

【図３７】システム側の機能を分割した実施の形態において、システム管理機関の構成を示すブロック図である。

【図３８】システム側の機能を分割した実施の形態において、ユーザ側の構成を示すブロック図である。

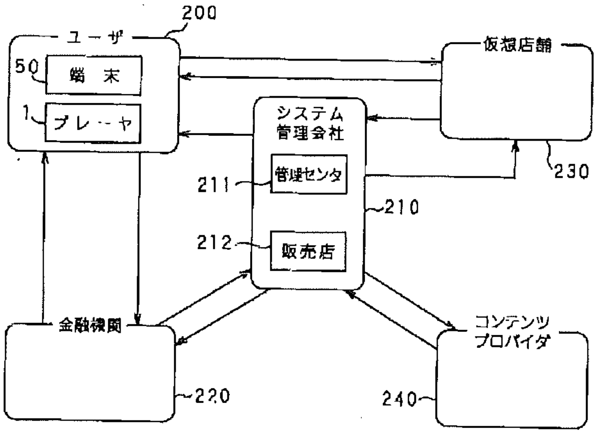
【図３９】プレーヤとユーザ端末の具体的な使用形態の一例の説明に用いる図である。

【図４０】プレーヤとユーザ端末の具体的な使用形態の他の例の説明に用いる図である。

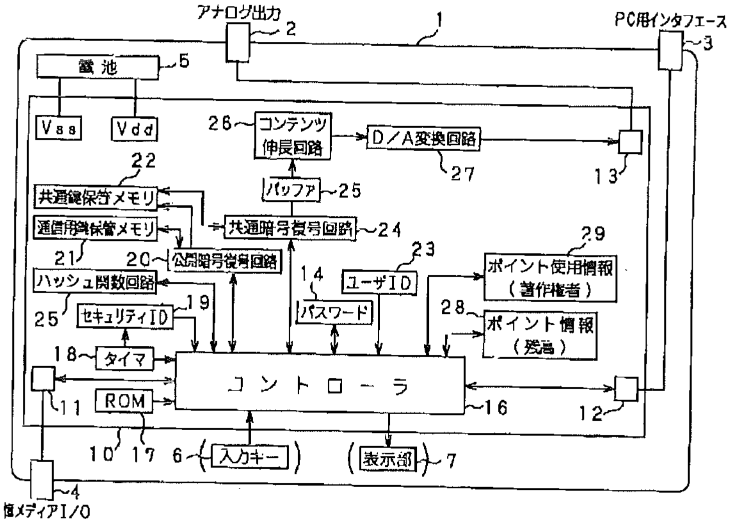
【符号の説明】

１　プレーヤ、　２　アナログ出力端子、　３　ＰＣ用インターフェース端子、　４　記憶メディア用Ｉ／Ｏ端子、　１６　コントローラ、　１９　セキュリティＩＤ発生回路、　２０　公開暗号復号回路、　２１　通信用鍵保管メモリ、２２　共通鍵保管メモリ、　２３　ユーザＩＤ格納メモリ、　２４　共通暗号復号回路、　２５　バッファメモリ、　２６　伸長回路、　２７　Ｄ／Ａ変換回路、　５０　ユーザ端末、　１００　コンテンツ管理機能ブロック、　１１０ユーザ管理機能ブロック、　１２０　使用情報管理機能ブロック、　１３０　管理機能ブロック、　２００　ユーザ側、　２１０　システム管理会社、　２１１管理センタ、　２２０　金融機関、　２３０　仮想店舗、　２４０　コンテンツプロバイダ

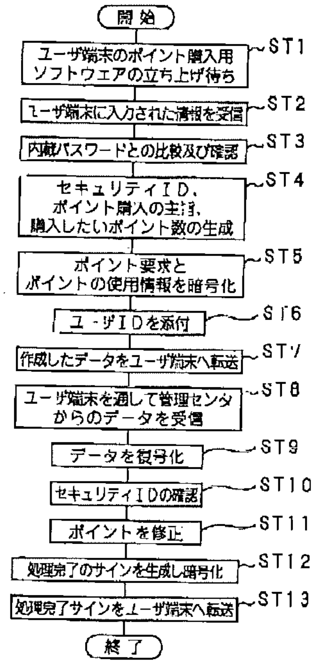
【図１】



【図２】

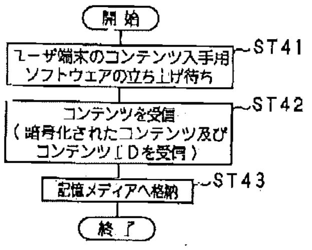


【図８】


開始
ユーザ端末のポイント購入用ソフトウェアの立ち上げ待ち
ST1
ユーザ端末に入力された情報を受信
ST2
内蔵パスワードとの比較及び確認
ST3
セキュリティID、ポイント購入の主旨、購入したいポイント数の生成
ST4
ポイント要求とポイントの使用情報を暗号化
ST5
ユーザIDを添付
ST6
作成したデータをユーザ端末へ転送
ST7
ユーザ端末を通して管理センタからのデータを受信
ST8
データを復号化
ST9
セキュリティIDの確認
ST10
ポイントを修正
ST11
処理完了のサインを生成し暗号化
ST12
処理完了サインをユーザ端末へ転送
ST13
終了

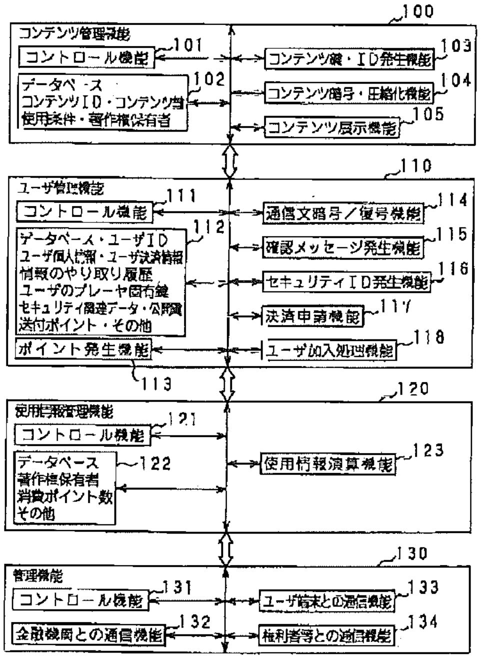
ポイント購入時のプレーヤのフローチャート

【図１２】

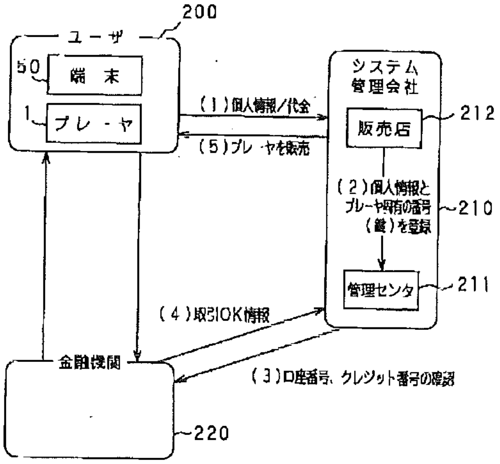

開始
ユーザ端末のコンテンツ入手用ソフトウェアの立ち上げ待ち
ST41
コンテンツを受信（暗号化されたコンテンツ及びコンテンツIDを受信）
ST42
記憶メディアへ格納
ST43
終了

コンテンツ入手時のプレーヤのフローチャート

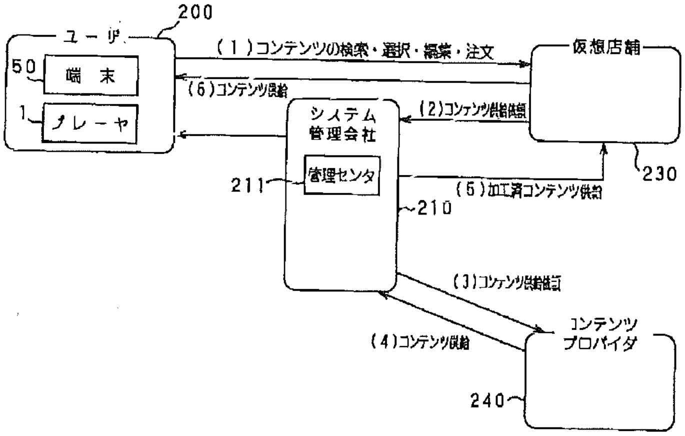
【図３】



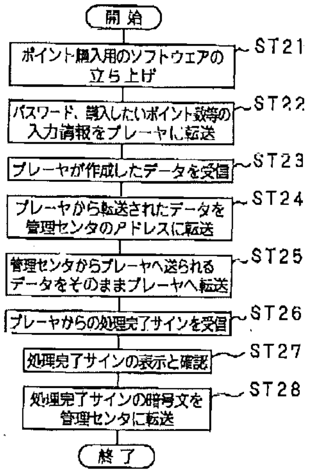
【図４】



【図５】

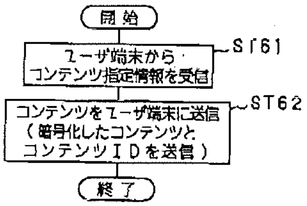


【図９】


開始
ポイント購入用のソフトウェアの立ち上げ
ST21
パスワード、購入したいポイント数等の入力情報をプレーヤに転送
ST22
プレーヤが作成したデータを受信
ST23
プレーヤから転送されたデータを管理センタのアドレスに転送
ST24
管理センタからプレーヤへ送られるデータをそのままプレーヤへ転送
ST25
プレーヤからの処理完了サインを受信
ST26
処理完了サインの表示と確認
ST27
処理完了サインの暗号文を管理センタに転送
ST28
終了

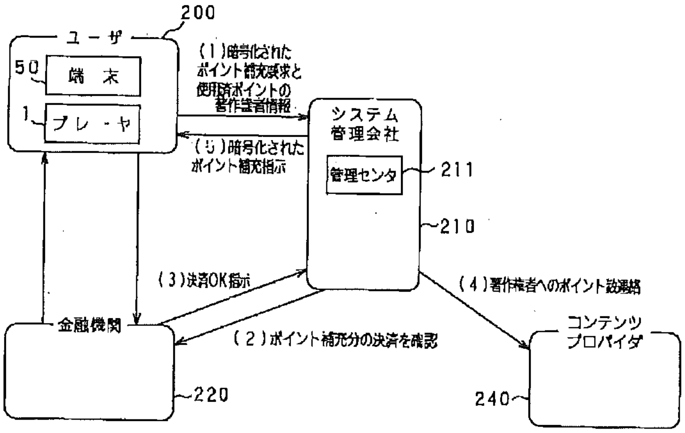
ポイント購入時のユーザ端末のフローチャート

【図１４】

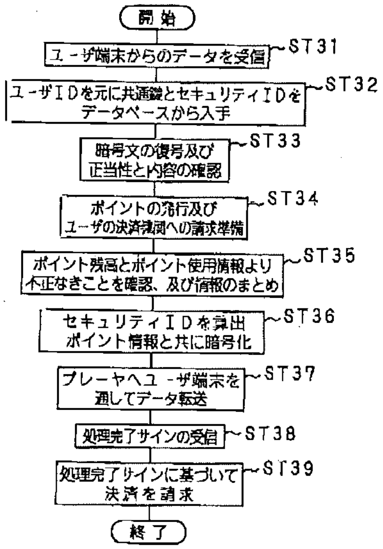


コンテンツ入手時の管理センタのフローチャート

【図６】

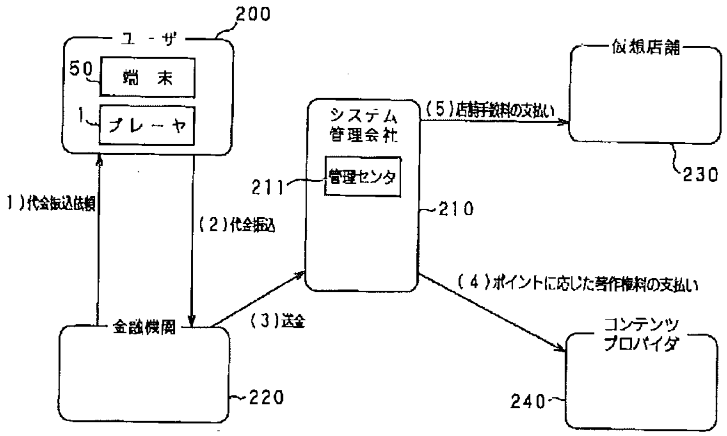


【図１０】

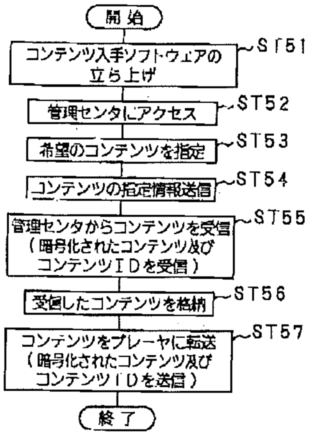


ポイント購入時の管理センタのフローチャード

【図７】

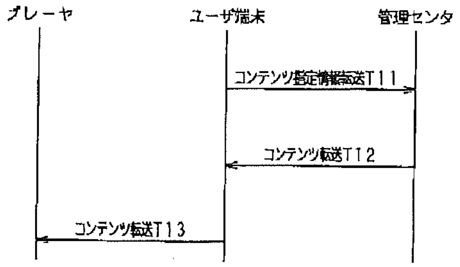


【図１３】



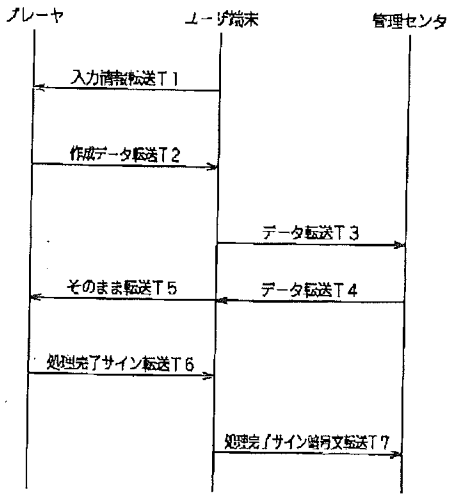
コンテンツ入手時のユーザ端末のフローチャ－ト

【図１５】



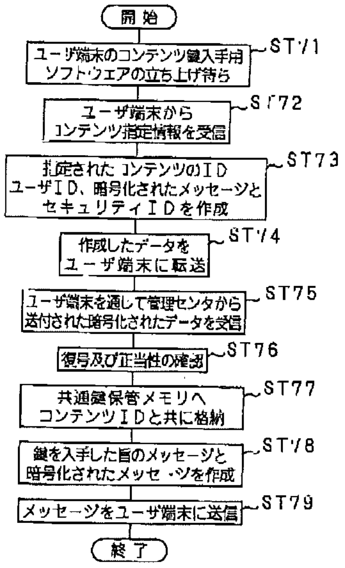
コンテンツ入手時のシーケンス

【図１１】



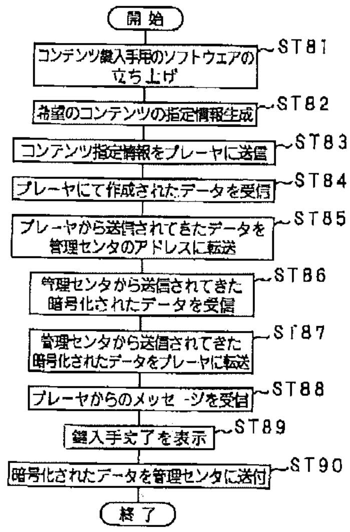
ポイント購入時のシーケンス

【図１６】


開始
ユーザ端末のコンテンツ鍵入手用ソフトウェアの立ち上げ待ち
ST71
ユーザ端末からコンテンツ指定情報を受信
ST72
指定されたコンテンツのＩＤ ユーザＩＤ、暗号化されたメッセージとセキュリティＩＤを作成
ST73
作成したデータをユーザ端末に転送
ST74
ユーザ端末を通して管理センタから送付された暗号化されたデータを受信
ST75
復号及び正当性の確認
ST76
共通鍵保管メモリへコンテンツＩＤと共に格納
ST77
鍵を入手した旨のメッセージと暗号化されたメッセ－ジを作成
ST78
メッセージをユーザ端末に送信
ST79
終了

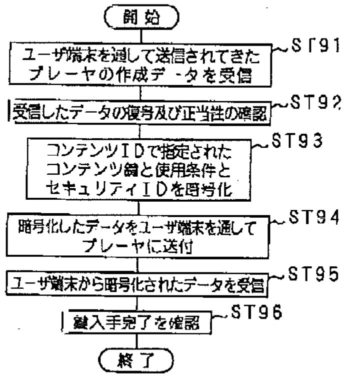
コンテンツ鍵・入手時のプレ－ヤのフローチャート

【図１７】


開始
コンテンツ鍵入手用のソフトウェアの立ち上げ
ST81
希望のコンテンツの指定情報生成
ST82
コンテンツ指定情報をプレーヤに送信
ST83
プレーヤにて作成されたデータを受信
ST84
プレーヤから送信されてきたデータを管理センタのアドレスに転送
ST85
管理センタから送信されてきた暗号化されたデータを受信
ST86
管理センタから送信されてきた暗号化されたデータをプレーヤに転送
ST87
プレーヤからのメッセ－ジを受信
ST88
鍵入手完了を表示
ST89
暗号化されたデータを管理センタに送付
ST90
終了

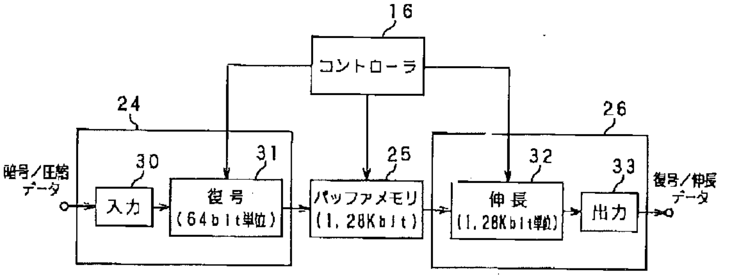
コンテンツ鍵・使用条件入手時のユ－ザ端末のフローチャ－ト

【図１８】

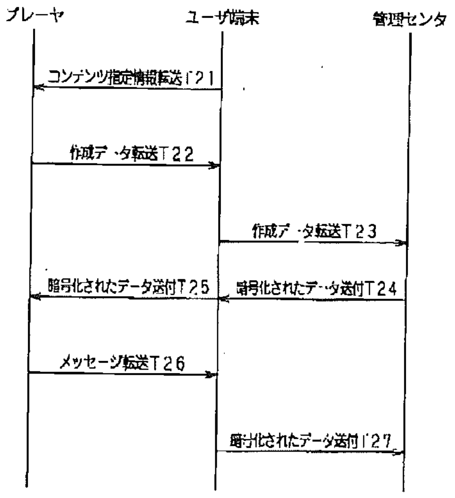

開始
ユーザ端末を通して送信されてきたプレーヤの作成デ－タを受信
ST91
受信したデータの復号及び正当性の確認
ST92
コンテンツＩＤで指定されたコンテンツ鍵と使用条件とセキュリティＩＤを暗号化
ST93
暗号化したデータをユーザ端末を通してプレーヤに送付
ST94
ユーザ端末から暗号化されたデータを受信
ST95
鍵入手完了を確認
ST96
終了

コンテンツ鍵・使用条件入手時の管理センタのフローチャート

【図２６】

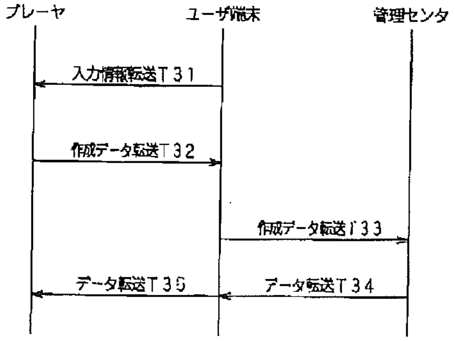


【図19】


プレーヤ
ユーザ端末
管理センタ
コンテンツ指定情報転送T21
作成デ・タ転送T22
作成デ・タ転送T23
暗号化されたデ・タ送付T24
暗号化されたデータ送付T25
メッセージ転送T26
暗号化されたデータ送付T27

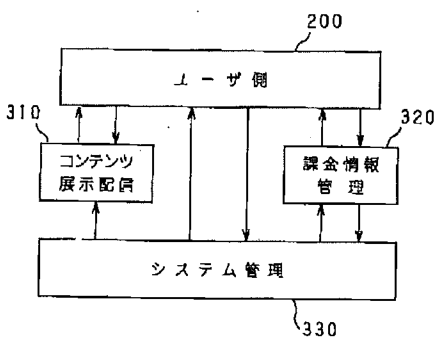
コンテンツ鍵・使用条件入手時のシーケンス

【図24】

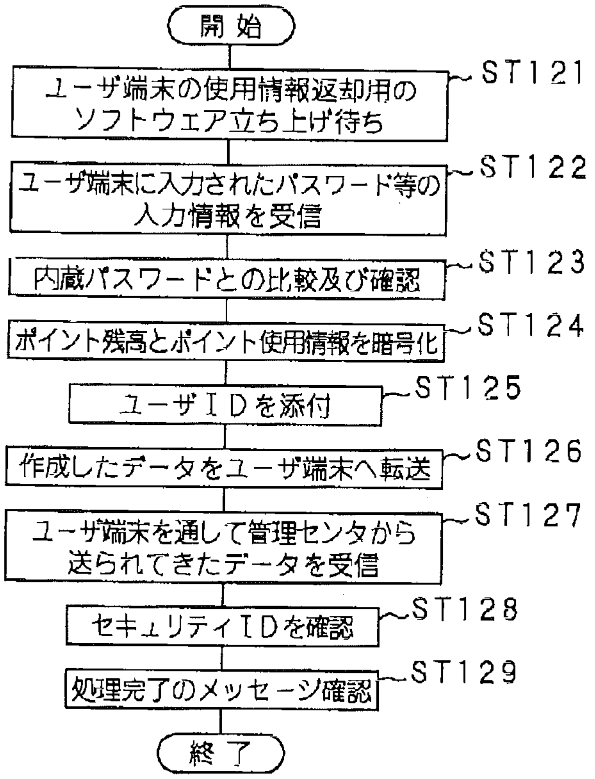


使用情報返却時のシーケンス

【図30】

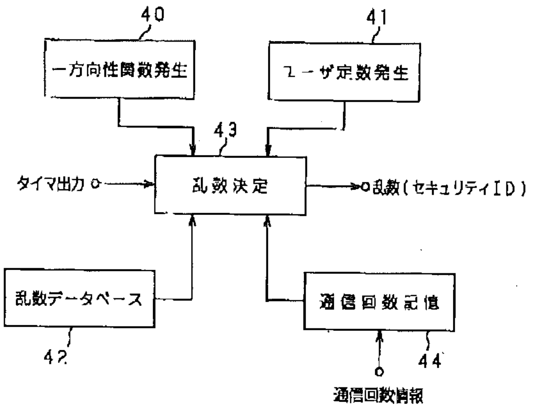


【図21】

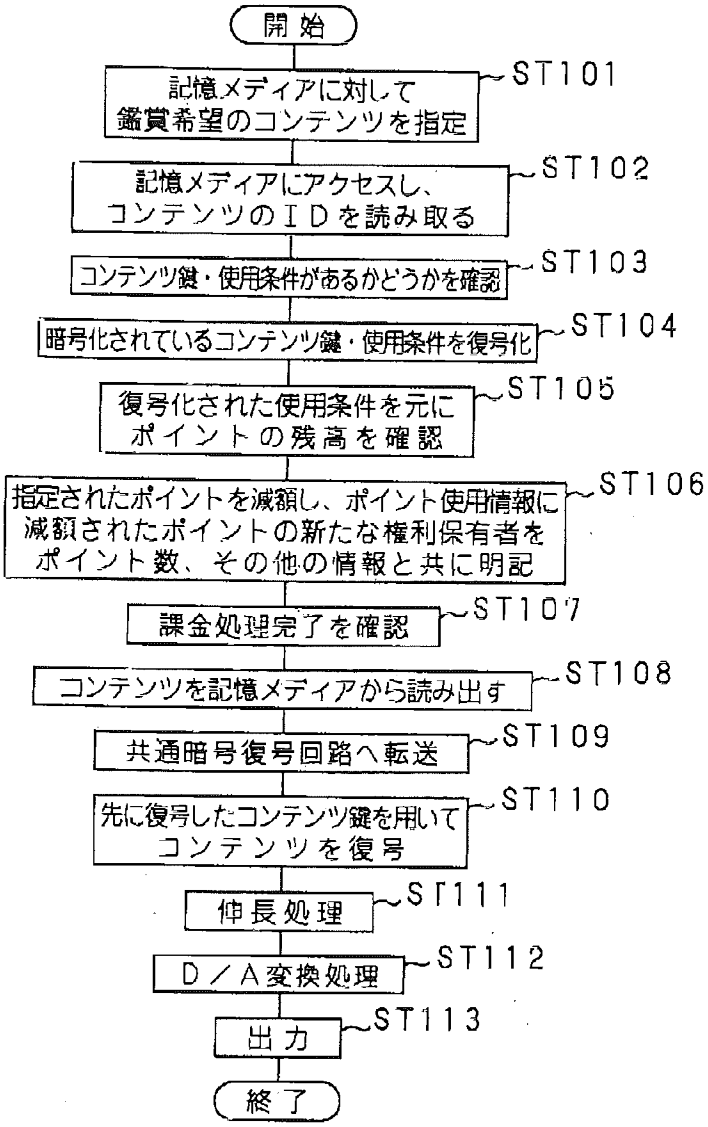

開始
ユーザ端末の使用情報返却用のソフトウェア立ち上げ待ち
ST121
ユーザ端末に入力されたパスワード等の入力情報を受信
ST122
内蔵パスワードとの比較及び確認
ST123
ポイント残高とポイント使用情報を暗号化
ST124
ユーザIDを添付
ST125
作成したデータをユーザ端末へ転送
ST126
ユーザ端末を通して管理センタから送られてきたデータを受信
ST127
セキュリティIDを確認
ST128
処理完了のメッセージ確認
ST129
終了

使用情報返却時のプレーヤのフローチャート

【図27】

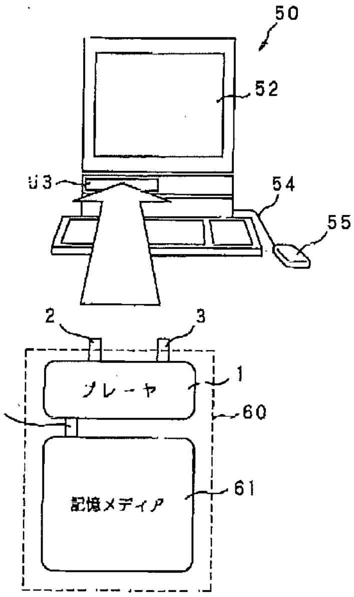


【図20】

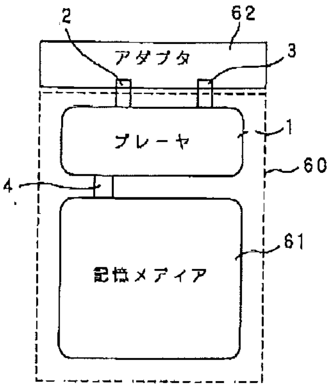

開始
記憶メディアに対して鑑賞希望のコンテンツを指定
ST101
記憶メディアにアクセスし、コンテンツのＩＤを読み取る
ST102
コンテンツ鍵・使用条件があるかどうかを確認
ST103
暗号化されているコンテンツ鍵・使用条件を復号化
ST104
復号化された使用条件を元にポイントの残高を確認
ST105
指定されたポイントを減額し、ポイント使用情報に減額されたポイントの新たな権利保有者をポイント数、その他の情報と共に明記
ST106
課金処理完了を確認
ST107
コンテンツを記憶メディアから読み出す
ST108
共通暗号復号回路へ転送
ST109
先に復号したコンテンツ鍵を用いてコンテンツを復号
ST110
伸長処理
ST111
Ｄ／Ａ変換処理
ST112
出力
ST113
終了

コンテンツ鑑賞時のプレーヤのフローチャート

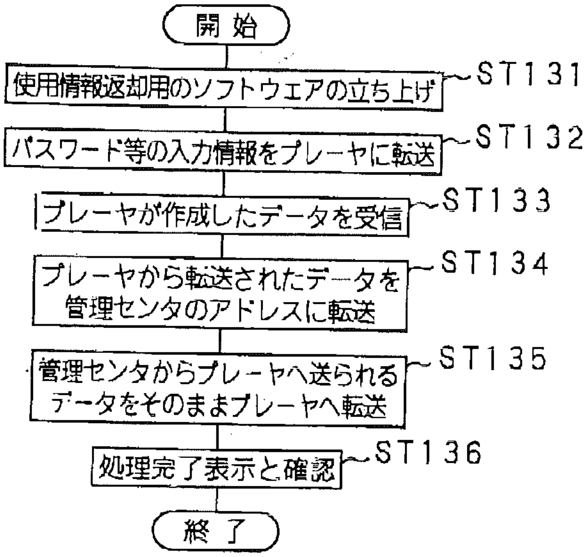
【図39】



【図40】

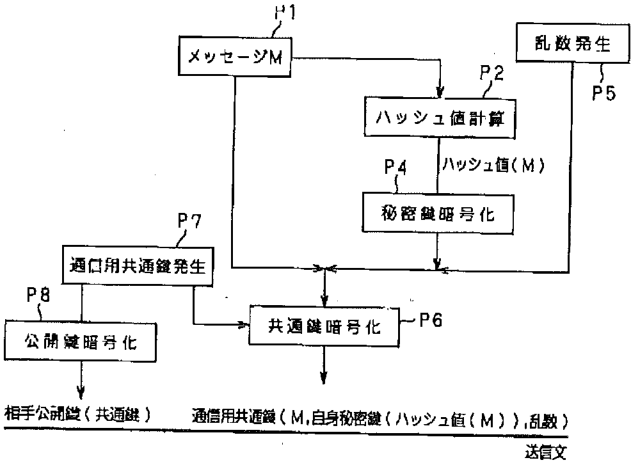


【図22】

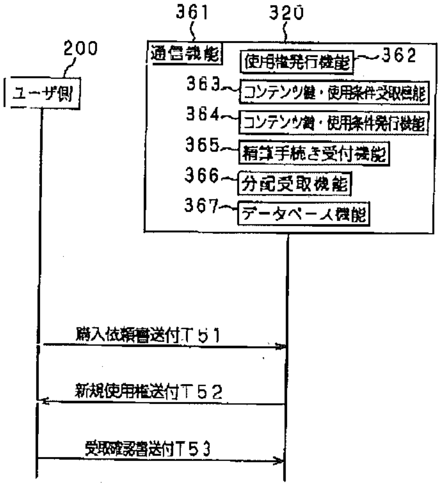

開 始
使用情報返却用のソフトウェアの立ち上げ
ST131
パスワード等の入力情報をプレーヤに転送
ST132
プレーヤが作成したデータを受信
ST133
プレーヤから転送されたデータを
管理センタのアドレスに転送
ST134
管理センタからプレーヤへ送られる
データをそのままプレーヤへ転送
ST135
処理完了表示と確認
ST136
終 了

使用情報返却時のユーザ端末のフローチャート

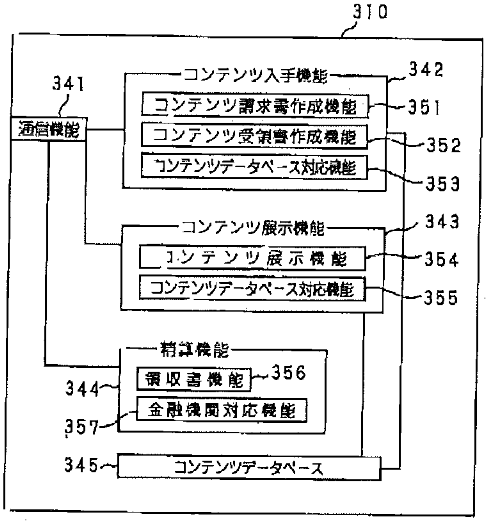
【図28】



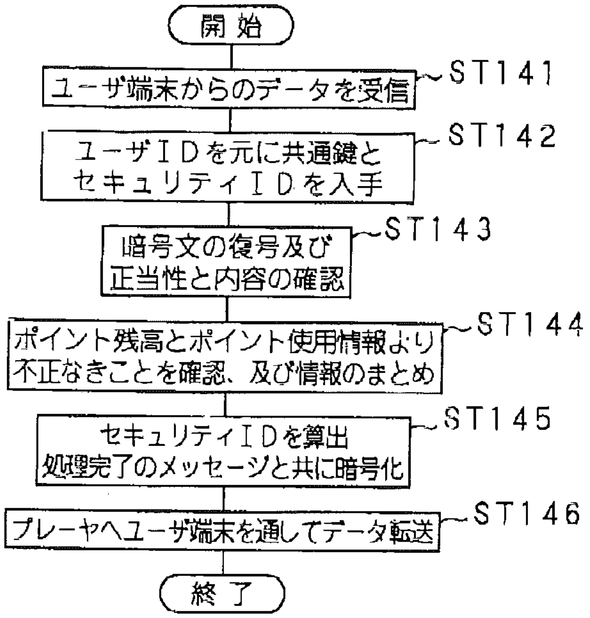
【図32】



【図35】

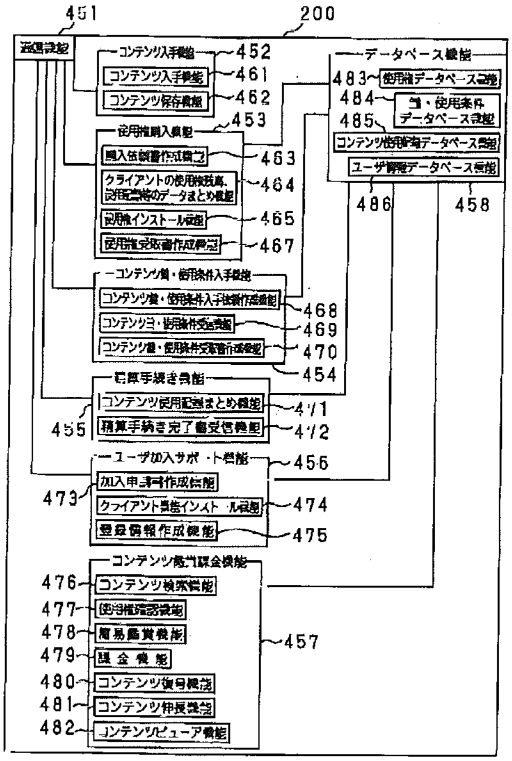


【図２３】

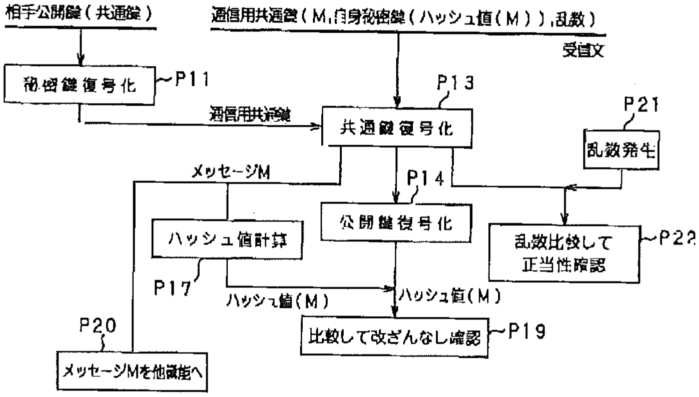

開 始
ユーザ端末からのデータを受信
ST141
ユーザＩＤを元に共通鍵と
セキュリティＩＤを入手
ST142
暗号文の復号及び
正当性と内容の確認
ST143
ポイント残高とポイント使用情報より
不正なきことを確認、及び情報のまとめ
ST144
セキュリティＩＤを算出
処理完了のメッセージと共に暗号化
ST145
プレーヤへユーザ端末を通してデータ転送
ST146
終 了

使用情報返却時の管理センタのフローチャート

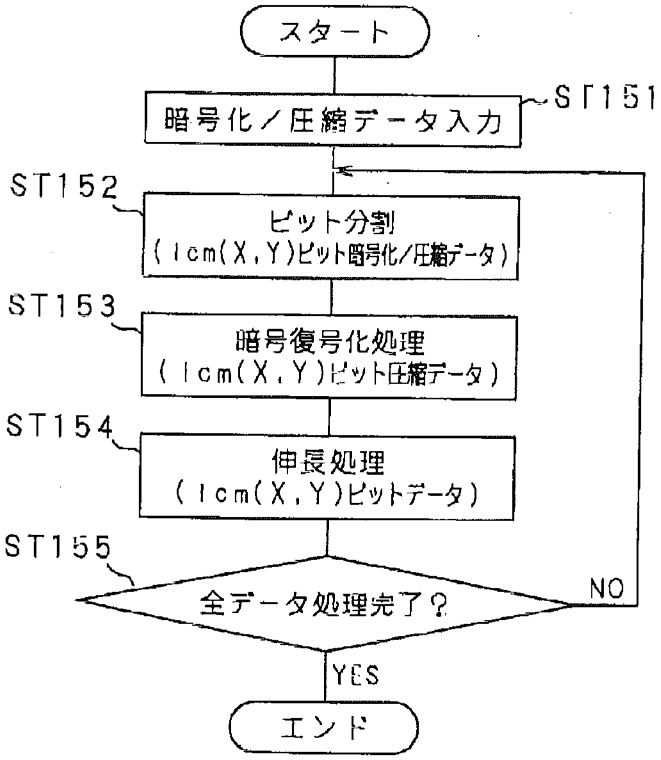
【図３８】



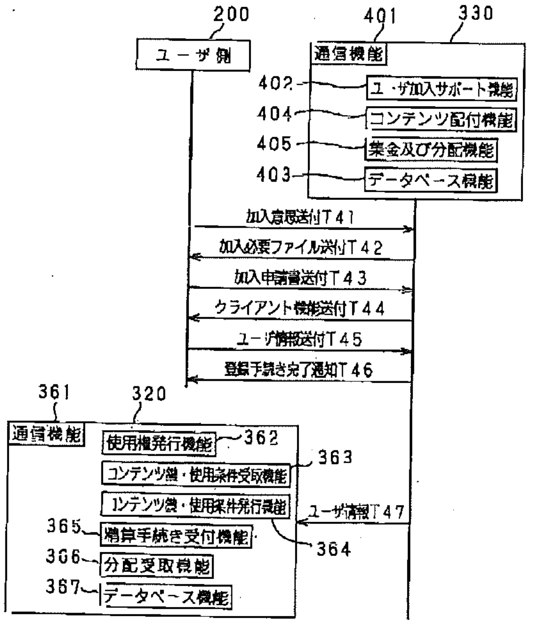
【図２９】



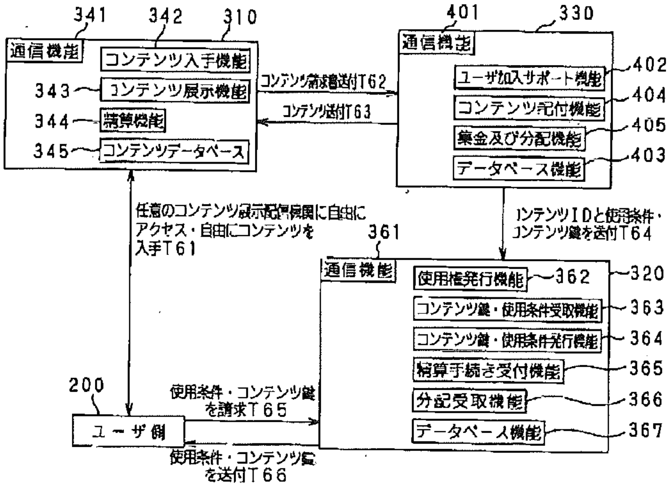
【図２５】


スタート
暗号化／圧縮データ入力
ST151
ST152
ビット分割
（1cm(X,Y)ビット暗号化/圧縮データ）
ST153
暗号復号化処理
（1cm(X,Y)ビット圧縮データ）
ST154
伸長処理
（1cm(X,Y)ビットデータ）
ST155
全データ処理完了？
NO
YES
エンド

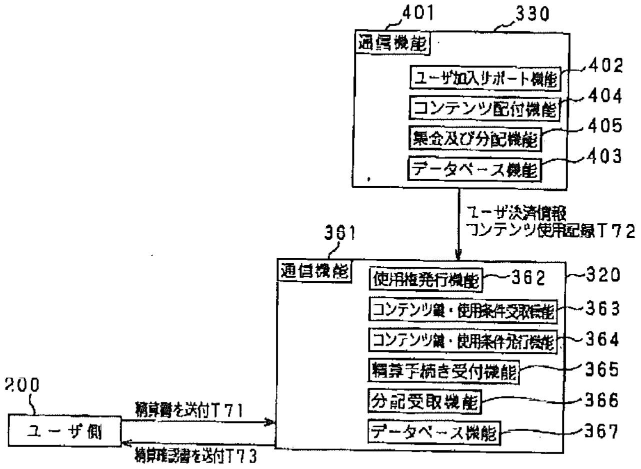
【図３１】


200
ユーザ側
401
330
通信機能
402
ユ・ザ加入サポート機能
404
コンテンツ配付機能
405
集金及び分配機能
403
データベース機能
加入意思送付T41
加入必要ファイル送付T42
加入申請書送付T43
クライアント機能送付T44
ユーザ情報送付T45
登録手続き完了通知T46
361
320
通信機能
使用権発行機能
362
コンテンツ鍵・使用条件受取機能
363
コンテンツ鍵・使用条件発行機能
ユーザ情報T47
365
精算手続き受付機能
364
366
分配受取機能
367
データベース機能

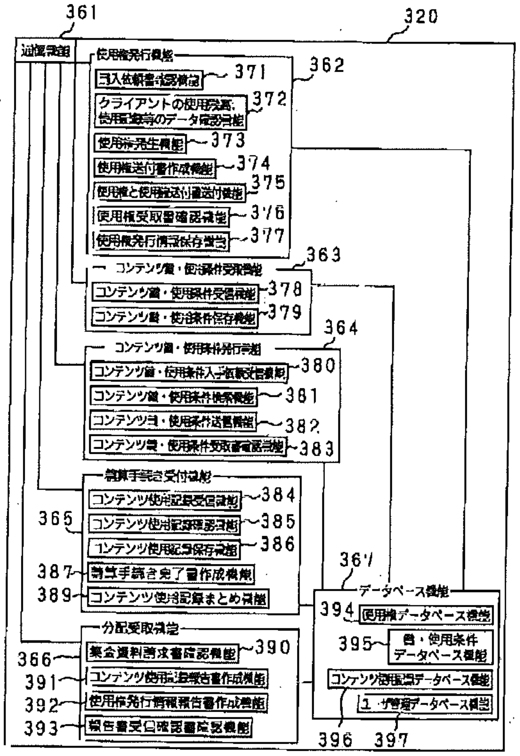
【図３３】


341
342
310
通信機能
コンテンツ入手機能
343
コンテンツ展示機能
344
精算機能
345
コンテンツデータベース
コンテンツ請求書送付T62
コンテンツ送付T63
401
330
通信機能
ユーザ加入サポート機能
402
コンテンツ配付機能
404
集金及び分配機能
405
データベース機能
403
任意のコンテンツ展示配信機関に自由に
アクセス・自由にコンテンツを
入手T61
コンテンツIDと使用条件・
コンテンツ鍵を送付T64
361
通信機能
使用権発行機能
362
320
コンテンツ鍵・使用条件受取機能
363
コンテンツ鍵・使用条件発行機能
364
精算手続き受付機能
365
分配受取機能
366
データベース機能
367
200
ユーザ側
使用条件・コンテンツ鍵
を請求T65
使用条件・コンテンツ鍵
を送付T66

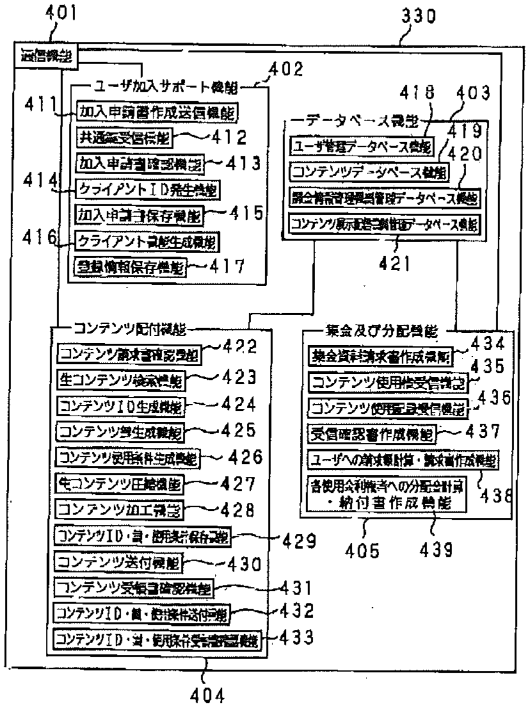
【図３４】


401
330
通信機能
ユーザ加入サポート機能
402
コンテンツ配付機能
404
集金及び分配機能
405
データベース機能
403
ユーザ決済情報
コンテンツ使用記録T72
361
通信機能
使用権発行機能
362
320
コンテンツ鍵・使用条件受取機能
363
コンテンツ鍵・使用条件発行機能
364
精算手続き受付機能
365
分配受取機能
366
データベース機能
367
200
ユーザ側
精算書を送付T71
精算確認書を送付T73

【図３６】


361
320
通信機能
使用権発行機能
購入依頼書確認機能
371
362
クライアントの使用残高、使用記録等のデータ確認機能
372
使用権発生機能
373
使用権送付書作成機能
374
使用権と使用権送付書送付機能
375
使用権受取書確認機能
376
使用権発行情報保存機能
377
コンテンツ鍵・使用条件受取機能
363
コンテンツ鍵・使用条件受信機能
378
コンテンツ鍵・使用条件保存機能
379
コンテンツ鍵・使用条件発行機能
364
コンテンツ鍵・使用条件入手依頼受信機能
380
コンテンツ鍵・使用条件検索機能
381
コンテンツ鍵・使用条件送信機能
382
コンテンツ鍵・使用条件受取書確認機能
383
精算手続き受付機能
コンテンツ使用記録受信機能
384
365
コンテンツ使用記録確認機能
385
コンテンツ使用記録保存機能
386
367
387
精算手続き完了書作成機能
389
コンテンツ使用記録まとめ機能
データベース機能
394
使用権データベース機能
395
鍵・使用条件データベース機能
分配受取機能
366
集金資料請求書確認機能
390
391
コンテンツ使用記録報告書作成機能
392
使用権発行情報報告書作成機能
393
報告書受信確認書確認機能
コンテンツ使用記録データベース機能
ユーザ管理データベース機能
396
397

【図３７】



フロントページの続き

(51)Int.Cl.^{6}		識別記号	ＦＩ		
Ｈ０４Ｌ	9/08		Ｈ０４Ｍ	15/00	Ｚ
	12/14		Ｇ０６Ｆ	15/21	Ｚ
Ｈ０４Ｍ	15/00		Ｈ０４Ｌ	9/00	６０１Ｅ
					６０１Ａ
				11/02	Ｆ

TRANSLATION DECLARATION

I, Daniel Dowdle, hereby declare:

1. I am a translator at MultiLing Corporation, a professional translation company incorporated in Delaware with its principal office located at 180 North University Avenue, Suite 600, Provo, Utah 84601-4474.

2. I am competent to translate between the Japanese and English languages.

3. At the request of Ropes & Gray, LLP, I translated JP Patent Application Publication No. H10-269289 (Maari) from Japanese to English.

4. To the best of my knowledge and belief, the attached English language document is a true, complete, and correct translation of JP Patent Application Publication No. H10-269289 (Maari).

5. I make this declaration of my own personal knowledge. If called to testify as to the truth of the matters stated herein, I could and would testify competently.

6. I declare under penalty of perjury that the foregoing is true and correct.

Executed this Tenth day of March, 2014, at Provo, Utah.

Daniel Dowdle
MultiLing Corporation
180 North University Avenue
Suite 600
Provo, Utah 84601-4474

(19) Japanese Patent Office (JP)	**(12) Publication of Unexamined Patent Application (A)**	**(11) Publication Number: H10-269289** (43) Date of Publication: October 9, 1998

(51) Int. Cl.^{6}	Identification Code	FI		
G06F 17/60		G06F	15/21	330
1/00	370		1/00	370 F
9/06	550		9/06	550 Z
15/00	330		15/00	330 Z
G09C 1/00	660	G09C	1/00	660 F
H04L 9/08		H04M	15/00	Z
12/14		G06F	15/21	Z
H04M 15/00		H04L	9/00	601 E
				601 A
			11/02	F

Examination Request Status: Not Yet Requested. No. of Claims: 37, OL (39 pages total)

(21) Application number: H09-74182

(22) Date of Application: March 26, 1997

(71) Applicant: 000002185
Sony Corporation
6-7-35 Kitashinagawa, Shinagawa-ku, Tokyo

(72) Inventor: Koichi MAARI
C/O Sony Corporation,
6-7-35 Kitashinagawa, Shinagawa-ku, Tokyo

(74) Agent: Akira KOIKE, Attorney (and two others)

(54) [Title of the Invention] METHOD OF CONTROLLING DIGITAL CONTENT DISTRIBUTION, A METHOD OF REPRODUCING DIGITAL CONTENT, AND AN APPARATUS USING THE SAME

(57) [Abstract]
[Problem]

To build a system that is portable, enables digital content to be enjoyed anywhere and anytime, provides adequate protection against copying and unauthorized use of the digital content, and is economical.

[Resolution Means]

A public-key encryption/decryption circuit 20 for decrypting an encrypted content key and encrypting a session key; a common key storage memory 22 for storing the content key and the session key; a communication key storage memory 21 for storing key information for a public-key encryption system; a point information storage memory 29 for storing point information; a point usage information storage memory 28 for storing point usage information; a common key encryption/decryption circuit 24 for decrypting encrypted digital content, decrypting encrypted point information, and encrypting point usage information; a decompressing circuit 26 for decompressing compressed digital content; and a D/A conversion circuit 27 for subjecting digital content to digital-to-analog conversion are integrated on a single chip.

What is Claimed is:

[Claim 1]

A method for controlling digital content distribution, the method comprising steps of:

digital content manipulation for encrypting and compressing digital content using a content key for each piece of relevant digital content;

content transmission for transmitting the manipulated digital content in accordance with a digital content transmission request from a communication partner; content key transmission for encrypting a content key for use in decrypting the manipulated digital content and transmitting same in accordance with a content key transmission request from a communication partner;

billing information transmission for encrypting billing information that is decremented each time the manipulated digital content is decrypted and transmitting same in accordance with a billing information transmission request from a communication partner;

content usage information reception for receiving and decrypting encrypted content usage information transmitted from a communication partner; and

usage fee distribution for distributing a usage fee, which is collected on the basis of the content usage information, to a proprietor of the digital content.

[Claim 2]

The method for controlling digital content distribution according to claim 1, wherein the content key is a common key.

[Claim 3]

The method for controlling digital content distribution according to claim 1, wherein the content key is encrypted using a public key of a communication partner.

[Claim 4]

The method for controlling digital content distribution according to claim 1 comprising a step of common key decryption for receiving and decrypting an encrypted common key transmitted from a communication partner.

[Claim 5]

The method for controlling digital content distribution according to claim 4, wherein the common key is a session key.

[Claim 6]

The method for controlling digital content distribution according to claim 4, wherein billing information is encrypted using the common key in the billing information transmission step.

[Claim 7]

The method for controlling digital content distribution according to claim 4, wherein the common key is used in decrypting the encrypted content usage information in the content usage information receiving step.

[Claim 8]

The method for controlling digital content distribution according to claim 1, wherein the encrypted content usage information transmitted from the communication partner in connection with the billing information transmission request from the communication partner is received in the content usage information receiving step.

[Claim 9]

The method for controlling digital content distribution according to claim 1, wherein information indicating a content use condition is transmitted together with the billing information in the billing information transmission step.

[Claim 10]

A method for reproducing digital content, the method comprising steps of:

content reception for receiving and storing digital content manipulated using encryption and compression processing;

content key request information generation for generating content key request information for requesting a content key required for decrypting the manipulated digital content;

content key request information transmission for encrypting and transmitting the content key request information;

content key reception for receiving a content key sent in accordance with the content key request;

content key decryption for decrypting the encryption that has been applied to the content key;

content key storage for storing either the encrypted content key or the post-decryption content key;

content decryption for decrypting the manipulated digital content using the content key;

billing information request information generation for generating billing information request information for requesting billing information that is decremented each time the manipulated digital content is decrypted;

billing information request information transmission for encrypting and transmitting the billing information request information;

billing information reception for receiving billing information transmitted in accordance with the billing information request, decrypting the encryption applied to the billing information, and storing same;

content decompression for decompressing the manipulated digital content;
content usage information storage for generating and storing content usage

information that corresponds to the decryption of the manipulated digital content; and

content usage information transmission for encrypting and transmitting the content usage information.

[Claim 11]

The method for reproducing digital content according to claim 10, wherein, in the content usage information storage step, a balance in the stored billing information is confirmed, the stored billing information is decremented in accordance with the decryption of the manipulated digital content, and content usage information including at least an amount of the billing information decrement is generated.

[Claim 12]

The method for reproducing digital content according to claim 10, comprising a step of digital/analog conversion for subjecting the decrypted and decompressed digital content to digital-to-analog conversion.

[Claim 13]

The method for reproducing digital content according to claim 10, wherein the manipulated digital content is stored in an external storage medium in the content reception step.

[Claim 14]

The method for reproducing digital content according to claim 10, wherein the content key is a common key.

[Claim 15]

The method for reproducing digital content according to claim 10, wherein the content key is decrypted using a unique secret key in the content key decryption step.

[Claim 16]

The method for reproducing digital content according to claim 10, comprising a step of common key transmission for generating a common key, and encrypting and transmitting the common key.

[Claim 17]

The method for reproducing digital content according to claim 16, wherein a session key is generated as the common key in the common key transmission step.

[Claim 18]

The method for reproducing digital content according to claim 16, wherein the billing information request information is encrypted using the common key in the billing information request information transmission step.

[Claim 19]

The method for reproducing digital content according to claim 16, wherein the common key is used in the encryption of the content usage information in the content usage information transmission step.

[Claim 20]

The method for reproducing digital content according to claim 10, wherein, in the content usage information transmission step, the encrypted content usage information is transmitted in connection with the billing information request resulting from the billing information request information generation step.

[Claim 21]

The method for reproducing digital content according to claim 10, wherein information indicating a use condition for content encrypted and transmitted together with the billing information is also received in the billing information reception step.

[Claim 22]

A digital content reproducing apparatus, comprising:

data communication means for performing data communications;

content storage control means for receiving digital content manipulated using encryption and compression processing and storing same in a storage medium;

content key decryption means for decrypting an encrypted content key; content key storage means for storing either the encrypted content key or the post-decryption content key;

content decryption means for decrypting the manipulated digital content using the content key;

billing information decryption means for decrypting the encryption applied to billing information that is decremented each time the manipulated digital content is decrypted;

billing information storage means for storing the decrypted billing information; content decompression means for decompressing the manipulated digital content;

content usage information generation means for generating content usage information that corresponds to the decryption of the manipulated digital content;

content usage information storage means for storing the content usage information; and

content usage information encryption means for encrypting the content usage information.

[Claim 23]

The digital content reproducing apparatus according to claim 22, comprising:

content key request information encryption means for encrypting content key request information for requesting a content key required for the decryption of the manipulated digital content; and

billing information request information encryption means for encrypting billing information request information for requesting billing information that is decremented each time the manipulated digital content is decrypted.

[Claim 24]

The digital content reproducing apparatus according to claim 22, wherein the content usage information generation means check a balance of billing information stored in the billing information storage means, decrement the stored billing information in accordance with the decryption of the manipulated digital content, and generate content usage information including at least an amount of the billing information decrement.

[Claim 25]

The digital content reproducing apparatus according to claim 22, comprising digital/analog conversion means for subjecting the decrypted and decompressed digital content to digital-to-analog conversion.

[Claim 26]

The digital content reproducing apparatus according to claim 22, wherein the content storage control means store the manipulated digital content in an external storage medium.

[Claim 27]

The digital content reproducing apparatus according to claim 22, wherein the content key is a common key.

[Claim 28]

The digital content reproducing apparatus according to claim 22, comprising unique key storage means for storing an apparatus-unique key, wherein in the content key decryption step, the encrypted content key is decrypted using an apparatus-unique secret key stored in the unique key storage means.

[Claim 29]

The digital content reproducing apparatus according to claim 22, comprising common key generation means for generating a common key, and common key encryption means for encrypting the common key.

[Claim 30]

The digital content reproducing apparatus according to claim 29, wherein the common key generation means generate a session key as the common key.

[Claim 31]

The digital content reproducing apparatus according to claim 29, wherein the billing information decryption means decrypt the billing information using the common key.

[Claim 32]

The digital content reproducing apparatus according to claim 29, wherein the content usage information encryption means encrypt the content usage information using the common key.

[Claim 33]

The digital content reproducing apparatus according to claim 22, wherein the content usage information encryption means encrypt the content usage information in connection with the encryption of the billing information request information by the billing information request information encryption means.

[Claim 34]

The digital content reproducing apparatus according to claim 22, wherein information indicating a use condition for encrypted content is also decrypted together with the billing information in the billing information decryption step.

[Claim 35]

The digital content reproducing apparatus according to claim 22 which is configured to be portable.

[Claim 36]

The digital content reproducing apparatus according to claim 22 comprising a card-shaped enclosure.

[Claim 37]

The digital content reproducing apparatus according to claim 22 comprising an integrated circuit.

[Detailed Description of the Invention]
[0001]
[Technical Field of the Invention]

The present invention generally relates to a digital content distribution control method suitable for a system for distributing digital content such as audio data and video data, for example, and for billing according to a usage quantity of the digital content, a digital content reproducing method, and an apparatus using the digital content reproducing method.

[0002]
[Background Art]

A software control method disclosed in Japanese Examined Patent Application Publication No. H6-19707, a software usage control method disclosed in Japanese Examined Patent Application Publication No. H6-28030, and a software control method disclosed in Japanese Examined Patent Application Publication No. H6-95302, for example, are known as advantageous techniques for simplifying the distribution of digital content such as computer programs, audio data, video data, and the like, exploiting potential demand for digital content, and expanding the market for this field. The software control method disclosed in Japanese Examined Patent Application Publication No. H6-19707 is designed so that, when using software such as computer programs and video data, which are intangible assets, it is possible for software proprietors and the like to ascertain the usage status of the software. Furthermore, the software usage control method disclosed in Japanese Examined Patent Application Publication No. H6-28030 is designed so that, when using software such as computer programs, video data, and the like, which are intangible assets, purchase prices are set for paid programs (available for use free-of-charge after purchase), and data indicating an amount of money available for purchase of paid programs is provided in a computer system. When purchasing a paid program, these programs are registered in a table as the names of software programs available in this computer system, and data indicating the amount of money available to purchase paid programs is decremented by the price of the purchased software. When deleting registered software from this table, the data indicating the amount of money available to purchase paid programs is incremented and updated in accordance with the circumstances. In addition, the software control method disclosed in Japanese Examined Patent Application Publication No. H6-95302 is designed to be effective in a system in a case where, in order to collect utilization fees according to the actual amount of usage (the number of times or the length of time used, or the like) for a paid program when using software such as computer programs, video data, and the like, which are intangible assets, the identifications of the programs that were used, "user identification codes, and fees are recorded" in advance, and, by retrieving this record, the program proprietor is able to ascertain the utilization fees for the programs owned by the program proprietor and to collect the utilization fees in accordance with the usage of the programs.

[0003]

[Problems to be Solved by the Invention]

However, the above-mentioned system for distributing digital content through a network is considered to be operated only on personal computers. Therefore, there is no system that is portable with ease and allows the digital content to be enjoyed anytime and anywhere.

[0004]The above-mentioned disclosed technique is advantageous in exploiting potential demands for digital content and expanding the market. However, this technique is insufficient in protecting digital content from illegal duplication or unauthorized use and provides no economical system.

[0005]

Accordingly, the present invention was conceived in light of the foregoing, and an object thereof is to provide a method of controlling digital content distribution, and a method and apparatus for reproducing digital content that make it possible to build a system that is portable, enables digital content to be enjoyed anywhere and anytime, provides adequate protection against the copying and unauthorized use of the digital content, and is economical.

[0006]

[Means to Solve the Problem]

According to the present invention, the digital content distributing side manipulates digital contents by encrypting and compressing the same, transmits the manipulated digital content, an encrypted content key, and encrypted billing information to a communication partner, and distributes to digital content proprietors the digital content usage fees collected based on digital content usage information received from the communication partner. On the other hand, the digital content reproducing side decrypts and decompresses the manipulated digital content for reproduction by the content key. At the same time, the reproducing side decrements the billing information according to the use of the content and generates content usage information to be transmitted to the content distributing side. Additionally, the digital content reproducing apparatus associated with the present invention is made portable. The present invention thereby solves the above-mentioned problems.

[0007]

[Description of the Preferred Embodiments]

The preferred embodiments of the present invention will be described below while referring to the drawings.

[0008]

Before describing specific contents and constitutions of a digital content distributing method, a digital content reproducing method, and a digital content reproducing apparatus according to the present invention, an outline constitution of an entire system to which the present invention is applied and an operating method of this system will be described for easier understanding of the above-mentioned distributing method, reproducing method and reproducing apparatus with reference to FIGS. 1 through 7.

[0009]

FIG. 1 shows a schematic constitution of the entire system.

[0010]

In FIG. 1, it is assumed that a user 200 has a digital content reproducing apparatus (hereafter referred to as a player 1) associated with the present invention and a so-called personal computer (hereafter referred to as a user terminal 50).

[0011]

The user terminal 50 is an ordinary personal computer that stores various software as application software, to be described later, for use in the present invention and connects to a display device providing display means, a speaker serving as sounding means, and a keyboard and a mouse serving as information inputting means. The user terminal 50 can be connected to a system administration company 210 through a network, for example, and also has interface means between the user terminal and the player 1 that allows data to be transferred.

[0012]

The player 1 has a constitution as shown in FIG. 2, for example.

[0013]

Details of the constitution shown in FIG. 2 will be described later. The player 1, as a main component of the processing route through which digital content flows, at least has a common key encryption/decryption circuit 24 for decrypting encrypted digital content by use of a content key, a decompressing circuit 26 serving as decompressing means for decompressing compressed digital content, and a D/A converting circuit 27 for converting digital data into an analog signal. It should be noted that the term decryption as used hereinbelow refers to undoing encryption.

[0014]

This player 1, as a main component for handling proprietary information and information indicating the usage status of digital content to be used (these pieces of information are hereafter referred to as point usage information) and data on an amount of money that must be held to use the digital content, namely billing data to be decremented every time the digital content is used (hereafter referred to as point information), has at least a point usage information storage memory 29 for storing the point usage information and a point information storage memory 28 for storing the point information.

[0015]

Further, the player 1 has a common key storage memory 22 and a communication key storage memory 21 as a constitution for storing various keys to be used for encryption and decryption to be described later and a common key encryption/decryption circuit 24 and a public-key encryption/decryption circuit 20 as a constitution for performing encryption and decryption by use of the keys stored in these memories. Still further, the player 1 has, as a constitution associated with the above-mentioned encryption and decryption, a security ID generating circuit 19 for generating random numbers to generate a security ID in operative association with a host computer of a system administration company 210, a timer 18, and a hash function circuit 25 for generating a so-called hash value to be described later.

[0016]

In addition, the player 1 has a controller 16 serving as control means for controlling, based on a program stored in a ROM 17, the digital content, various data, and components, and a battery 5 as operating power for the system when used in portable state.

[0017]

Herein, it is desirable, in terms of security, that the components of the player 1 shown in FIG. 2 be configured of a single chip of IC (Integrated Circuit) or LSI (Large Scale Integration). The components shown in FIG. 2 are all mounted on an integrated circuit 10. The player 1 has three terminals (an analog output terminal 2, a PC interface terminal 3, and a recording medium I/O terminal 4) as interfaces with the outside. These terminals are connected to terminals 13, 12, and 11 of the integrated circuit 10 respectively. It should be noted that these terminals may be integrated or may be provided as additional separate terminals.

[0018]

The system administration company 210 is composed of an administration center 211 that administers the overall system and a store 212 for selling the player 1, transfers information associated with the supply of digital content to be described later with the user terminal 50 of the user 200 through a virtual store 230, manipulates digital content that compresses and encrypts content owned by a content provider 240, supplies the encrypted digital content, and transfers information with a financial organization 220. Furthermore, information such as the bank account number, credit card number, name, contact address, and the like for the user 200 is confirmed and exchanged between the system administration company 210 and the financial organization 220 to determine whether or not a transaction is possible with the user 200. Processing such as the actual paying of fees and like is performed between the finance organization 220 and the user 200. It should be noted that the store 212 is not necessarily included in the system administration company 210, and the store may be an outside agent.

[0019]

The administration center 211 of the system administration company 210 has a constitution as shown in FIG. 3, for example. Details of the constitution shown in FIG. 3 will be described later. The administration center 211 at least has, as main components, a content administration function block 100 having functions for controlling digital content, displaying the digital content, performing manipulation processing such as encryption, compression, and the like, and generating a content key and ID which are key information for use in the encryption and decryption; a user administration function block 110 having functions for control of user information, encryption and decryption of a communication statement (including a message, point information, and the like), generating a confirmation message and a security ID, exchanging settlement information with the financial organization 230, generating points, and the like, and provided with a user subscription processing function 118 for processing user subscriptions and the like; a usage information administration function block 120 for controlling point usage information and the like; and an administration function block 130 for controlling the entire system and having a communication function.

[0020]

The following describes an example of the method described above of actually operating the system constituted as shown in FIG. 1, with reference to FIGS. 4 through 7. It should be noted that the operation method below is a procedure to be actually followed by the user 200, the system administration company 210, the financial organization 220, the content provider 240, and the like.

[0021]

The description of the method of operating the system will be made in the order, starting with a procedure of purchasing the player 1, followed by a procedure of operations from searching for digital content to installing the digital content into a storage medium of the player 1, a procedure of purchasing billing point information for making the digital content available and, if the digital content has been used, settling the fee for the usage, and finally a procedure of distributing fees collected from the user for the viewed digital content.

[0022]

In the procedure of purchasing the player 1, the user 200 purchases the player 1 from the store 212 by actually going to the store, by mail order, or the like, as shown in (1) and (5) of FIG. 4.

[0023]

As shown in (2) of FIG. 4, the store 212 registers into the administration center 211 of the system administration company 210 personal information (name, contact information, and the like) and settlement information (bank account number, credit number, and the like) obtained from the user 200 and the player-unique number of the player 1 (including the player-unique key and the like) at the purchase of the player 1.

[0024]

The administration center 211 confirms the bank account number, credit card number, and the like provided by the user 200 with the financial organization 220 as shown in (3) of FIG. 4, and obtains information from the financial organization 220 as to the possibility of a transaction as shown in (4) of FIG.4.

[0025]

Subsequently, in the procedure of operations from searching for digital content through installing the obtained digital content into the recording medium of the player 1, the user 200 who purchased the player 1 performs search, selection, editing, and order of a desired digital content by use of the user terminal 50 having the interface means for interfacing with the player 1 as shown in (1) of FIG. 5. The processing operations during this time from search through order are performed in relation to the virtual store 230 connected through a network, for example by use of search software stored in the user terminal 50 as application software.

[0026]

The virtual store 230 denotes a store that the administration center 211 virtually installs on the network, for example. In this virtual store 230, for example, information indicative of a plurality of contents are exhibited. Based on this information provided by the virtual store 230, the user 200 orders a desired content. The information indicative of the contents exhibited in the virtual store 230 includes, if the content is video data of a movie for example, the title, advertisement, or one scene of the movie, and the like, for example. If the content is audio data, the information includes the title, the name of the artist, or first several phrases (so-called intro) of the music, for example. Therefore, when the user terminal 50 of the user 200 accesses the virtual store 230, the plurality of contents of the virtual store 230 are virtually displayed on the user terminal 50, and the user 200 selects a desired one from the exhibited contents to place an order.

[0027]

When digital content is ordered from the user terminal 50 of the user 200, the virtual store 230 sends a request to the administration center 211 for supplying the ordered content as shown in (2) of FIG. 5.

[0028]

The administration center 211 that receives the request for supplying the digital content sends a request to the content provider 240 for distributing the requested content. In this manner, the content provider 240 distributes the requested digital content to the administration center 211 as shown in (4) of FIG. 5.

[0029]

The administration center 211 encrypts and compresses digital content distributed by the content provider 240 according to a predetermined compressing scheme and attaches, to this encrypted and compressed digital content, the ID of this content (the content ID), proprietor information such as copyright holder or the like for the content, the amount of fee to be billed when this content is used, the name of the virtual store that supplies this content to the user 200, and the like. It should be noted that the fee to be billed for the content is determined by the content provider 240 in advance.

[0030]

The content manipulated at the administration center 211 is transmitted to the virtual store 230 as shown in (5) of FIG. 5 and further supplied from the virtual store 230 to the user terminal 50 of the user 200 as shown in (6) of FIG. 5. In this manner, the desired content is supplied from the user terminal 50 to the player 1 and these contents are stored within that player 1.

[0031]

Note that the process flow shown in (2) through (5) of FIG. 5 can also be performed in advance. Namely, the virtual store 230 may not only exhibit information showing details of the plurality of contents but may also be made to prepare in advance the manipulated digital content corresponding to this display.

[0032]

Subsequently, in the above-mentioned procedure of purchasing billing point information for making available the digital content installed in the player 1 and of settling the fee if this digital content is used, the user terminal 50 confirms the shortage of the point information stored in the player 1 and sends a request from that user terminal 50 for replenishment of the point information.

[0033]

At this moment, as shown in (1) of FIG. 6, a request for replenishing the point information encrypted by the player 1 is transferred from the user terminal 50 to the administration center 211. At the same time, information about a proprietor, such as a copyright holder or the like, corresponding to digital content that has already been used, that is point usage information, is read out from the player 1, encrypted, and transmitted to the administration center 211 through the user terminal 50. Thus, the point usage information is transferred concurrently with the point information replenishment request, thereby saving the user 200 from accessing the administration center 211 only to transmit the point usage information to the administration center 211. Of course, the point usage information need not be transferred concurrently with purchasing of the point information, and the point usage information may be transferred independently.

[0034]

The administration center 211 receiving the encrypted point information replenishment request and point usage information decrypts the cryptograph to recognize the point information replenishment quantity requested by the user 200 and the contents of the point usage information. Further, the administration center 211 confirms with the financial organization 220 to see if the point replenishment can be settled or not as shown in (2) of FIG. 6. The financial organization 220 checks the account of the user 200, and if settlement is permissible, a settlement OK instruction is sent from the financial organization 220 to the administration center 211 as shown in (3) of FIG. 6.

[0035]

Moreover, at the same time, the administration center 211 notifies the content provider 240 of a point usage count, namely an amount of money, to be paid to the proprietor such as the copyright holder or the like as shown in (4) of FIG. 6.

[0036]

Thereafter, the administration center 211 encrypts a point replenishment information directive, and transmits this directive together with a security ID to the user terminal 50 as point replenishment instruction information as shown in (5) of FIG. 6. The point replenishment instruction information sent from the user terminal 50 to the player 1 is decrypted within that player 1. After the security ID is confirmed, the point information to be stored in the point information storage memory 28 is replenished and the notified proprietary information such as the copyright information and the like is deleted from the point usage information storage memory 29

[0037]

Subsequently, in the procedure of distributing the fee billed for viewing digital content, namely the fee to be drawn from the user account according to the point usage information, the financial organization 220 sends a request to the user 200 for sending the fee as shown in (1) of FIG. 7. At this moment, if there is an enough balance on the account of the user 200, the financial organization does not specially ask the user to send the fee. On the other hand, if there is not enough balance, the user 200 sends the fee to the financial organization 220 as shown in (2) of FIG. 7.

[0038]

The financial organization 220 subtracts a predetermined commission from the fee received from the user 200 and then sends the money to the administration center 211 as shown in (3) of FIG. 7. Namely, the administration center 211 collects the content manipulation fee, finance charge, system administration fee, and the like from the amount of money received from the financial organization 220. Moreover, the administration center 211 pays the copyright fee according to the point used to the content provider 240 as shown in (4) of FIG. 7 and a store commission to the virtual store 230 as shown in (5) of FIG. 7. The content provider 240 receiving the copyright fee pays the copyright fee to each copyright holder. The virtual store 230 receiving the store commission pays the commission per virtual store to each virtual store.

[0039]

As described, the fee paid by the user 200 is divided into the copyright fee, the store commission, the content manipulation commission, the settlement commission, and the system administration commission based on the point usage information. The copyright fee is paid to the content provider 240. The store commission is paid to the virtual store 230. The content manipulation commission is paid to the system administration company 210. The settlement commission is paid to the system administration company and the financial organization 220. The system administration commission is paid to the system administration company 210.

[0040]

It should be noted here that, in transferring data between the system of this embodiment, namely between the administration center 211 and the player 1, the data to be transferred is encrypted and decrypted to ensure the security of data communication. The preferred embodiment of the present invention is compatible with either a common key encryption system or a public-key encryption system.

[0041]

From the standpoint of processing speed, a common key encryption method is used in the preferred embodiment of the present invention as the encryption method when transferring the digital content, the point usage information, point information, messages, security IDs, and various other types of information. Different common keys are required for encryption and decryption of these various pieces of information. In the player 1 of FIG. 2, the common keys to be used for decryption of encrypted information received from the administration center 211 are stored in the common key storage memory 22. The common key encryption/decryption circuit 24 decrypts the encrypted information received from the administration center 211 by use of the common keys stored in this common key storage memory 22.

[0042]

On the other hand, as method of encryption when transmitting the common keys to be used for encryption and decryption of the various pieces of information, the encryption to be used varies depending on which type of encryption a player-unique key that is a unique key for the player 1 corresponds to. Namely, if the player-unique key corresponds to common key encryption, the common keys are encrypted by use of the player-unique key and the encrypted common keys are decrypted by use of the player-unique key. In contrast, if the player-unique key corresponds to public-key encryption, the common keys are encrypted by the public key of the other party and the encrypted common keys are decrypted by the secret key of the decrypting party.

[0043]

For example, when the common key (for example, a session key to be described later) is transmitted from the player 1 to the administration center 211, if the player-unique key corresponds to common key encryption, the common key encryption/decryption circuit 24 in the player 1 encrypts the common key by use of the player-unique key stored in the communication key storage memory 21. The administration center 211 decrypts the encrypted common key by use of the player-unique key stored in the administration center 211. Likewise, in transmission of the common key from the player 1 to the administration center 211, if the player-unique key corresponds to public-key encryption, for example, the public-key encryption/decryption circuit 20 encrypts the common key by use of the public key of the administration center 211 stored in the communication key storage memory 21 in the player 1. The administration center 211 decrypts the encrypted common key by use of the secret key stored in the administration center 211.

[0044]

Conversely, in transmission of the common key (for example, a content key) from the administration center 211 to the player 1, if the player-unique key corresponds to common key encryption, the common key is encrypted by use of the player-unique key stored in the administration center 211. In the player 1, the common key encryption/decryption circuit 24 decrypts the encrypted common key by use of the player-unique key stored in the communication key storage memory 21. Likewise, in transmission of the common key from the administration center 211 to the player 1, if the player-unique key corresponds to public-key encryption, for example, the common key is encrypted by use of the public key of player 1 stored in the administration center 211 and the public-key encryption/decryption circuit 20 decrypts the encrypted common key by use of the player-unique key stored in the communication key storage memory 21 in the player 1, that is, the secret key.

[0045]

The encryption method of the player-unique key itself as mentioned above is determined by whether sending (from the system administration company 210 to the player 1) of that player-unique key is easy or not. Namely, because common key encryption is more advantageous in cost, if sending of the player-unique key is easy, common key encryption is used, however, if sending of the player-unique key is difficult, public-key encryption is used although higher in cost. When implementing the player-unique key in hardware, common key encryption is used. When implementing the player-unique key in software, public-key encryption is used.

[0046]

An example in which the public-key encryption method is used in the preferred embodiment of the present invention will be given and described below taking into account compatibility in a case where the player-unique key itself is to be implemented in software as the encryption method. Namely, in transmission of the common key between the administration center 211 and the player 1, if the common key (the session key) is encrypted by the player 1, the encryption is performed by use of the public key of the administration center 211. The administration center 211 decrypts the encrypted common key by use of the player-unique key (namely the secret key). Conversely, if the common key (content key) is encrypted by the administration center 211, the encryption is performed by use of the public key of the player and the player 1 decrypts the encrypted common key by use of the player-unique key (namely the secret key).

[0047]

The following describes sequentially the actual operations of the player 1, the user terminal 50, and the administration center 211 that constitute a system operated by use of the above-mentioned procedures and encryption method.

[0048]

First, a processing flow in the player 1, the user terminal 50, and the administration center 10 at point replenishing or point purchasing will be described, using FIGS. 8 through 11, with reference to FIGS. 2 and 3.

[0049]

FIG. 8 shows a processing flow in the player 1 at point purchasing.

[0050]

Referring to FIG. 8, software installed in the user terminal 50, or a personal computer, for point purchasing is started in step ST1, during which the controller 16 of the player 1 waits until the software for point purchasing starts up.

[0051]

When the software for point purchasing gets started, the controller 16 of the player 1 receives from the user terminal 50 the information inputted in the user terminal 50 in step ST2. The information inputted in the user terminal 50 is what the user operating the user terminal 50 is required by the user terminal 50 to input according to the software for point purchasing described above. For example, this information includes a password, information about a point information count to be purchased, and the like.

[0052]

The information from the user terminal 50 is received by the controller 16 through the PC interface terminal 3 of the player 1 and the terminal 12 of the integrated circuit 10 mounted on one chip in the player 1. The controller 16 receiving the information from the user terminal 50 compares a password stored in the password storage memory 14 in the integrated circuit 10 of the player 1 with a password contained in the received information in step ST3 to confirm whether the received password is valid.

[0053]

If the received password is found valid, the controller 16 generates information indicating intention to purchase a point (intention of point purchasing), information about the point to be purchased, and other information in step ST4. At the same time, the controller 16 causes the security ID generating circuit 19 to generate a security ID and, in step ST5, causes the common key encryption/decryption circuit 24 to encrypt these pieces of information. In the next step ST6, the controller 16 reads the user ID from the user ID storage memory 23, adds the user ID to the encrypted information, and, in step ST7, transfers the resultant data to the user terminal 50 through the terminal 12 and the PC interface terminal 3. This generated data is then sent from the user terminal 50 to the administration center 211.

[0054]

At this time, because common key encryption is used for encryption of the generated data as described above, the common key is generated before the generated data is transmitted. Therefore, the controller 16 causes the security ID generating circuit 19, which is a random number generating means, for example, to generate a session key. Moreover, this common key (the session key) is sent from the player 1 to the administration center 211 before the transmission of the generated data. Because the common key is encrypted by public-key encryption as described above, the controller 16 sends the session key, which is the common key, to the public-key encryption/decryption circuit 20 and, at the same time, takes the public key of the administration center 211, which is stored in advance in the communication key storage memory 21, and sends this public key to the public-key encryption/decryption circuit 20. Consequently, the public-key encryption/decryption circuit 20 encrypts the common key (the session key) by use of the public key of the administration center 211. The session key thus encrypted is sent to the administration center 211 along with the user ID before the transmission of the generated data.

[0055]

It should be noted that, as described above, if the point usage information is transferred concurrently with the request for point information, the controller 16 reads the point usage information including the proprietor information and the like from the point usage information storage memory 29 and causes the common key encryption/decryption circuit 26 to encrypt the point usage information. The encrypted point usage information is transmitted along with the generated data. The balance of the point information may also be transferred in the same manner concurrently with the transfer of the point usage information.

[0056]

Thereafter, in step ST8, the controller 16 receives the encrypted data coming from the administration center 211 through the user terminal 50. This data sent from the administration center 211 is data obtained by first encrypting the point information corresponding to the point information count to be purchased that is transferred from the player 1 and information including the security ID, and the like, by use of the same common key as the session key.

[0057]

When the data is received from the administration center 211, the controller 16 sends the received data to the common key encryption/decryption circuit 24 in step ST9 and, at the same time, reads the common key generated in advance and stored in the common key storage memory 22 and sends this common key to the common key encryption/decryption circuit 24. The common key encryption/decryption circuit 24 decrypts the encrypted data coming from the administration center 211 by use of the common key.

[0058]

Subsequently, in step ST10, the controller 16 confirms the security ID of the decrypted data by comparing the same with the security ID coming from the security ID generating circuit 19. Then after the confirmation, in step ST11, the controller 16 updates the point information stored in the point information storage memory 28 with the newly sent point information.

[0059]

When the processing for updating the point information and the like has been completed, the controller 16, in step ST12, generates a processing completion sign, sends the sign to the common key encryption/decryption circuit 24 along with the common key read from the common key storage memory 22, and causes the common key encryption/decryption circuit 24 to encrypt the sign. Thereafter, in step ST13, the controller 16 transfers the processing completion sign that has been encrypted to the user terminal 50 through the terminals 12 and 3, and the sign is sent to the administration center 211.

[0060]

Thus, the processing flow in the player 1 for point purchasing is completed.

[0061]

The following describes a processing flow in the user terminal 50 for point purchasing with reference to FIG. 9.

[0062]

As shown in FIG. 9, the user terminal 50 starts up the software for point purchasing in step ST21. When the software for point purchasing starts up, the user terminal 50 sends a request to the user operating this user terminal 50 to input the password and the information such as a point count to be purchased and the like according to the software for point purchasing in step ST22. When these pieces of information have been inputted by the user, the inputted information is transferred to the player 1 as with step ST2 shown in FIG. 8.

[0063]

Thereafter, in step ST23, the user terminal 50 receives the data generated as shown in step ST7 shown in FIG. 8 from the player 1. In step ST24, the data transferred from the player 1 is sent to the address registered in advance, namely the administration center 211.

[0064]

The user terminal 50, after transferring the data, waits for a return from the administration center 211. When the data is returned from the administration center 211, the data returned from the administration center 211 is transferred to the player 1 without change in step ST25.

[0065]

In step ST26, when the user terminal 50 receives the processing completion sign from the player 1 as with step ST13 of FIG. 8, the processing completion sign is displayed on the display device in step ST27 to notify the user of completion of the processing such as point purchasing and the like.

[0066]

Thereafter, the user terminal 50 sends the cryptograph of the processing completion sign sent from the player 1 to the administration center 211.

[0067]

Thus, the processing flow in the user terminal 50 when purchasing a point is completed.

[0068]

The following describes a processing flow in the administration center 211 when point purchasing with reference to FIG. 10.

[0069]

As shown in FIG. 10, the administration center 211 receives the encrypted data from the player 1 transferred through the user terminal 50, as shown in step ST7 of FIG. 8 and in step ST24 of FIG. 9, through a communication function 133 of the administration function block 130 controlled by the control function 131 as shown in step ST31. The user administration function block 110 of the administration center 211 receiving this data obtains the common key from a database 112 and the security ID from a security ID generating function 116 based on the user ID attached to the received data under the control of a control function 111, as indicated by step ST32.

[0070]

Note that the common key at this moment is a session key sent from the player 1 in advance. This session key was encrypted by public-key encryption as described above. Therefore, at decryption of this encrypted session key, the user administration function block 110 of the administration center 211 takes out the secret key of public-key encryption of the administration center 211 and sends this secret key and the encrypted session key to a communication statement encryption/decryption function 114. The communication statement encryption/decryption function 114 decrypts the encrypted session key by use of the public key of the administration center 211. The session key (the common key) thus obtained is stored in the database 112.

[0071]

When the common key corresponding to the user ID is obtained from the database 112 and the security ID is obtained from the security ID generating function 116, as shown in step ST33, the communication statement encryption/decryption function 114 of the user administration function block 110 in the administration center 211 decrypts the encrypted data from the player 1 by use of the common key. Further, the control function 111 compares the security ID attached to the decrypted data with the security ID read from the security ID generating function 116 to confirm whether the user 200 (the player 1) that made the access is an authorized user or not.

[0072]

The administration center 211, upon confirming that the accessing party is authorized, causes a point generating function 113 of the user administration function block 110 to issue point information, as indicated in step ST34, according to the data sent from the user terminal 50 and causes a settlement billing function 117 to prepare billing for the settlement organization (the financial organization 220) of the user.

[0073]

Furthermore, as in step ST35, the administration center 211 causes the control function block 111, for example, to confirm that here is no illegality in the balance of the point information and the point usage information sent from the player 1 and reorganizes the information for later processing. Namely, it is confirmed from the balance of the point information and the actually used point information count that there is no illegal use, and the information is reorganized. It should be noted that the confirmation and reorganization need not always be performed, however, these are preferably performed.

[0074]

After the processing of step ST35, the user administration function block 110 of the administration center 211, as indicated by step ST36, causes the security ID generating function 115 to compute a new security ID for the player 1 (the user) based on random number generation, for example, and causes the control function 110 to encrypt the security ID along with the point information. The encryption at this time is also performed by use of the session key (the common key) sent from the player 1 in advance.

[0075]

When the encryption has been completed, the communication function 133 of the administration function block 130 in the administration center 211 sends the decrypted data to the player 1 through the user terminal 50 under the control of the control function 131 as indicated by step ST25 of FIG. 9 and step ST8 of FIG. 8.

[0076]

Thereafter, as indicated by step ST38, the communication function 133 of the administration center 211 receives the processing completion sign from the user terminal 50 shown in step ST28 of FIG. 9 and decrypts the received sign. In step ST39, the settlement billing function 117 of the user administration function block 110 in the administration center 211 sends a request to the financial organization 220 for settlement according to the decrypted processing completion sign. The settlement request to the financial organization 220 is issued from the communication function 132 of the administration function block 130.

[0077]

Thus, the processing flow in the administration center 211 for point purchasing is completed.

[0078]

The sequence of information transfer between the player 1, the user terminal 50, and the administration center 211 in the processing flow shown in FIGS. 8 through 10 can be represented as shown in FIG. 11.

[0079]

Namely, as shown in FIG. 11, in an input information transfer operation T1, the input information such as the password and the point count is transferred from the user terminal 50 to the player 1 as shown in step ST2 of FIG. 8 and step ST22 of FIG. 9.

[0080]

In a generated data transfer operation T2, the data generated by the player 1 is transferred from the player 1 to the user terminal 50 as shown in step ST7 of FIG. 8 and step ST23 of FIG. 9. Moreover, in a data transfer operation T3, the data generated by the player 1 is transferred from the user terminal 50 to the administration center 211 as shown in step ST24 of FIG. 9 and step ST31 of FIG. 10.

[0081]

In a data transfer operation T4, the data encrypted by the administration center 211 is transferred from the administration center 211 to the user terminal 50 as shown in step ST37 of FIG. 10 and step ST25 of FIG. 9. In a transfer operation T5, the data coming from the administration center 211 is transferred by the user terminal 50 to the player 1 without change as shown in step ST25 of FIG. 9 and step ST8 of FIG. 8.

[0082]

In a processing completion sign transfer operation T6, the processing completion sign is transferred from the player 1 to the user terminal 50 as shown in step ST13 of FIG. 8 and step ST26 of FIG. 9. Further, in a processing completion sign cryptograph transfer, the encrypted processing completion sign is transferred from the player 1 to the administration center 211 as shown in step ST28 of FIG. 9 and step ST38 of FIG. 10.

[0083]

The following describes a processing flow in the player 1, the user terminal 50, and the administration center 211 when obtaining the above-mentioned digital content, using FIGS. 12 through 15, with reference to FIGS. 2 and 3 .

[0084]

FIG. 12 shows a processing flow in the player 1 when obtaining digital content.

[0085]

As shown in FIG. 12 the controller 16, as in step ST41, waits until the software for obtaining digital content installed on the user terminal 50, or the personal computer, has started up.

[0086]

When the software for obtaining digital content is started, the controller 16 receives data including digital content from the administration center 211 through the user terminal 50, as in step ST42. The data to be received at this time from the user terminal 50 through the terminal 3 and 12 has at least the digital content encrypted by the content key (a specific common key for each specific content) and the content ID corresponding to the digital content. Therefore, use of this encrypted digital content requires that the content key be obtained from the administration center 211. A method of obtaining the content key will be described later.

[0087]

The controller 16, upon receiving the data from the user terminal 50, stores this data, namely the encrypted digital content, into a storage medium connected to the storage medium I/O terminal 4 through the terminal 11 of the integrated circuit 10. It should be noted that this storage medium may be a rewritable optical disk, a semiconductor memory device, or the like, and preferably, this storage medium is a device allowing random access.

[0088]

Thus, the processing flow in the player 1 for obtaining digital content is completed.

[0089]

The following describes a processing flow in the user terminal 50 for obtaining digital content with reference to FIG. 13.

[0090]

As shown in FIG. 13, in step ST51, the user terminal 50 starts up the software for obtaining digital content. When this software is started up, the user terminal 50 accesses, in step S52, the administration center 211 having a registered address according to the software for obtaining digital content.

[0091]

At this moment, the administration center 211 is displaying a plurality of digital contents by use of the virtual store 230. In step ST53, the user selects a desired digital content through the user terminal 50 from among the plurality of digital contents displayed in the virtual store 230. Namely, as in step ST54, the user terminal 50 sends content specification information for specifying a desired digital content from among the digital content displayed in the virtual store 230 to the administration center 211.

[0092]

In step ST55, when the data returned from the administration center 211 according to the above-mentioned content specification information, namely the data composed of the encrypted digital content and the content ID, is received, the user terminal 50 temporarily stores the received data in an internal storage means such as hard disk, memory device, or the like, as in step ST56.

[0093]

Thereafter, the user terminal 50 transfers the stored data (the encrypted digital content and the content ID) to the player 1 as shown in step ST42 of FIG. 12.

[0094]

Thus, the processing flow in the user terminal 50 for obtaining digital content is completed.

[0095]

The following describes a processing flow in the administration center 211 for obtaining digital content with reference to FIG. 14.

[0096]

At this point, the administration center 211 shown in FIG. 3 is displaying a plurality of contents in the virtual store 230 described above. To be more specific, the virtual store 230 is generated in the content administration function block 100 of the administration center 211. The plurality of digital contents are displayed in the generated virtual store 230.

[0097]

In a situation in which digital content is on display in a virtual store 230 like this, as in step ST61 of FIG. 14, content specification information is received from the user terminal 50 in accordance with step ST54 of FIG. 13.

[0098]

When the content specification information is received from the user terminal 50, the control function 101 of the content administration function block 100 sends the received content specification information to the administration function block 130. The control function 131 of the administration function block 130 transfers the content specification information received from the control administration function block 100 [sic] to the content provider 240 through the communication function 134 for the proprietor. Thus, the digital content requested by the content specification information comes from this content provider 240. The digital content obtained from the content provider 240 is then transferred from the administration function block 130 to the content administration function block 100 to be inputted in this content encryption and compressing function 104. At this moment, the control function 101 sends the content key generated by the content key and ID generating function 103 and stored in the database 102 to the content encryption and compressing function 104. The content encryption and compressing function 104 encrypts the digital content by use of the content key and compression processes the encrypted digital content in a predetermined manner. The control function 101 attaches the content ID taken from the database 102 to the encrypted and compression processed digital content and sends the result to the administration function block 130. It should be noted that, if the digital content is an audio signal, ATRAC (Adaptive TRansform Acoustic Coding), for example, is used for compression processing of the digital content. ATRAC is a technology for use in compressing data stored in a recently available storage medium called MD (Mini Disc, trademark). ATRAC considers the human auditory characteristic to compression process audio data highly efficiently.

[0099]

Thereafter, as shown in step ST62 of FIG. 14, the control block 131 of the administration function block 130 transmits the encrypted and compression processed digital content to which a content ID has been attached to the user terminal 50 through the communication function 133 interfacing with the user terminal.

[0100]

This completes the processing flow in the administration center 211 for obtaining digital content.

[0101]

The sequence of transferring information between the player 1, user terminal 50, and the administration center 211 in the processing flow shown in FIGS. 12 through 14 can be represented as shown in FIG. 15.

[0102]

Namely, in FIG. 15, in an input information transfer operation T11, as in step ST54 of FIG. 13, the content specification information is transferred from the user terminal 50 to the administration center 211. In a content transfer operation T12, the encrypted digital content and the content ID are transferred from the administration center 211 to the user terminal 50, as in step ST62 of FIG. 14.

[0103]

In a content transfer operation T13, the content ID and the encrypted digital content once stored in the user terminal 50 are transferred to the player 1, as in step ST57 of FIG. 13 and step ST42 of FIG. 12.

[0104]

The following describes a processing flow in the player 1, the user terminal 50, and the administration center 211 for obtaining a content key necessary for using the above-mentioned digital content and a use condition of the content key, using FIGS. 16 through 19, with reference to FIGS. 2 and 3.

[0105]

FIG. 16 shows a processing flow in the player 1 for obtaining the content key and the use condition.

[0106]

In step ST71 of FIG. 16, the controller 16 of the player 1 waits for the software installed in advance on the user terminal 50 for obtaining the content key and the use condition to start up.

[0107]

When the software of the user terminal 50 for obtaining the content key and the use condition has started up, information inputted in the user terminal 50 according to the software is received through the PC interface terminal 3 and the terminal 12 of the integrated circuit 10, as in step ST72. The input information supplied from the user terminal 50 is information for requesting a content key necessary for undoing the encryption of the encrypted digital content to be viewed. Note that in this example, for the content key requesting information, information for specifying the digital content that uses this content key is used.

[0108]

The controller 16 that receives the content specification information from the user terminal 50 generates the ID of the digital content specified by the content specification information and the security ID that comes from the security ID generating circuit 19 in step ST73 and causes the common key encryption/decryption circuit 24 to encrypt this generated data. Moreover, the controller 16 adds the user ID read from the user ID storage memory 23 to this generated data and sends the resultant data to the user terminal 50 through the terminal 12 and the PC interface terminal 3. This generated data is then sent from the user terminal 50 to the administration center 211.

[0109]

At this moment, because common key encryption is also used for encryption of the generated data, a common key is generated before the data is transmitted. Therefore, the controller 16 causes the security ID generating circuit 19, which is a random number generating means, for example, to generate a session key. Moreover, this common key (the session key) is sent from the player 1 to the administration center 211 before the transmission of the generated data. Because this common key is encrypted by public-key encryption as described above, the controller 16 sends the session key, which is the common key, to the public-key encryption/decryption circuit 20 and, at the same time, takes the public key of the administration center 211 from the communication key storage memory 21 to send this public key to the public-key encryption/decryption circuit 20. Consequently, the public-key encryption/decryption circuit 20 encrypts the common key (the session key) by use of the public key of the administration center 211. Thus, the session key thus encrypted is sent to the administration center 211 before transmission of the generated data.

[0110]

Thereafter, in step ST75, the controller 16 receives the encrypted data sent from the administration center 211 through the user terminal 50 as will be described later. The data transmitted from the administration center 211 at this time is data in which the content key, use condition, security ID and the like have been encrypted as will be described later.

[0111]

When the encrypted data has been received from the administration center 211, the player 1 decrypts the encrypted data and confirms the validity of the data, as in step ST76. Namely, the controller 16 confirms the validity by comparing the security ID of the decrypted data with the security ID from the security ID generating circuit 19.

[0112]

Herein, the content key is encrypted by public-key encryption and the use condition and security ID are encrypted by common key encryption as will be described. Therefore, in order to decrypt the encrypted content key, a secret key of public-key encryption is required. In the player 1 of the present embodiment, because a player-unique key is used as the secret key as described above, the player-unique key is taken from the communication key storage memory 21. This player-unique key is sent to the public-key encryption/decryption circuit 20 along with the encrypted content key. The public-key encryption/decryption circuit 20 decrypts the encrypted content key by use of the player-unique key. The decrypted content key is stored in the common key storage memory 22. On the other hand, in order to decrypt the use condition and security ID encrypted by common key encryption, these pieces of data are sent to the common key encryption/decryption circuit 24 and the common key is read from the common key storage memory 22 to send to the common key encryption/decryption circuit 24. The common key encryption/decryption circuit 24 decrypts the use condition and security ID by use of the common key. The decrypted use condition is stored in the point usage information memory 29. It should be noted here that the decrypted content key and use condition are not taken outside the player 1, specifically, these pieces of data are not taken outside the controller 16, the common key storage memory 22, and the point usage information storage memory 29 that are mounted on the integrated circuit 10 shown in FIG. 2.

[0113]

After confirming the validity, the controller 16 stores the decrypted content key in the common key storage memory 22 along with the content ID, as in step ST77.

[0114]

Thereafter, in step ST78, the controller 16 generates a message indicating that the content key has been obtained, sends this message to the common key encryption/decryption circuit 24, reads out the common key that was stored in advance in the common key storage memory 22, and sends this common key to the common key encryption/decryption circuit 24. The common key encryption/decryption circuit 24 encrypts the message by use of this common key.

[0115]

When the encryption of the message has been completed, the controller 16 sends the encrypted message to the user terminal 50 through the terminals 12 and 3, as in step ST79. This encrypted message is then transferred to the administration center 211.

[0116]

Thus, the processing flow in the player 1 for obtaining the content key and the use condition is completed.

[0117]

The following describes a processing flow in the user terminal 50 for obtaining a content key and a use condition with reference to FIG. 17.

[0118]

In FIG. 17, the user terminal 50 starts up the software for obtaining the content key and the use condition, in step ST81. When this software has started up, the user terminal 50 sends a request to the user operating the user terminal 50 to specify a desired content according to the software in step ST82. When the user specifies the desired content, the user terminal 50 generates the specification information. The user terminal 50 sends the content specification information to the player 1, in step ST83.

[0119]

Subsequently, in step ST84, when the data generated by the player 1 is received, as in step ST74 of FIG. 16, the user terminal 50 transfers, in step ST85, the data received from the player 1 to the administration center 211 the address of which has been registered in advance.

[0120]

The user terminal 50, after transferring the data to the administration center 211, waits for the return of data from the administration center 211, and in step ST86, when data in which the content key, use condition, security ID, and the like specified for the content key have been encrypted is returned from the administration center 211, in step ST87, the data from the administration center 211 is transferred without change to the player 1.

[0121]

The user terminal 50, after transferring the data to the player 1, waits for the return of data from the player 1, and in step ST88, when an encrypted message to the effect that the content key has been acquired is returned from the player 1 as in step ST79 of FIG. 16, the user is notified thereof, in step ST89, by a display to the effect that the content key acquisition has been completed, which is carried out on the display device connected to the user terminal 50.

[0122]

Thereafter, the encrypted message returned from the player 1 is sent to the administration center 211 in step ST90.

[0123]

Thus, the processing flow in the user terminal 50 for obtaining the content key and the use condition is completed.

[0124]

The following describes a processing flow in the administration center 211 for obtaining a content key and a use condition with reference to FIG. 18.

[0125]

In FIG. 18, the administration center 211 communication function 133 interfacing with the user terminal, in step ST91, receives the encrypted data of the content ID, user ID, message, and security ID transmitted from the player 1 through the user terminal 50 as in step ST74 of FIG. 16 and step ST85 of FIG. 17. The received data is then sent to the user administration function block 110.

[0126]

The control function 111 of the user administration function block 110 retrieves the common key for undoing the encryption from the database 112 on the basis of the user ID attached to the received encrypted data, and decrypts this encrypted data by the communication statement encryption/decryption function 114 by use of this common key. Moreover, the control function 111 confirms the validity of the decrypted data by use of the user ID read from the database 112 and the security ID read from the security ID generating function 116.

[0127]

Note that the common key at this moment is a session key sent from the player 1 in advance. This session key was encrypted by public-key encryption as described above. Therefore, at decryption of this encrypted session key, the secret key based on public-key encryption of the administration center 211 is taken into the administration center 211, as described above. The encrypted session key is decrypted by the communication statement encryption/decryption function 114 by use of this secret key. The session key (the common key) thus obtained is stored in the database 112.

[0128]

When the validity of the received data has been confirmed, the control function 111 sends a request to the content administration function block 100 for the content key and use condition specified by the content ID. The control function 101 of the requested content administration function block 100 reads the content key and use condition specified in the content ID from the database 102 and transfers the content key and use condition to the user administration function block 110 As shown in step ST93, the control function 111 sends the content key and use condition to the communication statement encryption/decryption function 114 along with the security ID.

[0129]

At this point, the content key is encrypted based on public-key encryption and the use condition and the security ID are encrypted based on common key encryption as described above. Therefore, at the time of encryption of the content key, the public key of the user 200 (the public key stored in advance corresponding to the player 1) is taken from the database 112 based on the user ID and this public key is sent to the communication statement encryption/decryption function 114. Using this public key, the communication statement encryption/decryption function 114 encrypts the content key. On the other hand, at the time of encryption of the use condition and the security ID, the common key (the session key) specified by the user ID is taken from the database 112 and this common key is sent to the communication statement encryption/decryption function 114. The communication statement encryption/decryption function 114 encrypts the use condition and the security ID by use of the common key.

[0130]

The encrypted content key, use condition, and security ID are sent to the administration function block 130 and then transmitted from the communication function 133 to the user terminal 50 as in step ST94. The data sent to the user terminal 50 is then sent to the player 1 through the user terminal 50 as in step ST87 of FIG. 17 and step ST75 of FIG. 16.

[0131]

Thereafter, the administration center 211 waits to receive the encrypted message generated in the player 1 and sent through the user terminal 50 as in step ST79 of FIG. 16 and step ST90 of FIG. 17. When the communication function 133 receives the encrypted message generated by the player 1 as in step ST95, the administration center 211 decrypts the encrypted message by use of the common key, and confirms that the player 1 has obtained the content key and the use condition as in step ST96.

[0132]

Thus, the operation flow in the administration center 211 for obtaining the content key and the use condition is completed.

[0133]

The sequence of information transfer between the player 1, the user terminal 50, and the administration center 211 in the processing flow shown in FIGS. 16 through 18 is represented as shown in FIG. 19.

[0134]

Namely, referring to FIG. 19, in a content specification information transfer operation T21, the content specification information is transferred from the user terminal 50 to the player 1 as in step ST83 of FIG. 17. In a generated data transfer operation T22, the data generated by the player 1 is transferred to the user terminal 50 as in step ST74. In a generated data transfer operation T23, the data generated by the player 1 is transferred from the user terminal 50 to the administration center 211. In an encrypted data sending operation T24, the data encrypted by the administration center 211 is sent to the user terminal 50 as in step ST94 of FIG. 18. Furthermore, in an encrypted data sending operation T25, this encrypted data is sent to the player 1.

[0135]

In a message transfer operation T26, data obtained by encrypting a message indicating that the content key has been obtained is transferred from the player 1 to the user terminal 50 as in step ST79 of FIG. 16. In an encrypted data sending operation T27, the encrypted message coming from the player 1 is sent from the user terminal 50 to the administration center 211.

[0136]

The following describes a processing flow in the player 1 that has received the point information, the digital content, and the content key as described above for actually viewing the received digital content by use of the user terminal 50, using FIG. 20, with reference to FIG. 2.

[0137]

It is assumed here that the terminal 4 of the player 1 is connected to a storage medium in which the digital content is stored.

[0138]

In this state, the user terminal 50 specifies the digital content to be viewed in the player 1, as in step ST101. At this moment, this specification is made by the user operating the user terminal 50, for example.

[0139]

At this moment, as in step ST102, the controller 16 of the player 1 accesses the storage medium according to the content specification information coming from the user terminal 50 to read the ID of the content.

[0140]

The controller 16, as in step ST103, accesses the common key storage memory 22, based on the content ID read from the storage medium, to confirm whether the content key is stored and, at the same time, accesses the point usage information storage memory 29 to confirm whether the use condition is stored.

[0141]

At this point, if the content key and the use condition are not confirmed to be stored in the common key storage memory 22 and the point usage information storage memory 29 respectively, the controller 16 sends information to the user terminal 50 indicating that the content key and the like do not exist. Based on this information, a message is displayed on the display device from the user terminal 50 prompting to obtain the content key and the like. In this case, the content key and the like are obtained as shown in the flowchart of obtaining the content key as described above. Thus, if the content key and the like are newly obtained, the encrypted content key and the like are decrypted as described above in step ST104.

[0142]

Subsequently, as shown in step ST105, based on the decrypted use condition, the controller 16 confirms whether there is a sufficient balance of the point information stored in the point information storage memory 28. If the balance of the point information stored in the point information storage memory 28 is insufficient, the controller 16 sends information to the user terminal 50 indicating that the balance of the point information is insufficient. Based on this information, the user terminal 50 displays a message on the display device, prompting obtaining the point information. In this case, the point information is obtained as indicated in the flowchart of obtaining the point information as described above.

[0143]

At this point, when actually viewing the digital content, the controller 16 decrements the point information count from the point information storage memory 28 according to the digital content to be viewed, as in step ST106, and stores the new point usage information corresponding to the usage state of this point information into the point usage information storage memory 29 (updates the point usage information). The point usage information to be newly stored thus in the point usage information storage memory 29 includes proprietor information (copyright holder and the like) corresponding to the viewed digital content, information about the decremented point information count, and the like.

[0144]

Thereafter, as in step ST107, the controller 16 confirms that the billing processing of decrementing the point information, newly storing the point usage information, and the like has been completed and then reads the digital content from the storage medium.

[0145]

Because the digital content read from the storage medium is encrypted, the controller 16 transfers this encrypted digital content to the common key encryption/decryption circuit 24, as in step ST109.

[0146]

Based on the instruction given by the controller 16, as in step ST110, the common key encryption/decryption circuit 24 decrypts the encrypted digital content by use of the content key decrypted and stored in advance in the common key storage memory 22.

[0147]

Moreover, because this digital content is compression processed in a predetermined manner as described above, the controller 16, as in step ST111, transfers the decrypted but still compression processed digital content from the common key encryption/decryption circuit 24 to the decompressing circuit 26, and the decompression processing corresponding to the compression processing is performed there.

[0148]

Thereafter, as in step ST112, the decompressed digital content is converted by the D/A conversion circuit 27 into an analog signal. The analog signal is outputted outside (for example, to the user terminal 50) through the terminal 13 of the integrated circuit 10 and the analog output terminal 2 of the player 1, as in step ST113.

[0149]

Thus, the processing flow in the player 1 for viewing digital content is completed, allowing the user to view the digital content.

[0150]

The following describes a processing flow in the player 1, the user terminal 50, and the administration center 310 for returning the point usage information newly stored in the point usage information storage medium 29 of the player 1 to the administration center 211 at the above-mentioned digital content viewing, using FIGS. 21 through 24, with reference to FIGS. 2 and 3.

[0151]

FIG. 21 shows a processing flow in the player 1 at returning the point usage information.

[0152]

In FIG. 21, as shown in step ST121, the controller 16 waits until the software installed in advance in the user terminal 50 for returning point usage information starts up.

[0153]

When the software of the user terminal 50 for returning point usage information has started up, information inputted in the user terminal 50 according to the software is received through the PC interface terminal 3 and the terminal 12 of the integrated circuit 10, as in step ST122. The input information supplied from the user terminal 50 at this time includes a password and the like to be inputted by the user.

[0154]

In step ST123, the controller 16 that has received this content specification information from the user terminal 50 compares the password supplied from the user terminal 50 with the password stored in the password storage memory 14 to confirm whether the supplied password is valid or not.

[0155]

If the password is found valid during the password confirmation, the controller 16 reads the balance of the point information stored in the point information storage memory 28 and the point usage information stored in the point usage information storage memory 29, as in step ST124, and encrypts these pieces of information.

[0156]

When the balance of the point information and the point usage information have been encrypted, the controller 16 reads the user ID from the user ID storage memory 23 and attaches this user ID to the encrypted data, as in the step ST125.

[0157]

The data attached with the user ID is transferred from the controller 16 to the user terminal 50 through the terminal 12 and the PC interface terminal 3, as in step ST126. This data is then transferred to the administration center 211.

[0158]

It should be noted that the encryption at this time is also based on common key encryption as described above. Namely, before transmission of the data, the common key is generated as described above, this generated common key is encrypted by public-key encryption (by encryption using the public key of the administration center 211), and the encrypted common key is sent to the administration center 211 along with the user ID.

[0159]

After the data is transferred to the user terminal 50 as described above, the controller 16 waits until the data to be described later comes from the administration center 211 through the user terminal 50.

[0160]

At this point, when the data from the administration center 211 has been received, as in step ST127, the player 1 decrypts, using the common key, the received data encrypted by use of common key encryption and confirms the validity of the decrypted data, as in step ST127 [sic - step 128?]. Namely, the controller 16 confirms the validity by comparing the security ID of the decrypted data with the security ID from the security ID generating circuit 19.

[0161]

Moreover, the data transferred from the administration center 211 includes a processing completion message encrypted by use of the common key indicating. Therefore, the controller 16, after confirming the validity of the security ID, sends the encrypted processing completion message to the common key encryption/decryption circuit 24, causes this circuit to decrypt the message by use of the common key, and receives the message with decryption processing completed, thereby confirming that the processing in the administration center 211 has been completed.

[0162]

Thus, the processing flow in the player 1 for returning the point usage information is completed.

[0163]

The following describes a processing flow in the user terminal 50 for returning point usage information with reference to FIG. 22.

[0164]

In FIG. 22, the user terminal 50 starts up the software for returning point usage information, as in step ST131. When this software starts up, the user terminal 50 sends a request in step 132, according to the software, to the user of the user terminal 50 to input a password and the like. When the password is inputted by the user, that password is transferred to the player 1.

[0165]

Subsequently, in step ST133, when the data generated by the player 1 is received, as in step ST126 of FIG. 21, the user terminal 50 transfers, in step ST134, the data received from the player 1 to the administration center 211 the address of which has been registered in advance.

[0166]

The user terminal 50, after transferring the data to the administration center

211, waits for return from the administration center 211. When the data sent from the administration center 211 to the player 1 is received, that data is transferred to the player 1 directly in step ST135.

[0167]

The user terminal 50, after transferring the data to the player 1, displays a processing completion message to the user on the display device and receives confirmation from the user.

[0168]

Thus, the processing flow in the user terminal 50 for returning the point usage information is completed.

[0169]

The following describes a processing flow in the administration center 211 for returning point usage information with reference to FIG. 23.

[0170]

As in step ST141, the communication function 133 of the administration center 211 interfacing the user terminal receives the data including point usage information and the like from the player 1 through the user terminal 50 in step ST126 of FIG. 21 and step ST134 of FIG. 22.

[0171]

When this data is received, as in step ST142, the user administration function block 110 of the administration center 211 obtains, from the database 112, the common key received and stored in advance, as well as the security ID based on the user ID attached to the received data under the control of the control function 111.

[0172]

When the common key and the security ID corresponding to the user ID have been obtained from the database 112, as shown in step ST143, the data including the encrypted point usage information coming from the player 1 is decrypted in the communication statement encryption/decryption function 114 of the user administration function block 110 in the administration center 211 by use of the common key. Further, in the control function 111, the security ID in the decrypted data is compared with the security ID read from the database 112 to confirm whether the accessing user 200 (the player 1) is valid or not.

[0173]

After the validity and data contents have been confirmed, the data is transferred to the usage information administration function block 120. A control function 121 of the usage information administration function block 120, as shown in step ST144, uses the point information balance and point usage information sent from the player 1 to confirm whether use by the user 200 is illegal or not using the information stored in the database 122. At the same time, an operation for summarizing the point information balance and point usage information is carried out in a usage information operation function 123 when it has been confirmed that no illegality is involved.

[0174]

Thereafter, as shown in step ST145, the control function 111 of the user administration function block 110 controls the security ID generating function 116 to compute the security ID, and controls a confirmation message generating function 115 to generate a processing completion message. The security ID and the processing completion message are encrypted by the communication statement encryption/decryption function 114 of the user administration function block 110 by use of the common key.

[0175]

As shown in step ST146, the generated encrypted data is sent from the communication function 133 to the user terminal 50 and then sent from the user terminal 50 to the player 1, as in step ST135 of FIG. 22 and step ST127 of FIG. 21.

[0176]

Thus, the processing flow in the administration center 211 for returning the point usage information is completed.

[0177]

The sequence of information transfer between the player 1, the user terminal 50, and the administration center 211 in the processing flow of FIGS. 21 through 23 described above can be represented as shown in FIG. 24.

[0178]

Namely, in FIG. 24, in an input information transfer operation T31, input information such as the password and the like is transferred from the user terminal 50 to the player 1 as in step ST132 of FIG. 22. In a generated data transfer operation T32, the data generated by the player 1 is transferred to the user terminal 50 as in step ST126 of FIG. 21. In a generated data transfer operation T33, the data generated by the player 1 is transferred from the user terminal 50 to the administration center 211 as in step ST134 of FIG. 22. In a data transfer operation T34, the data generated by the administration center 211 is transferred to the user terminal 50 as in step ST146 of FIG. 23. In a data transfer operation T35, the data generated by the administration center 211 is transferred to the player 1 through the user terminal 50 as in step ST127 of FIG. 21.

[0179]

The actual operations of the player 1, the user terminal 50, and the administration center 211 of the system of the present embodiment flow as described above.

[0180]

So far, the entire processing flow in the system of the present embodiment has been described. However, in the following, the operation of each main component of the system of the present embodiment will be described in detail.

[0181]

First, encryption and compressing operations and decompressing and decryption operations in the present embodiment of the invention will be described.

[0182]

When digital content is distributed using a network as in the system of the preferred embodiment described above, compression/decompression techniques are used to reduce the amount of this data, and encryption/compression techniques are used for protection against copying and/or for billing. Namely, the distributing side (in the above-mentioned example, the administration center 211) compresses and then performs encryption processing on digital content. When the digital content (encrypted and compressed data) generated by the distributing side (the administration center 211) as with the above-mentioned example is distributed through a network, the receiving side (in the above-mentioned example, the player 1) receives the encrypted and compressed digital content and then decrypts and decompresses the digital content. It should be noted that the order in which encryption and compressing are performed and the order in which decryption and decompressing are performed may be altered in some cases.

[0183]

If the digital content includes a copyright or the like, the receiving side is billed according to the intention of the holder of the copyright before decrypting and decompressing the digital content. This billing is performed mainly by purchasing the key for decryption, namely the content key, however, there are various methods by which this content key is purchased.

[0184]

Herein, if the processing procedure in which digital content is compressed and encrypted and then decrypted and decompressed as mentioned above is followed, a malicious user, for example, can obtain the decrypted and compressed data with comparative ease. That is, the capacity of the compressed data of the digital content generally is large, and therefore, for example, this compressed data is often stored in an inexpensive external memory rather than in the internal memory of an ordinary content playback device of the receiving side. For this reason, it is easy to illegally remove the compressed digital content either directly from the external memory or through the part that connects to the external memory.

[0185]

Moreover, the algorithms for decompressing the compressed data are made public in many cases. In addition, these decompressing algorithms are not ones that cannot be processed if hidden like general encryption keys. Furthermore, compared to the encrypted and compressed digital content distributed from the transmission side, the decrypted compressed digital content does not differ as far as the volume of the data, and therefore, it is easy to maliciously distribute the decrypted compressed digital content. That is, according to a system that distributes digital content that has been encrypted after being compressed, there is a serious risk of the compressed digital content, which anyone can easily decompress, being easily stolen by a user having malicious intent, and being distributed and decompressed yet again in places the copyright holders never intended.

[0186]

Therefore, in this embodiment of the present invention, in consideration of this situation, in order to allow the security of the digital content distributed through a network to be enhanced, the processing indicated by the flowchart shown in FIG. 25 is performed in the player 1 of FIG. 2.

[0187]

Namely, in the decryption processing by the common key encryption/decryption circuit 24 of the player 1 shown in FIG. 2 and the decompression processing by the decompressing circuit 26, the data of encryption and compression processed digital content read from the storage medium is first divided into units of least common multiple lcm(X,Y) of processing unit X bits of a decryption algorithm and processing unit Y bits of a decompression algorithm, as in step ST151.

[0188]

Subsequently, decryption processing is performed on the data of the encrypted and compressed digital content divided into least common multiple lcm(X,Y) units by the common key encryption/decryption circuit 24 in units of the least common multiple lcm(X,Y) as shown in step ST152.

[0189]

Regarding the digital content data compressed in units of the least common multiple lcm (X, Y) obtained by the decryption process, as shown in step ST154, decompression processing is performed by the decompressing circuit 26 for all the units of compressed data.

[0190]

Thereafter, the decryption and decompression processing in units of this least common multiple lcm(X,Y) are repeated until the processing of all data of the encrypted and compressed digital content has been completed. Namely, as shown in step ST155, it is determined whether decryption and decompression processing in units of least common multiple lcm(X,Y) have been completed on all data of the digital content. If the decryption and decompression processing are not completed, the process returns to step ST152, while if the decryption and decompression processing have been completed, the processing shown in the flowchart comes to an end

[0191]

Thus, the digital content with all data decrypted and decompressed can be obtained.

[0192]

It should be noted that, in the processing by the player 1 shown in the flowchart of FIG. 25, the decrypted data in units of least common multiple lcm(X,Y) exists, but the data quantity of this decrypted data is small. Thus, it is possible to store the data in relatively high priced but highly secure internal memory, thereby making the likelihood of the data being stolen extremely low, as when stored in the external memory described above.

[0193]

Moreover, in the player 1 of the present embodiment, a buffer memory 25 shown in FIG. 2 is provided as an internal memory for ensuring the data security between the common key encryption/decryption circuit 24 and the decompressing circuit 26. That is, this buffer memory 25 is provided in a single-chip integrated circuit 10, and is difficult to access from the outside, thereby preventing the data from being taken out.

[0194]

In the flowchart described above, the constitution is such that decryption and decompression processing are carried out for all data in units of the least common multiple lcm (X,Y). As a specific constitution, for example as shown in FIG. 26, first, the digital content data is divided into X bits, which is the unit of processing of the decryption process algorithm, the decryption process is performed on this X-bit data, the compressed data of the decryption processed X-bits is then reorganized into Y-bit parts, which is the unit of processing of the decompression process algorithm, and decryption and decompression processing in units of the least common multiple lcm (X,Y) are realized as described above by decompressing the Y-bit compressed data.

[0195]

The common key encryption/decryption circuit 24 of the player 1 for realizing the processing is composed of an input block 30 and an encryption/decryption block 31, and the decompressing circuit 26 is composed of a decompression block 32 and an output block 33. The buffer memory 25 is arranged between the common key encryption/decryption circuit 24 and the decompressing circuit 26.

[0196]

As a more specific example, if the encryption processing for the digital content is performed herein using DES (Data Encryption Standard) encryption for example, this encryption processing and the corresponding decryption processing are performed in units of 64 bits.

[0197]

The decompression processing for compressed digital content is currently often performed in units of 1K to 2K bits/channel, although this depends on a compression ratio and a sampling frequency thereof. It is assumed here for the sake of convenience that the decompression processing is performed in units of 1.28K bits.

[0198]

Therefore, in a system using the DES encryption method and the compression/decompression method in units of 1.28K bits, the least common multiple lcm becomes 1.28K.

[0199]

Under such conditions, the encrypted and compressed digital content is inputted in the input block 30 of the common key encryption/decryption circuit 24 of FIG. 26. In the input block 31 [sic], the encrypted and compressed digital content is divided into X bits of processing units of the algorithm of the decryption processing, namely 64 bits of data, which are then outputted to the encryption/decryption block 31.

[0200]

The encryption/decryption block 32 [sic] does decryption processing on the X-bit data, namely the 64-bit data, in units of 64 bits. The 64 bit compressed data that is obtained by this 64-bit decryption is sent to the buffer memory 25.

[0201]

According to an instruction from the controller 16, the buffer memory 25 outputs in a batch the 1.28K bits of compressed data when Y bits of processing unit of the algorithm of decompression processing, namely 1.28K bits of compressed data have been accumulated. This compressed data is sent to the decompression block 32 of the decompressing circuit 26.

[0202]

The decompressing circuit 26 decompresses the inputted 1.28K bits of compressed data and outputs this decompressed data to the output block 33.

[0203]

Moreover, the controller 16 controls the processing in the decryption block 31 and the processing in the decompression block 32 while monitoring the amount of data accumulated in the buffer memory 25

[0204]

It should be noted that, in this case, performing the decryption processing in units of 20 (=1280/64) concurrently provides a faster processing system.

[0205]

In addition, unlike the hardware constitutions as shown in FIGS. 2 and 26, if the processing is performed based on a programmable device, the controller 16 for example, performs the processing based on a decryption program or a decompression program according to the status of the buffer memory 25.

[0206]

In the description made so far, the example in which the compressed and then encrypted digital content is supplied to the player 1 and the player 1 decrypts and then decompresses this digital content was used. However, the same effect as described above can be obtained by decompressing and decrypting the encrypted and then compressed digital content.

[0207]

Moreover, the present invention is not limited to compression/decompression and encryption/decryption algorithms, and is valid for all sorts of methods.

[0208]

Thus, according to the present invention, the security of digital content transferred through a network is enhanced.

[0209]

The following describes the operation of generating the security ID.

[0210]

In the method, such as in the present embodiment, in which point information is obtained in advance and the obtained point information is decremented according to the viewing of digital content, as described above, the administration center 211 on the network receives a request for point information purchase from the user terminal 50 of the user 200, makes a desired confirmation with the financial organization 220 and others, encrypts that point information, and sends the encrypted point information to the player 1 of the user 200 through the network.

[0211]

In the method, such as in the present embodiment, in which point information is obtained in advance and the obtained point information is decremented according to the viewing of digital content, transfer of similar data (for example, encrypted information "request for replenishment of 3,000 yen of point information" and corresponding information "3,000 yen of point information") between the administration center 211 and the player 1 (the user terminal 50) every time point information is purchased poses a problem of money replenishment based on so-called "spoofing" the financial organization 220 by a malicious person, for example. "Spoofing" the financial organization herein denotes that a malicious person disguises himself as an authentic user (the user 200 in the present embodiment) to illegally obtain point information, for example.

[0212]

Namely, if similar data is transferred every time point information is purchased, for example, a malicious person could tap that data from the communication line, generate the similar data, and send a request to the administration center 211 to send point information to that malicious person. In this case, there is a risk that the malicious person can get point information, and furthermore, that the fee for the obtained point information will be billed to the authentic user 200.

[0213]

At that point, in order to prevent such an illegal act, the system according to the present embodiment uses random numbers generated by a random number generating capability operatively associated with both the receiving side (the player 1) and the distributing side (the administration center 211) in order to increase the security. In the present embodiment, the security ID is generated as these random numbers. It should be noted that the random number generation can be operatively associated between the receiving side and the distributing side by synchronizing the operations of both side by initializing the timer 18, for example, at the user registration sequence, for example.

[0214]

Namely, an operation of obtaining point information, for example, by the player 1 from the administration center 211 by use of this random number (the security ID) is performed as follows.

[0215]

Data to be sent from the administration center 211 to the player 1 when purchasing point information includes point information encrypted by the common key (the session key) previously obtained from the player 1 and the security ID generated as described above, for example.

[0216]

The controller 16 of the player 1 sends the data received from the administration center 211 to the common key encryption/decryption circuit 24 as described above for the decryption processing by use of the common key. Thus, the point information and the security ID sent from the administration center 211 can be obtained.

[0217]

Thereafter, the controller 16 of the player 1 compares the security ID sent from the administration center 211 with the security ID generated by the security ID generating circuit 19 of the controller 16. If a match is found between the security ID from the administration center 211 and the security ID generated by the security ID generating circuit 19 of the controller 16, the point information sent from the administration center 211 is stored in the point information storage memory 28.

[0218]

Thus, only the player 1 of the approved user 200 can obtain the point information. In other words, even if a malicious person, who has the same kind of player as the player 1 of an authorized user 200, attempts to obtain point information illegally using the spoofing, the security ID of the player possessed by the malicious person will not match the security ID sent from the administration center 211, therefore making it impossible for this person with malicious intent to illegally obtain the point information by spoofing.

[0219]

Of course, the security ID generated by the player 1 of the user 200 is generated in the security ID generating circuit 19 installed in the integrated circuit 10 of the player 1, and cannot be accessed from the outside, and therefore a malicious person cannot steal this security ID.

[0220]

Various constitutions are available that generate a random number as the security ID. One of these constitutions is shown in FIG. 27 as an example. The constitution shown in FIG. 27 is a specific example of the security ID generating circuit 19 shown in FIG. 2.

[0221]

In FIG. 27, a unidirectional function generating circuit 40 generates a so-called unidirectional function. The unidirectional function is a function that is comparatively easy to calculate, however, calculating the inverse function is far more difficult. This unidirectional function can also be received in advance by confidential communication or the like and can be stored in the unidirectional function generating circuit 40. It should be noted that the unidirectional function generating circuit 40 can also be adapted to generate the unidirectional function by use of time information from the timer 18 in the integrated circuit 10 of FIG. 2 as an input function. The unidirectional function is then sent to a random number decision circuit 43.

[0222]

Moreover, a user constant generating circuit 41 generates a predetermined user constant specified for each user. This user constant is sent in advance by confidential communication or the like and stored in the user constant generating circuit 41. It should be noted that, for this user constant, the user ID stored in the user ID storage memory 23 can be used, for example.

[0223]

A random number database 42 stores random numbers. For example, 99 random numbers are stored.

[0224]

A communication count storage circuit 44 stores communication count information sent from the controller 16, for example. The communication count information is information indicating the number of times communication has been made between the player 1 and the administration center 211.

[0225]

The unidirectional function, user constant, and communication count information are sent to the random number decision circuit 43. The random number decision circuit 43, based on the time information received from the timer 18, for example, generates random numbers in a range (for example, 99 random numbers) stored in the random number database 42 from the unidirectional function and user constant.

[0226]

Namely, if the communication count information indicates a first communication, for example, the random number decision circuit 43 takes the 99th random number from the random number database 42. Moreover, if the communication count information indicates an nth communication, for example, the random number decision circuit 43 takes the 100-nth random number from the random number database 42. The obtained random number is then outputted as the security ID.

[0227]

The constitution of this security ID generation is the same on both the player 1 and the administration center 211.

[0228]

Note that, when the random numbers stored in the random number database 42 have all been used, 100th to 199th random numbers are newly computed in the random number decision circuit 42 [sic] or new random numbers or unidirectional functions are sent by confidential communication or the like, and these are stored in the random number database 42 or the unidirectional functions are incorporated in the unidirectional function generating circuit 40.

[0229]

Moreover, in the above description, the security of every communication is enhanced by generating random numbers (the security ID). However, in the present embodiment, a different common key (a session key) is programmably generated every time communication is made between the user 200 and the administration center 211, thereby enhancing the security still further.

[0230]

The following describes a manner in which a random number is inserted in a send statement (for example, a message or the like) to be actually transmitted, and in which this statement is encrypted by the session key, and a manner in which the random number is taken out of the received statement to confirm the validity, with reference to FIGS. 28 and 29. It should be noted that, in the examples of FIGS. 28 and 29, a signature (namely, a digital signature) is attached to the send statement.

[0231]

In FIG. 28, first, as the flow for encrypting and transmitting the common key using public-key encryption, the session key is generated as a common key for use in communications in a communication common key generating process P7, and this common key is encrypted by the receiving side public key in a public-key encryption process P8. The encrypted common key is then sent to the receiving side.

[0232]

Meanwhile, as the flow when a message is encrypted using common key encryption and transmitted as a send statement, for example, a message M is generated in a message generating process P1, and, in addition, a random number (the security ID) is generated in a random number generating process P5. The message M and the random number are sent to common key encryption process P6. In this common key encryption process P6, the message M and the random number are encrypted by use of the common key generated in the communication common key generating process P7.

[0233]

Furthermore, if the digital signature is to be attached, the message M is sent to a hash value computing process P2. In the hash value computing process P2, a so-called hash value is computed from the message M. It should be noted that a hash value is address information obtained by a hashing method. In the hashing method, a predetermined computation is performed on one part (a keyword) of data (in this case, the message M) and the result thereof is used as an address. A hash value (M) obtained from this message is sent to secret key encryption process P4 as a digital signature. In this secret key encryption process P4, the digital signature is encrypted by the secret key of the sending side. The encrypted digital signature is sent to common key encryption process P6. In the common key encryption process P6, the digital signature is encrypted by use of the common key generated in the communication common key generating process P7.

[0234]

The message M, digital signature, and random number are sent to the receiving side.

[0235]

The following describes a processing flow in the receiving side corresponding to FIG. 28, using FIG. 29.

[0236]

In FIG. 29, first, as the process flow for decryption of the common key by public key encryption, the common key sent from the sending side is decrypted by the secret key of the receiving side in secret key decryption process P11.

[0237]

Meanwhile, as the flow for decryption of the message M encrypted using the common key encryption method, in a common key decryption process P13, the sent message M is decrypted by the secret key decryption process P11 using the decrypted common key. This decrypted message M is sent to another process by an other function transmission process P20.

[0238]

Moreover, in the flow for decrypting a digital signature, a hash value decrypted by the common key decryption process P13 is decrypted by a public key decryption process P14 using the sending side public key. At the same time, in a hash value computing process P17, a hash value is computed from the message M. The hash value decrypted by the public key decryption process P14 and the hash value computed by the hash value computing process P17 are compared by a comparing process P19 to confirm that there has been no tampering.

[0239]

In addition, regarding the sent random numbers, a random number decrypted by the common key decryption process P13 and a random number generated by a receiving side random number generating process P21 are compared by a validity confirming process P22 to confirm that they are valid.

[0240]

Now then, in the system of the present embodiment shown in the above-mentioned FIG. 1, as the system for the user 200, a system administration company 210, a virtual store 230, and a content provider 240 are provided.

Furthermore, the financial organization 220 of FIG.1, for example, is an external bank or the like.

[0241]

The administration center 210 [sic] of the system administration company 210 performs almost all of the important system tasks, such as managing the display and distribution of the digital content in the virtual store 230, collecting, distributing, and managing user 200 billing information and various other information between the user 200 and the financial organization 220, encrypting digital content from the content provider 240, managing security for the information that is handled, and the like.

[0242]

However, in a system that uses a network to distribute digital content such as that described above, communications will become concentrated on the system side when the user is obtaining digital content from the system and when billing for the use of the digital content, raising fears that the user will not be able to obtain a satisfactory response.

[0243]

Accordingly, in another embodiment of the present invention, it is possible to prevent the concentration of communications as described above and to enhance communication response by dividing up the functions of the system administration company 210, more specifically, the functions of the administration center 211, as follows.

[0244]

That is, in another preferred embodiment of the present invention, as shown in FIG. 30, the constitution of the system for the user 200 is divided into a content display and distribution organization 310 having functions for displaying and distributing digital content, a billing information administration organization 320 having functions for managing the billing information of users in fixed regions, and a system administration organization 330 having functions for managing security for the entire system, such as generating data for the encryption of digital content and the like, distributing the generated data to the content display and distribution organization 310, collecting information from the billing information administration organization 320, and distributing revenues. Each of the organizations 310, 320, and 330 is able to communicate with the user 200 independently.

[0245]

In a constitution like that in FIG. 30, a plurality of content display and distribution organizations 310 can be dispersedly arranged over a worldwide network, making it possible for the user 200 to access a content display and distribution organization 310 in any region as long as the communications charges are paid. For example, when the user 200 wants to obtain digital content, access is made from the user 200 to the content display and distribution organization 310 to obtain the digital content. The digital content at this time has been encrypted and the like by the system administration organization 330, that is, the digital content has been placed in a state capable of being transmitted directly to the user 200 using the network.

[0246]

Moreover, the billing information administration organization 320 handles billing information, and therefore, from the standpoint of security management, preferably should not take on too many users. Therefore, a billing information administration organization 320 is established for each of a moderate number of users. However, it is preferable that this number be optimized since the trade-off is an increase in a number of attack points (billing information administration organizations 320) for third-parties harboring malicious intent. For example, when the user 200 carries out communications related to billing, the user 200 accesses the billing information administration organization 320.

[0247]

The system administration organization 330 improves security by collectively managing information that is important from the standpoint of security, such as subscriptions to the user system, the registration of settlement methods, the collection of money from users, and the distribution of profits to the proprietors, the content display and distribution organizations 310, the billing information administration organizations 320, and other such profit recipients, and the like. However, the system administration organization 330 need not be provided in only one location in the world; rather it is preferable that it be established in certain coherent units, for example, in units such as countries or the like. For example, when the user 200 is to carry out a communication that is important from the standpoint of security, such as subscribing to this system, registering a settlement method, or the like, the user 200 does so by accessing the system administration organization 330. The collection of money from the relevant user and the distribution of profits to the profit recipients are performed collectively by the system administration organization 330, which obtains the information from the billing information administration organization 320. Furthermore, the source data, that is, the content possessed by the copyright holder and the like is supplied to the system administration organization 330, converted to encrypted digital content at this point, and distributed to the content display and distribution organization 310.

[0248]

For example, by allocating the system functions to the three organizations

310, 320, and 330, and enabling the user 200 to directly access each of the organizations 310, 320 and 330 as described above, it becomes possible to prevent the concentration of communications and to improve communication response. Furthermore, the content display and distribution organization 310 also makes it possible to deal with things that already exist, such as so-called virtual malls, and is effective for sales promotion as well, making it attractive to users. Separating the billing information administration organization 320 is beneficial for preventing fraud in collusion with content display and sales functions. Moreover, because the users being managed are held to a fixed number, administrative functions aimed at fraud are also more effective.

[0249]

The flow of information when a user subscribes to the system, purchases point information, obtains a content key for use in decrypting encrypted digital content, and the like, the flow when distributing content and information for viewing the content, and the flow of billing information according to content use in the system of another preferred embodiment of the present invention shown in FIG. 30 will be explained below.

[0250]

First, the important portions of the flow when a user subscribes to the system will be explained using FIG. 31.

[0251]

When a user subscribes to the system and is registered, the registration operation is performed by a user subscription support function block 402 of the system administration organization 330 in accordance with the following procedures.

[0252]

First of all, information indicating an intention to subscribe to the system, such as a subscription intention sending operation T41, is sent from the user 200, that is, the player 1 and user terminal 50, to the system administration organization 330 through the network. The subscription intention information, which was inputted into a communication function block 401 of the system administration organization 330, is sent to the user subscription support function block 402.

[0253]

The user subscription support function block 402, upon receiving the subscription intention information, sends file information that is required for subscription, such as a file required for subscription sending operation T42, to the user 200 through the communication function block 401.

[0254]

On the basis of the file required for subscription sent from the system administration organization 330, the user 200 prepares a subscription application in accordance with a prescribed format. The prepared subscription application is sent to the system administration organization 330, such as in subscription application sending operation T43.

[0255]

The user subscription support function block 402, upon receiving the subscription application, sends information explaining the client function to the user 200, such as in client function sending operation T44.

[0256]

The user 200, upon receiving the client function information, sends to the system administration organization 330 user information, such as a user information sending operation T45, including, for example, the above-mentioned bank account number, credit card number, name, contact address, and the like.

[0257]

The user subscription support function block 402, upon receiving the user information that was sent, notifies the user 200 of information indicating that the subscription registration procedure has been completed, such as a registration procedure complete notification T46.

[0258]

Furthermore, after this user subscription registration procedure has been completed, the user subscription support function block 402 of the system administration organization 330 transfers the user information, such as a user information sending operation T47, to the billing information administration organization 320 through the communication function block 401. The billing information administration organization 320, which receives this user information, stores the user information in a database function block 367.

[0259]

Thus, the main flow at the time a user subscribes to the system is completed. Furthermore, the other constitutions included in FIG. 31 will be explained later.

[0260]

Hereafter, the main portions of the flow of information when purchasing point information, obtaining a key for decrypting encrypted digital content, and the like will be explained using FIG. 32. Furthermore, information on the purchase of point information and/or a content key for decrypting encrypted digital content is information needed for using content, and as such, this information will be abbreviated as use right information in the explanation that follows.

[0261]

When a user obtains important information for use in the system (as used here, a content use right), the user 200 accesses the billing information administration organization 320, which has been assigned in advance to be in charge of each of the users 200. A use right issuing function block 362 of the billing information administration organization 320 responds to the access requesting content use right information sent from the user 200, and issues a use right in accordance with the following procedure.

[0262]

First, the user 200 sends the billing information administration organization 320 information, such as a purchase request sending operation T51, indicating a desire to purchase a use right. The information indicating that the user desires to purchase a use right is purchase request information from the user 200 conforming to a prescribed format. The purchase request information, which is inputted to a communication function block 361 of the billing information administration organization 320 through the network, is sent to a use right issuing function block 362.

[0263]

The use right issuing function block 362, upon receiving the purchase request information, generates new use right information based on the user information stored in a database function block 367, and sends the use right information to the user 200, such as a new use right sending operation T52.

[0264]

The user 200, upon confirming the receipt of the new use right, prepares a receipt confirmation in accordance with a prescribed format, and sends this receipt confirmation to the use right issuing function block 362 of the billing information administration organization 320, such as a receipt confirmation sending operation T53.

[0265]

Thus, the main flow at the time of use right purchase is completed. Furthermore, the other constitutions included in FIG. 32 will be explained later.

[0266]

Hereafter, the main portions of the flow when distributing content and information for viewing the content (as used here, a use condition and a content key) will be explained using FIG. 33.

[0267]

First, a content obtaining function block 342 of the content display and distribution organization 310 sends a bill to the system administration organization 330 for digital content, such as a content bill sending operation T62.

[0268]

The system administration organization 330, which receives the content bill, manipulates the requested content in a content distributing function block 404 so that it can be distributed. That is, the content distributing function block 404 generates digital content that is in a state capable of being sent to the user 200 (encrypted digital content). This manipulated digital content is sent to the content display and distribution organization 310, such as in a content sending operation T63.

[0269]

The content display and distribution organization 310 stores the manipulated digital content in a content database function block 345.

[0270]

Furthermore, as content-viewing information, the content distributing function block 404 of the system administration organization 330 sends a content ID, use condition, and content key for decrypting encrypted content to the billing information administration organization 320, such as in an information for viewing content sending operation T64 .

[0271]

The billing information administration organization 320 receives the content viewing information in a content key/use condition receiving function block 363, and stores this information in the database function block 367.

[0272]

Subsequently, the user 200 accesses the content display and distribution organization 310 and obtains content, such as in a content-obtaining request T61. That is, the content display and distribution organization 310, upon receiving a request to obtain content from the user 200 through a communication function block 341, reads out encrypted digital content stored in a content database function block 354, and sends the digital content that has been read out to the user 200.

[0273]

Thereafter, the user 200 accesses the billing information administration organization 320 using an information for viewing content request T65 and obtains content viewing information, such as in an information for viewing content sending operation T66. That is, the billing information administration organization 320, upon receiving a request from the user 200 for a use condition and content key as content viewing information from the user through the communication function block 361, issues the content key and use condition from the content key/use condition issuing function block 364 and sends the same to the user 200 through the communication function block 361.

[0274]

Thus, the flow when distributing content and content viewing information is completed. Furthermore, the other constitutions included in FIG. 33 will be explained later.

[0275]

Hereafter, the main portion of the flow of a settlement when content has actually been viewed, that is, content use fee settlement will be explained using FIG. 34.

[0276]

First, after the user 200 has viewed the content, the user 200, for example, sends point usage information, that is, a record of content usage as described above, to the billing information administration organization 320, such as a settlement statement sending operation T71. Upon receiving the content usage record that has been sent from the user 200 through the communication function block 361 in this manner, a settlement procedure accepting function block 365 of the billing information administration organization 320 receives the content usage record and issues a settlement confirmation corresponding thereto. The settlement confirmation is sent to the user 200 through the same communication function block 361, such as a settlement confirmation sending operation T73. This makes it possible for the user 200 to learn that settlement has been performed.

[0277]

Subsequently, a settlement procedure accepting function block 365 of the billing information administration organization 320 issues information for issuing a use right from a use right issuing function block 362. This use right issuing information is sent, together with a content usage record sent from the user 200, to the system administration organization 330 through the communication function block 361 such as a user settlement/content usage record sending operation T74.

[0278]

The system administration organization 330, using a collection and distribution function block 405, summarizes the information sent from the billing information administration organizations 320 scattered in various regions, tabulates the collection amounts and collection destinations with the money distribution destinations, and settles accounts through an actual financial institution.

[0279]

Thus, the flow of the settlement of content usage fees comes to an end. Furthermore, the other constitutions included in FIG. 34 will be explained later.

[0280]

In the above explanations from FIG. 30 to FIG. 34, it goes without saying that encryption and decryption are performed the same as described above in the sending and receiving of data between the user 200 and the content display and distribution organization 310, the billing information administration organization 320, and the system administration organization 330 and/or the sending and receiving of data between the content display and distribution organization 310, the billing information administration organization 320, and the system administration organization 330. Moreover, either a public-key encryption system or a common key encryption system may be used in this encryption and decryption, and as was described above, the public-key encryption system can be used as the encryption system for the content key and the common key, and the common key encryption system can be used as the encryption system for messages, various documents, and the like. Furthermore, it is also possible to use procedures for improving security by using the random number together with these encryptions, and the least common multiple as the processing unit for encryption and compressing when handling content.

[0281]

Hereafter, the specific constitutions of the organizations 310, 320, and 330 will be briefly explained.

[0282]

First, the constitution of the content display and distribution organization 310 will be explained using FIG. 35.

[0283]

In FIG. 35, the content display and distribution organization 310 broadly includes: a communication function block 341 that is in charge of communication functions with the user 200 and the system administration organization 330; a content obtaining function block 342 that is in charge of content obtaining functions; a content displaying function block 343 that is in charge of content displaying functions; a settlement function block 344 that is in charge of settlements; and a content database function block 345 for storing content.

[0284]

The content obtaining function block 342 includes: a content bill request generating function 351 that is in charge of generating a bill request when billing the system administration organization 330 for content; a content receipt generating function 352 that is in charge of generating a receipt when content has been received from the system administration organization 330; and a content database corresponding function 353 that is in charge of making sure the handled content corresponds to the content being stored in the content database function block 345.

[0285]

The content displaying function block 343 includes: a content displaying function 354 that is in charge of functions for actually displaying content in virtual stores; and a content database corresponding function 355 that is in charge of making sure the displayed content corresponds to the content being stored in the content database function block 345.

[0286]

The settlement function block 344 includes: a receipt issuing function 356 that is in charge of functions for issuing receipts; and a financial organization corresponding function 357 that is in charge of correspondence with the financial organization 220.

[0287]

Hereafter, the constitution of the billing information administration organization 320 will be explained using FIG. 36.

[0288]

In FIG. 36, the billing information administration organization 320 broadly includes: a communication function block 361 that is in charge of communication functions with the user 200 and the system administration organization 330; a use right issuing function block 362 that is in charge of functions for issuing use rights; a content key/use condition receiving function block 363 that is in charge of receiving a content key and a use condition; a content key/use condition issuing function block 364 that is in charge of issuing a content key and a use condition; a settlement procedure accepting function block 365 that is in charge of functions for accepting a settlement procedure; a distribution and receiving function block 366 that is in charge of distributing and receiving functions; and a database function block 376.

[0289]

The use right issuing function block 362 includes: a purchase request confirming function 371 that is in charge of functions for confirming a purchase request; a point data confirming function 372 that is in charge of confirming data, such as the use right balance (point information balance) of a client, that is, a user 200, a usage record (point usage information), and the like; a use right generating function 373 that is in charge of functions for generating use rights; a use right sending notice generating function 374 that is charge of functions for generating a use right sending notice; a sending function 375 that is in charge of functions for actually sending a use right and a use right sending notice; a use right reception confirming function 376 that is in charge of confirming a use right receipt; and use right issue information storing function 377 that is in charge of functions for storing information on issued use rights.

[0290]

The content key/use condition receiving function block 363 includes: a receiving function 378 that is in charge of receiving a content key and a use condition; and a storing function 379 for storing content keys and use conditions.

[0291]

The content key/use condition issuing function block 364 includes: a receiving function 380 that is in charge of functions for receiving requests to obtain content keys and use conditions; a searching function 381 that is in charge of functions for searching for and retrieving content keys and use conditions from the database function block 367; a sending function 382 that is in charge of functions for encrypting and sending content keys and use conditions; and a confirming function 383 that is in charge of functions for confirming the receipt of content keys and use conditions.

[0292]

The settlement procedure accepting function block 365 includes: a content usage record receiving function 384 that is in charge of functions for receiving and decrypting encrypted content usage records (point usage information); a content usage record confirming function 385 that is in charge of confirming content usage records; a content usage record storing function 386 that is in charge of functions for storing content usage records in the database function block 367; a completion notice generating function 387 that is in charge of functions for generating completion notices for settlement procedures; and a summarizing function 389 that is in charge of functions for collectively editing content usage records.

[0293]

The distribution and receiving function block 366 includes: a bill confirming function 390 that is in charge of functions for confirming document bills for billing for documents when carrying out collection; a usage record report generating function 391 that is in charge of functions for generating content usage record reports to be submitted to the system administration organization 330; a use right issue report generating function 392 that is in charge of functions for generating use right issue information reports to be submitted to the system administration organization 330; and a certificate confirming function 393 that is in charge of functions for confirming a certificate of report reception.

[0294]

The database function block 367 includes: a use right data function 394 that is in charge of functions for storing use right data; a content key/use right database function 395 that is in charge of functions for storing content key and use condition data; a content usage record database function 396 for storing content usage records; and a user administration database function 397 for storing information related to users.

[0295]

Hereafter, the constitution of the system administration organization 330 will be explained using FIG. 37.

[0296]

In FIG. 37, the system administration organization 330 broadly includes: a communication function block 401 that is in charge of functions for communicating with the user 200, the content display and distribution organization 310, and the billing information administration organization 320; a user subscription support function block 402 that provides support at the time of user subscription; a content distributing function block 404 that is in charge of the distribution of content; a database function block 403; and a collection and distribution function block 405 that is in charge of money collection and distribution functions.

[0297]

The user subscription support function block 402 includes: a subscription application generating and sending function 411 that is in charge of generating and sending subscription applications; a common key receiving function 412 that is in charge of functions for receiving and decrypting encrypted common keys; a subscription application confirming function 413 that is in charge of functions for confirming subscription applications sent from users 200; an ID generating function 414 that is in charge of functions for generating client IDs, that is, user IDs; a subscription application storing function 415 that is in charge of functions for storing subscription applications in the database function block 403; a client function generating function 416 for generating client functions; and a registration information storing function 417 that is in charge of functions for storing registration information in the database function block 403.

[0298]

The database function block 403 includes: a user administration database function 418 for storing and managing user information; a content database function 419 for storing content; a billing information administration organization database function 420 for storing and managing billing information administration organization 320 information; and a content display and distribution organization database function 421 for storing and managing content display and distribution organization 310 information.

[0299]

The content distributing function block 404 includes: a bill confirming function 422 that is in charge of functions for confirming content bills; a content searching function 423 that is in charge of functions for searching raw content, that is, content prior to manipulation (source data) from the content database function 419 of the database function block 403; a content ID generating function 424 for generating content IDs; a content key generating function 425 for generating content keys; a content use condition generating function 426 for generating content use conditions; a content compressing function 427 for compressing raw content, that is, content prior to manipulation; a content manipulating function 428 for encrypting content; a storing function 429 that is in charge of functions for storing content IDs, content keys, and use conditions in the content database function 419 of the database function block 403; a content sending function 430 that is in charge of functions for sending content through the communication function block 401; a content receipt confirming function 431 that is in charge of functions for confirming content receipts; an ID/key/use condition sending function 432 that is in charge of functions for sending content IDs, content keys, and use conditions through the communication function block 401; and an ID/key/use condition receipt confirming function 433 that is in charge of functions for confirming receipts for content IDs, content keys, and use conditions.

[0300]

The collection and distribution function block 405 includes: a document bill generating function 434 for generating document bills for use in collection; a content use right receiving function 435 that is in charge of functions for receiving content use rights through the communication function block 401; a content usage record receiving function 436 that is in charge of functions for receiving content usage records through the communication function block 401; a reception confirmation generating function 437 that is in charge of functions for generating reception confirmations; a calculating and bill generating function 438 for calculating charges to be billed to users and generating bills; and a calculating and delivery notice generating function 439 for calculating dividends when distributing usage fees collected in accordance with usage to proprietors and generating delivery notices.

[0301]

Hereafter, the constitution of the user 200 corresponding to the system of the other preferred embodiment will be explained using FIG. 38. Furthermore, FIG. 38 collectively represents the functions of the player 1 and user terminal 50.

[0302]

In FIG. 38, the constitution of the user 200 side broadly includes: a communication function block 451 that is in charge of functions for communicating with the system administration organization 330, the content display and distribution organization 310, and the billing information administration organization 320; a content obtaining function block 452 that is in charge of obtaining content; a use right purchasing function block 453 that is in charge of purchasing point information, content keys, use conditions, and other such use rights; a content key/use condition obtaining function block 454 that is in charge of obtaining content keys and use conditions; a settlement procedure function block 455 that is in charge of settlement procedures; a user subscription support function block 456 that is in charge of functions for supporting subscriptions to the system; a content-viewing billing function block 457 that is in charge of functions for billing for viewing content; and a database function block 458.

[0303]

The content obtaining function block 452 includes: a content obtaining function 461 that is in charge of functions for actually obtaining content; and a content storing function 462 that is in charge of functions for storing content in storage media.

[0304]

The use right purchasing function block 453 includes: a purchase request generating function 463 for generating purchasing requests for use rights; a summarizing function 464 that is in charge of summarizing data, such as a client (user) use right balance (point balance), usage records (point usage information), and the like; a use right installing function 465 that is in charge of functions for installing various information as use rights; and a use right receipt generating function 467 for generating a use right receipt.

[0305]

The content key/use condition obtaining function block 454 includes: an obtain request generating function 468 for generating requests for obtaining content keys and use conditions; a receiving function 469 that is in charge of receiving content keys and use conditions; and a receipt generating function 470 for generating receipts for content keys and use conditions.

[0306]

The settlement procedure function block 455 includes: a summarizing function 471 for summarizing content usage records (point usage information); and a completion notice receiving function 472 that is in charge of receiving completion notices for settlement procedures.

[0307]

The user subscription support function block 456 includes: a subscription application generating function 473 that is in charge of generating a subscription application; a client function installing function 474 that is in charge of installing the client functions, that is, initializing the player 1 of the user; and a registration information generating function 475 that is in charge of functions for generating registration information.

[0308]

The content viewing billing function block 457 includes: a content searching function 476 that is in charge of searching for content stored in storage media; a use right confirming function 477 that is in charge of confirming use rights; a simplified content viewing function 478 for simply playing back content when, for example, content is being selected; a billing function 479 for managing billing information (point information); a content decryption function 480 for decrypting encrypted content; a content decompressing function 481 for decompressing compressed content; and a content viewer function 482 for making the details of content stored, for example, in storage media recognizable.

[0309]

The database function block 458 includes: a use right database function 483 for storing use right data; a content key/use condition database function 484 for storing content keys and use conditions; a content usage record database function 485 for storing content usage records; and a user information database function 486 for storing user information.

[0310]

Hereafter, the specific utilization configurations of the player 1 and user terminal 50 of the respective preferred embodiments as described above will be explained using FIGS. 39 and 40.

[0311]

As shown in FIG. 39, the analog output terminal 2, PC interface terminal 3, and storage medium I/O terminal 4 of the player 1 are arranged in a state protruding out from the player 1 enclosure, and a storage medium 61 is connected through the storage medium I/O terminal 4. Furthermore, the player 1 and storage medium 61 are formed, for example, so as to be able to be housed inside a case 60, and the analog output terminal 2 and PC interface terminal 3 of the player 1 are arranged, for example, at one end of this case 60.

[0312]

The case 60 in which the player 1 and storage medium 61 are housed is formed so as to be insertably connected to the input/output port 53 of the personal computer 50 serving as the user terminal 50 from the side on which the analog output terminal 2 and PC interface terminal 3 of the player 1 are arranged.

[0313]

The personal computer 50 has an ordinary constitution including a computer main unit with a display device 52, a keyboard 54, and a mouse 55, and interfaces corresponding to the player 1 analog output terminal 2 and PC interface terminal 3 are formed inside the input/output port 53. Therefore, the player 1 analog output terminal 2 and PC interface terminal 3 are connected to the personal computer 50 by simply inserting the case 60 housing the player 1 and storage medium 61 into the input/output port 53 of the personal computer 50.

[0314]

In the example of FIG. 39, interfaces corresponding to the player 1 analog output terminal 2 and PC interface terminal 3 are formed inside the input/output port 53 of the personal computer 50, however, for example, as shown in FIG. 40, it is also possible to arrange an adapter 62, which is capable of supporting a general-purpose input/output port interface of the personal computer 50, between the player 1 analog output terminal 2 and PC interface terminal 3.

[0315]

Based on the description given above, in a system of the preferred embodiment of the present invention, because digital content is encrypted using the content key, which is the system common key, as long as a user (player 1) is registered in the system of the preferred embodiment, the user can freely copy this encrypted content, and is able to view this content by simply obtaining the content key. Therefore, this content (encrypted content) can be easily installed in the storage medium. Alternatively, because a terminal device that does not conform to the system of the preferred embodiment is not able to decrypt the encrypted digital content, the content copyrights and the rights of the content proprietors are protected.

[0316]

Furthermore, according to the system according to the preferred embodiment of the present invention, point information is replenished using a prepaid system (a prepayment system) and point information is decremented when the content is viewed, and use information regarding these points is collected. Therefore it is possible for the proprietors (copyright holders, and the like) who hold the rights to used points, the content stores, and the like to collect viewing charges.

[0317]

In addition, security is improved since the previously described encryption is performed when point information and point usage information data are exchanged. For example, even if someone were to attempt to steal point information for billing by forging data that is exactly the same as previous data, as described above, interlinked random numbers (security IDs) are used by the system side and the player side and a transaction is carried out after confirming that the two random numbers match, thereby making it safe.

[0318]

In addition to that, the major components of the player are integrated onto a single chip, making it impossible to extract the key information and decrypted digital content to the outside. The player 1 is provided with a tamper-resistance function in the player 1 itself to prevent data from being stolen by destroying the player 1.

[0319]

As mentioned above, according to the preferred embodiments of the present invention, a high-security digital content distribution system is built.

[0320]

Furthermore, examples of the digital content can include various types of digital video data in addition to digital audio data. When using moving picture image data (including audio data) as the digital video data, for example, the Moving Picture Image Coding Experts Group (MPEG) and other such compression techniques can be used as the compression technique. Furthermore, the above-mentioned MPEG is the vernacular term for the video encryption system compiled in Working Group (WG) 11 of Sub-Committee (SC) 29 of the Joint Technical Committee (JTC) 1 of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC), and includes MPEG 1, MPEG 2, MPEG 4, and the like.

[0321]

Furthermore, as was described above, for example, the encryption technique called the Data Encryption Standard (DES) can be used as the encryption technique. Note that DES is the standard encryption technique (encryption algorithm) announced in 1976 by the National Institute of Standards and Technology (NIST) of the United States. Specifically, DES performs data conversion for each 64-bit block of data, and repeats the conversion, which uses a function, 16 times. The digital content, point information, and the like are encrypted by using a so-called common key technique using the DES. Furthermore, the common key technique is a technique in which the key data for encryption (encryption key data) and the key for decryption (decryption key data) are the same.

[0322]

Moreover, for example, so-called Electrically Erasable Programmable Read-Only Memory (EEPROM) can be used in the common key storage memory 22, communication key storage memory 21, point usage information storage memory 29, point information storage memory 28, and the like of the player 1 of FIG. 1.

[0323]

Other storage media available includes recording media such as a hard disk, a floppy disk, a magneto-optical disk, and phase-alternating magneto-optical disk, and storage media such as a semiconductor memory (IC card and the like).

[0324]

In the above-mentioned preferred embodiments, the keyboard 54, the mouse 55, and the display device 52 of the user terminal 50 are used to select content or check content displayed in the virtual store 230. However the keyboard, mouse, and display device may be simplified in function and installed on the player 1. Namely, the input section 6 and the display section 7 may be provided on the player 1, as in FIG. 2.

[0325]

[Effects of the Invention]

As is clear from the above explanation, according to the present invention, it is possible to build a system that is portable and enables digital content to be enjoyed anywhere and anytime, and also provides adequate protection against the copying and unauthorized use of the digital content and is economical.

[Brief Description of the Drawings]

FIG. 1 is a configuration diagram illustrating an entire constitution of a digital content distributing system according to an embodiment of the present invention.

FIG. 2 is a block circuit diagram illustrating a specific constitution of a player of the system according to an embodiment of the present invention.

FIG. 3 is a block circuit diagram illustrating a specific constitution of an administration center of the system according to an embodiment of the present invention.

FIG. 4 is a diagram for describing a procedure in which the player is purchased in the system according to an embodiment.

FIG. 5 is a diagram for describing a procedure for processing to be performed from digital content search to installation of digital content on a storage medium for the player in the system according to an embodiment.

FIG. 6 is a diagram for describing a procedure of purchasing point information for charging and of settlement to be made when digital content concerned has been used in the system according to an embodiment.

FIG. 7 is a diagram for describing a procedure of distributing charged fees in the system according to an embodiment.

FIG. 8 is a flowchart illustrating a processing flow in the player at the time of point purchase in the system according to an embodiment.

FIG. 9 is a flowchart illustrating a processing flow at a user terminal at the time of point purchase in the system according to an embodiment.

FIG. 10 is a flowchart illustrating a processing flow at an administration center at the time of point purchase in the system according to an embodiment.

FIG. 11 is a diagram illustrating a sequence of information transfer at the time of point purchase in the system according to an embodiment.

FIG. 12 is a flowchart illustrating a processing flow at the player at the time of acquiring digital content in the system according to an embodiment.

FIG. 13 is a flowchart illustrating a processing flow at the user terminal at the time of acquiring digital content in the system according to an embodiment.

FIG. 14 is a flowchart illustrating a processing flow at the administration center at the time of acquiring digital content in the system according to an embodiment.

FIG. 15 is a diagram illustrating a sequence of information transfer to be performed when acquiring digital content in the system according to an embodiment.

FIG. 16 is a flowchart illustrating a processing flow at the player at the time of acquiring a content key and a condition of use in the system according to an embodiment.

FIG. 17 is a flowchart illustrating a processing flow at the user terminal at the time of acquiring a content key and a condition of use in the system according to an embodiment.

FIG. 18 is a flowchart illustrating a processing flow at the administration center at the time of acquiring a content key and a condition of use in the system according to an embodiment.

FIG. 19 is a diagram illustrating a sequence of information transfer to be performed at the time of acquiring a content key and a condition of use in the system according to an embodiment of the present invention

FIG. 20 is a flowchart illustrating a processing flow in which digital content is actually viewed by use of the player and the user terminal in the system according to an embodiment.

FIG. 21 is a flowchart illustrating a processing flow at the player at the time of returning point usage information in the system according to an embodiment.

FIG. 22 is a flowchart illustrating a processing flow at the user terminal at the time of returning point usage information in the system according to an embodiment.

FIG. 23 is a flowchart illustrating a processing flow at the administration center at the time of returning point usage information in the system according to an embodiment.

FIG. 24 is a diagram illustrating a sequence for information transfer at the time of returning point usage information in the system according to an embodiment.

FIG. 25 is a flowchart illustrating a processing flow of performing decryption and decompression by the least common multiple of the processing unit of encryption and compression.

FIG. 26 is a block circuit diagram illustrating a constitution for performing decryption and decompression per unit of the least common multiple of the processing unit of encryption and compression.

FIG. 27 is a block circuit diagram illustrating a specific constitution for generating random numbers as a security ID, FIG. 28 is a diagram for illustrating an operation in which random numbers are inserted when encrypting a common key by public key encryption to transmit the encrypted common key.

FIG. 29 is a diagram for illustrating an operation in which random numbers are extracted from a received statement for confirming validity.

FIG. 30 is a diagram for describing each organization when the system functionality is divided.

FIG. 31 is diagram for describing a main portion of a processing flow at the time of user subscription to the system in an embodiment in which the system functionality is divided.

FIG. 32 is a diagram for describing a main portion of an information flow at the time of purchasing point information and acquiring a key for decrypting encrypted digital content in the embodiment in which the system functionality is divided.

FIG. 33 is a diagram for describing a main portion of a processing flow of distributing content and information for viewing the content in the embodiment in which the system functionality is divided.

FIG. 34 is a diagram for describing a main portion of a flow of fee settlement when content has been actually viewed in the embodiment in which the system functionality is divided.

FIG. 35 is a block diagram illustrating a constitution of a content display distributing organization in the embodiment in which the system functionality is divided.

FIG. 36 is a block diagram illustrating a constitution of a billing information control organization in the embodiment in which the system functionality is divided.

FIG. 37 is a block diagram illustrating a constitution of a system control organization in the embodiment in which the system functionality is divided.

FIG. 38 is a block diagram illustrating a constitution of the user side in the embodiment in which the system functionality is divided.

FIG. 39 is a diagram for describing one example of a specific usage form of the player and the user terminal.

FIG. 40 is a diagram for describing another example of a specific usage form of the player and the user terminal.

[Reference Numerals]

1 Player

2 Analog output terminal

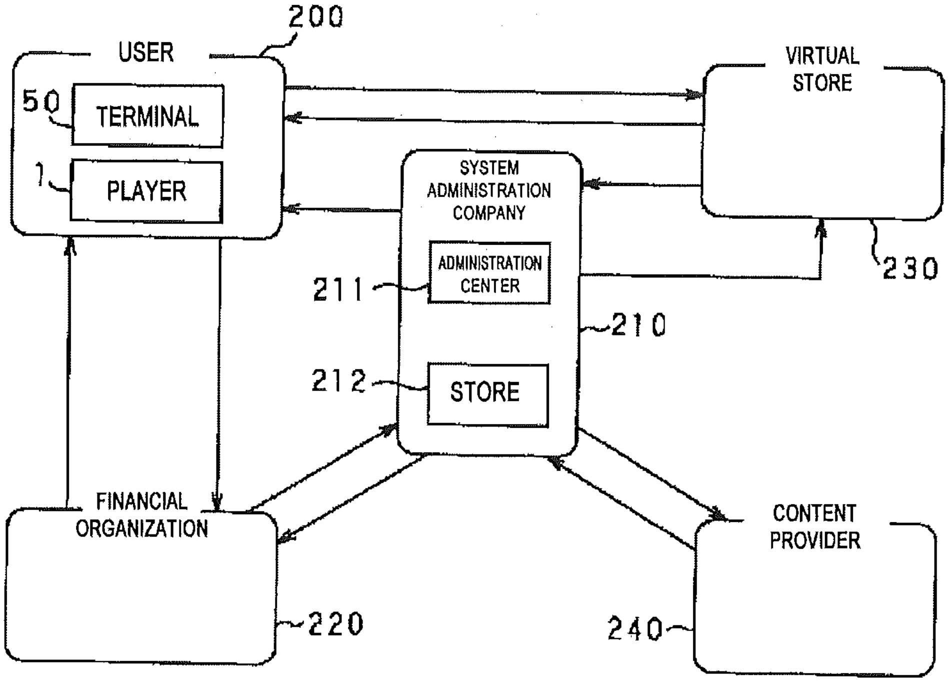
3 PC interface terminal

4 Storage medium I/O terminal

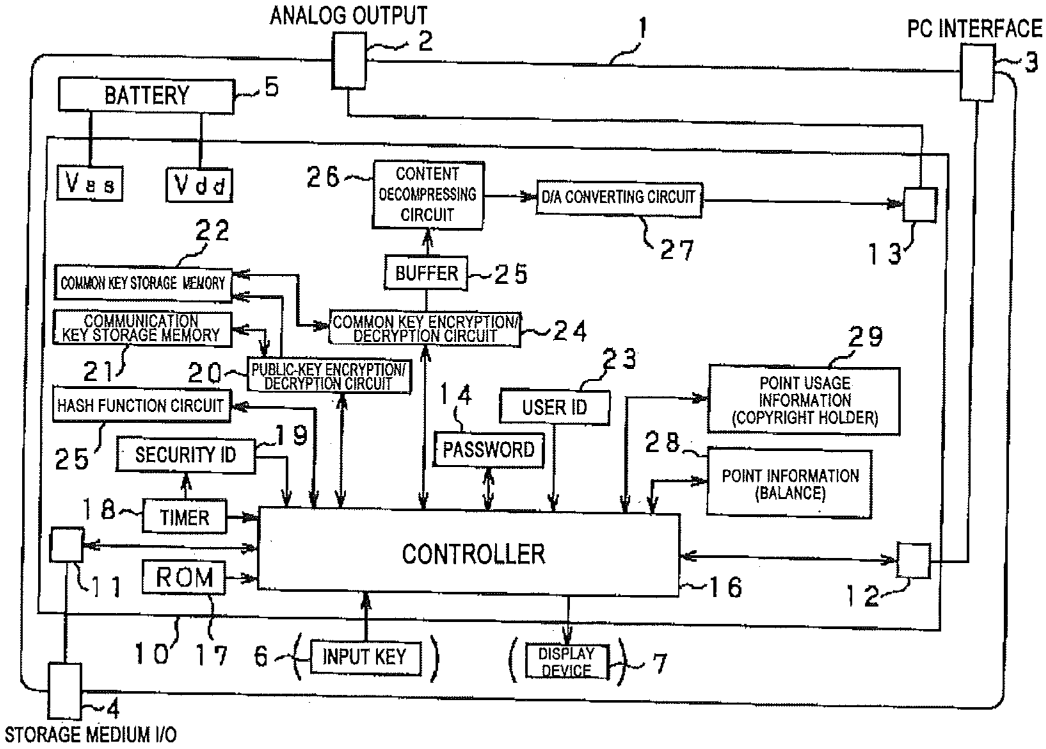
16 Controller

19 Security ID generation circuit
20 Public-key encryption/decryption circuit
21 Communication key storage memory
22 Common key storage memory
23 User ID storage memory
24 Common key encryption/decryption circuit
25 Buffer memory
26 decompressing circuit
27 D/A conversion circuit
50 User terminal
100 Content administration function block
110 User administration function block
120 Usage information administration function block
130 Administration function block
200 User side
210 System administration company
211 Administration center
220 Financial organization
230 Virtual store
240 Content provider

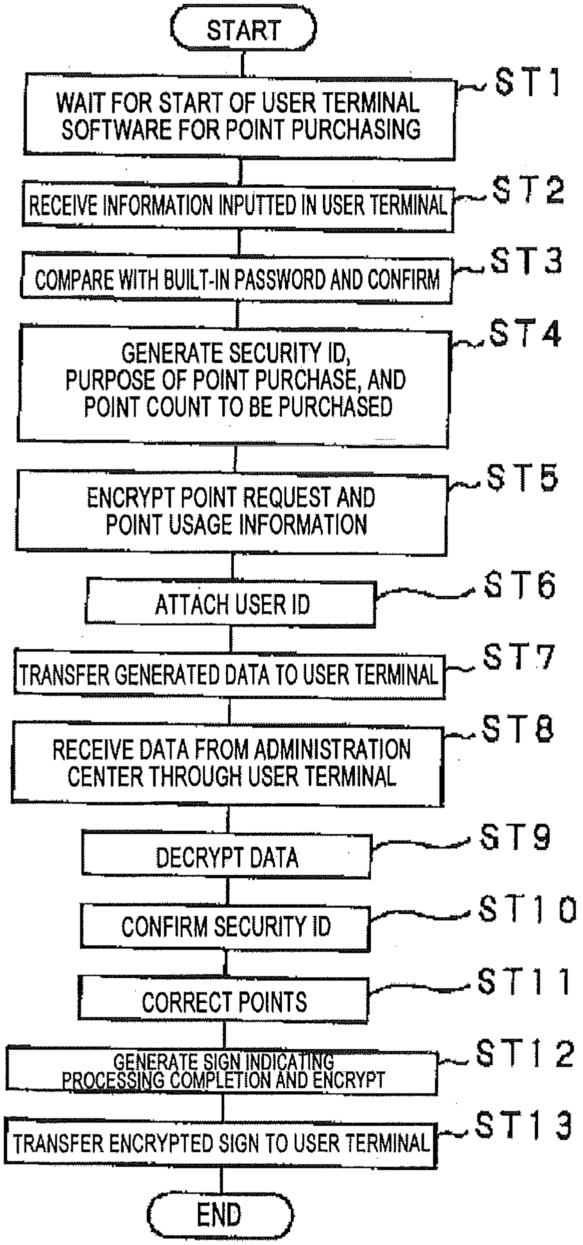
[FIG. 1]


200
USER
50
TERMINAL
1
PLAYER
VIRTUAL STORE
230
SYSTEM ADMINISTRATION COMPANY
ADMINISTRATION CENTER
211
210
212
STORE
FINANCIAL ORGANIZATION
220
CONTENT PROVIDER
240

[FIG. 2]

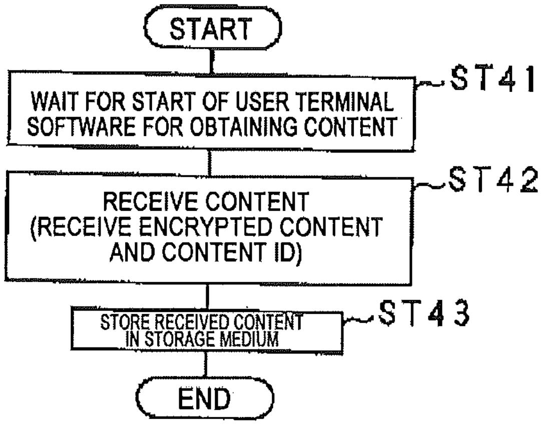

ANALOG OUTPUT
2
1
PC INTERFACE
3
BATTERY
5
Vss
Vdd
26
CONTENT DECOMPRESSING CIRCUIT
D/A CONVERTING CIRCUIT
27
13
22
COMMON KEY STORAGE MEMORY
BUFFER
25
COMMUNICATION KEY STORAGE MEMORY
COMMON KEY ENCRYPTION/ DECRYPTION CIRCUIT
24
21
20
PUBLIC-KEY ENCRYPTION/ DECRYPTION CIRCUIT
23
29
HASH FUNCTION CIRCUIT
14
USER ID
POINT USAGE INFORMATION (COPYRIGHT HOLDER)
25
SECURITY ID
19
PASSWORD
28
POINT INFORMATION (BALANCE)
18
TIMER
CONTROLLER
11
ROM
16
12
10
17
6
INPUT KEY
DISPLAY DEVICE
7
4
STORAGE MEDIUM I/O

[FIG. 8]


START
WAIT FOR START OF USER TERMINAL SOFTWARE FOR POINT PURCHASING
ST1
RECEIVE INFORMATION INPUTTED IN USER TERMINAL
ST2
COMPARE WITH BUILT-IN PASSWORD AND CONFIRM
ST3
GENERATE SECURITY ID, PURPOSE OF POINT PURCHASE, AND POINT COUNT TO BE PURCHASED
ST4
ENCRYPT POINT REQUEST AND POINT USAGE INFORMATION
ST5
ATTACH USER ID
ST6
TRANSFER GENERATED DATA TO USER TERMINAL
ST7
RECEIVE DATA FROM ADMINISTRATION CENTER THROUGH USER TERMINAL
ST8
DECRYPT DATA
ST9
CONFIRM SECURITY ID
ST10
CORRECT POINTS
ST11
GENERATE SIGN INDICATING PROCESSING COMPLETION AND ENCRYPT
ST12
TRANSFER ENCRYPTED SIGN TO USER TERMINAL
ST13
END

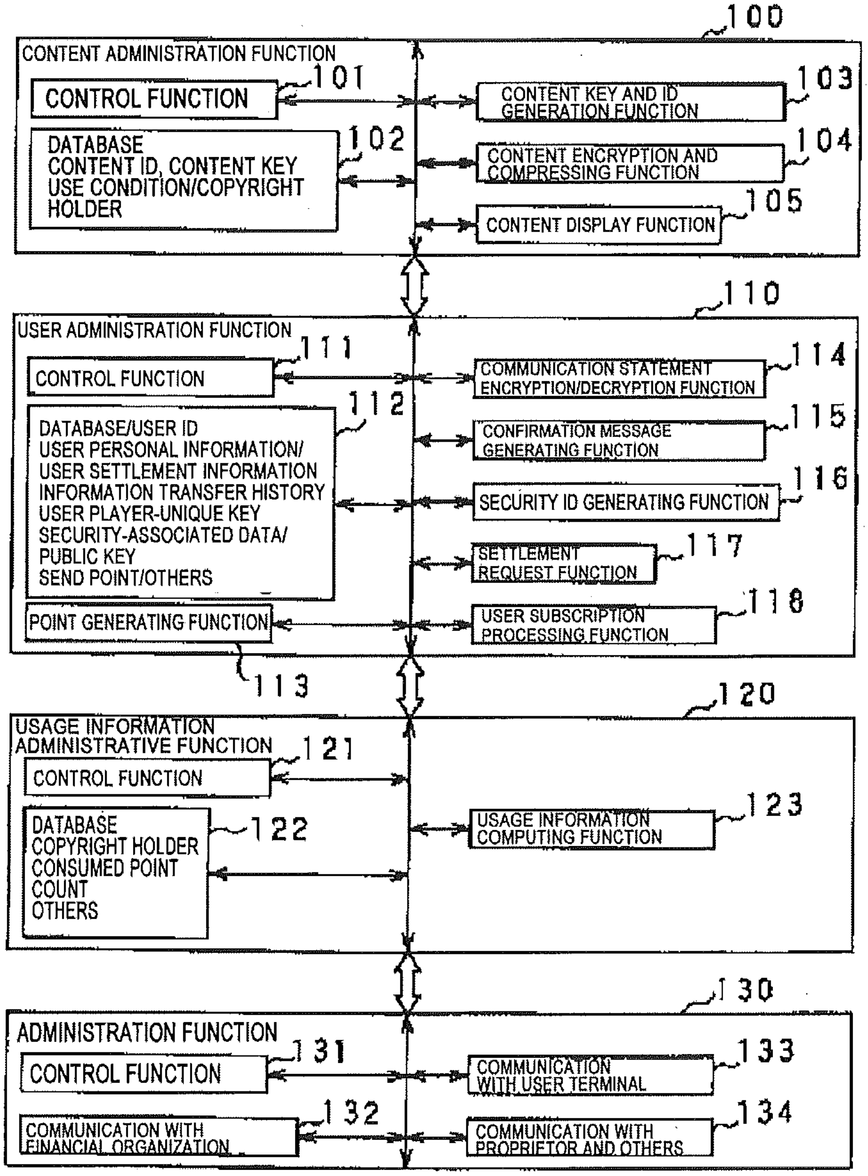
FLOWCHART FOR PLAYER WHEN PURCHASING POINTS

[FIG. 12]


START
WAIT FOR START OF USER TERMINAL SOFTWARE FOR OBTAINING CONTENT
ST41
RECEIVE CONTENT (RECEIVE ENCRYPTED CONTENT AND CONTENT ID)
ST42
STORE RECEIVED CONTENT IN STORAGE MEDIUM
ST43
END

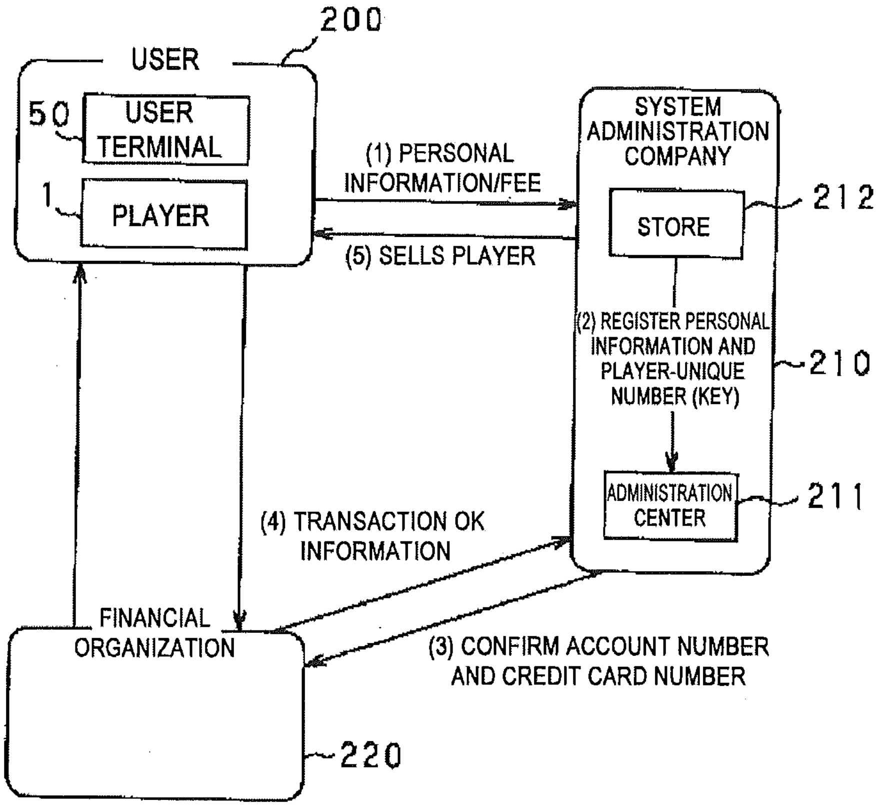
FLOWCHART FOR PLAYER WHEN OBTAINING CONTENT

[FIG. 3]

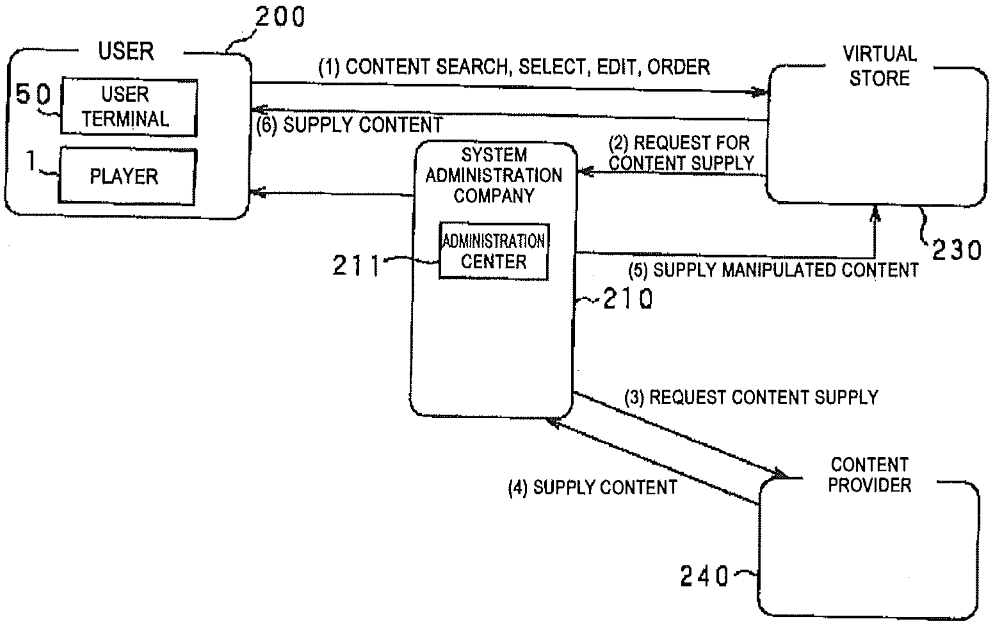


100
CONTENT ADMINISTRATION FUNCTION
101
CONTROL FUNCTION
102
DATABASE
CONTENT ID, CONTENT KEY
USE CONDITION/COPYRIGHT
HOLDER
103
CONTENT KEY AND ID GENERATION FUNCTION
104
CONTENT ENCRYPTION AND COMPRESSING FUNCTION
105
CONTENT DISPLAY FUNCTION
110
USER ADMINISTRATION FUNCTION
111
CONTROL FUNCTION
112
DATABASE/USER ID
USER PERSONAL INFORMATION/
USER SETTLEMENT INFORMATION
INFORMATION TRANSFER HISTORY
USER PLAYER-UNIQUE KEY
SECURITY-ASSOCIATED DATA/
PUBLIC KEY
SEND POINT/OTHERS
POINT GENERATING FUNCTION
113
114
COMMUNICATION STATEMENT ENCRYPTION/DECRYPTION FUNCTION
115
CONFIRMATION MESSAGE GENERATING FUNCTION
116
SECURITY ID GENERATING FUNCTION
117
SETTLEMENT REQUEST FUNCTION
118
USER SUBSCRIPTION PROCESSING FUNCTION
120
USAGE INFORMATION ADMINISTRATIVE FUNCTION
121
CONTROL FUNCTION
122
DATABASE
COPYRIGHT HOLDER
CONSUMED POINT
COUNT
OTHERS
123
USAGE INFORMATION COMPUTING FUNCTION
130
ADMINISTRATION FUNCTION
131
CONTROL FUNCTION
132
COMMUNICATION WITH FINANCIAL ORGANIZATION
133
COMMUNICATION WITH USER TERMINAL
134
COMMUNICATION WITH PROPRIETOR AND OTHERS

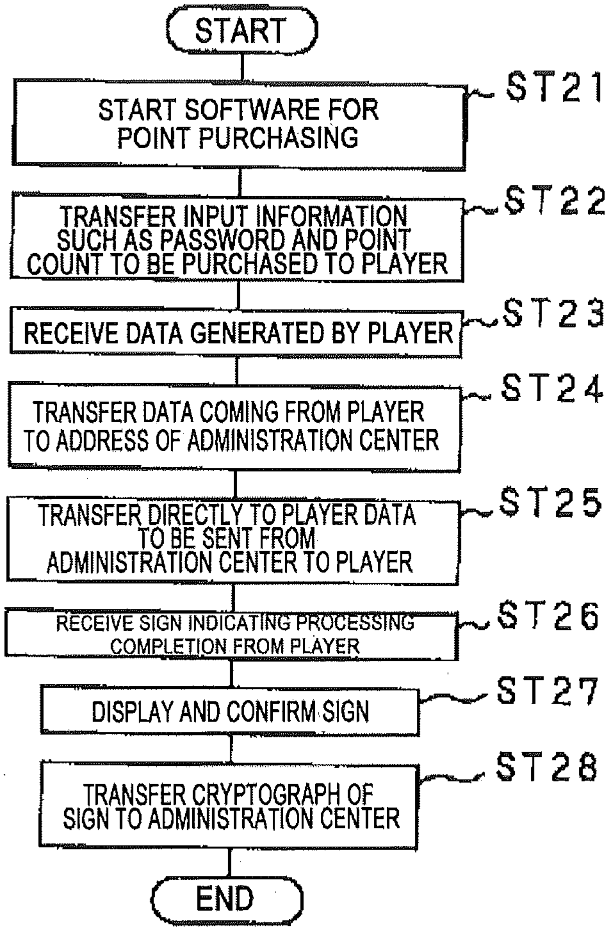
[FIG. 4]


200
USER
50
USER TERMINAL
1
PLAYER
(1) PERSONAL INFORMATION/FEE
(5) SELLS PLAYER
SYSTEM ADMINISTRATION COMPANY
STORE
212
(2) REGISTER PERSONAL INFORMATION AND PLAYER-UNIQUE NUMBER (KEY)
210
ADMINISTRATION CENTER
211
(4) TRANSACTION OK INFORMATION
FINANCIAL ORGANIZATION
(3) CONFIRM ACCOUNT NUMBER AND CREDIT CARD NUMBER
220

[FIG. 5]

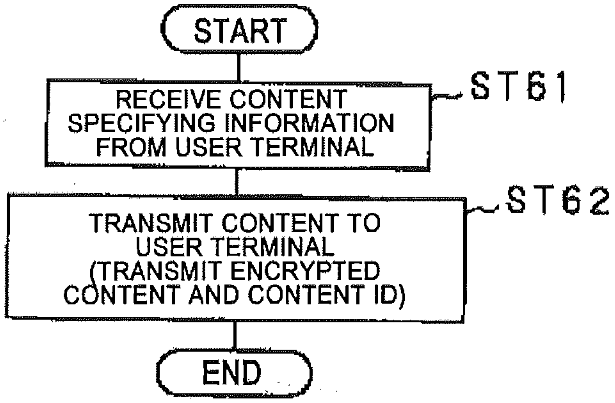

200
USER
50
USER TERMINAL
1
PLAYER
(1) CONTENT SEARCH, SELECT, EDIT, ORDER
VIRTUAL STORE
(6) SUPPLY CONTENT
SYSTEM ADMINISTRATION COMPANY
(2) REQUEST FOR CONTENT SUPPLY
ADMINISTRATION CENTER
211
230
(5) SUPPLY MANIPULATED CONTENT
210
(3) REQUEST CONTENT SUPPLY
(4) SUPPLY CONTENT
CONTENT PROVIDER
240

[FIG. 9]


START
START SOFTWARE FOR POINT PURCHASING
ST21
TRANSFER INPUT INFORMATION SUCH AS PASSWORD AND POINT COUNT TO BE PURCHASED TO PLAYER
ST22
RECEIVE DATA GENERATED BY PLAYER
ST23
TRANSFER DATA COMING FROM PLAYER TO ADDRESS OF ADMINISTRATION CENTER
ST24
TRANSFER DIRECTLY TO PLAYER DATA TO BE SENT FROM ADMINISTRATION CENTER TO PLAYER
ST25
RECEIVE SIGN INDICATING PROCESSING COMPLETION FROM PLAYER
ST26
DISPLAY AND CONFIRM SIGN
ST27
TRANSFER CRYPTOGRAPH OF SIGN TO ADMINISTRATION CENTER
ST28
END

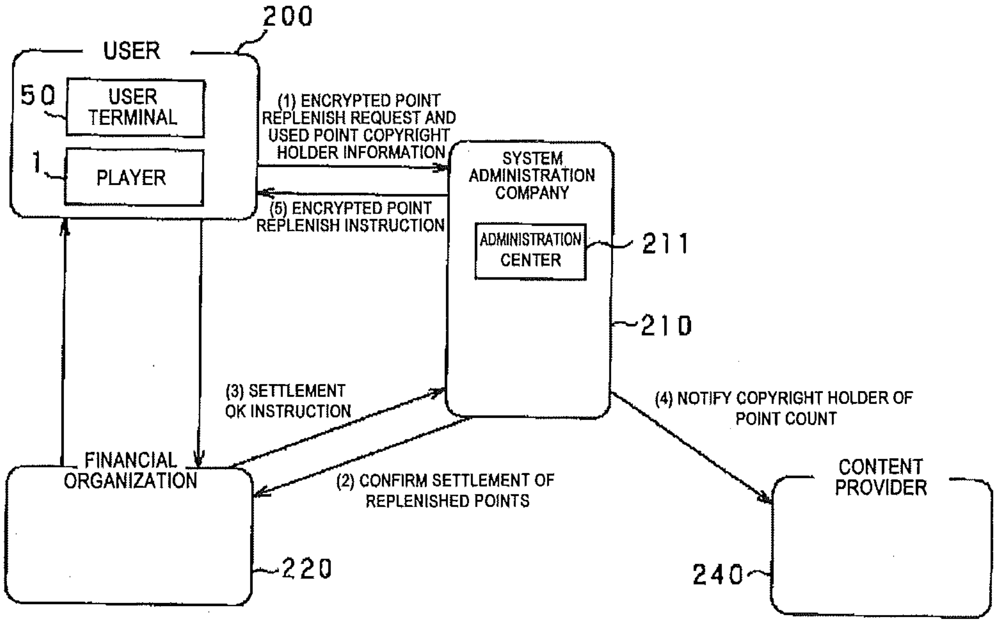
FLOWCHART FOR USER TERMINAL WHEN PURCHASING POINTS

[FIG. 14]

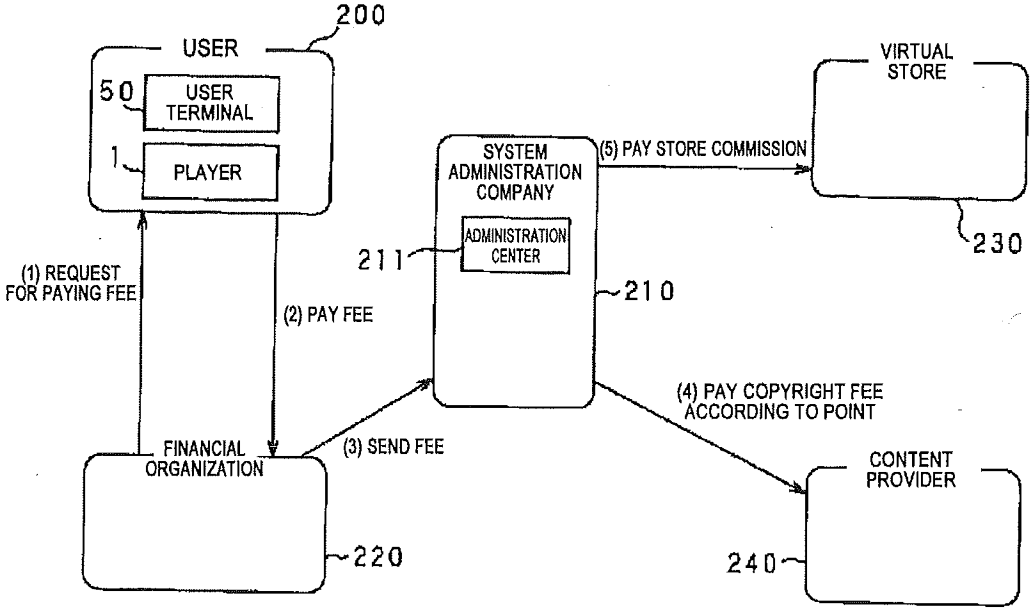

START
RECEIVE CONTENT SPECIFYING INFORMATION FROM USER TERMINAL
ST61
TRANSMIT CONTENT TO USER TERMINAL (TRANSMIT ENCRYPTED CONTENT AND CONTENT ID)
ST62
END

FLOWCHART FOR ADMINISTRATION CENTER WHEN PURCHASING CONTENT

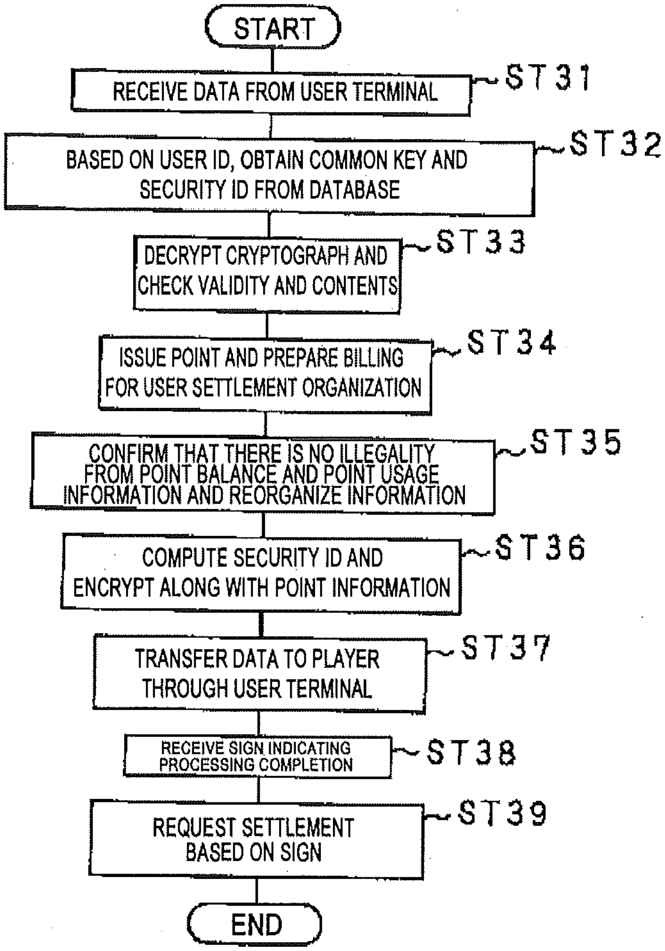
[FIG. 6]


200
USER
50
USER TERMINAL
1
PLAYER
(1) ENCRYPTED POINT REPLENISH REQUEST AND USED POINT COPYRIGHT HOLDER INFORMATION
(5) ENCRYPTED POINT REPLENISH INSTRUCTION
SYSTEM ADMINISTRATION COMPANY
ADMINISTRATION CENTER
211
210
(3) SETTLEMENT OK INSTRUCTION
(4) NOTIFY COPYRIGHT HOLDER OF POINT COUNT
FINANCIAL ORGANIZATION
(2) CONFIRM SETTLEMENT OF REPLENISHED POINTS
CONTENT PROVIDER
220
240

[FIG. 7]

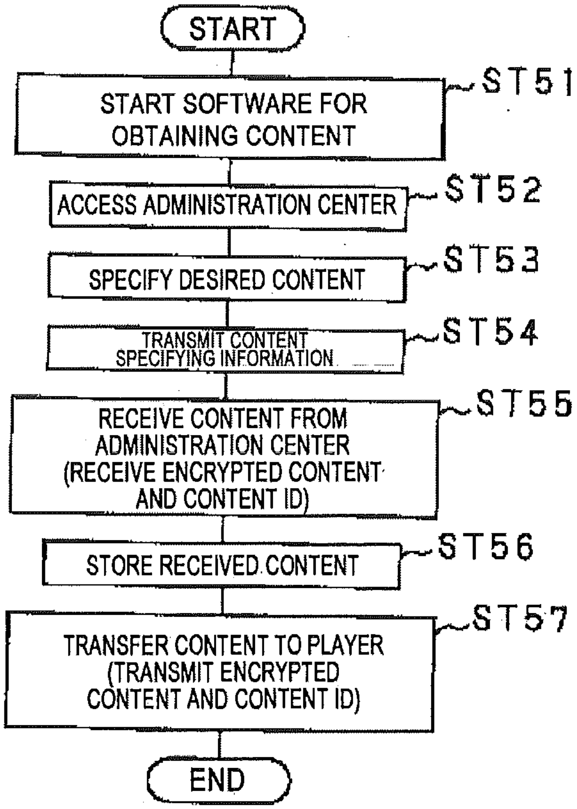

200
USER
50
USER TERMINAL
1
PLAYER
SYSTEM ADMINISTRATION COMPANY
211
ADMINISTRATION CENTER
210
VIRTUAL STORE
(5) PAY STORE COMMISSION
230
(1) REQUEST FOR PAYING FEE
(2) PAY FEE
(3) SEND FEE
(4) PAY COPYRIGHT FEE ACCORDING TO POINT
FINANCIAL ORGANIZATION
220
CONTENT PROVIDER
240

[FIG. 10]


START
RECEIVE DATA FROM USER TERMINAL
ST31
BASED ON USER ID, OBTAIN COMMON KEY AND SECURITY ID FROM DATABASE
ST32
DECRYPT CRYPTOGRAPH AND CHECK VALIDITY AND CONTENTS
ST33
ISSUE POINT AND PREPARE BILLING FOR USER SETTLEMENT ORGANIZATION
ST34
CONFIRM THAT THERE IS NO ILLEGALITY FROM POINT BALANCE AND POINT USAGE INFORMATION AND REORGANIZE INFORMATION
ST35
COMPUTE SECURITY ID AND ENCRYPT ALONG WITH POINT INFORMATION
ST36
TRANSFER DATA TO PLAYER THROUGH USER TERMINAL
ST37
RECEIVE SIGN INDICATING PROCESSING COMPLETION
ST38
REQUEST SETTLEMENT BASED ON SIGN
ST39
END

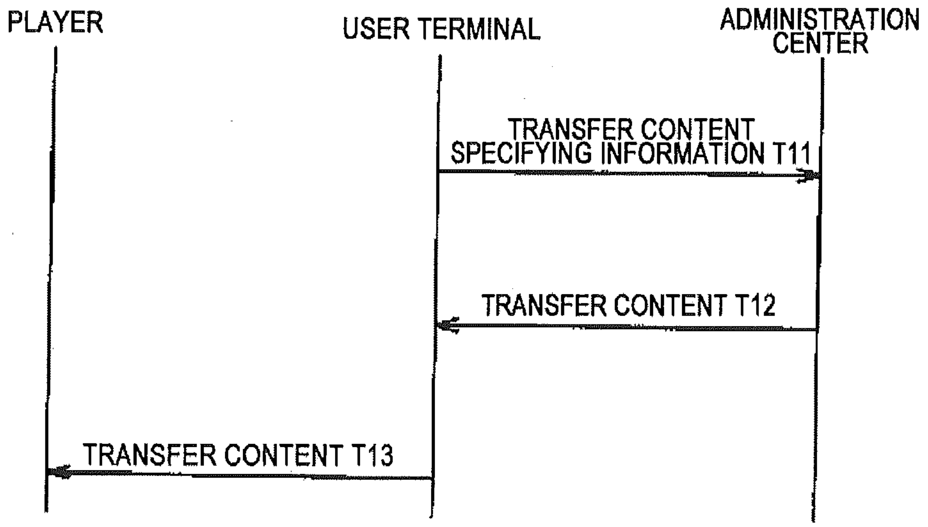
FLOWCHART FOR ADMINISTRATION CENTER WHEN PURCHASING POINTS

[FIG. 13]


START
START SOFTWARE FOR OBTAINING CONTENT
ST51
ACCESS ADMINISTRATION CENTER
ST52
SPECIFY DESIRED CONTENT
ST53
TRANSMIT CONTENT SPECIFYING INFORMATION
ST54
RECEIVE CONTENT FROM ADMINISTRATION CENTER (RECEIVE ENCRYPTED CONTENT AND CONTENT ID)
ST55
STORE RECEIVED CONTENT
ST56
TRANSFER CONTENT TO PLAYER (TRANSMIT ENCRYPTED CONTENT AND CONTENT ID)
ST57
END

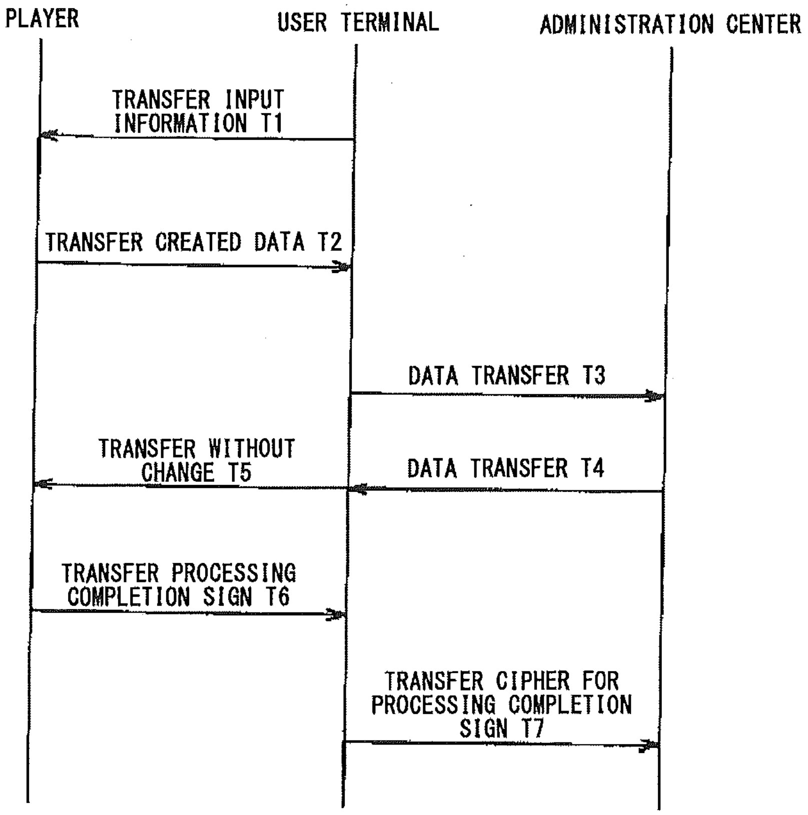
FLOWCHART FOR USER TERMINAL
WHEN OBTAINING CONTENT

[FIG. 15]



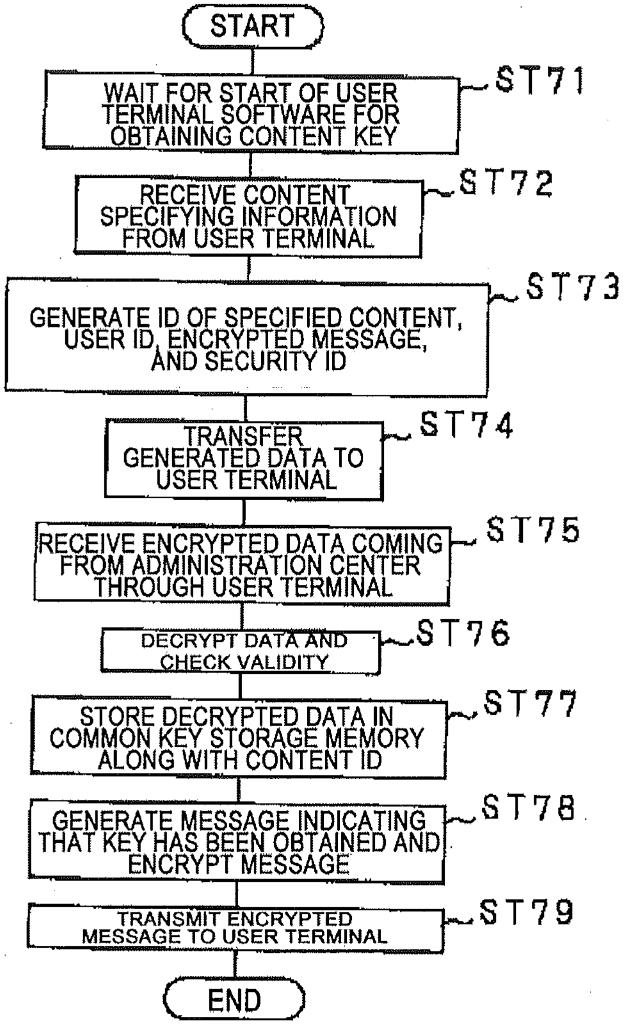
SEQUENCE WHEN OBTAINING CONTENT

[FIG. 11]


PLAYER
USER TERMINAL
ADMINISTRATION CENTER
TRANSFER INPUT INFORMATION T1
TRANSFER CREATED DATA T2
DATA TRANSFER T3
TRANSFER WITHOUT CHANGE T5
DATA TRANSFER T4
TRANSFER PROCESSING COMPLETION SIGN T6
TRANSFER CIPHER FOR PROCESSING COMPLETION SIGN T7

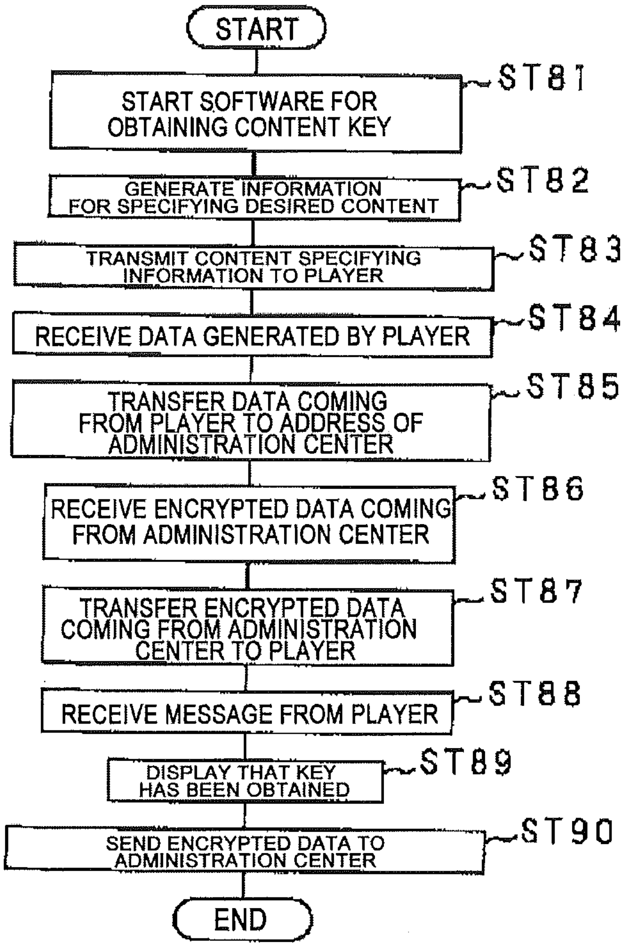
SEQUENCE WHEN PURCHASING POINTS

[FIG. 16]


START
WAIT FOR START OF USER TERMINAL SOFTWARE FOR OBTAINING CONTENT KEY
ST71
RECEIVE CONTENT SPECIFYING INFORMATION FROM USER TERMINAL
ST72
GENERATE ID OF SPECIFIED CONTENT, USER ID, ENCRYPTED MESSAGE, AND SECURITY ID
ST73
TRANSFER GENERATED DATA TO USER TERMINAL
ST74
RECEIVE ENCRYPTED DATA COMING FROM ADMINISTRATION CENTER THROUGH USER TERMINAL
ST75
DECRYPT DATA AND CHECK VALIDITY
ST76
STORE DECRYPTED DATA IN COMMON KEY STORAGE MEMORY ALONG WITH CONTENT ID
ST77
GENERATE MESSAGE INDICATING THAT KEY HAS BEEN OBTAINED AND ENCRYPT MESSAGE
ST78
TRANSMIT ENCRYPTED MESSAGE TO USER TERMINAL
ST79
END

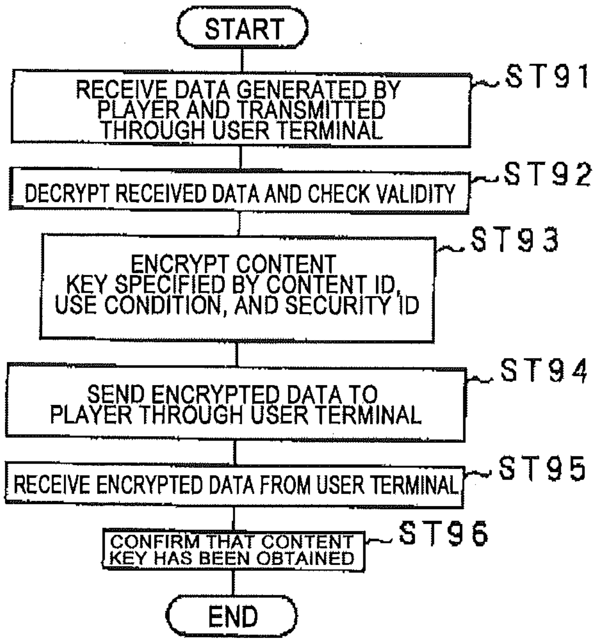
FLOWCHART FOR PLAYER WHEN OBTAINING CONTENT KEY

[FIG. 17]


START
START SOFTWARE FOR OBTAINING CONTENT KEY
ST81
GENERATE INFORMATION FOR SPECIFYING DESIRED CONTENT
ST82
TRANSMIT CONTENT SPECIFYING INFORMATION TO PLAYER
ST83
RECEIVE DATA GENERATED BY PLAYER
ST84
TRANSFER DATA COMING FROM PLAYER TO ADDRESS OF ADMINISTRATION CENTER
ST85
RECEIVE ENCRYPTED DATA COMING FROM ADMINISTRATION CENTER
ST86
TRANSFER ENCRYPTED DATA COMING FROM ADMINISTRATION CENTER TO PLAYER
ST87
RECEIVE MESSAGE FROM PLAYER
ST88
DISPLAY THAT KEY HAS BEEN OBTAINED
ST89
SEND ENCRYPTED DATA TO ADMINISTRATION CENTER
ST90
END

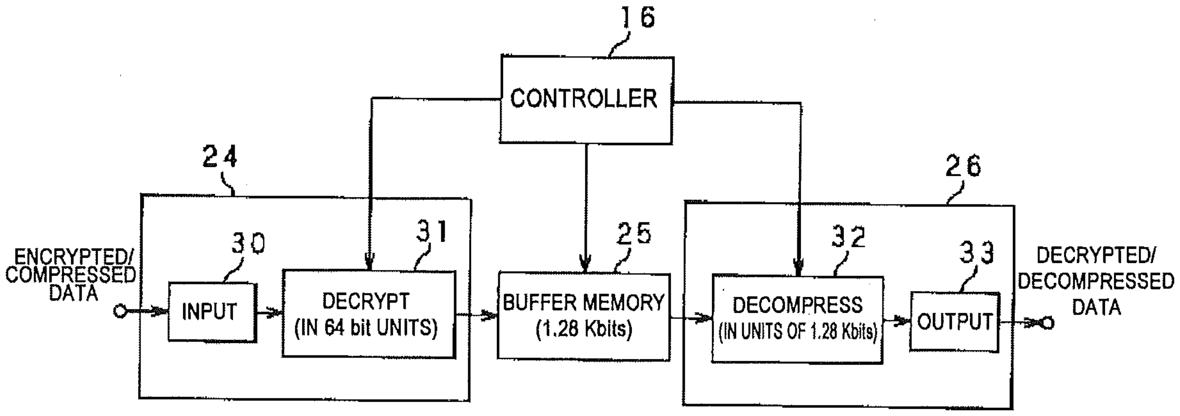
FLOWCHART FOR USER TERMINAL
WHEN OBTAINING CONTENT KEY/USE CONDITION

[FIG. 18]

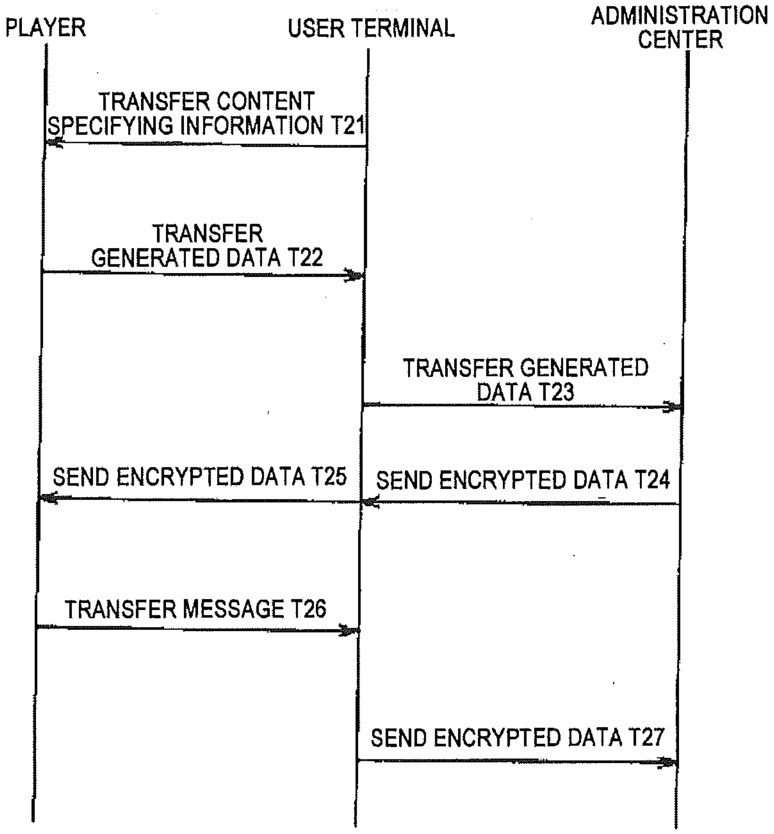

START
RECEIVE DATA GENERATED BY PLAYER AND TRANSMITTED THROUGH USER TERMINAL
ST91
DECRYPT RECEIVED DATA AND CHECK VALIDITY
ST92
ENCRYPT CONTENT KEY SPECIFIED BY CONTENT ID, USE CONDITION, AND SECURITY ID
ST93
SEND ENCRYPTED DATA TO PLAYER THROUGH USER TERMINAL
ST94
RECEIVE ENCRYPTED DATA FROM USER TERMINAL
ST95
CONFIRM THAT CONTENT KEY HAS BEEN OBTAINED
ST96
END

FLOWCHART FOR ADMINISTRATION CENTER WHEN OBTAINING CONTENT KEY/USE CONDITION

[FIG. 26]

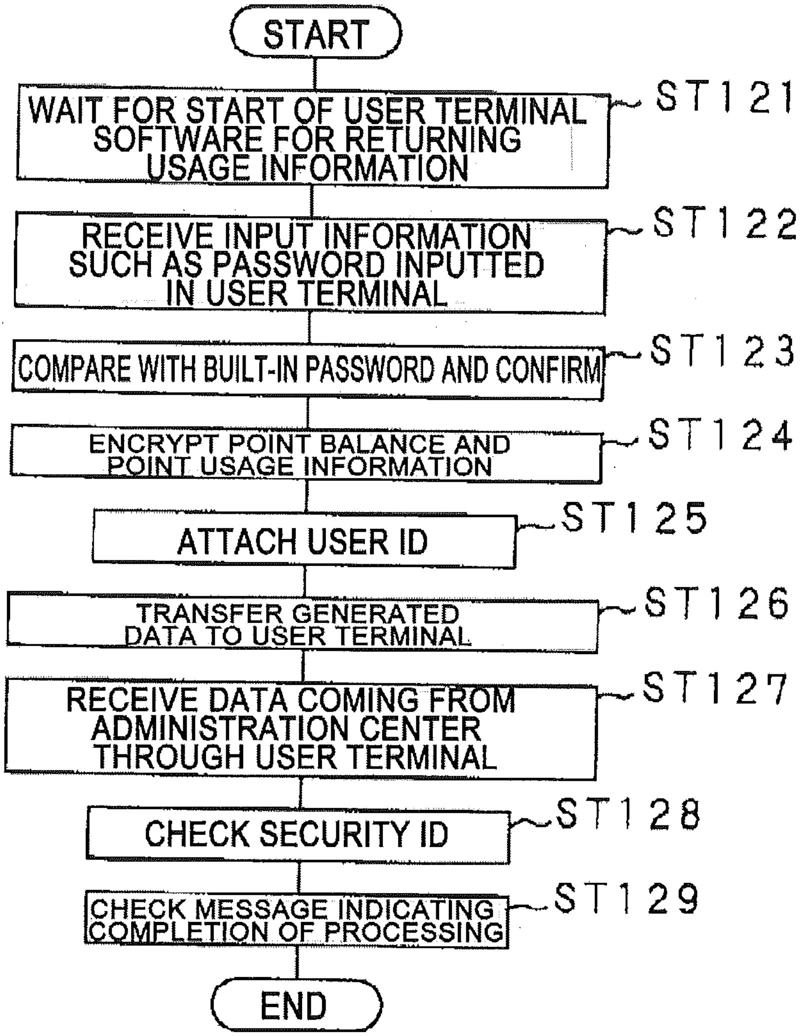

16
CONTROLLER
24
26
ENCRYPTED/ COMPRESSED DATA
30
INPUT
31
DECRYPT (IN 64 bit UNITS)
25
BUFFER MEMORY (1.28 Kbits)
32
DECOMPRESS (IN UNITS OF 1.28 Kbits)
33
OUTPUT
DECRYPTED/ DECOMPRESSED DATA

[FIG. 19]



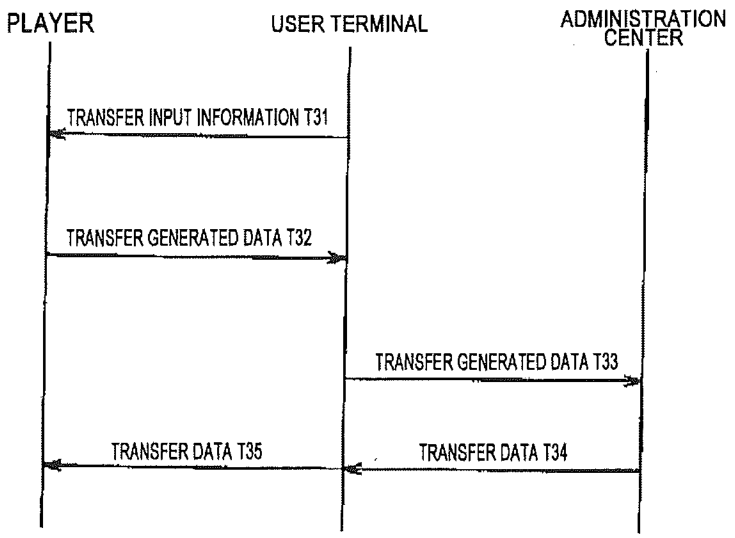
SEQUENCE WHEN OBTAINING CONTENT KEY/USE CONDITION

[FIG. 21]


START
WAIT FOR START OF USER TERMINAL SOFTWARE FOR RETURNING USAGE INFORMATION
ST121
RECEIVE INPUT INFORMATION SUCH AS PASSWORD INPUTTED IN USER TERMINAL
ST122
COMPARE WITH BUILT-IN PASSWORD AND CONFIRM
ST123
ENCRYPT POINT BALANCE AND POINT USAGE INFORMATION
ST124
ATTACH USER ID
ST125
TRANSFER GENERATED DATA TO USER TERMINAL
ST126
RECEIVE DATA COMING FROM ADMINISTRATION CENTER THROUGH USER TERMINAL
ST127
CHECK SECURITY ID
ST128
CHECK MESSAGE INDICATING COMPLETION OF PROCESSING
ST129
END

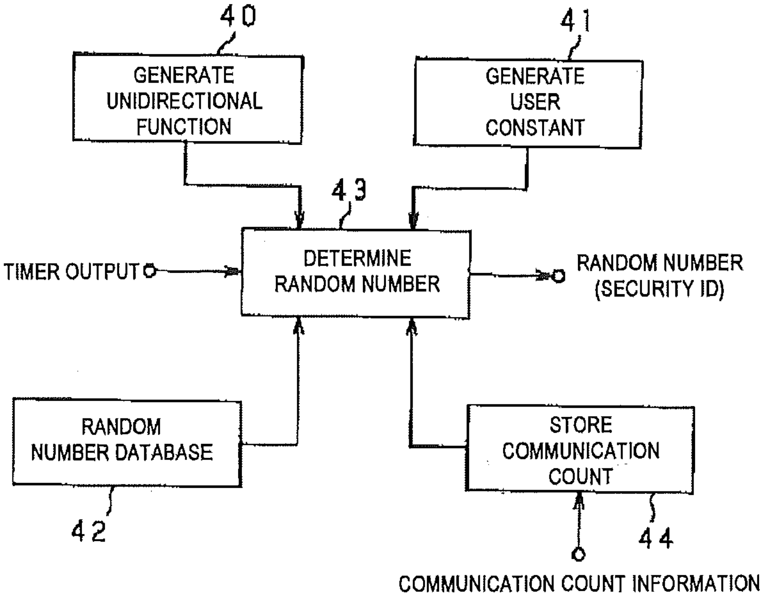
FLOWCHART FOR PLAYER WHEN RETURNING USAGE INFORMATION

[FIG. 24]

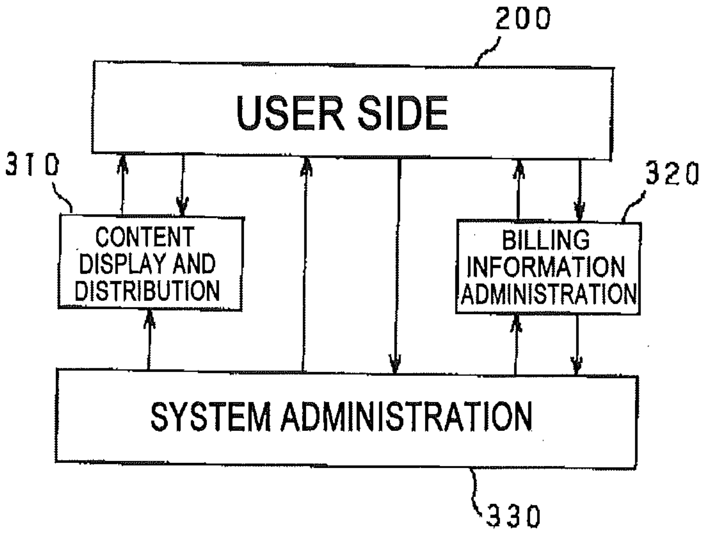

PLAYER
USER TERMINAL
ADMINISTRATION CENTER
TRANSFER INPUT INFORMATION T31
TRANSFER GENERATED DATA T32
TRANSFER GENERATED DATA T33
TRANSFER DATA T35
TRANSFER DATA T34

SEQUENCE WHEN RETURNING USAGE INFORMATION

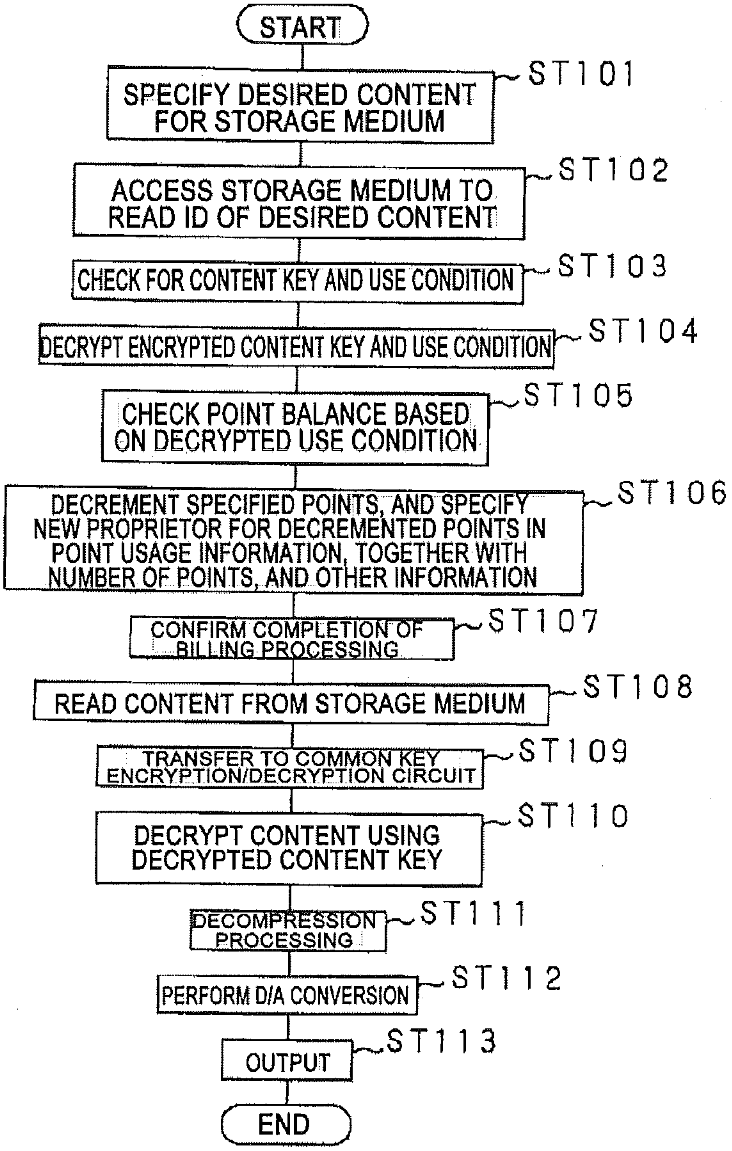
[FIG. 27]


40
GENERATE UNIDIRECTIONAL FUNCTION
41
GENERATE USER CONSTANT
43
TIMER OUTPUT
DETERMINE RANDOM NUMBER
RANDOM NUMBER (SECURITY ID)
RANDOM NUMBER DATABASE
42
STORE COMMUNICATION COUNT
44
COMMUNICATION COUNT INFORMATION

[FIG. 30]

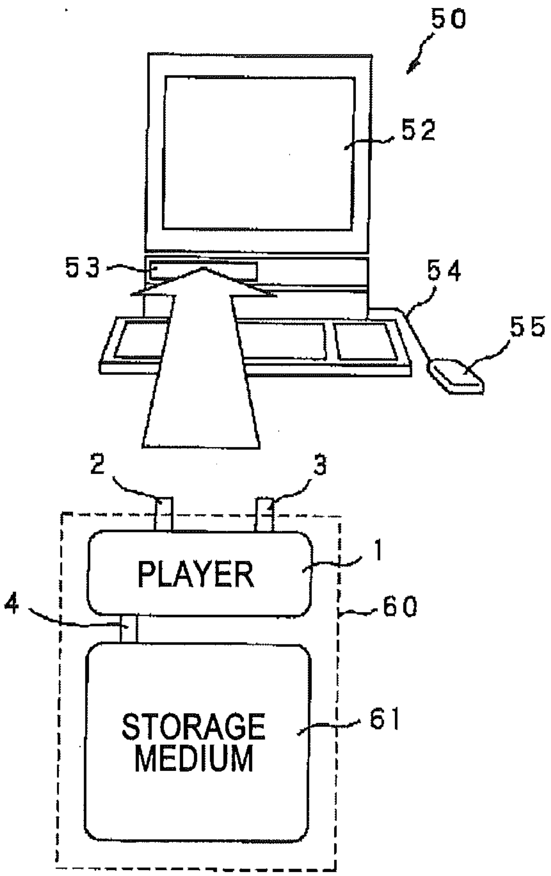

200
USER SIDE
310
CONTENT DISPLAY AND DISTRIBUTION
320
BILLING INFORMATION ADMINISTRATION
SYSTEM ADMINISTRATION
330

[FIG. 20]

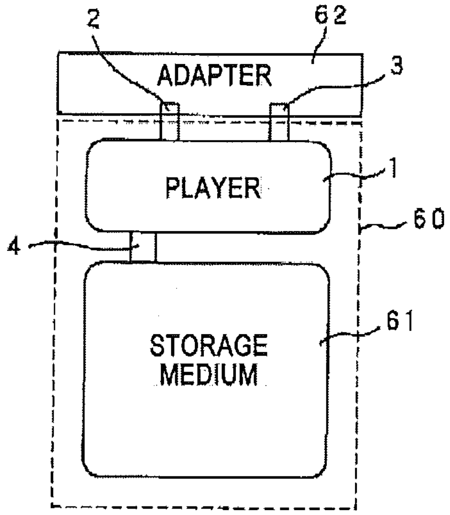

START
SPECIFY DESIRED CONTENT FOR STORAGE MEDIUM
ST101
ACCESS STORAGE MEDIUM TO READ ID OF DESIRED CONTENT
ST102
CHECK FOR CONTENT KEY AND USE CONDITION
ST103
DECRYPT ENCRYPTED CONTENT KEY AND USE CONDITION
ST104
CHECK POINT BALANCE BASED ON DECRYPTED USE CONDITION
ST105
DECREMENT SPECIFIED POINTS, AND SPECIFY NEW PROPRIETOR FOR DECREMENTED POINTS IN POINT USAGE INFORMATION, TOGETHER WITH NUMBER OF POINTS, AND OTHER INFORMATION
ST106
CONFIRM COMPLETION OF BILLING PROCESSING
ST107
READ CONTENT FROM STORAGE MEDIUM
ST108
TRANSFER TO COMMON KEY ENCRYPTION/DECRYPTION CIRCUIT
ST109
DECRYPT CONTENT USING DECRYPTED CONTENT KEY
ST110
DECOMPRESSION PROCESSING
ST111
PERFORM D/A CONVERSION
ST112
OUTPUT
ST113
END

FLOWCHART FOR PLAYER WHEN VIEWING CONTENT

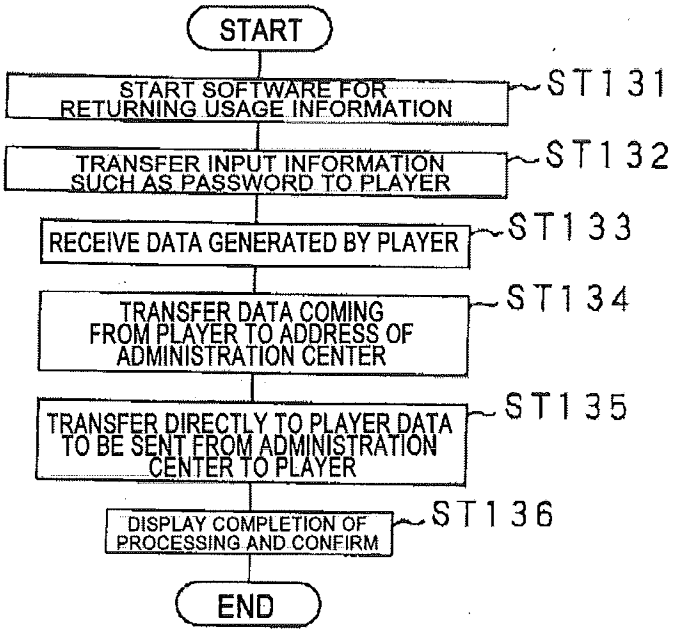
[FIG. 39]


50
52
53
54
55
2
3
PLAYER
1
4
60
STORAGE MEDIUM
61

[FIG. 40]

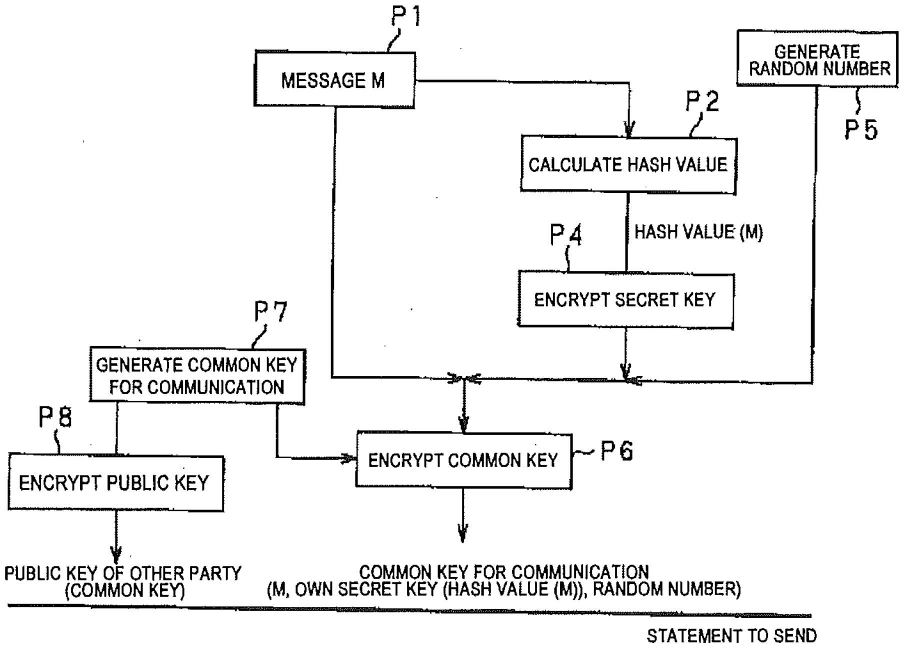

2
62
ADAPTER
3
PLAYER
1
4
60
STORAGE MEDIUM
61

[FIG. 22]

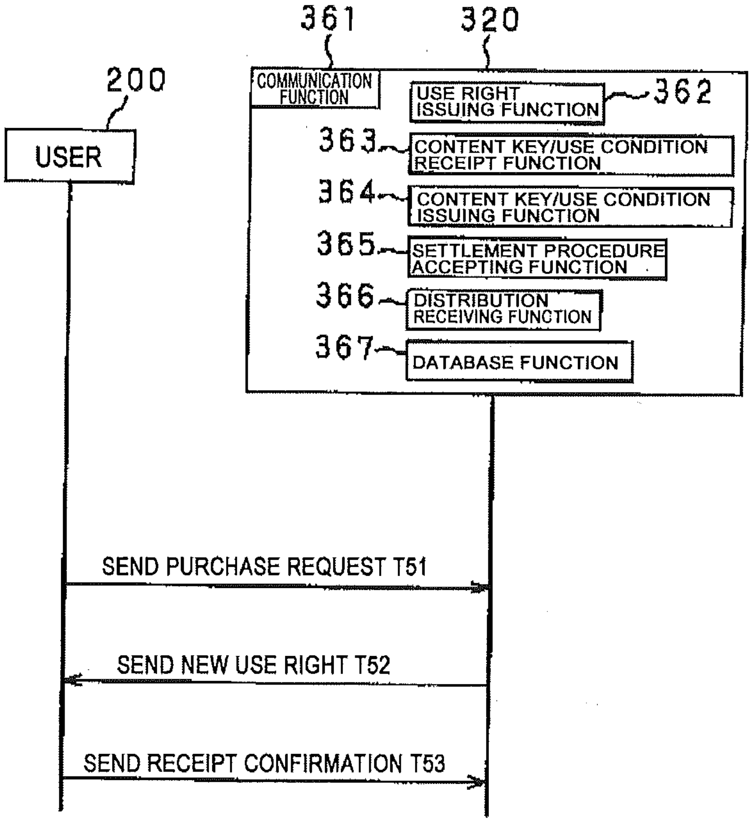

START
START SOFTWARE FOR RETURNING USAGE INFORMATION
ST131
TRANSFER INPUT INFORMATION SUCH AS PASSWORD TO PLAYER
ST132
RECEIVE DATA GENERATED BY PLAYER
ST133
TRANSFER DATA COMING FROM PLAYER TO ADDRESS OF ADMINISTRATION CENTER
ST134
TRANSFER DIRECTLY TO PLAYER DATA TO BE SENT FROM ADMINISTRATION CENTER TO PLAYER
ST135
DISPLAY COMPLETION OF PROCESSING AND CONFIRM
ST136
END

FLOWCHART FOR USER TERMINAL WHEN RETURNING USAGE INFORMATION

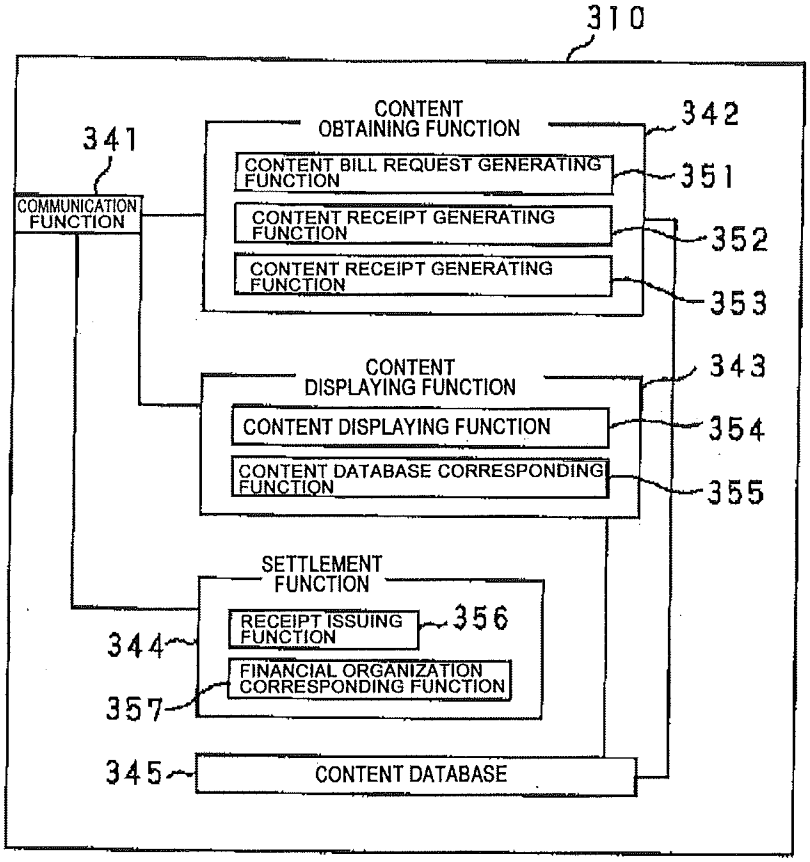
[FIG. 28]


P1
MESSAGE M
GENERATE RANDOM NUMBER
P5
P2
CALCULATE HASH VALUE
HASH VALUE (M)
P4
ENCRYPT SECRET KEY
P7
GENERATE COMMON KEY FOR COMMUNICATION
P8
ENCRYPT PUBLIC KEY
ENCRYPT COMMON KEY
P6
PUBLIC KEY OF OTHER PARTY (COMMON KEY)
COMMON KEY FOR COMMUNICATION (M, OWN SECRET KEY (HASH VALUE (M)), RANDOM NUMBER)
STATEMENT TO SEND

[FIG. 32]

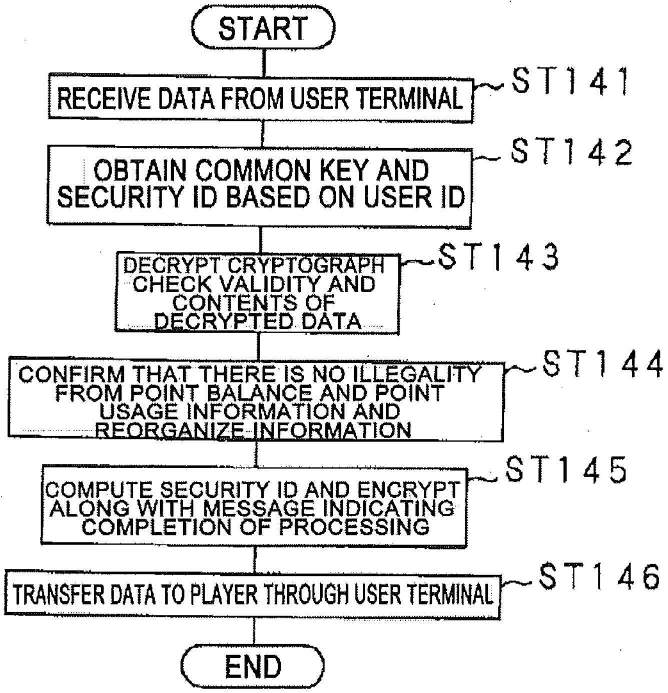

361
320
200
USER
COMMUNICATION FUNCTION
USE RIGHT ISSUING FUNCTION
362
363
CONTENT KEY/USE CONDITION RECEIPT FUNCTION
364
CONTENT KEY/USE CONDITION ISSUING FUNCTION
365
SETTLEMENT PROCEDURE ACCEPTING FUNCTION
366
DISTRIBUTION RECEIVING FUNCTION
367
DATABASE FUNCTION
SEND PURCHASE REQUEST T51
SEND NEW USE RIGHT T52
SEND RECEIPT CONFIRMATION T53

[FIG. 35]



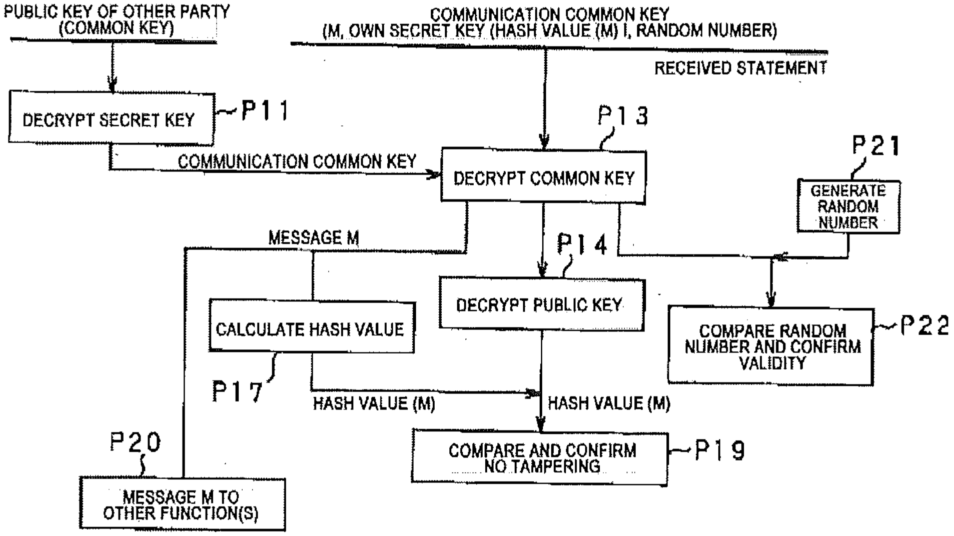
310
CONTENT OBTAINING FUNCTION
342
341
CONTENT BILL REQUEST GENERATING FUNCTION
351
COMMUNICATION FUNCTION
CONTENT RECEIPT GENERATING FUNCTION
352
CONTENT RECEIPT GENERATING FUNCTION
353
CONTENT DISPLAYING FUNCTION
343
CONTENT DISPLAYING FUNCTION
354
CONTENT DATABASE CORRESPONDING FUNCTION
355
SETTLEMENT FUNCTION
RECEIPT ISSUING FUNCTION
356
344
FINANCIAL ORGANIZATION CORRESPONDING FUNCTION
357
345
CONTENT DATABASE

[FIG. 23]

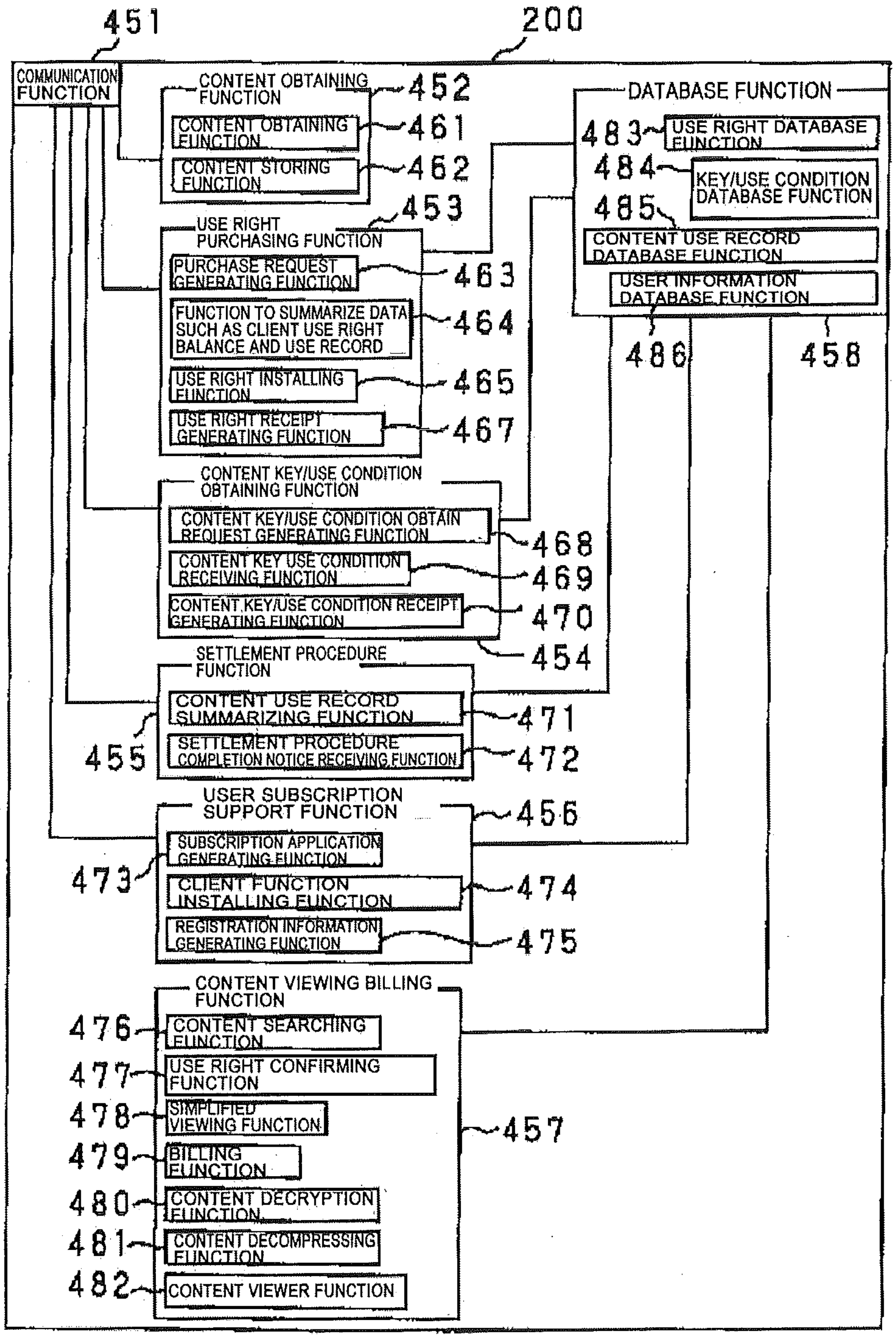

START
RECEIVE DATA FROM USER TERMINAL
ST141
OBTAIN COMMON KEY AND SECURITY ID BASED ON USER ID
ST142
DECRYPT CRYPTOGRAPH CHECK VALIDITY AND CONTENTS OF DECRYPTED DATA
ST143
CONFIRM THAT THERE IS NO ILLEGALITY FROM POINT BALANCE AND POINT USAGE INFORMATION AND REORGANIZE INFORMATION
ST144
COMPUTE SECURITY ID AND ENCRYPT ALONG WITH MESSAGE INDICATING COMPLETION OF PROCESSING
ST145
TRANSFER DATA TO PLAYER THROUGH USER TERMINAL
ST146
END

FLOWCHART FOR ADMINISTRATION CENTER WHEN RETURNING USAGE INFORMATION

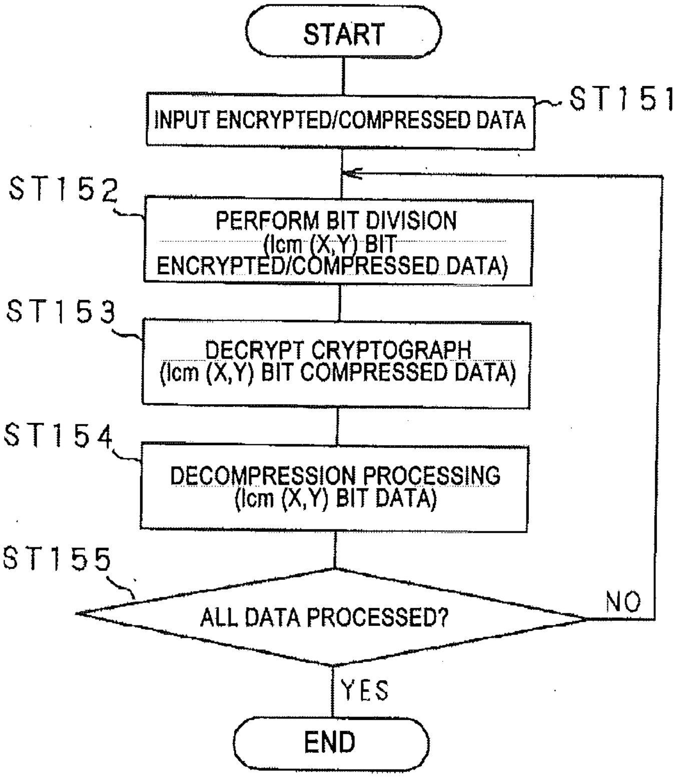
[FIG. 29]


PUBLIC KEY OF OTHER PARTY (COMMON KEY)
COMMUNICATION COMMON KEY (M, OWN SECRET KEY (HASH VALUE (M) I, RANDOM NUMBER)
RECEIVED STATEMENT
DECRYPT SECRET KEY
P11
COMMUNICATION COMMON KEY
P13
DECRYPT COMMON KEY
P21
GENERATE RANDOM NUMBER
MESSAGE M
P14
DECRYPT PUBLIC KEY
CALCULATE HASH VALUE
COMPARE RANDOM NUMBER AND CONFIRM VALIDITY
P22
P17
HASH VALUE (M)
HASH VALUE (M)
P20
COMPARE AND CONFIRM NO TAMPERING
P19
MESSAGE M TO OTHER FUNCTION(S)

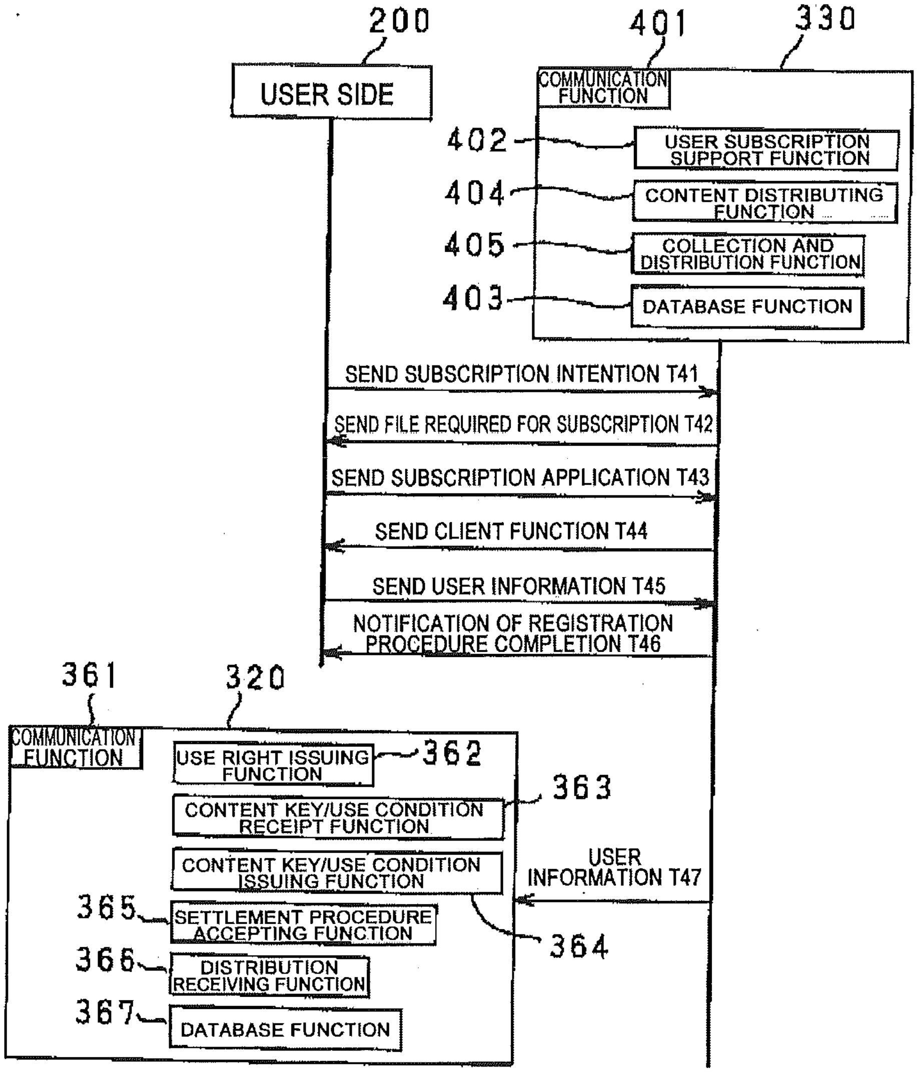
[FIG. 38]


451
200
COMMUNICATION FUNCTION
CONTENT OBTAINING FUNCTION
452
CONTENT OBTAINING FUNCTION
461
CONTENT STORING FUNCTION
462
453
USE RIGHT PURCHASING FUNCTION
PURCHASE REQUEST GENERATING FUNCTION
463
FUNCTION TO SUMMARIZE DATA SUCH AS CLIENT USE RIGHT BALANCE AND USE RECORD
464
USE RIGHT INSTALLING FUNCTION
465
USE RIGHT RECEIPT GENERATING FUNCTION
467
DATABASE FUNCTION
483
USE RIGHT DATABASE FUNCTION
484
KEY/USE CONDITION DATABASE FUNCTION
485
CONTENT USE RECORD DATABASE FUNCTION
USER INFORMATION DATABASE FUNCTION
486
458
CONTENT KEY/USE CONDITION OBTAINING FUNCTION
CONTENT KEY/USE CONDITION OBTAIN REQUEST GENERATING FUNCTION
468
CONTENT KEY USE CONDITION RECEIVING FUNCTION
469
CONTENT KEY/USE CONDITION RECEIPT GENERATING FUNCTION
470
454
SETTLEMENT PROCEDURE FUNCTION
CONTENT USE RECORD SUMMARIZING FUNCTION
471
455
SETTLEMENT PROCEDURE COMPLETION NOTICE RECEIVING FUNCTION
472
USER SUBSCRIPTION SUPPORT FUNCTION
456
SUBSCRIPTION APPLICATION GENERATING FUNCTION
473
CLIENT FUNCTION INSTALLING FUNCTION
474
REGISTRATION INFORMATION GENERATING FUNCTION
475
CONTENT VIEWING BILLING FUNCTION
476
CONTENT SEARCHING FUNCTION
477
USE RIGHT CONFIRMING FUNCTION
478
SIMPLIFIED VIEWING FUNCTION
457
479
BILLING FUNCTION
480
CONTENT DECRYPTION FUNCTION
481
CONTENT DECOMPRESSING FUNCTION
482
CONTENT VIEWER FUNCTION

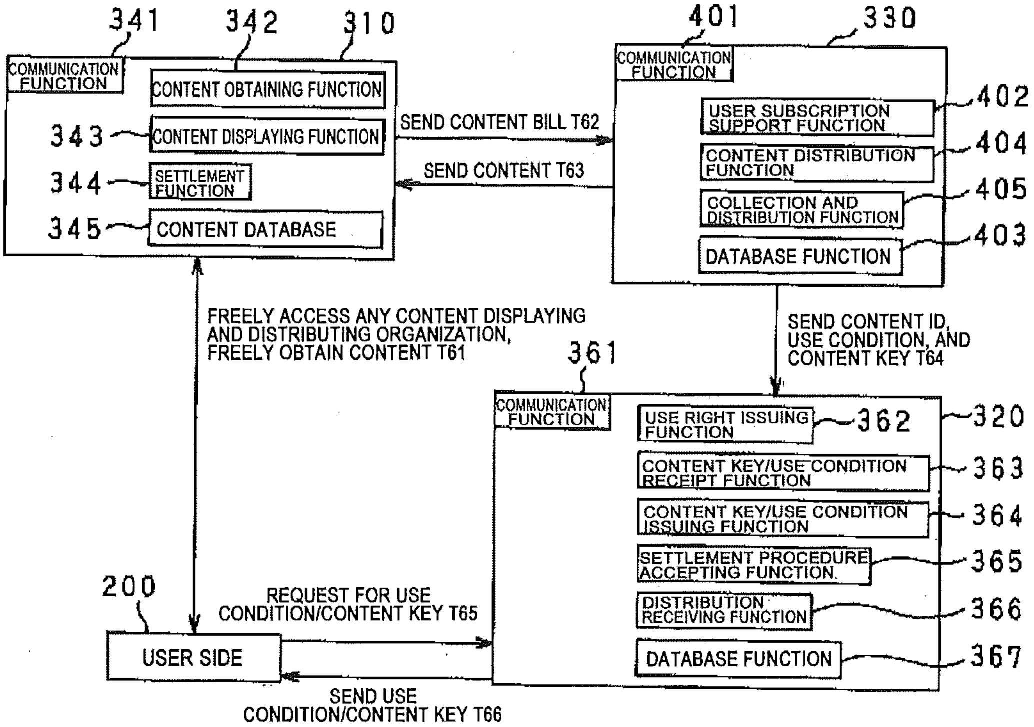
[FIG. 25]


START
INPUT ENCRYPTED/COMPRESSED DATA
ST151
ST152
PERFORM BIT DIVISION (lcm (X,Y) BIT ENCRYPTED/COMPRESSED DATA)
ST153
DECRYPT CRYPTOGRAPH (lcm (X,Y) BIT COMPRESSED DATA)
ST154
DECOMPRESSION PROCESSING (lcm (X,Y) BIT DATA)
ST155
ALL DATA PROCESSED?
NO
YES
END

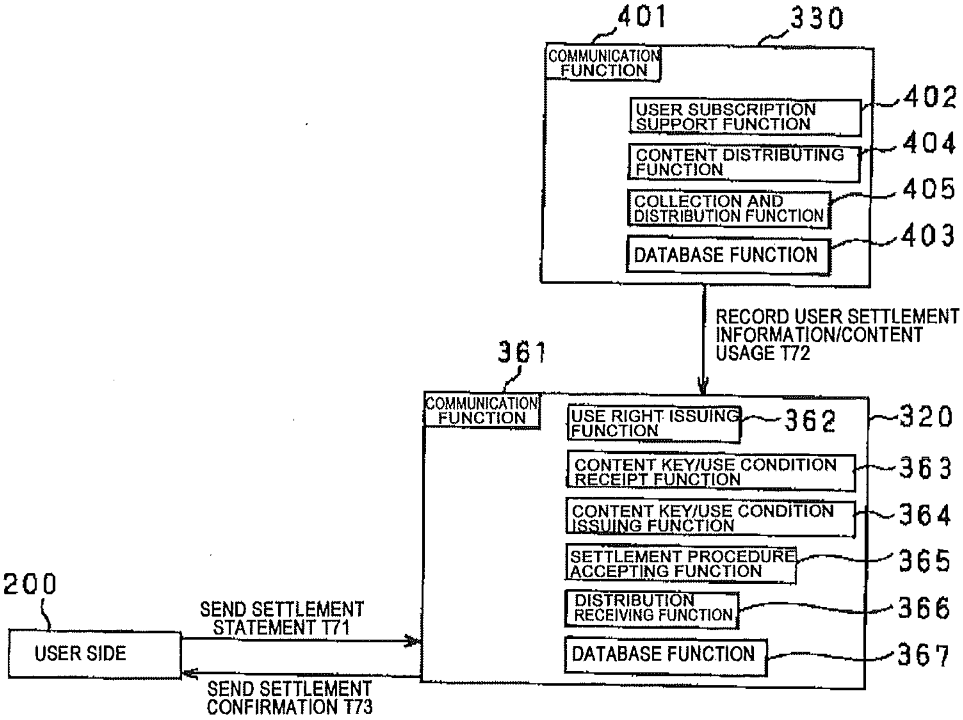
[FIG. 31]


200
USER SIDE
401
330
COMMUNICATION FUNCTION
402
USER SUBSCRIPTION SUPPORT FUNCTION
404
CONTENT DISTRIBUTING FUNCTION
405
COLLECTION AND DISTRIBUTION FUNCTION
403
DATABASE FUNCTION
SEND SUBSCRIPTION INTENTION T41
SEND FILE REQUIRED FOR SUBSCRIPTION T42
SEND SUBSCRIPTION APPLICATION T43
SEND CLIENT FUNCTION T44
SEND USER INFORMATION T45
NOTIFICATION OF REGISTRATION PROCEDURE COMPLETION T46
361
320
COMMUNICATION FUNCTION
USE RIGHT ISSUING FUNCTION
362
CONTENT KEY/USE CONDITION RECEIPT FUNCTION
363
CONTENT KEY/USE CONDITION ISSUING FUNCTION
USER INFORMATION T47
365
SETTLEMENT PROCEDURE ACCEPTING FUNCTION
364
366
DISTRIBUTION RECEIVING FUNCTION
367
DATABASE FUNCTION

[FIG. 33]

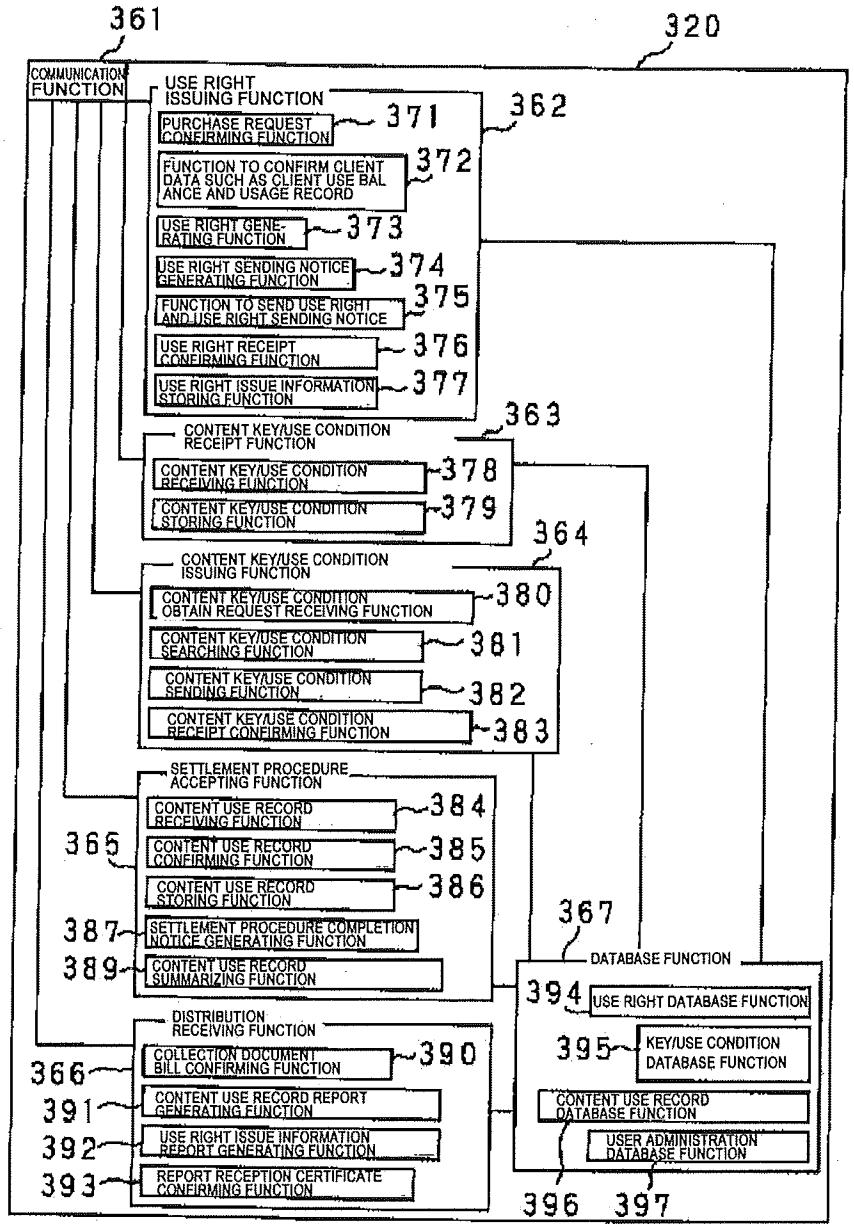

341
342
310
COMMUNICATION FUNCTION
CONTENT OBTAINING FUNCTION
343
CONTENT DISPLAYING FUNCTION
344
SETTLEMENT FUNCTION
345
CONTENT DATABASE
SEND CONTENT BILL T62
SEND CONTENT T63
401
330
COMMUNICATION FUNCTION
USER SUBSCRIPTION SUPPORT FUNCTION
402
CONTENT DISTRIBUTION FUNCTION
404
COLLECTION AND DISTRIBUTION FUNCTION
405
DATABASE FUNCTION
403
FREELY ACCESS ANY CONTENT DISPLAYING AND DISTRIBUTING ORGANIZATION, FREELY OBTAIN CONTENT T61
SEND CONTENT ID, USE CONDITION, AND CONTENT KEY T64
361
COMMUNICATION FUNCTION
USE RIGHT ISSUING FUNCTION
362
320
CONTENT KEY/USE CONDITION RECEIPT FUNCTION
363
CONTENT KEY/USE CONDITION ISSUING FUNCTION
364
SETTLEMENT PROCEDURE ACCEPTING FUNCTION
365
DISTRIBUTION RECEIVING FUNCTION
366
DATABASE FUNCTION
367
200
USER SIDE
REQUEST FOR USE CONDITION/CONTENT KEY T65
SEND USE CONDITION/CONTENT KEY T66

[FIG. 34]

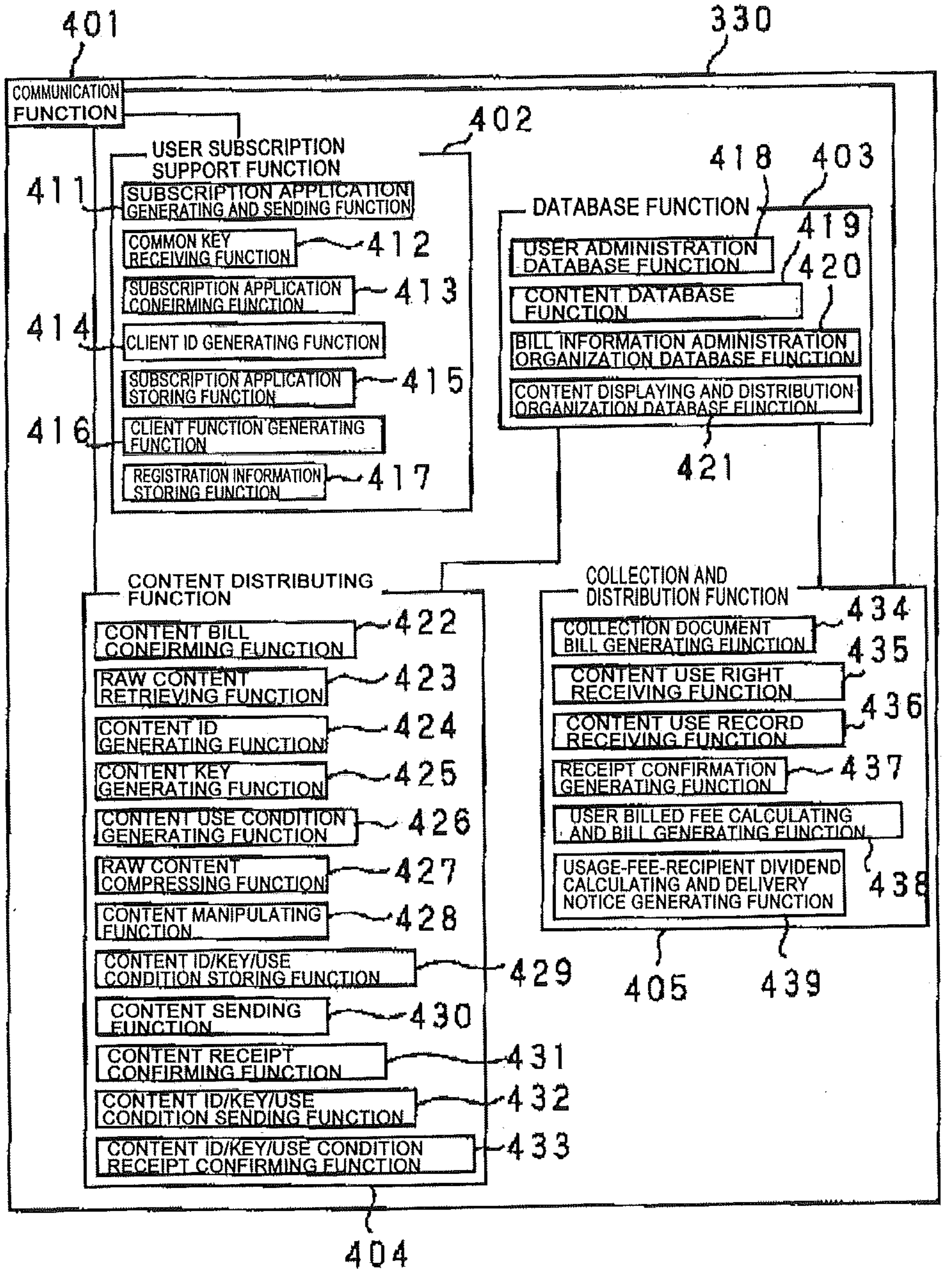


401
330
COMMUNICATION FUNCTION
USER SUBSCRIPTION SUPPORT FUNCTION
402
CONTENT DISTRIBUTING FUNCTION
404
COLLECTION AND DISTRIBUTION FUNCTION
405
DATABASE FUNCTION
403
RECORD USER SETTLEMENT INFORMATION/CONTENT USAGE T72
361
COMMUNICATION FUNCTION
USE RIGHT ISSUING FUNCTION
362
320
CONTENT KEY/USE CONDITION RECEIPT FUNCTION
363
CONTENT KEY/USE CONDITION ISSUING FUNCTION
364
SETTLEMENT PROCEDURE ACCEPTING FUNCTION
365
DISTRIBUTION RECEIVING FUNCTION
366
DATABASE FUNCTION
367
200
USER SIDE
SEND SETTLEMENT STATEMENT T71
SEND SETTLEMENT CONFIRMATION T73

[FIG. 36]


361
320
COMMUNICATION FUNCTION
USE RIGHT ISSUING FUNCTION
362
PURCHASE REQUEST CONFIRMING FUNCTION
371
FUNCTION TO CONFIRM CLIENT DATA SUCH AS CLIENT USE BAL ANCE AND USAGE RECORD
372
USE RIGHT GENE-RATING FUNCTION
373
USE RIGHT SENDING NOTICE GENERATING FUNCTION
374
FUNCTION TO SEND USE RIGHT AND USE RIGHT SENDING NOTICE
375
USE RIGHT RECEIPT CONFIRMING FUNCTION
376
USE RIGHT ISSUE INFORMATION STORING FUNCTION
377
CONTENT KEY/USE CONDITION RECEIPT FUNCTION
363
CONTENT KEY/USE CONDITION RECEIVING FUNCTION
378
CONTENT KEY/USE CONDITION STORING FUNCTION
379
CONTENT KEY/USE CONDITION ISSUING FUNCTION
364
CONTENT KEY/USE CONDITION OBTAIN REQUEST RECEIVING FUNCTION
380
CONTENT KEY/USE CONDITION SEARCHING FUNCTION
381
CONTENT KEY/USE CONDITION SENDING FUNCTION
382
CONTENT KEY/USE CONDITION RECEIPT CONFIRMING FUNCTION
383
SETTLEMENT PROCEDURE ACCEPTING FUNCTION
365
CONTENT USE RECORD RECEIVING FUNCTION
384
CONTENT USE RECORD CONFIRMING FUNCTION
385
CONTENT USE RECORD STORING FUNCTION
386
387
SETTLEMENT PROCEDURE COMPLETION NOTICE GENERATING FUNCTION
389
CONTENT USE RECORD SUMMARIZING FUNCTION
367
DATABASE FUNCTION
394
USE RIGHT DATABASE FUNCTION
395
KEY/USE CONDITION DATABASE FUNCTION
CONTENT USE RECORD DATABASE FUNCTION
USER ADMINISTRATION DATABASE FUNCTION
396
397
DISTRIBUTION RECEIVING FUNCTION
366
COLLECTION DOCUMENT BILL CONFIRMING FUNCTION
390
391
CONTENT USE RECORD REPORT GENERATING FUNCTION
392
USE RIGHT ISSUE INFORMATION REPORT GENERATING FUNCTION
393
REPORT RECEPTION CERTIFICATE CONFIRMING FUNCTION

[FIG. 37]


401
330
COMMUNICATION FUNCTION
402
USER SUBSCRIPTION SUPPORT FUNCTION
411
SUBSCRIPTION APPLICATION GENERATING AND SENDING FUNCTION
COMMON KEY RECEIVING FUNCTION
412
SUBSCRIPTION APPLICATION CONFIRMING FUNCTION
413
414
CLIENT ID GENERATING FUNCTION
SUBSCRIPTION APPLICATION STORING FUNCTION
415
416
CLIENT FUNCTION GENERATING FUNCTION
REGISTRATION INFORMATION STORING FUNCTION
417
418
403
DATABASE FUNCTION
419
USER ADMINISTRATION DATABASE FUNCTION
420
CONTENT DATABASE FUNCTION
BILL INFORMATION ADMINISTRATION ORGANIZATION DATABASE FUNCTION
CONTENT DISPLAYING AND DISTRIBUTION ORGANIZATION DATABASE FUNCTION
421
CONTENT DISTRIBUTING FUNCTION
CONTENT BILL CONFIRMING FUNCTION
422
RAW CONTENT RETRIEVING FUNCTION
423
CONTENT ID GENERATING FUNCTION
424
CONTENT KEY GENERATING FUNCTION
425
CONTENT USE CONDITION GENERATING FUNCTION
426
RAW CONTENT COMPRESSING FUNCTION
427
CONTENT MANIPULATING FUNCTION
428
CONTENT ID/KEY/USE CONDITION STORING FUNCTION
429
CONTENT SENDING FUNCTION
430
CONTENT RECEIPT CONFIRMING FUNCTION
431
CONTENT ID/KEY/USE CONDITION SENDING FUNCTION
432
CONTENT ID/KEY/USE CONDITION RECEIPT CONFIRMING FUNCTION
433
404
COLLECTION AND DISTRIBUTION FUNCTION
COLLECTION DOCUMENT BILL GENERATING FUNCTION
434
CONTENT USE RIGHT RECEIVING FUNCTION
435
CONTENT USE RECORD RECEIVING FUNCTION
436
RECEIPT CONFIRMATION GENERATING FUNCTION
437
USER BILLED FEE CALCULATING AND BILL GENERATING FUNCTION
438
USAGE-FEE-RECIPIENT DIVIDEND CALCULATING AND DELIVERY NOTICE GENERATING FUNCTION
405
439

(19)日本国特許庁（ＪＰ）　　(12) 公開特許公報（Ａ）　　(11)特許出願公開番号

特開平11－164058

(43)公開日　平成11年(1999) 6 月18日

(51)Int.Cl.^{6}	識別記号	ＦＩ	
Ｈ０４Ｍ　11/08		Ｈ０４Ｍ　11/08	
Ｈ０４Ｂ　7/24		Ｈ０４Ｂ　7/24	Ｃ
Ｈ０４Ｍ　3/42		Ｈ０４Ｍ　3/42	Ｚ

審査請求　有　　請求項の数４　ＯＬ　（全 ５ 頁）

(21)出願番号　　特願平9－324660

(22)出願日　　平成９年(1997)11月26日

(71)出願人　000233491
日立電子サービス株式会社
神奈川県横浜市戸塚区品濃町504番地２

(72)発明者　佐藤　登
神奈川県横浜市戸塚区品濃町504番地２
日立電子サービス株式会社内

(72)発明者　山岸　令和
神奈川県横浜市戸塚区品濃町504番地２
日立電子サービス株式会社内

(72)発明者　武貞　睦治
神奈川県横浜市戸塚区品濃町504番地２
日立電子サービス株式会社内

(74)代理人　弁理士　沼形　義彰　（外３名）

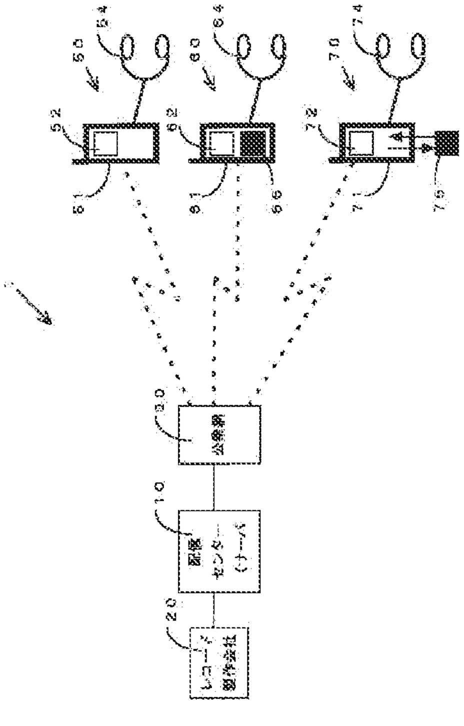
最終頁に続く

(54)【発明の名称】　携帯型音楽選曲視聴システム

(57)【要約】

【課題】　携帯電話機を利用してユーザが選曲する音楽のソフトを提供するシステムを提供する。

【解決手段】　サーバ機能を有する配信センター１０は、レコード製作会社２０から音楽ソフトの提供を受ける。配信センター１０に対して公衆回線網３０を介して接続される携帯用音楽選曲視聴機５０は、ボタン入力部を有する本体５１と、本体５１に設けられるディスプレイ５２と、音楽用レシーバ５４を有し、ユーザが選曲して音楽ソフトを配信センターに要求し、受信した音楽ソフトの音声をレシーバ５４に出力するとともに、歌詞等をディスプレイ５２に出力する。



【特許請求の範囲】

【請求項１】　音楽ソフトを製作するレコード会社と、レコード会社から音楽の提供を受けるサーバー機能を有する配信センターと、公衆回線網を介して配信センターに接続される携帯型音楽選曲視聴機とを備え、

携帯型音楽選曲視聴機は、入力された選曲情報を公衆回線網を介して配信センターに伝達し、配信センターから送られてくる音楽ソフトを音声と文字情報として出力する手段を備える携帯型音楽選曲視聴システム。

【請求項２】　携帯型音楽選曲視聴機は、電源部と、総合制御部と、電話番号登録用記憶部と、ボタン入力部と、ディスプレイ表示部と、送話・受話制御部と、送話器および受話器と、電波送受信制御部と、アンテナと、音楽用制御部と、音楽用増幅部と、音楽用レシーバを備える請求項１記載の携帯型音楽選曲視聴システム。

【請求項３】　携帯型音楽選曲視聴機は、受信した音楽ソフトを記憶する音楽用記憶部を備える請求項２記載の携帯型音楽選曲視聴システム。

【請求項４】　携帯型音楽選曲視聴機は、音楽ソフトを記憶する着脱自在の音楽用記憶媒体を備える請求項２および請求項３記載の携帯型音楽選曲視聴システム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、携帯型の電話機を用いた音楽の配信システムに関する。

【０００２】

【従来の技術】例えば、携帯型のラジオやテレビを利用して、地上局やサテライトからの放送電波を受信して音楽ソフトを楽しむことができる。この放送電波の受信は、放送局からの一方通行のサービスであって、ユーザが選曲することはできない。また、通信カラオケシステム等にあっては、有線回線を利用してユーザが選曲した音楽をセンターに要求し、サービスを受けることができる。

【０００３】

【発明が解決しようとする課題】携帯型の電話システムの普及に伴い、ユーザに対して電話サービスの他にも各種のサービスを提供することが可能となっている。本発明は、無線の公衆回線網を利用する音楽選曲視聴システムを提供するものである。

【０００４】

【課題を解決するための手段】本発明の音楽選曲視聴システムは、基本的な手段として、音楽ソフトを製作するレコード会社と、レコード会社から音楽の提供を受けるサーバー機能を有する配信センターと、公衆回線網を介して配信センターに接続される携帯型音楽選曲視聴機とを備える。そして、携帯型音楽選曲視聴機は、入力された選曲情報を公衆回線網を介して配信センターに伝達し、配信センターから送られてくる音楽ソフトを音声と文字情報として出力する手段を備えるものである。また、携帯型音楽選曲視聴機は、具体的な手段として、電源部と、総合制御部と、電話番号登録用記憶部と、ボタン入力部と、ディスプレイ表示部と、送話・受話制御部と、送話器および受話器と、電波送受信制御部と、アンテナと、音楽用制御部と、音楽用増幅部と、音楽用レシーバを備える。さらに、携帯型音楽選曲視聴機は、受信した音楽ソフトを記憶する音楽用記憶部を備えるか、または、音楽ソフトを記憶する着脱自在の音楽用記憶媒体を備えることができる。

【０００５】

【発明の実施の形態】図１は、本発明の携帯型音楽選曲視聴システムの全体構成図である。全体を符号１で示すシステムは、サーバーである配信センター１０を有し、配信センター１０はレコード製作会社２０から音楽コンテンツの供給を受ける。この配信センター１０に対して、公衆回線網３０を介して携帯型音楽選曲視聴機５０，６０，７０が接続される。

【０００６】携帯型音楽選曲視聴機５０は、例えば携帯電話機と同様の構造を有し、本体５１に必要なプッシュボタン等と、ディスプレイ５２を装備する。本体５１に対してはレシーバ５４が接続される。携帯型音楽選曲視聴機５０を有するユーザは、本体５１上のプッシュボタン等を操作して公衆回線網３０を経由して配信センター１０を呼び出し、希望する音楽ソフトを公衆回線網３０を介して受信する。受信した音楽ソフトは、携帯型音楽選曲視聴機５０の本体５１内に装備されたアンプで増幅され、レシーバ５４に出力される。

【０００７】レシーバ５４を装着したユーザは、音楽を楽しむとともに、必要に応じてディスプレイ５２に歌詞を表示して、カラオケとしても楽しむことができる。この携帯型音楽選曲視聴機５０は、メモリ機能等を備えない簡素化されたものであって、回線接続中にのみ音楽の供給を受けることができる。

【０００８】携帯型音楽選曲視聴機６０は、本体６１内に記憶装置６６を内蔵するモデルを示す。このモデルの携帯型音楽選曲視聴機６０にあっては、本体６１のプッシュボタン等を操作して配信センター１０を呼び出して、供給を受けた音楽ソフトは、レシーバ６４とディスプレイ６２に出力されるとともに、記憶装置６６により記憶される。したがって、ユーザは公衆回線３０の接続を遮断した後にも、記憶装置６６内の音楽ソフトを再生させて楽しむことができる。

【０００９】携帯型音楽選曲視聴機７０は、本体７１に対して着脱可能な記憶装置７６を備える。この記憶装置７６は、例えば磁気カード、磁気テープ、ＣＤ、ＤＶＤ、ＩＣカードのようなメモリカードである。ユーザは、本体７１のプッシュボタン等を操作して、携帯型音楽選曲視聴機７０の記憶装置（媒体）７６に音楽ソフトをダウンロードすると、この音楽ソフトを携帯型音楽選曲視聴機７０のディスプレイ７２やレシーバ７４で楽し

むことできるとともに、この記憶装置（媒体）を抜き出して、他のオーディオユニットに挿入し、より高品質な再生音楽を楽しむことができる。また、他のオーディオユニットで記憶装置７６内に音楽ソフトを記憶させ、この記憶装置７６を、この携帯型音楽選曲視聴機７０に挿入して音楽を楽しむこともできる。

【００１０】図2は、携帯電話の機能を有する本発明の携帯型音楽選曲視聴機の構成図である。全体を符号１００で示す携帯型音楽選曲視聴機は、電源部１３０に接続される総合制御部１１０を有し、総合制御部１１０は電話番号登録用記憶部１２０が接続される。ユーザが操作するボタン入力部１８２を有するボタン入力制御部１８０は、総合制御部１１０に信号を送り、総合制御部１１０は、ディスプレイ制御部１６０を介してディスプレイ表示部１６２に操作内容を表示するとともに、電波送受信制御部１４０、アンテナ１５０を介して公衆回線網にアクセルする。ユーザが相手の電話を呼び出すのであれば、交換機は相手電話を呼び出し、送話・受話制御部１７０に接続される送話器１７４と受話器１７２を用いてユーザは相手と通話することができる。

【００１１】音楽用制御部２００は、総合制御部１１０、電源部１３０、電波送受信制御部１４０、ボタン入力制御部１８０、ディスプレイし御部１６０に接続される。ユーザは、ボタン入力部１８２を操作して配信センターを呼び出す指令を出力すると、その内容はディスプレイ表示部１６２に表示されるとともに、電波送受信制御部１４０は、アンテナ１５０を介して公衆回線網を経由して配信センターにアクセスする。

【００１２】配信センターにアクセスができると、ユーザは選曲を指令し、配信センターは選曲された音楽ソフトを送り返す。この音楽ソフトを受信した音楽用制御部２００は、音楽用増幅部で信号を増幅し、レシーバジャック２２０に差し込まれる音楽用レシーバ２３０に音声を出力する。この音声出力は、受話器１７２へも出力することができる。歌詞等の情報は、ディスプレイ表示部１６２に表示される。

【００１３】音楽用制御部２００に接続される音楽用記憶部２４０は、音楽ソフトを記憶する。磁気カード、磁気テープ、ＣＤ、ＤＶＤ、ＩＣカードのようなメモリカードのような音楽用記憶媒体２５０は、音楽ソフトを記憶するとともに、この記憶媒体２５０を取り出して、他のオーディオユニット等で使用することもできる。

【００１４】図3は、本発明のシステムによる処理のフロー図である。ステップＳ１０では、レコード製作会社２０から配信センター１０へ、曲・歌詞・画像の登録（変更・削除）を行う。ステップＳ１１では、ユーザは携帯型音楽選曲視聴機のディスプレイとプッシュボタンで、曲名・歌手名・ジャンル・曲名No.・作曲家名等を選択できる。ステップＳ１２は、選曲した曲と歌詞・画像を配信センターからユーザに回線を通して送信する。

【００１５】ステップＳ１３は、ユーザのレシーバから音声、ディスプレイに歌詞と画像を表示する。記憶装置付きの場合は記録し、通話終了後も再生可能とする。ステップＳ１４は、配信センターから送信が終了したら課金処理を行う。課金処理についてはＮＴＴのＱ２方式に準拠して行う。ステップＳ１５では、記憶装置から再生中に電話着信があったら、割り込みの通知または表示を行なう。

【００１６】

【発明の効果】本発明は以上のように、携帯電話機を利用してユーザが選曲した音楽ソフトを楽しむことができるので、公衆回線網の利用も拡大され、サービスも向上するものである。

【図面の簡単な説明】

【図１】本発明の携帯型音楽選曲視聴システムの構成図。

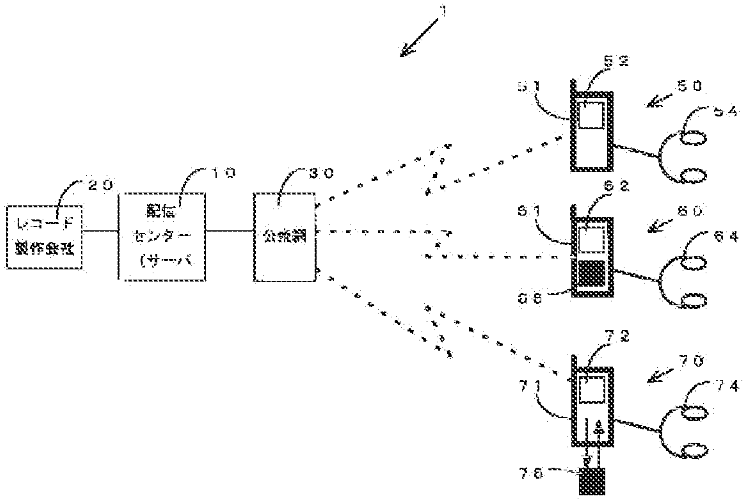
【図２】本発明の携帯用音楽選曲視聴機の構成図。

【図３】本発明の携帯型音楽選曲視聴システムのフロー図。

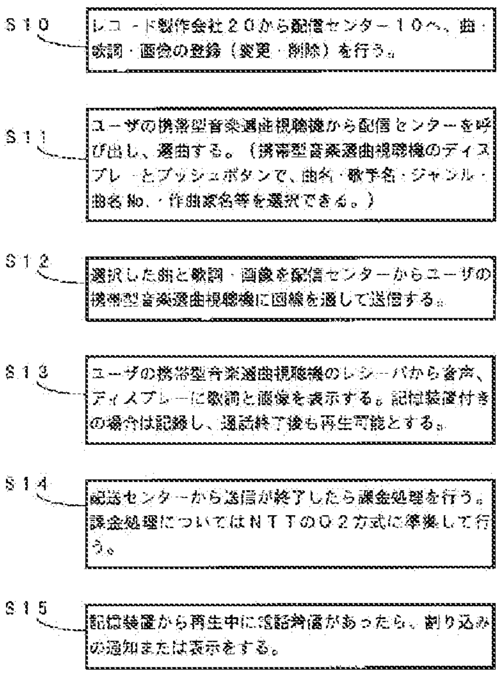
【符号の説明】

１０	配信センター
２０	レコード製作会社
３０	公衆回線網
５０，６０，７０	携帯用音楽選曲視聴機

【図１】



【図３】



Ｓ１０　レコード製作会社２０から配信センター１０へ、曲・歌詞・画像の登録（変更・削除）を行う。

Ｓ１１　ユーザの携帯型音楽選曲視聴機から配信センターを呼び出し、選曲する。（携帯型音楽選曲視聴機のディスプレーとプッシュボタンで、曲名・歌手名・ジャンル・曲名No.・作曲家名等を選択できる。）

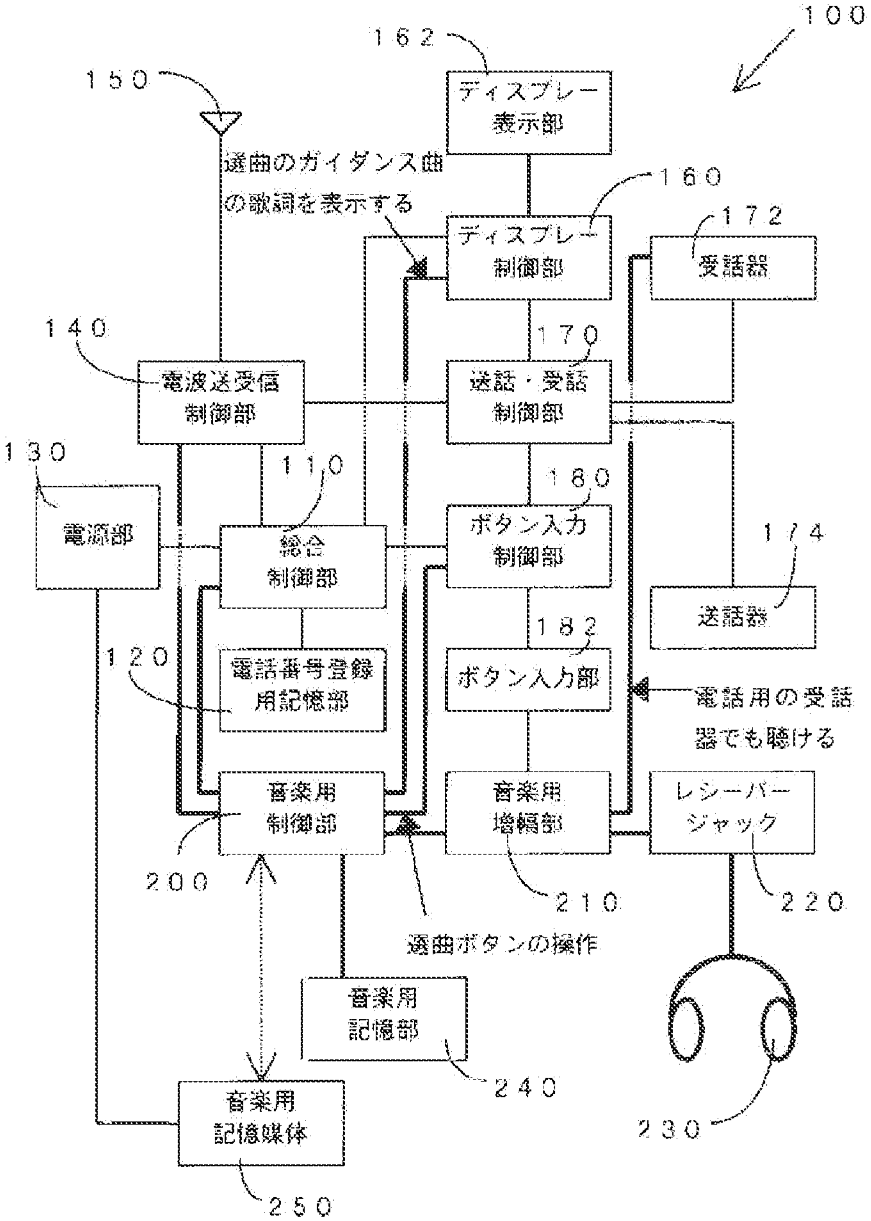
Ｓ１２　選択した曲と歌詞・画像を配信センターからユーザの携帯型音楽選曲視聴機に回線を通して送信する。

Ｓ１３　ユーザの携帯型音楽選曲視聴機のレシーバから音声、ディスプレーに歌詞と画像を表示する。記憶装置付きの場合は記録し、通話終了後も再生可能とする。

Ｓ１４　配送センターから送信が終了したら課金処理を行う。課金処理についてはＮＴＴのＱ２方式に準拠して行う。

Ｓ１５　記憶装置から再生中に通話着信があったら、割り込みの通知または表示をする。

【図２】



フロントページの続き

(72)発明者　岩崎　美奈
神奈川県横浜市戸塚区品濃町504番地２
日立電子サービス株式会社内

TRANSLATION DECLARATION

I, David Baldwin, hereby declare:

1. I am a translator for MultiLing Corporation, a professional translation company incorporated in Delaware with its principal office located at 180 North University Avenue, Suite 600, Provo, Utah 84601-4474.

2. I am competent to translate between the Japanese and English languages.

3. At the request of Ropes & Gray, LLP, I translated JP Patent Application Publication No. H11-164058 (Sato) from Japanese to English.

4. To the best of my knowledge and belief, the attached English language document is a true, complete, and correct translation of JP Patent Application Publication No. H11-164058 (Sato).

5. I make this declaration of my own personal knowledge. If called to testify as to the truth of the matters stated herein, I could and would testify competently.

6. I declare under penalty of perjury that the foregoing is true and correct.

Executed this 7th day of March, 2014, at Meridian, ID.

David Baldwin
MultiLing Corporation
180 North University Avenue
Suite 600
Provo, Utah 84601-4474

(19) Japanese Patent Office (JP)

(12) Publication of Unexamined Patent Application (A)

(11) Disclosure Number: Unexamined Application H11-164058

(43) Date of Disclosure: June 18, 1999

(51) Int. Cl.^{6}	Identification Code	FI	
H04M 11/08		H04M 11/08	
H04B 7/24		H04B 7/24	C
H04M 3/42		H04M 3/42	Z

Examination Request Status: Not Yet Requested. No. of Claims: 4, OL (5 pages total)

(21) Filing Number: Patent Application No. H9-324660

(22) Date of Application: November 26, 1997

(71) Applicant: 000233491
Hitachi Electron Service Co., Ltd.
504-2 Shinano-cho, Totsuka-ku, Yokoharna-shi, Kanagawa-ken

(72) Inventor: Noboru SATO
c/o Hitachi Electron Service Co., Ltd.
504-2 Shinano-cho, Totsuka-ku, Yokohama-shi, Kanagawa-ken

(72) Inventor: Norikazu YAMAGISHI
c/o Hitachi Electron Service Co., Ltd.
504-2 Shinano-cho, Totsuka-ku, Yokohama-shi, Kanagawa-ken

(72) Inventor: Mutsuharu TAKESADA
c/o Hitachi Electron Service Co., Ltd.
504-2 Shinano-cho, Totsuka-ku, Yokohama-shi, Kanagawa-ken

(72) Inventor: Mina IWASAKI
c/o Hitachi Electron Service Co., Ltd.
504-2 Shinano-cho, Totsuka-ku, Yokohama-shi, Kanagawa-ken

(74) Agent: Yoshiaki NUMAGATA, Attorney (plus 3 others)

(54) Title of the Invention: PORTABLE MUSIC SELECTION AND VIEWING SYSTEM

(57) [ABSTRACT]

[Problem]

To provide a system providing music software where a user can select a song using a mobile phone.

[Resolution Means]

A distribution center 10 having a server functionality receives a service for music software from a record production company 20. A portable music selection and viewing device 50 connected to the distribution center 10 via a public communications network 30 has a main body 51 having a button input part, a display 52 provided on the main body 51, and a music receiver 54, wherein a user selects a song and requests music software from the distribution center, received audio of the music software is output to a receiver 54, and lyrics and the like are output to the display 52.

[Scope of the Patent Claims]

[Claim 1]

A portable music selection and viewing system comprising:

a record company that produces music software, a distribution center having a server function that receives a music service from the record company, and a portable music selection and viewing device connected to the distribution center via a public communications network;

the portable music selection and viewing device comprising means to transfer inputted song selection information to the distribution center via a public communications network and to output the music software sent from the distribution center as audio and text information.

[Claim 2]

The portable music selection and viewing system according to claim 1, wherein the portable music selection and viewing device comprises a power source unit, a comprehensive control unit, a telephone number registration storage unit, a button input part, a display unit, a transmitting and receiving control unit, a transmitter and a receiver, a radio wave transmitting and receiving control unit, an antenna, a music control unit, a music amplifier, and a music receiver.

[Claim 3]

The portable music selection and viewing system according to claim 2, wherein the portable music selection and viewing device comprises a music storage unit that stores received music software.

[Claim 4]

The portable music selection and viewing system according to claims 2 and 3, wherein the portable music selection and viewing device comprises a detachable music storage medium that stores music software.

[Detailed Description of the Invention]

[0001]

[Technical Field of the Invention]

The present invention relates to a music distribution system which uses a mobile phone.

[0002]

[Related Art]

For example, music software may be enjoyed by receiving broadcast waves from a satellite or a ground station using a portable radio or television. The receiving of this broadcast wave is a one-way service from a broadcast station, and the user cannot select songs. Further, a service can be received in a communication karaoke system or the like where music selected by a user can be requested to a center using a wired line.

[0003]

[Problems to be Solved by the Invention]

In conjunction with the spread of mobile phone systems, it is now possible to offer users a variety of services in addition to phone service. The present invention provides a music selection viewing system that uses a wireless public communications network.

[0004]

[Summary of the Invention]

The music selection viewing system of the present invention provides, as specific means, a record company that produces music software, a distribution center having a server function that receives a music service from the record company, and a portable music selection and viewing device connected to the distribution center via a public communications network. Additionally, the portable music selection and viewing device provides means to transfer inputted song selection information to the distribution center via a public communications network, and to output the music software sent from the distribution center as audio and text information. In addition, the portable music selection and viewing device is provided with, as specific means, a power source unit, a comprehensive control unit, a telephone number registration storage unit, a button input part, a display unit, a transmitting and receiving control unit, a transmitter and a receiver, a radio wave transmitting and receiving control unit, an antenna, a music control unit, a music amplifier, and a music receiver. In addition, the mobile music selection device provides a music memory unit that stores received music software, or it can provide a detachable music storage medium that stores the music software.

[0005]

[Description of the Preferred Embodiment]

FIG. 1 is an overall configuration diagram of the portable music selection and viewing system of the present invention. The overall system shown by numeral 1 has a distribution center 10, which is a server, and this distribution center 10 receives a supply of music content from a record production company 20. Portable music selection and viewing devices 50, 60, and 70 are connected to this distribution center 10 via a public communications network 30.

[0006]

The portable music selection and viewing device 50 has a similar structure to a mobile phone, for example, and is provided with a display 52 and necessary push buttons or the like on a main body 51. A receiver 54 is connected to the main body 51. Users having the portable music selection and viewing device 50 operate the push buttons or the like on the main body 51 to call the distribution center 10 via the public communications network 30 and receive the desired music software via the public communications network 30. The received software is amplified by an amp mounted in the main body 51 of the portable music selection and viewing device 50 and is output to a receiver 54.

[0007]

Users with the receiver 54 installed can display lyrics on the display 52 as necessary and can enjoy karaoke in addition to enjoying music. This portable music selection and viewing device 50 is simplified, having no memory function or the like provided and can receive a supply of music only when connected online.

[0008]

The portable music selection and viewing device 60 shows a model incorporating a storage device 66 in a main body 61. With this model of portable music selection and viewing device 60, the push buttons or the like of the main body 61 are operated to call the distribution center 10, and the supplied music software received is output to a receiver 64 and to a display 62 and stored in the storage device 66. Therefore, the user can play and enjoy the music software in the storage device 66 even after the connection to the public communications network 30 is disconnected.

[0009]

The portable music selection and viewing device 70 provides a removable storage device 76 on a main body 71. This storage device 76 is a memory card similar to, for example, a magnetic card, a magnetic tape, a CD, a DVD, or an IC card. The user, after downloading the music software to the storage device (medium) 76 of the portable music selection and viewing device 70 by operating the push buttons or the like on the main body 71, can enjoy this music software on a display 72 or a receiver 74 of the portable music selection and viewing device 70, and can also enjoy higher quality music playback by removing this storage device (medium) and inserting it into another audio unit. Further, the user can store the music software from another audio unit into the storage device 76 and enjoy music by inserting this storage unit 76 into this portable music selection and viewing device 70.

[0010]

FIG. 2 is a configuration diagram of the portable music selection and viewing device of the present invention having the functionality of a mobile phone. The portable music selection and viewing device shown overall by numeral 100 has a comprehensive control unit 110 connected to a power source unit 130, and a telephone number registration storage unit 120 is connected to the comprehensive control unit 110. A button input control unit 180, having a button input part 182 which the user operates, sends a signal to the comprehensive control unit 110, and the comprehensive control unit 110 displays operation content on a display unit 162 via a display control unit 160 and accesses the public communications network via a radio wave transmitting and receiving control unit 140 and an antenna 150. If the user calls a phone of another party, the switchboard calls the other party's phone, and the user can communicate with the other party's phone by using a transmitter 174 and a receiver 172 connected to a transmitting and receiving control unit 170.

[0011]

A music control unit 200 is connected to the comprehensive control unit 110, the power source unit 130, the radio wave transmitting and receiving control unit 140, the button input control unit 180, and a display control unit 160. After the user operates the button input part 182 and outputs a command to call the distribution center, the content is displayed on the display unit 162, and the radio wave transmitting and receiving control unit 140 accesses the distribution center via the public communications network through the antenna 150.

[0012]

Once the distribution center is accessed, the user commands a song selection, and the distribution center sends back the selected music software. The music control unit 200 which has received this music software amplifies the signal using a music amplifier and outputs audio to a music receiver 230 inserted into a receiver jack 220. This audio output can also be output to a receiver 172. Information about lyrics or the like is displayed on the display unit 162.

[0013]

A music storage device 240 connected to the music control unit 200 stores the music software. A music storage medium 250 such as a magnetic card, magnetic tape, a CD, a DVD, or a memory card such as an IC card stores the music software, and this storage medium 250 can be removed and used on other audio units.

[0014]

FIG. 3 is a flow chart of a process according to the system of the present invention. In step S10, registration (modification/deletion) of a song, lyrics, and images is performed with the distribution center 10 from the record production company 20. In step S11, the user can select a song name, artist name, genre, song number, composer name and the like using the display and push buttons of the portable music selection and viewing device. In step S12, the selected song, lyrics, and images are transmitted from the distribution center to the user through a line.

[0015]

In step S13, the audio is played from the receiver of the user and the lyrics and images are displayed on the display. With a storage device, it is possible to save and then playback after a call has ended. In step S14, a billing process is performed after the transmission from the distribution center is complete. The billing process is performed in compliance with the Q2 method of NTT. In step S15, if there is an incoming call during playback from the storage device, an interruption notification or display is performed.

[0016]

[Effect of the Invention]

The present invention as described above allows a user to enjoy selected music software using a mobile phone, and thereby expands the use of public communications networks and also improves service.

[Brief Description of the Drawings]

FIG. 1 is a configuration diagram of the portable music selection and viewing system of the present invention.

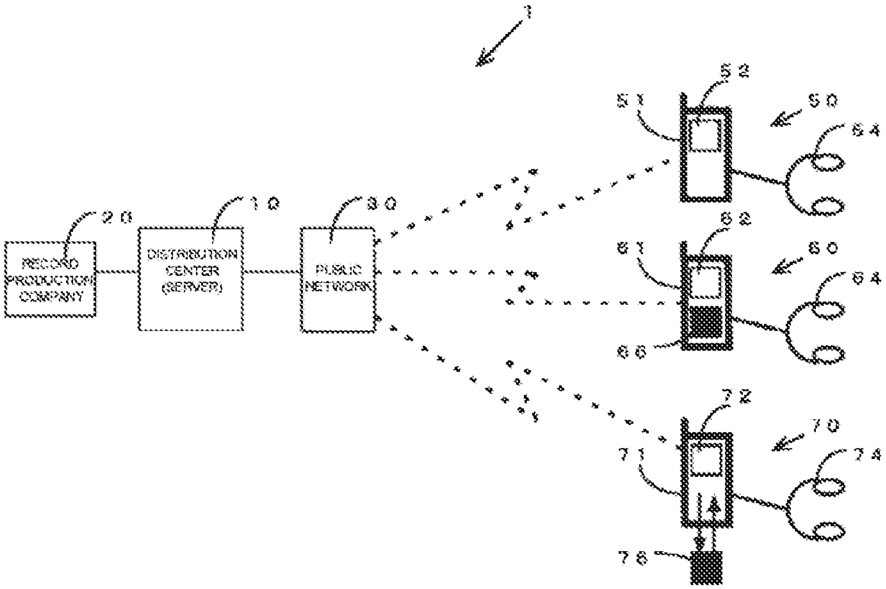
FIG. 2 is a configuration diagram of the portable music selection and viewing device of the present invention.

FIG. 3 is a flow diagram of the portable music selection and viewing system of the present invention.

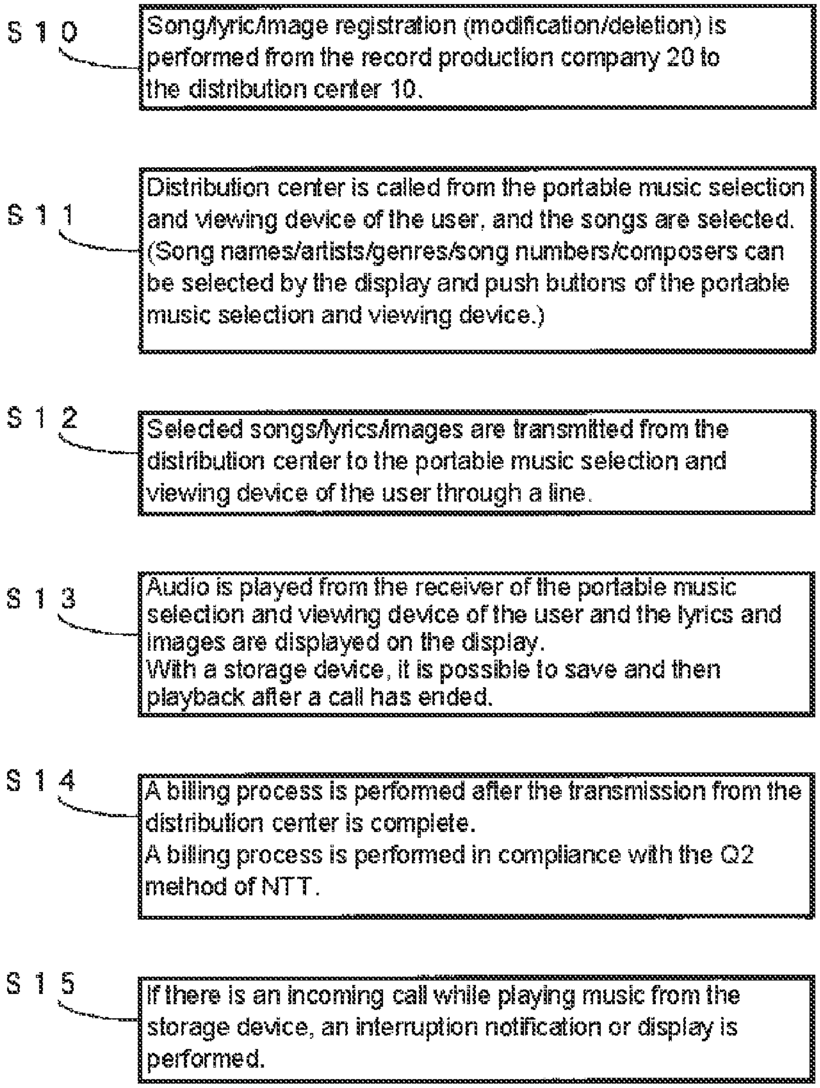
[Reference Numerals]

10 Distribution center
20 Record production company
30 Public communications network
50, 60, 70 Portable music selection and viewing device

[FIG. 1]



[FIG. 3]



S10 Song/lyric/image registration (modification/deletion) is performed from the record production company 20 to the distribution center 10.

S11 Distribution center is called from the portable music selection and viewing device of the user, and the songs are selected. (Song names/artists/genres/song numbers/composers can be selected by the display and push buttons of the portable music selection and viewing device.)

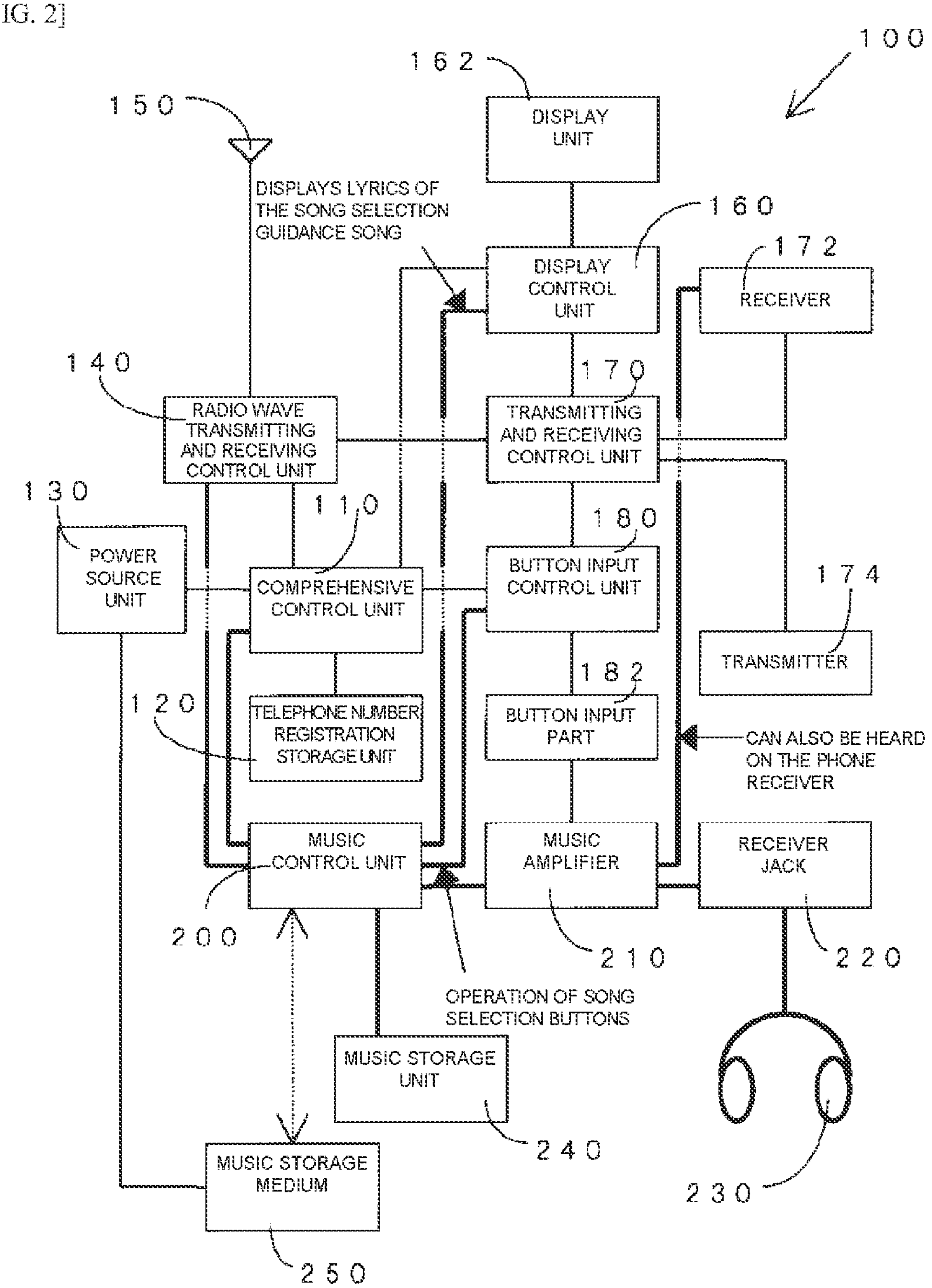
S12 Selected songs/lyrics/images are transmitted from the distribution center to the portable music selection and viewing device of the user through a line.

S13 Audio is played from the receiver of the portable music selection and viewing device of the user and the lyrics and images are displayed on the display.
With a storage device, it is possible to save and then playback after a call has ended.

S14 A billing process is performed after the transmission from the distribution center is complete.
A billing process is performed in compliance with the Q2 method of NTT.

S15 If there is an incoming call while playing music from the storage device, an interruption notification or display is performed.

[FIG. 2]



100
162
DISPLAY UNIT
150
DISPLAYS LYRICS OF THE SONG SELECTION GUIDANCE SONG
160
DISPLAY CONTROL UNIT
172
RECEIVER
140
170
RADIO WAVE TRANSMITTING AND RECEIVING CONTROL UNIT
TRANSMITTING AND RECEIVING CONTROL UNIT
130
110
180
POWER SOURCE UNIT
COMPREHENSIVE CONTROL UNIT
BUTTON INPUT CONTROL UNIT
174
TRANSMITTER
120
182
TELEPHONE NUMBER REGISTRATION STORAGE UNIT
BUTTON INPUT PART
CAN ALSO BE HEARD ON THE PHONE RECEIVER
MUSIC CONTROL UNIT
MUSIC AMPLIFIER
RECEIVER JACK
200
210
220
OPERATION OF SONG SELECTION BUTTONS
MUSIC STORAGE UNIT
240
230
MUSIC STORAGE MEDIUM
250

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ^{6} :
H04M 1/72, 1/60

A1

(11) International Publication Number: **WO 99/43136**

(43) International Publication Date: 26 August 1999 (26.08.99)

(21) International Application Number: PCT/US99/00570

(22) International Filing Date: 11 January 1999 (11.01.99)

(30) Priority Data:
09/025,395 18 February 1998 (18.02.98) US

(71) Applicant: ERICSSON, INC. [US/US]; P.O. Box 13969, Research Triangle Park, NC 27709-3969 (US).

(72) Inventors: RYDBECK, Nils, R., C.; 202 Rutherglen, Cary, NC 27511 (US). FUSSELL, John, P.; 2844 Mattlyn Court, Raleigh, NC 27613 (US).

(74) Agents: BENNETT, David, E. et al.; Rhodes, Coats & Bennett, LLP, P.O. Box 5, Raleigh, NC 27602 (US).

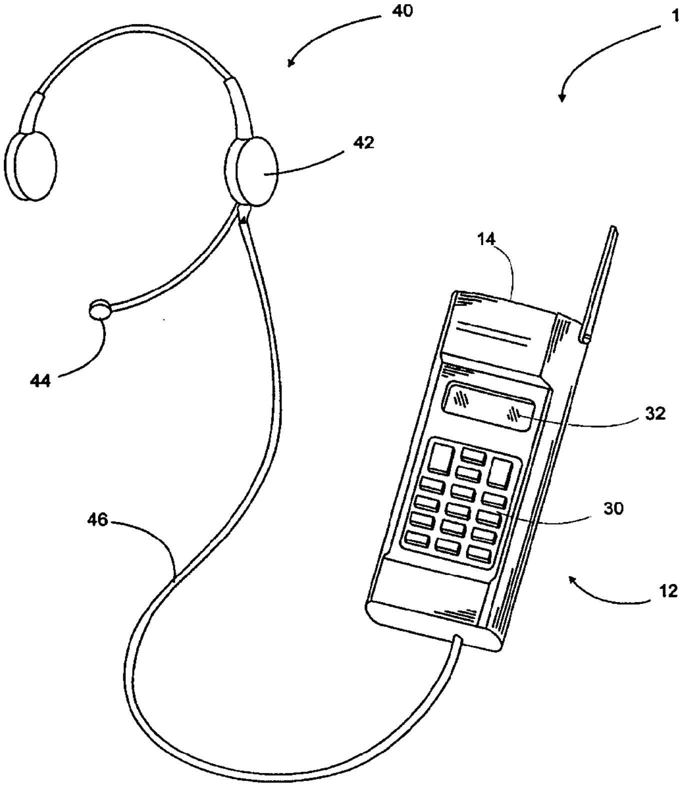
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN,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

Published
With international search report.

(54) Title: CELLULAR PHONE WITH EXPANSION MEMORY FOR AUDIO AND VIDEO STORAGE

(57) **Abstract**

A cellular telephone includes an internally integrated digital entertainment module. The telephone includes a transceiver unit and a headset which is connected to the transceiver unit by wired or wireless link. The entertainment module includes an interchangeable ROM and/or expansion RAM for storing music or other audio signals for playback through the telephone's headset. Music or other audio signals in digitized form is stored in the interchangeable ROM or is loaded into the expansion RAM from a CD player, computer, or other source of digitized audio signals. Under control of the cellular telephone's microprocessor, the digitally stored audio signal is played back through the telephone's headset. The entertainment module may be located in the transceiver unit, a removable battery pack, or in a separate adapter.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

CELLULAR PHONE WITH EXPANSION MEMORY FOR AUDIO AND VIDEO STORAGE

FIELD OF THE INVENTION

The present invention relates generally to mobile communication devices, and more particularly to portable radio communication devices having an integral entertainment module including RAM or ROM for storing audio, video and/or still images.

BACKGROUND OF THE INVENTION

In the past two decades, advances in digital electronic technology have led to a rapid growth in the area of entertainment oriented consumer electronic devices. In particular, portable electronic devices such as audio CD players, FM/AM radio receivers, and even television or video tape/disc players have become increasingly popular among consumers as they have become small, lightweight, and easy for an individual to carry.

While quite popular with consumers, the mass storage type devices (audio CD, video tape/disc) typically suffer from motion induced distortion otherwise known as bouncing or skipping. These problems arise, in part, as a result of the required motion of the mass storage medium during normal operation. That is, in the case of an audio CD or a video disc, the disc which comprises the storage medium is typically spun or rotated at a relatively high speed while the information stored on the disc is read by an associated read head. Proper and precise alignment of the read head with respect to the spinning storage medium must be maintained at all times in order to insure error free reading of the stored data. Such precise alignment is often difficult to maintain when the audio or video player is being used in manner which is conducive to extreme vibration or mechanical shock. In practice, mechanically harsh activities such as jogging or running are common among users of portable electronics, particularly with regard to the use of portable audio CD players. In such cases, skipping or bouncing artifacts induced in the CD player can seriously impair the overall performance of the player.

With further regard to the recreational athletic activities of portable electronics consumers, it is often the case such consumers will carry not only an audio CD player for entertainment purposes, but also a cellular telephone for safety and security. Although such equipment provides the desired entertainment/security services to the athletically active consumer, the need to carry multiple pieces of equipment is generally viewed as inhibiting or impairing to their athletic endeavors.

Therefore, there is and continues to be a need for a practical and efficient technique for incorporating the functionality of audio and/or video playing devices within wireless communications devices such as cellular telephones.

SUMMARY OF THE INVENTION

The present invention is a cellular telephone particularly adapted for leisure activities. The cellular telephone of the present invention includes a portable transceiver unit and a headset which can be worn by the user during leisure activities such as jogging, biking, gardening, etc. The transceiver unit includes a fully functional transceiver capable of sending and receiving voice and data signals via an RF carrier. The transceiver unit has an integral digital entertainment module including a memory for storing music or other audio signals for playback through the headset. For purposes of this application, memory means all forms of computer memory but dies not include disk storage, tape storage or other memory requiring electromechanical read systems. The memory may be in the form of a removable ROM cartridge and/or an expansion RAM. In those embodiments having an expansion RAM, an input port is provided for loading music or other audio signals into the expansion RAM from a CD player, computer, or other source of digitized audio.

Under the control of the transceiver unit's microprocessor, the digitally stored audio signal is played out through the telephone's headset, which in the preferred embodiment comprises stereo headphones. The headset may be connected to the phone by a wired or wireless link. Because of its integration into the cellular phone, the digital entertainment module can share components already present in the cellular phone. Such savings would not be available if a CD player were simply aggregated with the phone. Further, the use of solid state RAM or ROM, as opposed to disc storage, eliminates the need for bounce control circuitry. This enables the disclosed invention to provide cellular communications and entertainment during leisure activities.

In another aspect of the present invention, the digital entertainment module could be located in a removable battery pack which attaches to the transceiver unit, or in a separate adapter which plugs into the transceiver unit. Locating the digital entertainment module in either a battery pack or separate adapter allows the manufacturer to offer the digital entertainment module as an optional accessory which does not need to be purchased at the same time the cellular phone is purchased. This allows consumers who purchase a phone without the digital entertainment module to later purchase the battery pack or adapter as an upgrade to the existing phone.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a perspective view of the portable communication device of the present invention.

Figure 2 is a block diagram of the portable communication device.

Figure 3 is a block diagram of the entertainment module contained in the portable communication device.

Figure 4 is a perspective view of a second embodiment of the portable communication device in which the digital entertainment module is located in a removable battery pack.

Figure 5 is a block diagram showing the second embodiment of the portable communication device in which the entertainment module is located in a removable battery pack.

Figure 6 is a perspective view of a third embodiment of the portable communication device in which the digital entertainment module is located in a separate adapter with attaches to the transceiver unit.

Figure 7 is a block diagram showing the third embodiment of the portable communication device in which the entertainment module is located in a separate adapter.

DETAILED DESCRIPTION OF THE INVENTION

Referring now to the drawings, and particularly to Figures 1 through 3, the cellular phone of the present invention is shown therein and indicated generally by the numeral 10. The cellular phone 10 of the present invention is particularly adapted for use during leisure activities such as jogging, hiking, gardening, etc.

The cellular phone 10 includes a transceiver unit 12 and a headset 40 which can be worn on the head by the user. The transceiver unit 12 includes a main housing 14 and a removable battery pack 16 containing a rechargeable battery 28. Housing 14 of the transceiver unit 12 contains an RF transceiver 18, control logic 20, program memory 22, and audio processing unit 24 which are operatively connected by a system bus 26. The RF transceiver 18 may be, for example, a class 1 mobile phone transceiver capable of transmitting and receiving radio signals containing voice and/or data. Audio processing unit 24 processes voice and data signals that are transmitted and received by the transceiver 18. Audio processing unit 24 may include voice recognition circuitry to enable activation and use of the phone 10 by voice commands for truly hands-free operation. The control logic 20 controls the operation of the transceiver 18 according to instructions stored in program memory 22. A keypad 30 and display 32 provide a user interface. Keypad 30 enables the user to enter dialing instructions and commands to initiate a call, and to select options. The display 32 displays the number dialed and call status information to the user. Display 32 may also display instructions or options to the user. Unlike a conventional cellular phone, the transceiver unit 12 of the present invention does not include an internal microphone and speaker, though such is within the scope of the contemplated invention.

The headset 40 includes stereo speakers 42 and microphone 44 that are connected to the transceiver unit 12 by a cable 46. Cable 46 may include a plug (not shown) which removably mates with a corresponding jack on the transceiver unit 12. The cable 46 connects to the system bus 26 which routes audio signals from the audio processing unit 24 to and from the headset 40 under the control of the microprocessor 20. The jack could also connect directly to audio processing circuit 24. Alternatively, the headset 40 could communicate wirelessly with the transceiver unit 12, for example, by means of an infrared carrier, low power RF carrier or magnetic link.

The portable telephone 10 of the present invention includes a built-in digital entertainment module 50 (DEM) which allows music or other audio signals to be "played-back through the cellular telephone's headset 40. The entertainment module 50 includes extended RAM and/or removable memory cartridges for storing music or other audio signals which can be played back through the headset 40 of the phone 10.

Referring now to Figure 3, a schematic diagram of the digital entertainment module 50 is shown. The digital entertainment module 50 includes a secondary bus 52, extended random access memory (RAM) 54, removable ROM 56, and an input 58. The extended RAM 54 may, for example, be a flash EPROM chip capable of storing digitized audio. Digitized audio is loaded into the flash EPROM via input 58. The input 58 may be a serial port, parallel port, infra-red data port, modem, or any other type of input device capable of interfacing with a source of digitized audio, such as a CD player, or computer. It is also contemplated that audio may be obtained from the transceiver unit 12 in an "internet-enabled" phone 10. The removable ROM 56 is preferably in the form of a cartridge which fits into a slot in the transceiver unit 12. The ROM cartridge 54 would contain pre-recorded music which could be purchased by the user. In the preferred embodiment, the data format of both the extended RAM 54 and removable ROM 56 would be organized according to CD-ROM standards, which is 14 bits per sample and 44.1 k samples per second.

In operation, the user would insert a removable ROM cartridge 56 into the transceiver unit 12 or load audio into the extended RAM 54 from a CD player, computer, or other source of digitized audio. The transceiver unit 12 is attached the belt or other article of clothing worn by the user. The headset 40 is placed on the user's head and connected to the transceiver unit 12. Playback of audio in the extended RAM 54 or removable ROM 56 could be activated via the keypad 30, or alternately, by voice command. The audio would be played back through the headset 40 under control of the microprocessor 20 while the user engages in leisure activities. When an incoming call is received, the microprocessor 20 automatically mutes or stops the playback of audio from the digital entertainment module 50 until the call is terminated. Preferably, the transceiver unit 12 includes a preferred caller list stored in a screening memory which may be part of program memory 22 or separate therefrom but in communication with the control logic 20. This preferred caller list is used to screen incoming calls such that only calls from callers on the preferred caller list cause the playback of audio from the digital entertainment module 50 to be muted or stopped; calls from callers not on the preferred caller list preferably do not cause such response. Upon termination of the call, the microprocessor 20 would unmute or restart the playback of audio from the digital entertainment module 50.

A significant advantage of the present invention is that audio is played back from solid state RAM or ROM memory thus eliminating the need for bounce control circuitry which is commonly used in portable CD players. Further, because of its integration into the cellular phone 10, there is no need for the user to carry both a portable audio player and a cellular phone. Moreover, integration of the entertainment module 50 into the cellular phone 10 allows the entertainment module 10 to share components with the cellular phone 10 to take advantage of the phone's communication capability to load the RAM 54. Thus, the present invention could replace both a conventional cellular phone and portable audio player at lower cost than a conventional walk-man and telephone.

Referring now to Figures 4 and 5, a second embodiment of the present invention is shown. The second embodiment is similar to the first embodiment and, therefore, the same reference numerals will be used to identify similar components. As shown in Figures 4 and 5, the second embodiment of the phone 10 includes a transceiver unit 12 with a removable battery pack 14, and a headset 40 connected to the transceiver unit 12. The transceiver unit 12 includes a transceiver 18, microprocessor 20, program memory 22, audio processing circuits 24, keypad 30 and display 32 as previously described. Similarly, the headset 40 includes stereo speakers 42 and microphone 44. The second embodiment differs from the first in that the digital entertainment module 50 is contained within the removable battery pack 14. The entertainment module 50 connects to a secondary bus in the battery pack 14. When the battery pack 14 is attached to the transceiver unit 12, a connection is made between the secondary bus in the battery pack 14 and the main bus 26 of the transceiver unit 12. The main bus 26 and secondary bus enable the routing of audio signals between the entertainment module 50 and audio processing circuits 24 under the control of the microprocessor 20.

Figures 6 and 7 show a third embodiment of the present invention. The third embodiment is similar to the first and second embodiments and therefore the same reference numbers will be used to identify similar components. As shown in Figures 6 and 7, the third embodiment includes a transceiver unit 12, headset 40, and adapter 70. The transceiver unit 12 includes a transceiver 18, microprocessor 20, program memory 22, audio processing circuits 24, keypad 30, and display 32. In addition, the transceiver unit 12 in the third embodiment includes an internal microphone and speaker 34 and 36 respectively. Thus, the transceiver unit 12 can be used without the headset 40.

The headset 40 includes a pair of stereo speakers 42 and microphone 44. The headset 40 is connected by a cable 46 to the adapter 70. The entertainment module 50 is contained in the adapter 70. The adapter 70 includes a secondary bus 72 which connects to the main bus 26 on the transceiver unit when the adapter 70 is plugged into the transceiver unit 12. An input/output circuit 74 routes audio signals to and from the headset 40.

When the transceiver unit 12 is used without the adapter 70, audio signals are routed under the control of the microprocessor from the audio processing circuits 24 to the internal microphone and speaker 34 and 36. When the adapter 70 is plugged into the transceiver unit 12, the audio signals are routed to the microphone 44 and speakers 42 on the headset 40.

The configuration of the phone 10 shown in Figures 6 and 7 is advantageous in that it allows the transceiver unit 12 to be sold without the digital entertainment module 50 and later upgraded by the consumer. The adapter 70 and headset 40 could be sold separately as an accessory or at a later time as an upgrade. Thus, a single phone could be manufactured for use both with and without the digital entertainment module 50.

It will be apparent to those skilled in the art that the digital entertainment module 50 could also be used to store video or still images which could be output to the display 32 of the transceiver unit 12. Any sound accompanying the video would be played back through the headset 40 or internal speaker. It should also be apparent that the digital entertainment module 50 could include a broadcast receiver for receiving conventional radio and TV broadcasts in addition to its entertainment memory.

The present invention may, of course, be carried out in other specific ways than those herein set forth without departing from the spirit and essential characteristics of the invention. The present embodiments are, therefore, to be considered in all respects as illustrative and not restrictive, and all changes coming within the meaning and equivalency range of the appended claims are intended to be embraced therein.

CLAIMS

What is claimed is:

1. A mobile radio communication device comprising:

a. a transceiver unit for transmitting and receiving audio signals;

b. a speaker operatively connected to said transceiver unit for converting audio signals received by said transceiver unit into audible signals which can be heard by a user;

c. a microphone operatively connected to said transceiver unit for converting the user's voice into audio signals for transmission by said transceiver;

d. memory operatively connected to said transceiver unit for storing pre-recorded audio for subsequent playback through said speaker.

2. The mobile radio communication device according to claim 1 wherein said memory is an erasable memory.

3. The mobile radio communication device according to claim 1 wherein said memory is an unerasable memory.

4. The mobile radio communication device according to claim 1 wherein said memory is contained in said transceiver unit.

5. The mobile radio communication device according to claim 1 further including a removable cartridge insertable into said transceiver unit, wherein said memory is contained in said removable cartridge.

6. The mobile radio communication device according to claim 1 further including a removable battery pack attachable to said transceiver unit, said memory being located in said battery pack.

7. The mobile communication device according to claim 1 further including a detachable adapter for attaching to said transceiver unit, said memory being located in said adapter.

8. The mobile radio communication device according to claim 1 further including a headset, wherein said speaker and microphone are mounted to said headset.

9. The mobile radio communication device according to claim 1 further including a input port operatively connected to said memory for loading audio into said memory.

10. The mobile radio communication device of claim 1 further including a screening memory in communication with said transceiver for storing a list of preferred callers and wherein when an incoming call is received during playback of said pre-recorded audio, playback continues unless said incoming call is from a caller on said list of preferred callers.

11. A cellular telephone having an entertainment module for playing pre-recorded audio and video signals comprising:

a. a transceiver for transmitting and receiving audio and data signals;

b. a microprocessor for controlling the operation of said transceiver;

c. a signal processing circuit operatively connected to the transceiver and microprocessor for processing signals transmitted and received by the transceiver; and

d. an entertainment module with a memory operatively connected to the microprocessor and signal processing circuits for storing audio and video signals for subsequent playback under the control of said microprocessor.

12. The cellular telephone of claim 11 wherein said memory comprises an erasable and programmable memory for storing and playing audio and video signals.

13. The cellular telephone of claim 12 including an input coupled to the erasable and programmable memory for downloading and storing audio and video signals into said erasable and programmable memory.

14. The cellular telephone of claim 11 wherein said memory comprises a permanent memory which is removable from said cellular telephone for storing and playing audio and video signals.

15. The cellular telephone of claim 11 wherein the entertainment module includes a first memory which is programmable and erasable, an input coupled to said first memory for downloading and storing audio and video signals into said first memory, and a second permanent memory having pre-recorded audio and video signals stored therein.

16. The cellular telephone according to claim 15 wherein said second memory is a removable and interchangeable memory cartridge.

17. The cellular telephone of claim 11 wherein the first and second memories are coupled to a headset port in the cellular telephone, thereby permitting audio signals to be directed from the memories to a headset coupled to the cellular telephone via the headset port.

18. The cellular telephone of claim 11 wherein the microprocessor is pre-programmed to preempt output from said first and second memories in response to an incoming call or the initiation of an outgoing call.

19. The cellular telephone of claim 11 further including a screening memory in communication with said microprocessor for storing a list of preferred callers and wherein said output from said first and second memories is not preempted in response to an incoming call unless said incoming call is from a caller on said list of preferred callers.

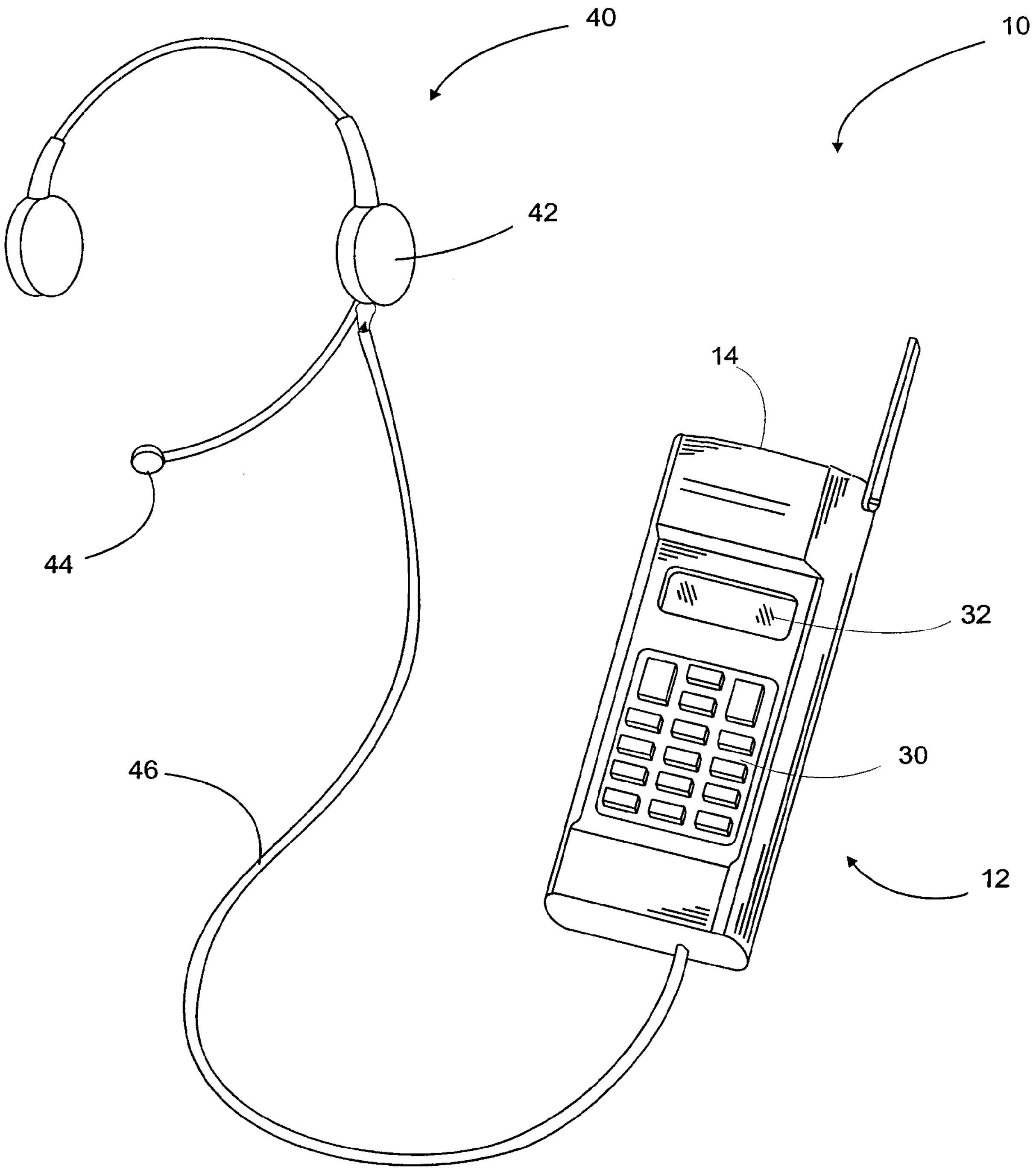


FIG. 1

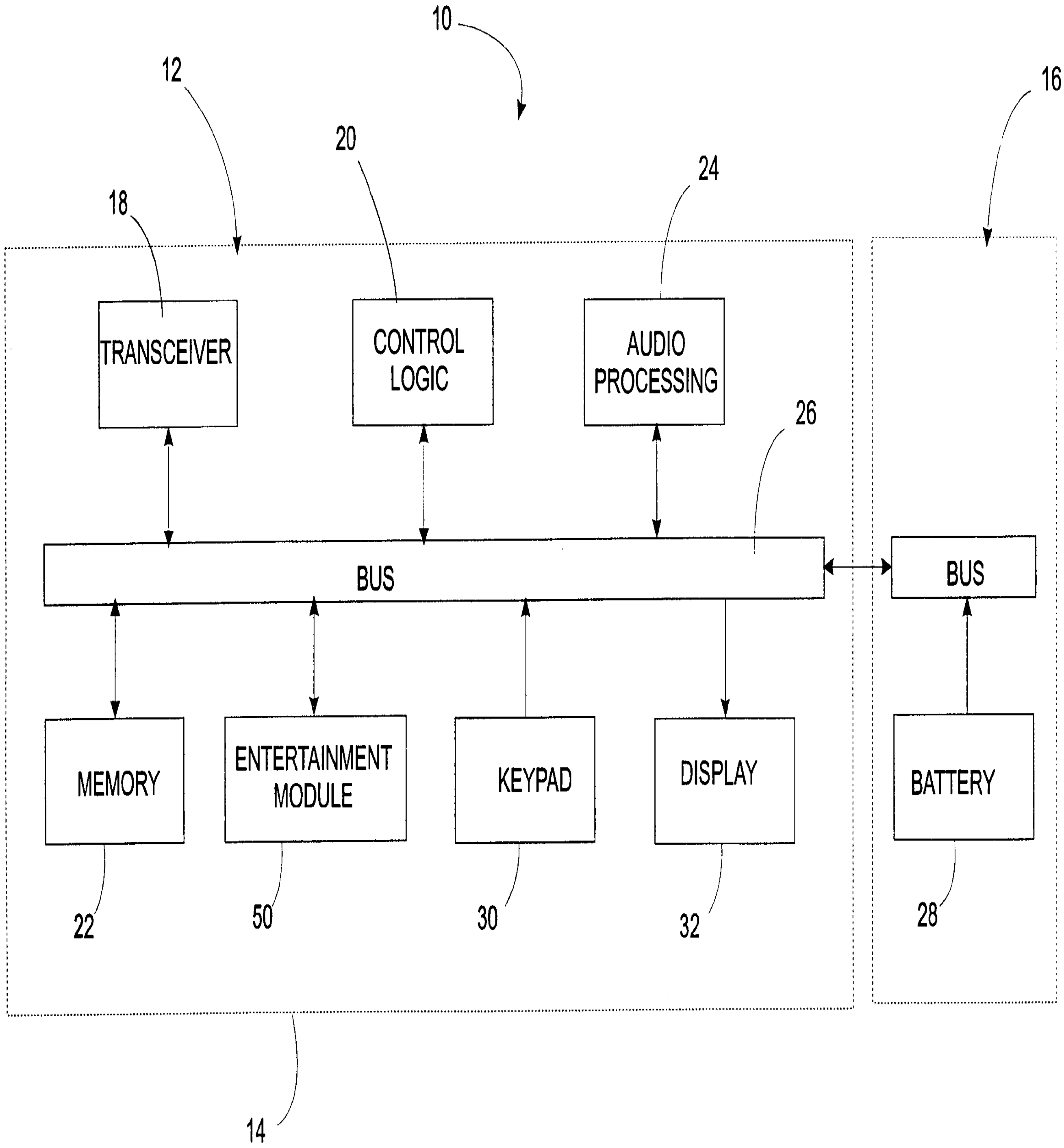

10
12
20
24
16
18
TRANSCEIVER
CONTROL LOGIC
AUDIO PROCESSING
26
BUS
BUS
MEMORY
ENTERTAINMENT MODULE
KEYPAD
DISPLAY
BATTERY
22
50
30
32
28
14

FIG. 2

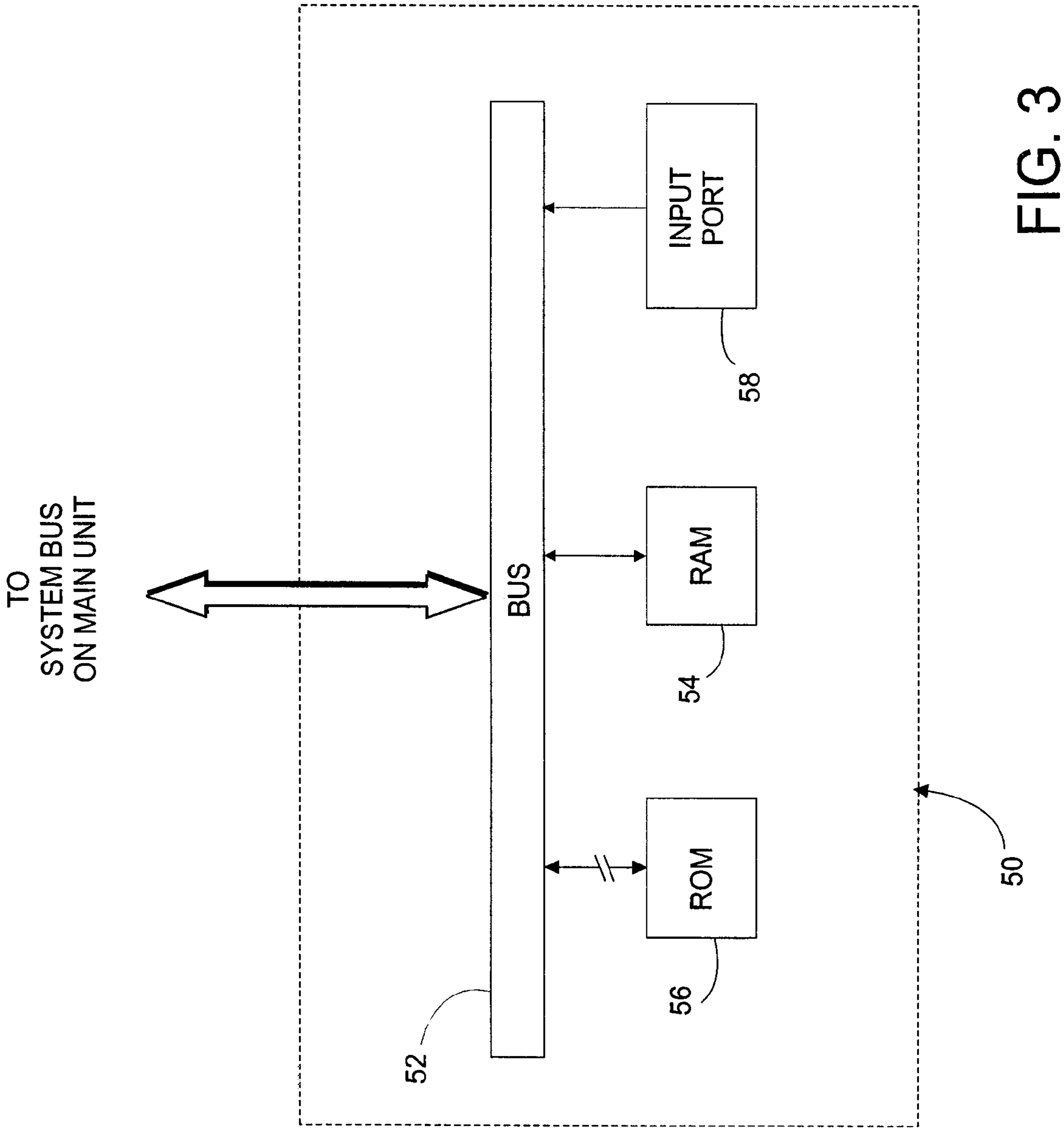


FIG. 3

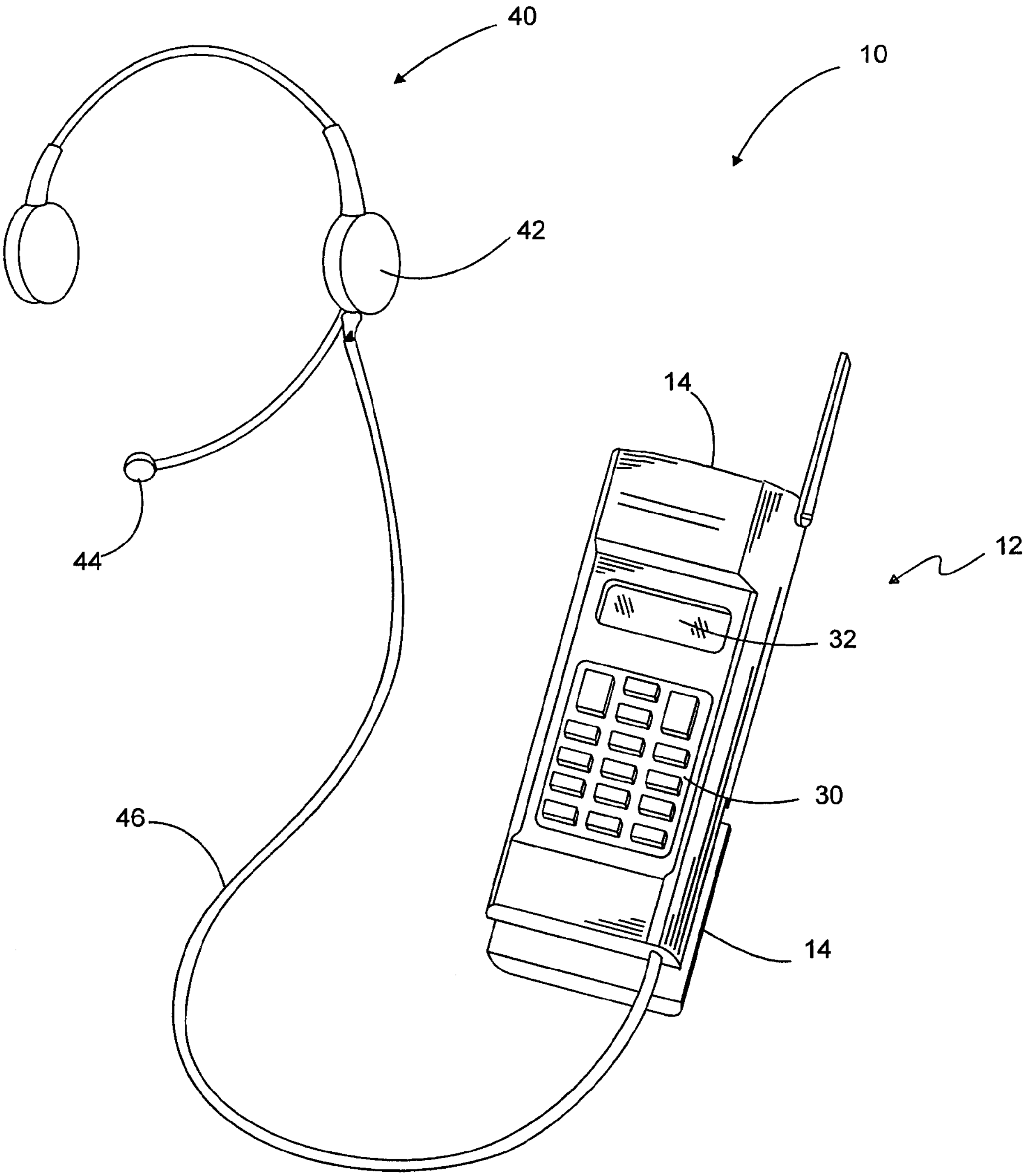


FIG. 4

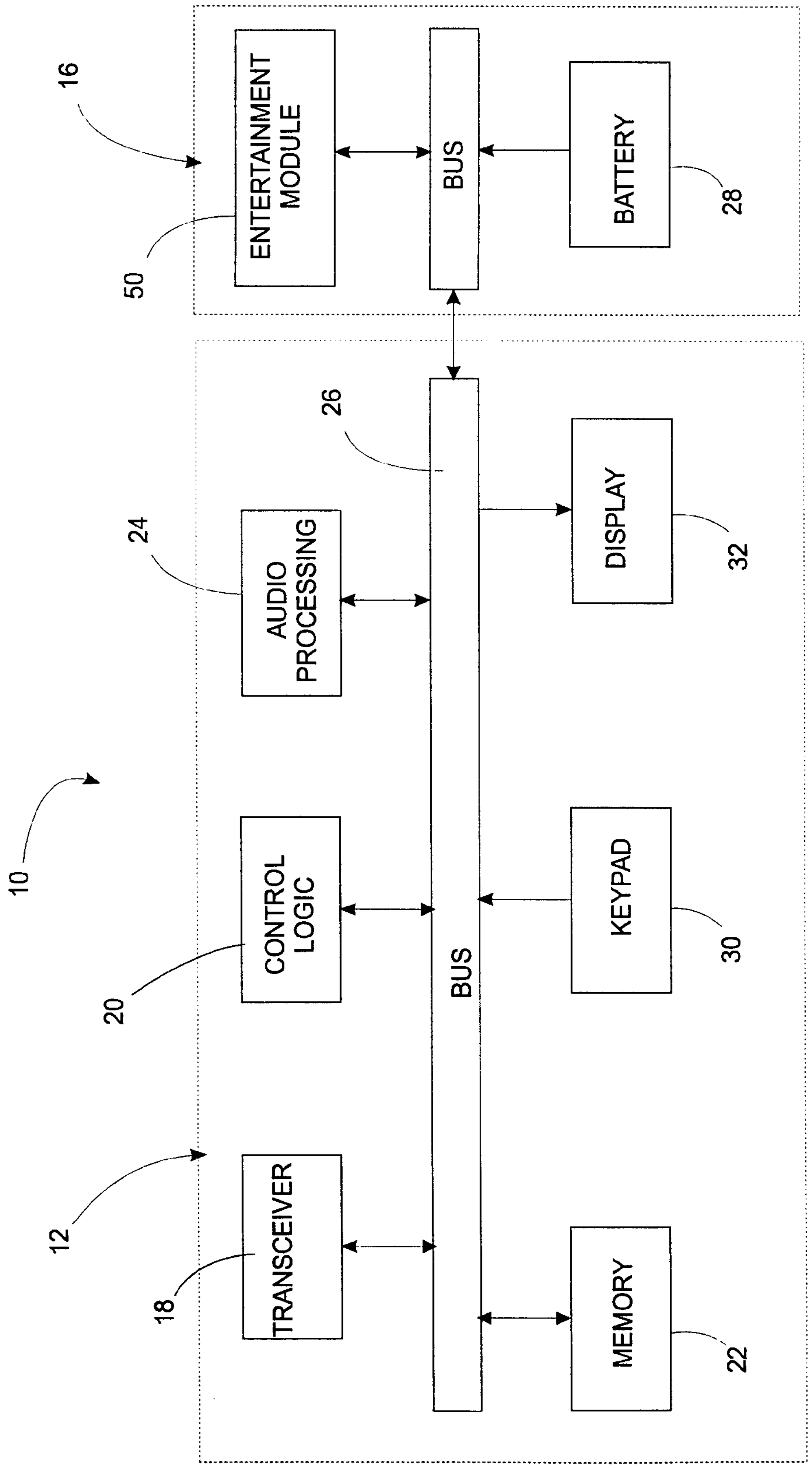

16
ENTERTAINMENT MODULE
50
BUS
BATTERY
28
26
AUDIO PROCESSING
24
DISPLAY
32
10
CONTROL LOGIC
20
BUS
KEYPAD
30
12
TRANSCEIVER
18
MEMORY
22

FIG. 5

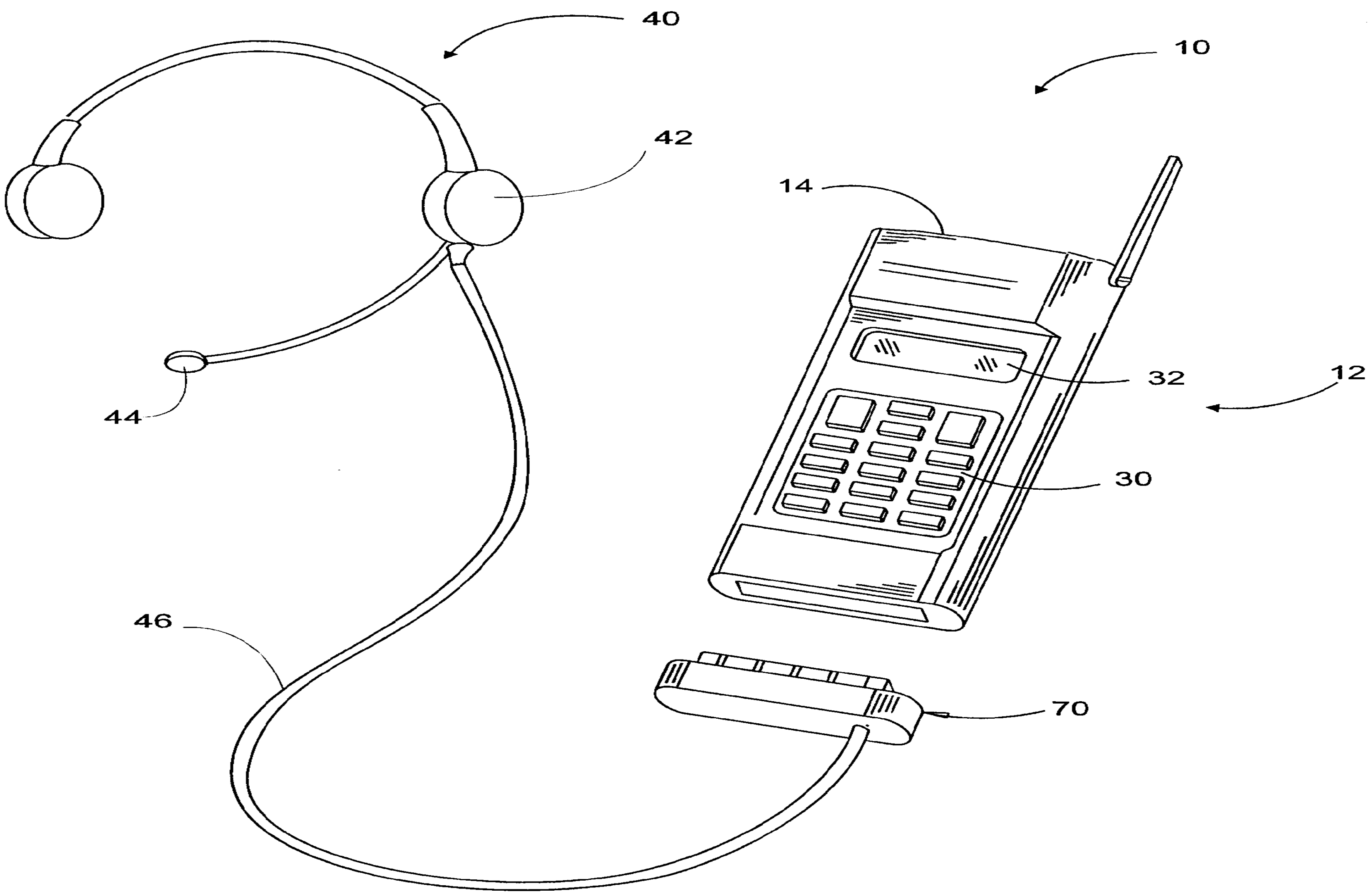


FIG. 6

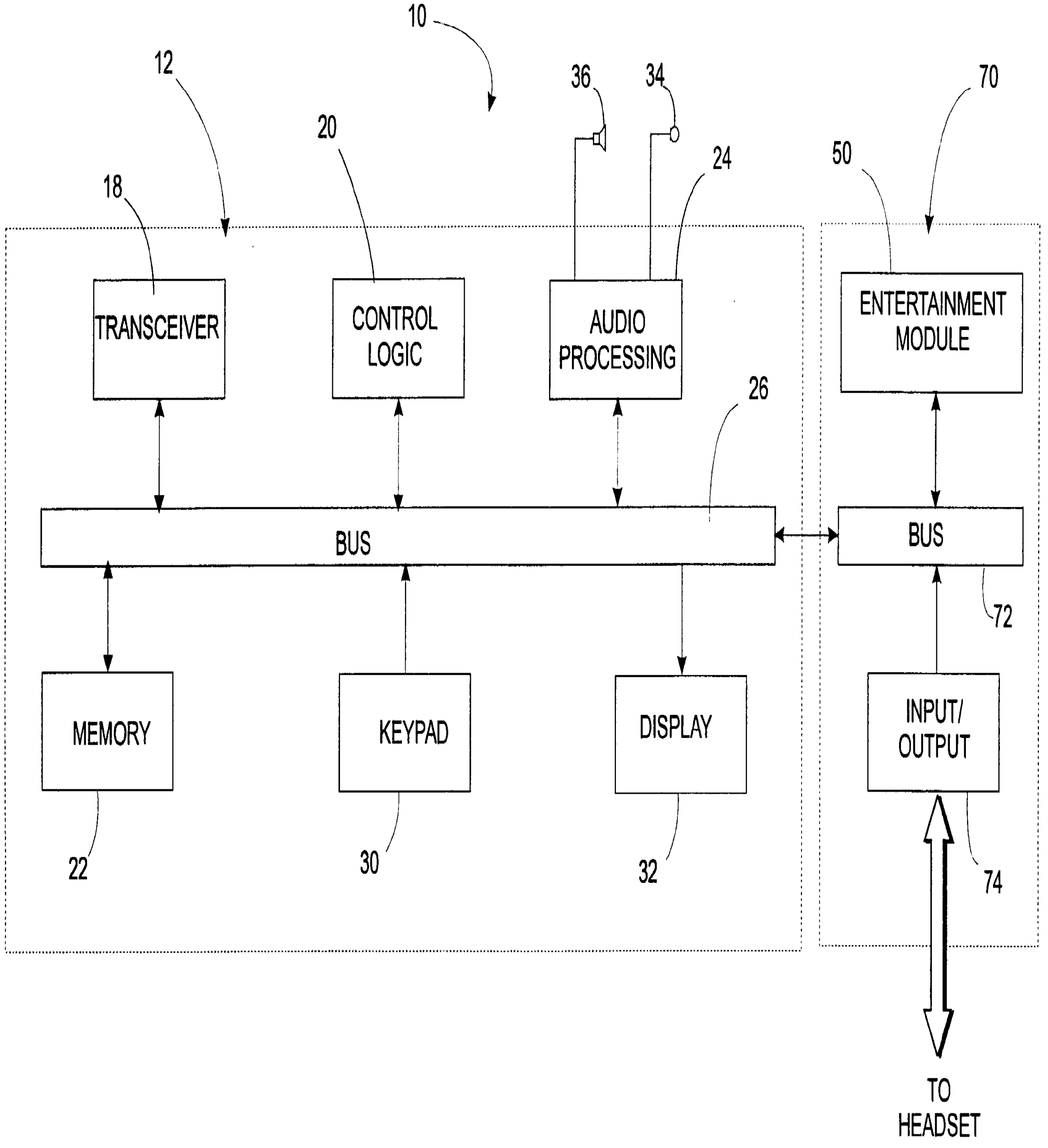

10
12
20
36
34
24
18
50
70
TRANSCEIVER
CONTROL LOGIC
AUDIO PROCESSING
ENTERTAINMENT MODULE
26
BUS
BUS
72
MEMORY
KEYPAD
DISPLAY
INPUT/ OUTPUT
22
30
32
74
TO HEADSET

FIG. 7

INTERNATIONAL SEARCH REPORT

Ir. ational Application No
PCT/US 99/00570

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04M1/72 H04M1/60

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04M H04B

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	GB 2 308 775 A (NIPPON ELECTRIC CO) 2 July 1997	1,2,5,8
A	see page 6, line 1 - line 26 see page 8, line 9 - page 11, line 17 see page 14, line 6 - line 11 see figures 1,2 ---	10,11,18
X	US 4 481 382 A (VILLA-REAL) 6 November 1984	1,2,4,5
A	see column 2, line 29 - line 34 see column 12, line 39 - line 56 see column 13, line 34 - line 50 see figures 6-8 --- -/--	11

☒ 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 21 April 1999	Date of mailing of the international search report 29/04/1999
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Fragua, M

1

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 99/00570

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	US 4 875 229 A (SPEAR ET AL) 17 October 1989 see abstract see column 1, line 25 - line 36 see column 2, line 32 - line 55 see figure 1 ---	1,2,5
A	WO 97 26744 A (ROBB) 24 July 1997 see abstract see page 1, line 8 - page 2, line 6 see page 3, line 5 - page 4, line 4 see page 9, line 18 - page 10, line 17 see page 12, line 12 - line 22 see page 15, line 4 - line 16 see page 16, line 20 - line 25 see page 17, line 10 - page 18, line 6 see page 21, line 13 - page 22, line 26 see page 24, line 1 - line 12 see page 25, line 10 - line 22 see page 26, line 1 - page 27, line 5 see figures 3,9 ---	1-5, 11-16
A	GB 2 289 555 A (NOKIA MOBILE PHONES LTD) 22 November 1995 see page 1, line 1 - line 6 see page 4, line 18 - line 23 see page 5, line 20 - page 6, line 2 see page 7, line 1 - page 8, line 9 see page 10, line 5 - line 7 see page 14, line 1 - line 16 see figures 1-3 ---	1,2,4,7, 11-13
A	DE 195 28 424 A (SIEMENS AG) 21 November 1996 see column 1, line 29 - line 34 see column 4, line 68 - column 5, line 64 see column 7, line 36 - line 60 see column 8, line 43 - line 53 see figures 1,2,6 ---	1,2,4, 11,12
A	US 5 550 754 A (WILLIAMS ET AL) 27 August 1996 see abstract see column 7, line 24 - column 8, line 9 see column 11, line 1 - line 34 see column 12, line 24 - column 13, line 27 see column 20, line 54 - column 22, line 4 see figures 8,16,30,31 --- -/--	1,7-9, 11,17

1

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

INTERNATIONAL SEARCH REPORT

In ıtional Application No
PCT/US 99/00570

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 665 655 A (HEWLETT PACKARD CO) 2 August 1995 see abstract see column 3, line 29 - column 4, line 5 see column 4, line 44 - column 5, line 1 see figures 3B,6 ---	1,2,6,9
A	US 5 661 788 A (CHIN) 26 August 1997 see abstract see column 2, line 47 - column 3, line 3 see column 3, line 38 - column 4, line 3 see column 4, line 59 - line 67 see figure 1 -----	1,4,10, 11,18,19

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

In tional Application No
PCT/US 99/00570

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
GB 2308775 A	02-07-1997	JP 9187060 A	15-07-1997
		AU 7540296 A	03-07-1997
US 4481382 A	06-11-1984	NONE	
US 4875229 A	17-10-1989	CA 1287374 A	06-08-1991
WO 9726744 A	24-07-1997	AU 1363697 A	11-08-1997
		CA 2243244 A	24-07-1997
		EP 0875109 A	04-11-1998
GB 2289555 A	22-11-1995	FI 942334 A	20-11-1995
DE 19528424 A	21-11-1996	WO 9635288 A	07-11-1996
		EP 0824820 A	25-02-1998
US 5550754 A	27-08-1996	NONE	
EP 0665655 A	02-08-1995	US 5446783 A	29-08-1995
		JP 7226807 A	22-08-1995
US 5661788 A	26-08-1997	KR 135777 B	27-04-1998
		CN 1136753 A	27-11-1996

Electronic Acknowledgement Receipt

EFS ID:	18670421
Application Number:	13438754
International Application Number:	
Confirmation Number:	3525
Title of Invention:	DATA STORAGE AND ACCESS SYSTEMS
First Named Inventor/Applicant Name:	Patrick Sandor Racz
Customer Number:	42624
Filer:	Michael R. Casey
Filer Authorized By:	
Attorney Docket Number:	4037-0003
Receipt Date:	03-APR-2014
Filing Date:	03-APR-2012
Time Stamp:	19:51:34
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	Transmittal Letter	20140403_IDS_Transmittal.pdf	134802 97fa1321981eec11a011182407bfe7562f998cd7	no	2

Warnings:

Information:

2	Information Disclosure Statement (IDS) Form (SB08)	20140403_1449.pdf	62744 448695ffb20d4c8842b6a3e2016fe2d5c19c1ea0	no	4
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	Foreign Reference	F0000.pdf	14392701 a67842f2e2c965e0881c38029e2747c7153fd2c7	no	142
Warnings:					
Information:					
4	Foreign Reference	F0001.pdf	2396771 33806c44e85b0dc6c9b39957268fc3ac0035bffe	no	13
Warnings:					
Information:					
5	Foreign Reference	F0002.pdf	1413146 6b163dba9cb5ea9b6adedf8bc5bfd123d6ba661b	no	27
Warnings:					
Information:					
6	Non Patent Literature	NP0000.pdf	4818094 526b7a17ab5733befd244d85ea649504c0d84daf	no	244
Warnings:					
Information:					
7	Non Patent Literature	NP0001.pdf	4077659 ade813838fa779322667769cb182e026706a8a87	no	229
Warnings:					
Information:					
8	Non Patent Literature	NP0002.pdf	4409368 f7ab0429d4265416e9f10e558c66129779177a42	no	245
Warnings:					
Information:					
9	Non Patent Literature	NP0003.pdf	4593970 cded3d9e471db72858be14d57e266c5b7ab1a0c3	no	254
Warnings:					
Information:					
10	Non Patent Literature	NP0004.pdf	7005801 eadb8ed8802a60ad4ad998c42b1cfdd959ce7294	no	396

Warnings:					
Information:					
11	Non Patent Literature	NP0005.pdf	4183215 35dc3c2a8b25ae4eec2c2641fe85cb2e8a66cdc5	no	216
Warnings:					
Information:					
12	Non Patent Literature	NP0006.pdf	11419394 d3b48d09c49db792d3df9d33e7da5cf9a3546f75	no	190
Warnings:					
Information:					
13	Non Patent Literature	NP0007.pdf	12705124 c2ad42e4bae42a481d37f703e1c7cd6525f61f6b	no	217
Warnings:					
Information:					
14	Non Patent Literature	NP0008.pdf	2171808 2afa487f9fee72569ca8303c8aa3a4a147c13f91	no	13
Warnings:					
Information:					
			Total Files Size (in bytes):	73784597	

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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re PATENT Application of:		Confirmation No.:	3525
Patrick Sandor Racz		Attorney Docket:	4037-0003
Appl. S.N.:	13/438,754	Group Art Unit:	2887
Filing Date:	April 3, 2012	Examiner:	Le, Thien Minh
Title:	DATA STORAGE AND ACCESS SYSTEMS	Date:	04/03/2014

INFORMATION DISCLOSURE STATEMENT

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

Commissioner:

Pursuant to 37 C.F.R. § 1.56, the attention of the Patent and Trademark Office is hereby directed to the reference(s) listed on the attached PTO-1449. One copy of each non-U.S. Patent reference is attached. It is respectfully requested that the information be expressly considered during the prosecution of this application, and that the reference(s) be made of record therein and appear among the "References Cited" on any patent to issue therefrom.

The submission of any document herewith, which is not a statutory bar, is not intended that any such document constitutes prior art against any of the claims of the present application or is considered to be material to patentability as defined in 37 C.F.R. § 1.56(b). Applicants do not waive any rights to take any action which would be appropriate to antedate or otherwise remove as a competent reference against the claims of the present application.

This Information Disclosure Statement (IDS) is being filed within three (3) months of the U.S. filing date OR before the mailing date of a first Office Action on the merits after an RCE. No certification or fee is required.

CHARGE STATEMENT: Deposit Account No. 501860, order no. **4037-0003**.

The Commissioner is hereby authorized to charge any fee specifically authorized hereafter, or any missing or insufficient fee(s) filed, or asserted to be filed, or which should have been filed herewith or concerning any paper filed hereafter, and which may be required under Rules 16-18 (missing or insufficiencies only) now or hereafter relative to this application and the resulting Official Document under Rule 20, or credit any overpayment, to our Accounting/Order Nos. shown above, for which purpose a duplicate copy of this sheet is attached

This CHARGE STATEMENT does not authorize charge of the issue fee until/unless an issue fee transmittal sheet is filed.

CUSTOMER NUMBER
42624

Davidson Berquist Jackson & Gowdey LLP
4300 Wilson Blvd., 7th Floor,
Arlington Virginia 22203
Main: (703) 894-6400 ● FAX: (703) 894-6430

Respectfully submitted,

By: / Michael R. Casey /

Michael R. Casey
Registration No.: 40,294

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 1 of 6	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	1-1	"Delkin Breaks 400MB Flash Memory Barrier for MP3 Players", Richard Menta, MP3newswire.net, October 2, 2000, available at http://www.mp3newswire.net/stories/2000/delkin2.html	
	1-2	"First MP3 Portable with 128MB Built-in Flash Review: The Soul", Richard Menta, MP3newswire.net, December 7, 1999, available at http://www.mp3newswire.net/stories/2000/soul.html	
	1-3	"New_RaveMP_MP3_Players Debut", Richard Menta, MP3Newswire.net, June 30, 2000, available at http://www.mp3newswire.net/stories/2000/drive.html	
	1-4	"Pirates Beware: We're Watching", Wired.com, January 3, 2001, available at http://archive.wired.com/science/discoveries/news/2001/01/40866	
	1-5	"SDMI Executive Director Challenges MP3.com Editorial", Rich Menta, November 4, 1999, available at http://www.mp3newswire.net/stories/sdmi.html	
	1-6	"SDMI: Divide or Conquer?", Wired.com, November 18, 1999, available at http://archive.wired.com/science/discoveries/news/1999/11/32513	
	1-7	"Smart Cards: A Case Study", IBM International Technical Support Organization, October 1998, available at http://www.redbooks.ibm.com/redbooks/pdfs/sg245239.pdf	

Examiner Signature		Date Considered	

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 2 of 6	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	2-1	"Smart cards: A primer", JavaWorld, December 1997, available at http://www.javaworld.com/article/2077101/learnjava/smart-cards--a-primer.html	
	2-2	"The End of SDMI", Eric Scheirer, Technology Correspondent, MP3.com, October 15, 1999, available at ftp://ftp.gwdg.de/pub/eff/cafe/scheirer1.html	
	2-3	"Web Sites and Recording Labels at Impasse on Fees", Richtel, Matt, The New York Times, November 29, 1999, available at http://www.nytimes.com/library/tech/99/11/biztech/articles/29tune.html	
	2-4	American Heritage College Dictionary (3rd Edition 1997): (definition of "payment" and "pay")	
	2-5	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00110 (U.S. Pat. No. 8,336,772), dated April 3, 2014 (including Declarations)	
	2-6	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00111 (U.S. Pat. No. 8,336,772), dated April 3, 2014 (including Declarations)	
	2-7	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00112 (U.S. Pat. No. 7,942,317), dated April 3, 2014 (including Declarations)	

Examiner Signature		Date Considered	

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

FORM PTO-1449 (modified)

Sheet 3 of 6

Application No.	13/438,754
Filing Date	April 3, 2012
First Named Inventor	Patrick Sandor Racz
Group Art Unit	2887
Examiner Name	Le, Thien Minh
Attorney Docket No.	4037-0003
Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	3-1	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00113 (U.S. Pat. No. 7,942,317), dated April 3, 2014 (including Declarations)	
	3-2	Freedman & Glossbrenner, The Internet Glossary and Quick Reference Guide, 1998, pgs. 79, 246.	
	3-3	Hartel, "Formalizing the Safety of Java, the Java Virtual Machine". ACM Comp. Surv. Vol.33 No.4, Dec. 2001 p517-558.	
	3-4	Herreweghen and Wille, Risks and Potentials of using EMV for Internet Payments, USENIX Workshop on Smartcard Technology, May 10-11, 1999, pp.163-173	
	3-5	IBM Dictionary of Computing, 10th Ed. 1994, pgs. 297, 533 and 637.	
	3-6	Kyu Ha Lee' et al., "AN ARCHITECTURE AND IMPLEMENTATION OF MPEG AUDIO LAYER III DECODER USING DUAL-CORE DSP." IEEE Transactions on Consumer Electronics, Vol. 47, No. 4, NOVEMBER 2001.	
	3-7	Lawrence Haynes, "Theatre Medical Data Store." IEEE (pub), 1998.	

Examiner Signature		Date Considered	

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
FORM PTO-1449 (modified)

Sheet 4 of 6

Application No.	13/438,754
Filing Date	April 3, 2012
First Named Inventor	Patrick Sandor Racz
Group Art Unit	2887
Examiner Name	Le, Thien Minh
Attorney Docket No.	4037-0003
Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	4-1	McGraw-Hill Dictionary of Scientific and Technical Terms (4th Edition 1989) (definition of “data carrier”)	
	4-2	Merriam-Webster Collegiage Dictionary (10th Edition 1997) : (definition of “pay” and “payment”)	
	4-3	Microsoft Press Computer User’s Dictionary, 1998. P. 157, 227, 367	
	4-4	Scheuermann, D., "The Smart Card as a mobile security device." Ch 4, Chris Mitchell (ed.), Security for Mobility. Institution of Engineering and Technology (pub), 2004	
	4-5	Scheuermann, D., "The Smart Card as a mobile security device." Security for Mobility. Electronics and Communications Engineering Journal, Vol.14 No. 5, Oct 2002	
	4-6	Smart Cards: Seizing Strategic Business Opportunities Smart Card Forum; Hardcover (including but not limited to definitions in “Glossary of Terms”)	
	4-7	Smartflash LLC et al. v. Apple Inc. et al., Civil Action 6:13-CV-00447-MHS-KNM, Defendants' Preliminary Claim Constructions and Extrinsic Evidence, dated April 1, 2014	

Examiner Signature		Date Considered	

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 5 of 6	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	5-1	Smartflash LLC et al. v. Apple Inc. et al., Civil Action 6:13-CV-00447-MHS-KNM, Plaintiffs' Preliminary Claim Constructions and Extrinsic Evidence, dated April 1, 2014	
	5-2	Smartflash LLC et al. v. Samsung Electronics et al., Civil Action 6:13-CV-00448-MHS-KNM, Defendants' Preliminary Claim Constructions and Extrinsic Evidence, dated April 1, 2014	
	5-3	Smith,M., "Smart cards: Integrating for portable complexity", IEEE Computer, 1998	
	5-4	Sony Press Release, "Sony Announces 'Memory Stick' Recordable IC Memory Card Products, July 30, 1998, available at http://www.sony.net/SonyInfo/News/Press_Archive/199807/98-067/	
	5-5	The IEEE Standard Dictionary of Electrical and Electronics Terms (6th Edition 1996): (definitions of "data carrier" and "data medium")	
	5-6	THE JAVA CARD 3 PLATFORM, White Paper, August 2008, Oracle Corp.	
	5-7	The New IEEE Standard Dictionary of Electrical and Electronic Terms (5th Edition 1993), pgs. 305, 533, 1011, 1252	

Examiner Signature		Date Considered	

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 6 of 6	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	6-1	Webster's New World Dictionary, Third College Edition, 1991, pg. 1173	
	6-2		
	6-3		
	6-4		
	6-5		
	6-6		
	6-7		

Examiner Signature		Date Considered	

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

Electronic Acknowledgement Receipt

EFS ID:	18733464
Application Number:	13438754
International Application Number:	
Confirmation Number:	3525
Title of Invention:	DATA STORAGE AND ACCESS SYSTEMS
First Named Inventor/Applicant Name:	Patrick Sandor Racz
Customer Number:	42624
Filer:	Michael R. Casey
Filer Authorized By:	
Attorney Docket Number:	4037-0003
Receipt Date:	10-APR-2014
Filing Date:	03-APR-2012
Time Stamp:	18:34:02
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	Information Disclosure Statement (IDS) Form (SB08)	20140410_1449.pdf	93997 42cffc8391268689ab90a88cd028188f8609f3bf	no	6

Warnings:

Information:

This is not an USPTO supplied IDS fillable form

2	Non Patent Literature	NP0000.pdf	314438 9fe00d99c6eb6fad4f67c383c7e77615e262faf6	no	2
Warnings:					
Information:					
3	Non Patent Literature	NP0001.pdf	431957 2a65b4f5c4640aed4f6d4bd97463cb5fa770ae2c	no	4
Warnings:					
Information:					
4	Non Patent Literature	NP0002.pdf	283343 3cd61f0a003202445c4321cb08ece1754d757320	no	2
Warnings:					
Information:					
5	Non Patent Literature	NP0003.pdf	176727 284814f6576cde31d0a77eef2617c39ddf079947	no	1
Warnings:					
Information:					
6	Non Patent Literature	NP0004.pdf	297109 a82b0e894d5623bbc532eeb2d801bbc5a77e1481	no	3
Warnings:					
Information:					
7	Non Patent Literature	NP0005.pdf	202677 4d9f4bf4e2e288ea7d560dc4d7a0d3642ec7c855	no	1
Warnings:					
Information:					
8	Non Patent Literature	NP0006.pdf	1383418 28744efd52f2502e69fb3b2e090c5bd987d345d5	no	234
Warnings:					
Information:					
9	Non Patent Literature	NP0007.pdf	894522 352ffe3a4063e47bac91386db795891bd9eeb13e	no	8
Warnings:					
Information:					
10	Non Patent Literature	NP0008.pdf	539163 2ca66a31ed0bd9a888c6336e493a90d355d7079b	no	5

Warnings:					
Information:					
11	Non Patent Literature	NP0009.pdf	920252 b1e9496bbff3e1a601c832ab24a8a9cfbe655111	no	5
Warnings:					
Information:					
12	Non Patent Literature	NP0011.pdf	5586135 70a2d74906de93056ea3f56ef904afc510bda709	no	283
Warnings:					
Information:					
13	Non Patent Literature	NP0012.pdf	5557450 62f539627cc2b2b474c1ff1d33c731da59b89802	no	292
Warnings:					
Information:					
14	Non Patent Literature	NP0013.pdf	4356651 3c555f6bc7893a648cbb25805f6a6d53102ef558	no	224
Warnings:					
Information:					
15	Non Patent Literature	NP0014.pdf	4358547 6e4a36616b97b18654db898a2387b497d2a87ed8	no	231
Warnings:					
Information:					
16	Non Patent Literature	NP0016.pdf	354987 f3a74c98e015598ac708454da4489353ccbff910	no	42
Warnings:					
Information:					
17	Non Patent Literature	NP0017.pdf	2444198 692ed0bb5c57639ade7eb7324ad78ea7416b2cab	no	13
Warnings:					
Information:					
18	Non Patent Literature	NP0019.pdf	1066401 7cb4cb934d92e3045b3e3df3c99b2fc096eb2a92	no	6
Warnings:					
Information:					
19	Non Patent Literature	NP0020.pdf	625591 28e4bc41bc5c0a8e2dc314deb233c08aabb8c17d	no	8

Warnings:					
Information:					
20	Non Patent Literature	NP0025.pdf	1217551 ec8b1eb825c859ae5ee87e7efca6a202e0ba3038	no	6
Warnings:					
Information:					
21	Non Patent Literature	NP0026.pdf	11900023 43822eba914865fa60be5301af07792f54b61b58	no	18
Warnings:					
Information:					
22	Non Patent Literature	NP0027.pdf	4557418 d9442b8c8ac9f820c674ba49af4e07a446dacc65	no	51
Warnings:					
Information:					
23	Non Patent Literature	NP0029.pdf	8212293 a321e145244d6e34f12f0cd733d1e9379cf346c8	no	87
Warnings:					
Information:					
24	Non Patent Literature	NP0030.pdf	751649 d36cc325728bda2b485a07f3257dd74335c70af2	no	4
Warnings:					
Information:					
25	Non Patent Literature	NP0031.pdf	340373 9b76c14917e24c064a29226faaac0a7d82bf6825	no	3
Warnings:					
Information:					
26	Non Patent Literature	NP0033.pdf	2757271 ee7fbb8f7e0ba351c9dd0630d7861c536b5a1138	no	34
Warnings:					
Information:					
27	Non Patent Literature	NP0015.pdf	1150937 b857ff255746b652ebf63063bee50782a132e746	no	4
Warnings:					
Information:					
28	Non Patent Literature	NP0010.pdf	4325283 4953e8f4c66cb9a212610c0741f90d431e82d975	no	4

Warnings:					
Information:					
29	Non Patent Literature	NP0018.pdf	1370708 bb1ab24af05e140541a2f063e17a3e45033922fd	no	5
Warnings:					
Information:					
30	Non Patent Literature	NP0021.pdf	856948 fba66de6b620b7ea02a511e4035b6b53eba78052	no	3
Warnings:					
Information:					
31	Non Patent Literature	NP0022.pdf	5093250 45b9beb6af4d2419aaa0794a8983352923ab5f27	no	4
Warnings:					
Information:					
32	Non Patent Literature	NP0023.pdf	1435625 8ebb9c2d903967e08cccebca5c3c8afbd008215a	no	5
Warnings:					
Information:					
33	Non Patent Literature	NP0024.pdf	712970 219cdf487c058b9788cfdba3bca5117241b63e52	no	12
Warnings:					
Information:					
34	Non Patent Literature	NP0032.pdf	1093416 8d25969f357dda921b2683976dc772b09f0dcf5f	no	3
Warnings:					
Information:					
35	Non Patent Literature	NP0034.pdf	1852905 f56009b9be6e28e69cd6553a74f6a919c5e6bf38	no	10
Warnings:					
Information:					
36	Non Patent Literature	NP0035.pdf	1117024 886ec505b0eb19a290083548dd8c6ecd13691d9d	no	3
Warnings:					
Information:					
37	Non Patent Literature	NP0028.pdf	17261151 b2adc11eacb54914ffc8d83d4c1488b15275f139	no	375

Warnings:	
Information:	
Total Files Size (in bytes):	95894358

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.

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

NOTICE OF ALLOWANCE AND FEE(S) DUE

42624 7590 04/11/2014
DAVIDSON BERQUIST JACKSON & GOWDEY LLP
4300 WILSON BLVD., 7TH FLOOR
ARLINGTON, VA 22203

EXAMINER	
LE, THIEN MINH	
ART UNIT	PAPER NUMBER
2887	

DATE MAILED: 04/11/2014

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/438,754	04/03/2012	Patrick Sandor Racz	4037-0003	3525

TITLE OF INVENTION: DATA STORAGE AND ACCESS SYSTEMS

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	$0	$0	$1780	$0	07/11/2014

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.

If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.

If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".

For purposes of this notice, small entity fees are 1/2 the amount of undiscounted fees, and micro entity fees are 1/2 the amount of small entity fees.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

42624 7590 04/11/2014
DAVIDSON BERQUIST JACKSON & GOWDEY LLP
4300 WILSON BLVD., 7TH FLOOR
ARLINGTON, VA 22203

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/438,754	04/03/2012	Patrick Sandor Racz	4037-0003	3525

TITLE OF INVENTION: DATA STORAGE AND ACCESS SYSTEMS

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	$0	$0	$1780	$0	07/11/2014

EXAMINER	ART UNIT	CLASS-SUBCLASS
LE, THIEN MINH	2887	235-380000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

❑ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

❑ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) The names of up to 3 registered patent attorneys or agents OR, alternatively,

(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 ______________________

2 ______________________

3 ______________________

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ❑ Individual ❑ Corporation or other private group entity ❑ Government

4a. The following fee(s) are submitted:

❑ Issue Fee

❑ Publication Fee (No small entity discount permitted)

❑ Advance Order - # of Copies ______________

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**

❑ A check is enclosed.

❑ Payment by credit card. Form PTO-2038 is attached.

❑ The Director is hereby authorized to charge the required fee(s), any deficiency, or credits any overpayment, to Deposit Account Number ________ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)

❑ Applicant certifying micro entity status. See 37 CFR 1.29

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

❑ Applicant asserting small entity status. See 37 CFR 1.27

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

❑ Applicant changing to regular undiscounted fee status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature ______________________ Date ______________________

Typed or printed name ______________________ Registration No. ______________________

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/438,754	04/03/2012	Patrick Sandor Racz	4037-0003	3525

42624 7590 04/11/2014

DAVIDSON BERQUIST JACKSON & GOWDEY LLP
4300 WILSON BLVD., 7TH FLOOR
ARLINGTON, VA 22203

EXAMINER	
LE, THIEN MINH	
ART UNIT	PAPER NUMBER
2887	

DATE MAILED: 04/11/2014

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)

(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 0 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 0 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

OMB Clearance and PRA Burden Statement for PTOL-85 Part B

The Paperwork Reduction Act (PRA) of 1995 requires Federal agencies to obtain Office of Management and Budget approval before requesting most types of information from the public. When OMB approves an agency request to collect information from the public, OMB (i) provides a valid OMB Control Number and expiration date for the agency to display on the instrument that will be used to collect the information and (ii) requires the agency to inform the public about the OMB Control Number's legal significance in accordance with 5 CFR 1320.5(b).

The information collected by PTOL-85 Part B is required by 37 CFR 1.311. 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, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450. 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.

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 disclosure of these records is required by the Freedom of Information Act.
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 inspection 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.

Notice of Allowability	**Application No.** 13/438,754	**Applicant(s)** RACZ ET AL.	
	Examiner Thien M. Le	**Art Unit** 2887	**AIA (First Inventor to File) Status** No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--
All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *a request for RCE filed on 3/21/2014*.

   □ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on______.

2. □ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

3. ☒ The allowed claim(s) is/are *40-60 and 64-70*. As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov .

4. □ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   Certified copies:

   a) □ All b) □ Some *c) □ None of the:

   1. □ Certified copies of the priority documents have been received.
   2. □ Certified copies of the priority documents have been received in Application No. _____ .
   3. □ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. □ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.

   □ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. □ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. □ Notice of References Cited (PTO-892)
2. ☒ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date 3/21/2014; 12/31/2013; 4/3/2014
3. □ Examiner's Comment Regarding Requirement for Deposit of Biological Material
4. □ Interview Summary (PTO-413), Paper No./Mail Date _____ .
5. □ Examiner's Amendment/Comment
6. ☒ Examiner's Statement of Reasons for Allowance
7. □ Other _____.

DETAILED ACTION

The present application is being examined under the pre-AIA first to invent provisions.

The information disclosure statement filed on 3/21/2014 has been entered. The IDS filed on 12/31/2013 has been reconsidered to provide the date to the NPL document on page 4. Claims 40-60 and 64-70 are presented for examination.

Allowable Subject Matter

Claims 40-60 and 64-70 are allowed.

The following is a statement of reasons for the indication of allowable subject matter: The prior art fails to disclose a handheld multimedia terminal comprising: a wireless interface, a non-volatile memory, a program store, a processor, a user interface, a display, and having the processor control codes as recited in claims 40. Claim 41 recites a data supply server having similar limitations as recited in claim 40. The prior art also fails to disclose a computer system and a method of providing and downloading multimedia contents having limitations as recited in claims 50, 57 and 64.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Thien M. Le whose telephone number is (571)272-2396. The examiner can normally be reached on Monday - Friday from 7:30am - 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Steve S. Paik can be reached on (571) 272-2404. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Thien M. Le/
Primary Examiner, Art Unit 2887

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 1 of 4	Confirmation No.	3525

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.	Document No.	Publication/ Issue Date	Name of Patentee or Applicant of Cited Document
	1-1	US-4200770	Apr-80	Hellman et al.
	1-2	US-4218582	Aug-80	Hellman et al.
	1-3	US-4272810	Jun-81	Gates et al.
	1-4	US-4405829	Sep-83	Rivest et al.
	1-5	US-4424414	Jan-84	Hellman et al.
	1-6	US-4463387	Jul-84	Hashimoto et al.
	1-7	US-4528643	Jul-85	Freeny, Jr.
	1-8	US-4731840	Mar-88	Mniszewski et al.
	1-9	US-4757534	Jul-88	Matyas et al.
	1-10	US-4782529	Nov-88	Shima
	1-11	US-4796220	Jan-89	Wolfe
	1-12	US-4803725	Feb-89	Horne et al.
	1-13	US-4809327	Feb-89	Shima
	1-14	US-4825306	Apr-89	Robers
	1-15	US-4868687	Sep-89	Penn et al.
	1-16	US-4868877	Sep-89	Fischer
	1-17	US-4878246	Oct-89	Pastor et al.
	1-18	US-4879747	Nov-89	Leighton et al.
	1-19	US-4905163	Feb-90	Garber et al.
	1-20	US-4926479	May-90	Goldwasser et al.
	1-21	US-4944006	Jul-90	Citta et al.
	1-22	US-4995082	Feb-91	Schnorr
	1-23	US-5005200	Apr-91	Fischer
	1-24	US-5130792	Jul-92	Tindell et al.
	1-25	US-5159634	Oct-92	Reeds, III
	1-26	US-5191573	Mar-93	Hair

Examiner Signature	/Thien Le/	Date Considered	04/08/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 2 of 4		
	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.	Document No.	Publication/ Issue Date	Name of Patentee or Applicant of Cited Document
	2-1	US-5214702	May-93	Fischer
	2-2	US-5220604	Jun-93	Gasser et al.
	2-3	US-5224163	Jun-93	Gasser et al.
	2-4	US-5224166	Jun-93	Hartman, Jr.
	2-5	US-5260788	Nov-93	Takano et al.
	2-6	US-5261002	Nov-93	Perlman et al.
	2-7	US-5276901	Jan-94	Howell et al.
	2-8	US-5315658	May-94	Micali
	2-9	US-5319705	Jun-94	Halter et al.
	2-10	US-5347580	Sep-94	Molva et al.
	2-11	US-5355302	Oct-94	Martin et al.
	2-12	US-5369705	Nov-94	Bird et al.
	2-13	US-5371794	Dec-94	Diffie et al.
	2-14	US-5388211	Feb-95	Hornbuckle
	2-15	US-5412717	May-95	Fischer
	2-16	US-5420927	May-95	Micali
	2-17	US-5497421	Mar-96	Kaufman et al.
	2-18	US-5509071	Apr-96	Petrie, Jr. et al.
	2-19	US-5519778	May-96	Leighton et al.
	2-20	US-5537475	Jul-96	Micali
	2-21	US-5557541	Sep-96	Schulhof et al.
	2-22	US-5581479	Dec-96	McLaughlin et al.
	2-23	US-5588060	Dec-96	Aziz
	2-24	US-5592664	Jan-97	Starkey
	2-25	US-5604804	Feb-97	Micali
	2-26	US-5606617	Feb-97	Brands

Examiner Signature	/Thien Le/	Date Considered	04/08/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 3 of 4	Confirmation No.	3525

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.	Document No.	Publication/ Issue Date	Name of Patentee or Applicant of Cited Document
	3-1	US-5636139	Jun-97	McLaughlin et al.
	3-2	US-5646992	Jul-97	Subler et al.
	3-3	US-5646998	Jul-97	Stambler
	3-4	US-5649187	Jul-97	Hornbuckle
	3-5	US-5666420	Sep-97	Micali
	3-6	US-5673316	Sep-97	Auerbach et al.
	3-7	US-5675734	Oct-97	Hair
	3-8	US-5706347	Jan-98	Burke et al.
	3-9	US-5710887	Jan-98	Chelliah et al.
	3-10	US-5745574	Apr-98	Muftic
	3-11	US-5765152	Jun-98	Erickson
	3-12	US-5796841	Aug-98	Cordery et al.
	3-13	US-5864620	Jan-99	Pettitt
	3-14	US-5892900	Apr-99	Ginter et al.
	3-15	US-5915025	Dec-99	Taguchi et al.
	3-16	US-5925127	Jul-99	Ahmad
	3-17	US-5982892	Nov-99	Hicks et al.
	3-18	US-5991399	Nov-99	Graunke et al.
	3-19	US-5999629	Dec-99	Heer et al.
	3-20	US-6064739	May-00	Davis
	3-21	US-6098056	Aug-00	Rusnak et al.
	3-22	US-6275936	Aug-01	Kyojima et al.
	3-23			
	3-24			
	3-25			
	3-26			

Examiner Signature	/Thien Le/	Date Considered	04/08/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 4 of 4	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	4-1	Smartflash LLC v. Apple Inc., Civil Action 6:13-CV-00447-MHS-KNM, Subpoena to Testify 5/2013	
	4-2		
	4-3		
	4-4		
	4-5		
	4-6		
	4-7		

Examiner Signature	/Thien Le/	Date Considered	04/08/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

Index of Claims	**Application/Control No.** 13438754	**Applicant(s)/Patent Under Reexamination** HULST ET AL.
	Examiner THIEN M LE	**Art Unit** 2887

✓	**Rejected**		**-**	**Cancelled**		**N**	**Non-Elected**		**A**	**Appeal**
=	**Allowed**		**÷**	**Restricted**		**I**	**Interference**		**O**	**Objected**

□ **Claims renumbered in the same order as presented by applicant** □ **CPA** ☒ **T.D.** □ **R.1.47**

CLAIM		**DATE**								
Final	Original	12/21/2012	04/01/2013	09/04/2013	01/15/2014	03/31/2014				
1	40	✓	=	=	=	=				
5	41	✓	=	=	=	=				
6	42	✓	=	=	=	=				
7	43	✓	=	=	=	=				
8	44	✓	=	=	=	=				
9	45	✓	=	=	=	=				
10	46	✓	=	=	=	=				
11	47	✓	=	=	=	=				
12	48	✓	=	=	=	=				
13	49	✓	=	=	=	=				
14	50	✓	=	=	=	=				
15	51	✓	=	=	=	=				
16	52	✓	=	=	=	=				
17	53	✓	=	=	=	=				
18	54	✓	=	=	=	=				
19	55	✓	=	=	=	=				
20	56	✓	=	=	=	=				
21	57	✓	=	=	=	=				
22	58	✓	=	=	=	=				
23	59	✓	=	=	=	=				
24	60	✓	=	=	=	=				
	61	✓	-	-	-	-				
	62	✓	-	-	-	-				
	63	✓	-	-	-	-				
25	64	✓	=	=	=	=				
26	65	✓	=	=	=	=				
27	66	✓	=	=	=	=				
28	67	✓	=	=	=	=				
2	68		=	=	=	=				
3	69		=	=	=	=				
4	70		=	=	=	=				
	71									
	72									
	73									
	74									
	75									

Index of Claims	**Application/Control No.** 13438754	**Applicant(s)/Patent Under Reexamination** HULST ET AL.
	Examiner THIEN M LE	**Art Unit** 2887

✓	**Rejected**		-	**Cancelled**		N	**Non-Elected**		A	**Appeal**
=	**Allowed**		÷	**Restricted**		I	**Interference**		O	**Objected**

□ **Claims renumbered in the same order as presented by applicant** □ **CPA** ☒ **T.D.** □ **R.1.47**

CLAIM		DATE								
Final	Original	12/21/2012	04/01/2013	09/04/2013	01/15/2014	03/31/2014				
	76									
	77									
	78									
	79									
	80									
	81									
	82									
	83									
	84									
	85									
	86									
	87									
	88									
	89									
	90									
	91									
	92									
	93									
	94									
	95									
	96									
	97									
	98									
	99									
	100									
	101									
	102									
	103									
	104									
	105									
	106									

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

FORM PTO-1449 (modified)

Sheet 1 of 4

Application No.	13/438,754
Filing Date	April 3, 2012
First Named Inventor	Patrick Sandor Racz
Group Art Unit	2887
Examiner Name	Le, Thien Minh
Attorney Docket No.	4037-0003
Confirmation No.	3525

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.	Document No.	Publication/ Issue Date	Name of Patentee or Applicant of Cited Document
	1-1	US-4878245	1989/10/31	Bradley et al.
	1-2	US-4999806	1991/03/12	Chernow
	1-3			
	1-4			
	1-5			
	1-6			
	1-7			
	1-8			
	1-9			
	1-10			
	1-11			
	1-12			
	1-13			
	1-14			
	1-15			
	1-16			
	1-17			
	1-18			
	1-19			
	1-20			
	1-21			
	1-22			
	1-23			
	1-24			
	1-25			
	1-26			

Examiner Signature	/Thien Le/	Date Considered	04/08/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 2 of 4	Confirmation No.	3525

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.	Document No.	Publication Date	Name of Patentee or Applicant of Cited Document	Notes
	2-1	JP 10-269289	1998/10/09	Kouichi	
	2-2	JP11-164058	1999/06/18	Sato et al.	
	2-3	WO 99/43136	1999/08/26	Rydbeck et al.	
	2-4				
	2-5				
	2-6				
	2-7				
	2-8				
	2-9				
	2-10				
	2-11				
	2-12				
	2-13				
	2-14				
	2-15				
	2-16				
	2-17				
	2-18				
	2-19				
	2-20				
	2-21				
	2-22				
	2-23				
	2-24				
	2-25				
	2-26				

Examiner Signature	/Thien Le/	Date Considered	04/08/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 3 of 4	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	3-1	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00102 (U.S. Pat. No. 8,118,221), dated March 31, 2014 (including Declarations)	
	3-2	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00103 (U.S. Pat. No. 8,118,221), dated March 31, 2014 (including Declarations)	
	3-3	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00104 (U.S. Pat. No. 7,334,720), dated March 31, 2014 (including Declarations)	
	3-4	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00105 (U.S. Pat. No. 7,334,720), dated March 31, 2014 (including Declarations)	
	3-5	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00106 (U.S. Pat. No. 8,033,458), dated March 31, 2014 (including Declarations)	
	3-6	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00107 (U.S. Pat. No. 8,033,458), dated March 31, 2014 (including Declarations)	
	3-7	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00108 (U.S. Pat. No. 8,061,598), dated April 1, 2014 (including Declarations)	

Examiner Signature	/Thien Le/	Date Considered	04/08/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 4 of 4	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	4-1	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00109 (U.S. Pat. No. 8,061,598), dated April 1, 2014 (including Declarations)	
	4-2	Eberhard von Faber, Robert Hammelrath, and Franz-Peter Heider, "The Secure Distribution of Digital Contents," IEEE (1997)	
	4-3		
	4-4		
	4-5		
	4-6		
	4-7		

Examiner Signature	/Thien Le/	Date Considered	04/08/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

Issue Classification	**Application/Control No.** 13438754	**Applicant(s)/Patent Under Reexamination** HULST ET AL.
	Examiner THIEN M LE	**Art Unit** 2887

CPC						
Symbol					**Type**	**Version**
			/			
			/			
			/			
			/			
			/			
			/			
			/			
			/			
			/			
			/			
			/			
			/			
			/			
			/			
			/			

CPC Combination Sets								
Symbol					**Type**	**Set**	**Ranking**	**Version**
			/					
			/					

NONE (Assistant Examiner)	(Date)	**Total Claims Allowed:** 28	
/THIEN M LE/ Primary Examiner.Art Unit 2887 (Primary Examiner)	03/31/2014 (Date)	O.G. Print Claim(s) 1	O.G. Print Figure 5

Issue Classification	**Application/Control No.** 13438754	**Applicant(s)/Patent Under Reexamination** HULST ET AL.
	Examiner THIEN M LE	**Art Unit** 2887

US ORIGINAL CLASSIFICATION	
CLASS	**SUBCLASS**
235	380

CROSS REFERENCE(S)

CLASS	**SUBCLASS (ONE SUBCLASS PER BLOCK)**				
235	382				

INTERNATIONAL CLASSIFICATION									
CLAIMED					**NON-CLAIMED**				
G	0	6	K	5 / 00 (2006.01.01)					

NONE (Assistant Examiner)	(Date)	**Total Claims Allowed:** 28	
/THIEN M LE/ Primary Examiner.Art Unit 2887 (Primary Examiner)	03/31/2014 (Date)	O.G. Print Claim(s) 1	O.G. Print Figure 5

Issue Classification	**Application/Control No.** 13438754	**Applicant(s)/Patent Under Reexamination** HULST ET AL.
	Examiner THIEN M LE	**Art Unit** 2887

□ **Claims renumbered in the same order as presented by applicant** □ **CPA** ☒ **T.D.** □ **R.1.47**

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	40	20	56												
5	41	21	57												
6	42	22	58												
7	43	23	59												
8	44	24	60												
9	45	25	64												
10	46	26	65												
11	47	27	66												
12	48	28	67												
13	49	2	68												
14	50	3	69												
15	51	4	70												
16	52														
17	53														
18	54														
19	55														

NONE (Assistant Examiner)	(Date)	**Total Claims Allowed:** 28	
/THIEN M LE/ Primary Examiner.Art Unit 2887 (Primary Examiner)	03/31/2014 (Date)	O.G. Print Claim(s) 1	O.G. Print Figure 5

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 3525

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
13/438,754	04/03/2012 RULE	235	2887	4037-0003

APPLICANTS

INVENTORS
Patrick Sandor Racz, Saint Heller, UNITED KINGDOM;
Hermen-ard Hulst, Amsterdam, NETHERLANDS;

** **CONTINUING DATA** *************************
This application is a CON of 13/212,047 08/17/2011 PAT 8336772
which is a CON of 12/943,872 11/10/2010 PAT 8118221
which is a CON of 12/014,558 01/15/2008 PAT 7942317
which is a CON of 11/336,758 01/19/2006 PAT 7334720
which is a CON of 10/111,716 09/17/2002 ABN
which is a 371 of PCT/GB2000/004110 10/25/2000

** **FOREIGN APPLICATIONS** *************************
UNITED KINGDOM 9925227.2 10/25/1999

** **IF REQUIRED, FOREIGN FILING LICENSE GRANTED** **
04/16/2012

		STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
Foreign Priority claimed ☑ Yes □ No 35 USC 119(a-d) conditions met ☑ Yes □ No Verified and Acknowledged /THIEN MINH LE/ Examiner's Signature	□ Met after Allowance Initials	UNITED KINGDOM	17	28	6

ADDRESS

DAVIDSON BERQUIST JACKSON & GOWDEY LLP
4300 WILSON BLVD., 7TH FLOOR
ARLINGTON, VA 22203
UNITED STATES

TITLE

DATA STORAGE AND ACCESS SYSTEMS

FILING FEE RECEIVED 1591	FEES: Authority has been given in Paper No.____________ to charge/credit DEPOSIT ACCOUNT No.____________ for following:	□ All Fees □ 1.16 Fees (Filing) □ 1.17 Fees (Processing Ext. of time) □ 1.18 Fees (Issue) □ Other ________________ □ Credit

BIB (Rev. 05/07).

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 1 of 28	Confirmation No.	3525

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.	Document No.	Publication/ Issue Date	Name of Patentee or Applicant of Cited Document
	1-1	US-2002/0032059	2002/03/14	Sugimura et al.
	1-2	US-4367402	1983/01/04	Giraud et al.
	1-3	US-4677657	1987/06/30	Nagata et al.
	1-4	US-4755660	1988/07/05	Nakano
	1-5	US-4822984	1989/04/18	Remery et al.
	1-6	US-4827512	1989/05/02	Hirokawa et al.
	1-7	US-4885788	1989/12/05	Takaragi et al.
	1-8	US-4887234	1989/12/12	Iijima
	1-9	US-4910393	1990/03/20	Gercekci et al.
	1-10	US-4959788	1990/09/25	Nagata et al.
	1-11	US-4968873	1990/11/06	Dethloff et al.
	1-12	US-5122643	1992/06/16	Gamou et al.
	1-13	US-5126541	1992/06/30	Shinagawa
	1-14	US-5140517	1992/08/18	Nagata et al.
	1-15	US-5185798	1993/02/09	Hamada et al.
	1-16	US-5200600	1993/04/06	Shinagawa
	1-17	US-5212369	1993/05/18	Karlisch et al.
	1-18	US-5247163	1993/09/21	Ohno et al.
	1-19	US-5252812	1993/10/12	Nakamura
	1-20	US-5276903	1994/01/04	Shinagawa
	1-21	US-5283885	1994/02/01	Hollerbauer
	1-22	US-5285055	1994/02/08	Oonakahara et al.
	1-23	US-5293424	1994/03/08	Holtey et al.
	1-24	US-5349649	1994/09/20	Iijima
	1-25	US-5365047	1994/11/15	Yamaguchi
	1-26	US-5379344	1995/01/03	Larsson et al.

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 2 of 28	Confirmation No.	3525

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.	Document No.	Publication/ Issue Date	Name of Patentee or Applicant of Cited Document
	2-1	US-5401945	1995/03/28	Buschmann et al.
	2-2	US-5420912	1995/05/30	Kopp et al.
	2-3	US-5426432	1995/06/20	Sanemitsu
	2-4	US-5442704	1995/08/15	Holtey
	2-5	US-5511000	1996/04/23	Kaloi, et al.
	2-6	US-5511023	1996/04/23	Schrenk
	2-7	US-5523794	1996/06/04	Mankovitz et al.
	2-8	US-5557679	1996/09/17	Julin et al.
	2-9	US-5606143	1997/02/25	Young
	2-10	US-5610774	1997/03/11	Hayashi et al.
	2-11	US-5636276	1997/06/03	Brugger
	2-12	US-5664228	1997/09/02	Mital
	2-13	US-5686714	1997/11/11	Abe et al.
	2-14	US-5687398	1997/11/11	Martineau
	2-15	US-5737571	1997/04/07	Fukuzumi
	2-16	US-5763869	1998/06/09	Moll et al.
	2-17	US-5802325	1998/09/01	Le Roux
	2-18	US-5825875	1998/10/20	Ugon
	2-19	US-5825882	1998/10/20	Kowalski et al.
	2-20	US-5841979	1998/11/24	Schulhof, et al.
	2-21	US-5844281	1998/12/01	Kawan et al.
	2-22	US-5856699	1999/01/05	Drupsteen et al.
	2-23	US-5892975	1999/04/06	Barnes
	2-24	US-5896507	1999/04/20	Martineau
	2-25	US-5911031	1999/06/08	Young-Man Lee
	2-26	US-5943423	1999/08/24	Sead Muftic

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 3 of 28	Confirmation No.	3525

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.	Document No.	Publication/ Issue Date	Name of Patentee or Applicant of Cited Document
	3-1	US-5960082	1999/09/28	Haenel
	3-2	US-5963980	1999/10/05	Coulier et al.
	3-3	US-5972738	1999/08/24	Cesarie et al.
	3-4	US-5995965	1999/11/30	Bettina Experton
	3-5	US-6003113	1999/12/14	Hoshino
	3-6	US-6005942	1999/12/21	Chan et al.
	3-7	US-6032857	2000/03/07	Kitagawa et al.
	3-8	US-6289711 B1	2004/12/07	Kwok et al
	3-9	US-6314409 B2	2001/11/06	Schneck, et al.
	3-10	US-6449684	2002/09/10	MacSmith et al.
	3-11	US-6532518	2003/03/11	MacSmith et al.
	3-12	US-6697944 B1	2004/02/24	Jones et al.
	3-13	US-6880761 B1	2005/04/19	Ritter et al.
	3-14	US-8033458	2011/10/11	Racz
	3-15	US-8061598	2011/11/22	Racz
	3-16	US-8118221	2012/02/21	Racz
	3-17	US-8336772	2012/12/25	Racz
	3-18			
	3-19			
	3-20			
	3-21			
	3-22			
	3-23			
	3-24			
	3-25			
	3-26			

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 4 of 28	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.	Document No.	Publication Date	Name of Patentee or Applicant of Cited Document	Notes
	4-1	CA 2294721	1998/12/23	Ritter et al.	
	4-2	EP 0159539 B1	1990/09/12	Siemens Aktiengesellshaft	PT/SOR
	4-3	EP 0162221 B1	1992/11/11	Yoshida et al.	
	4-4	EP 0275510 B1	1992/10/21	Abraham et al.	
	4-5	EP 0328289 B2	1998/03/04	Shinagawa	
	4-6	EP 0354793 B1	1996/10/23	Shinagawa	
	4-7	EP 0536792 B1	1998/03/18	Hayashi et al.	
	4-8	EP 0809221	1997/11/26	Poggio	
	4-9	EP 0819287 B1	1998/01/21	Heyns et al.	
	4-10	EP 0913789 A2	1999/05/06	Neilsen	
	4-11	EP 0949595 A2	1999/10/13	Pan et al.	
	4-12	EP 1003135 A1	2000/05/24	Sarradin	SOR
	4-13	EP 1004992 A2	2000/05/31	Chan et al.	
	4-14	GB 2082816 A	1982/03/10	Halpern	
	4-15	GB 2092353 B	1984/05/31	Chalmers	
	4-16	GB 2107500 B	1985/09/11	Bernstein	
	4-17	GB 2204973 A	1988/11/23	Steiner et al.	
	4-18	GB 2274009 B	1992/12/29	Payne	
	4-19	JP 10-269289	1998/10/09	Kouichi	SOR/T
	4-20	WO 01/031599	2001/05/03	Racz et al.	
	4-21	WO 9527955	1995/10/19	Hummerston et al.	
	4-22	WO 9619771	1996/06/27	Weiner	
	4-23	WO 9625724	1996/08/22	Heyns et al.	
	4-24	WO 97/02548	1997/01/23	Everett et al.	
	4-25	WO 9809257	1998/03/05	Lisimaque et al.	
	4-26	WO 9819237	1998/05/07	Wilkinson	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 5 of 28	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.	Document No.	Publication Date	Name of Patentee or Applicant of Cited Document	Notes
	5-1	WO 9825238	1998/06/11	Clewits	
	5-2	WO 9833343	1998/07/30	Lehmus	
	5-3	WO 9837526	1998/08/27	Everett et al.	
	5-4	WO 9843212	1998/10/01	Chan et al.	
	5-5	WO 9852152	1998/11/19	Richards et al.	
	5-6	WO 9852153	1998/11/19	Richards	
	5-7	WO 9852158	1998/11/19	Everett et al.	
	5-8	WO 9852159	1998/11/19	Everett et al.	
	5-9	WO 9852160	1998/11/19	Everett et al.	
	5-10	WO 9852161	1998/11/19	Richards	
	5-11	WO 9852162	1998/11/19	Everett et al.	
	5-12	WO 9852163	1998/11/19	Richards et al.	
	5-13	WO 9922516	1999/05/06	Sarfati	
	5-14	WO 9946682	1999/09/16	Lindley et al.	
	5-15	WO 9946727	1999/09/16	Bright	
	5-16	WO 9948250	1999/09/23	Higginson et al.	
	5-17	WO 9962210	1999/12/02	DiGiorgio et al.	
	5-18				
	5-19				
	5-20				
	5-21				
	5-22				
	5-23				
	5-24				
	5-25				
	5-26				

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 6 of 28	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	6-1	About MultiMedia Cards (http://www.mmca.org/whatisa.htm), dated April, 23, 1999	
	6-2	Actiontec to Deploy Wave Systems' EMBASSY E-Commerce Technology in Home Networking and Modem Cards, May 11, 1999	
	6-3	Aladdin and Wave Complete Security Technology Integration, Jan. 27, 1998	
	6-4	Amendment 1 to SDMI Portable Device Specification, Part I, Version 1.0, Sep. 1999	
	6-5	An introduction to BPI statistics - Printed on 6/27/2000	
	6-6	Answers to Freqently Asked Questions About Smart Cards, 1996	
	6-7	Atmel and Wave Systems Corp. Partner to Provide Advanced "E-Commerce on a Chip" Solution, May 10, 1999	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 7 of 28	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	7-1	Automatic Fare Collection (http://www.oti.co.il/automatic_fare_collection.html), printed Jan. 5, 2000	
	7-2	Breakthrough For E-Commerce Over DTV Proven in Test with New Jersey Public Television, Mar. 28, 2000	
	7-3	Capital Broadcasting Unit, DTV Plus & WaveXpress Form Strategic Partnership to Explore & Deliver Broadcast E-commerce, Mar. 30, 2000	
	7-4	ChipCASH: Making stored value work, printed Aug. 18, 2000	
	7-5	Creative Nomad World (http://www.nomadworld.com/products/welcome.html), printed on 11/22/99	
	7-6	Creative Secures MP3 Player, November 12, 1999	
	7-7	CREATIVE TECHNOLOGIES CHOOSES WAVE SYSTEMS CORP. FOR MICRO-TRANSACTION SELLING AND PURCHASING, dated March 12, 1997	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 8 of 28	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	8-1	Creative Unveils Next Generation Portable Digital Audio Player, October 6, 1999	
	8-2	Cyber-COMM Frequently Asked Questions (FAQ), printed Apr. 27, 2000	
	8-3	Cyber-COMM Selects Wave Systems to Provide Trusted E-Commerce Infrastructure in Europe, Sep. 29, 1999	
	8-4	Digital Rights Management - Printed on 10/17/2000	
	8-5	E-COMMERCE: Companies 'not investing enough', October 14, 1999	
	8-6	Electronic Purse (http://www.oti.co.il/electronic_purse.html), printed Jan. 5, 2000	
	8-7	empeg car - Printed on 11/22/99	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 9 of 28	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	9-1	Emphasis on Internet ID and Authentication Central to the Smart Card Forum Strategic Direction for 2000, September 22, 1999	
	9-2	ePanorama - Card Technology Technology Page, printed May, 1, 2000	
	9-3	Ericsson demonstrates plug-in MP3 player for mobile phones at Telecom 99 (http://www.mmca.org/Press/ericsson-mp3.htm), October 9, 1999	
	9-4	FAQ for alt.technology.smartcards (http://www.ioc.ee/atsc/faq.html), printed on 11/30/2000	
	9-5	Flash and EEPROM: the never-ending tradeoff - August 1997	
	9-6	Flash Card Market Will Boom To $1.3 Billion By 2002, Predicts IDC - December 3, 1998	
	9-7	Frequently Asked Questions (http://www.scia.org/knowledgebase/aboutSmartCards/faqs.htm), printed on 11/30/2000	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
FORM PTO-1449 (modified)

Sheet 10 of 28

Application No.	13/438,754
Filing Date	April 3, 2012
First Named Inventor	Patrick Sandor Racz
Group Art Unit	2887
Examiner Name	Le, Thien Minh
Attorney Docket No.	4037-0003
Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	10-1	Gadgets Galore from Comdex (http://www.wired.com/news/mp3/0,1285,32573,00.html), November 17, 1999	
	10-2	GlobalWave- Secure Digital Media Distribution; printed May 15, 2000	
	10-3	Growing Number of On-Line Users Turn Digital Content Into Cash Using MyPublish from Wave Systems, Feb. 22, 2000	
	10-4	~~Guide to the~~ SDMI Portable Device Specification - Part 1, Version 1.0 (date unknown) not dated	
	10-5	Heer, D.N.; Maher, D.P.; IEEE Transactions on Consumer Electronics, Volume: 41, Issue: 3, 1995, Pages: 869 - 874	
	10-6	Heer, D.N.; Maher, D.P.; Takahashi, R.; Proceedings of International Conference on Consumer Electronics, 1995, pgs. 382-383.	
	10-7	How DRM commerce works (http://www.intertrust.com/main/technology/howdrmcworks.html) - archived on 10/18/2000	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 11 of 28		
	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	11-1	HP and Wave Systems to Bring Trust And Security to the PC, Oct. 27, 1998	
	11-2	HP Offers secure PC for ecommerce, available by at least Apr. 27, 2000	
	11-3	HP, RSA Data Security and Wave Systems to Enable Enhanced Security for E-Mail and Internet Transactions, Jan 20, 1999	
	11-4	IBM forays into digital music software, February 8, 1999	
	11-5	IGST to Incorporate Wave Systems' EMBASSY Technology in Multimedia Chips for Set-Top Boxes, Dec. 2, 1998	
	11-6	I-Jam's New MP3 Internet Music Player will use SanDisk Multimedia Cards to Download and Store Music - June 14, 1999	
	11-7	Industry Specific Smart Card Specifications (http://www.scia.org/knowledgebase/aboutSmartCards/specs.html), printed on 11/30/2000	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 12 of 28	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	12-1	Internet and Smart Card Application Deployment (http://www.smartcardcentral.com/technical/articles/jsource/jsource_080999.asp), archived August 1999	
	12-2	Internet Audio Portables: A Christmas Buying Guide, printed Jan. 5, 2000	
	12-3	Intertrust technology - archived on 10/18/2000	
	12-4	ITE to Embed Wave System Corp.'s Embassy Technology into Integrated Circuits, Nov. 9, 1998	
	12-5	JazPiper FAQ (http://www.jazpiper.nl/en_ie/faq.htm), printed on 11/22/99	
	12-6	Joint Press Release by Infineon Technologies and Grundig - August 27, 1999	
	12-7	Joint Press Release by Infineon Technologies and Hitachi - April 22, 1999	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 13 of 28	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	13-1	Launch Discover New Music (http://www.launch.com/music/mymusic/pvn_content/0,3349,,00.html), January 4, 2000	
	13-2	Liquid Audio signs with EMI, June 17, 1999, pp. 1-2	
	13-3	Major Companies Join Forces to Solve Security, Trust, and Privacy Issues in Electronic Business, Aug. 24, 1999	
	13-4	Microsoft Ups Commitment to Digital Media (http://www.billboard.com/sites/archive/00/0526.asp), May 26, 2000	
	13-5	Ministry of Sound's Content Library to Become Available to RioPort Audience and Distribution Partners, November 15, 1999	
	13-6	More popular than Sex (http://www.wired.com/news/business/0,1367,31834,00.html), October 14, 1999	
	13-7	MP3 catalyzing Net music's future, April 27, 1999	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 14 of 28	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	14-1	MP3 device makers win key court ruling, June 15, 1999	
	14-2	MP3: on the road again, November 22, 1999	
	14-3	MP3s Anywhere You Are (http://www.wired.com/news/mp3/0,1285,32109,00.html), October 28, 1999	
	14-4	MultiMedia Card: The Storage Medium (http://www.pontis.de/site_e/produkte/ca_mmc_e.htm), printed Jan. 4, 2000	
	14-5	Music a la Card (http://www.mplayer3.com/site_e/ho_pl_e.htm), printed on 11/22/99	
	14-6	Music Battle Takes to the Hill (http://www.wired.com/news/politics/0,1283,32008,00.html), October 20, 1999	
	14-7	Music to Microsoft's Ears (http://www.wired.com/news/culture/0,1284,32379,00.html), November 15, 1999	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 15 of 28	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	15-1	NATIONAL SEMICONDUCTOR® AND WAVE SYSTEMS TO DEVELOP CO-PROCESSOR-BASED SECURITY SOLUTIONS FOR E-COMMERCE, dated Sep. 15, 1999	
	15-2	Net music changing copyrights, April 27, 1999	
	15-3	New Media Age - p. 23 - November 18, 1999	
	15-4	PC Free and Wave Systems to Deliver Bundles "E-Commerce PC", June 9, 1999	
	15-5	PCT/GB00/04110, International Search Report mailed February 13, 2000	
	15-6	Personal Jukebox PJB-100 - Printed on 11/22/99	
	15-7	Piracy (RIAA Webpage: http://www.riaa.com/piracy/piracy.htm), printed on 11/25/1999	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 16 of 28	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	16-1	Piracy Definitions - Printed on 6/27/2000	
	16-2	Portable MP3 Players and More (http://www.pontis.de/site_e/produkte/sh_prl_e.htm), printed Jan. 4, 2000	
	16-3	PortalPlayer Announces Collaboration with SanDisk Corporation to Enable New Classes of Consumer Electronic Devices for Music Playback - December 8, 1999	
	16-4	Prepaid smart card techniques: A brief introduction and comparison - dated 1994	
	16-5	Programming Plus Digital Content: A New Business Model for Television, printed Apr. 27, 2000	
	16-6	PSXAMP - Printed on 11/22/99	
	16-7	~~Reciprocal - Driving the content economy (date unknown)~~ not dated	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 17 of 28	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	17-1	Research and Intellectual Property (http://www.intertrust.com/main/technology/research-ip.html) - Printed on 10/18/2000	
	17-2	RIAA wins restraining order against MP3 recording device, RIAA Press Release - 10/16/1998	
	17-3	Rio 500 (from archive.org archived Nov 28, 1999 but original date unknown)	
	17-4	RIOPORT SELECTS MAGEX TO PROVIDE DIGITAL RIGHTS MANAGEMENT SOLUTION FOR DIGITAL AUDIO DELIVERY, PLAYBACK USING RIOPORT PLATFORM AND INTERTRUST TECHNOLOGY, November 15, 1999	
	17-5	RioPort Shows Secure Solution for Downloading and Playing Back Windows Media Format Music and Spoken Word Tracks from Internet to Desktop, November 15, 1999	
	17-6	RIOPORT, INC. COLLABORATES WITH MICROSOFT ON TECHNOLOGY TO SEAMLESSLY DELIVER SECURE DIGITAL AUDIO CONTENT TO A VARIETY OF EXTERNAL PLAYBACK DEVICES, November 10, 1999	
	17-7	Rioport: About us (http://www.rioport.com/RioAbout/0,1202,,00.htm), printed on 11/22/99	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 18 of 28	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	18-1	S3 hears sweet music with MP3 plans (http://news.cnet.com/news/0-1006-200-1711701.html), April 18, 2000	
	18-2	Samsung and Sony Announce Memory Stick Collaboration, http://www.samsung.com/news/samsung/2001/sec0803-20010803102150.html, pp. 1-2, Aug. 3, 2001	
	18-3	Samsung Yepp Features (http://www.samsungyepp.com/Yepp64.html), archived Mar. 4, 2000	
	18-4	SanDisk Announces Development of the World's First Floppy Disk Adapter for Multimediacards, July 19, 1999	
	18-5	SanDisk Will Supply I&C with Multimediacards for new portable MP3 Internet Music Players, June 23, 1999	
	18-6	SanDisk Will Supply Maycom with 32 MB Multimediacard for Use with New Merit, April 27, 1999	
	18-7	Sarnoff and Wave Systems to Form inTelecast, April 19, 1999	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 19 of 28	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	19-1	SDMI announces portable device technology will be available (https://www.sdmi.org/dscgl/ds.py/Get/File-646/nyc-99-09-24-press-release.htm), 9/24/1999	
	19-2	SDMI approves technology selection, 11/12/1999	
	19-3	SDMI fact sheet, printed on 11/25/1999	
	19-4	~~SDMI FAQ (date unknown)~~ not dated	
	19-5	SDMI IDENTIFIES AUDIO WATERMARK TECHNOLOGY FOR NEXT GENERATION PORTABLE DEVICES FOR DIGITAL MUSIC, 8/9/1999	
	19-6	~~SDMI Member~~ Company Statements On the SDMI Portable Device Specification Release (date ~~unknown)~~ not dated	
	19-7	SDMI Participant List (http://www.sdmi.org/participant_list.htm), dated May 24, 2000	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 20 of 28	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	20-1	SDMI Portable Device Specification - Part 1, Version 1.0, dated July 8, 1999	
	20-2	SDMI Publishes Open Standard for Portable Devices, July 13, 1999	
	20-3	SDMI Working group structure (http://www.sdmi.org/work_group_struct.htm), updated June 1, 2000	
	20-4	Shaw, I., Cash on Delivery - Mobile Consumer Management Goes Back to Basics (1994)	
	20-5	Smart Card Forum - http://www.smartcardforum.org/aboutscf/backgrd.htm, printed Nov. 30, 2000	
	20-6	Smart Card Overview (http://www.scia.org/knowledgebase/aboutSmartCards/primer.htm), printed on 11/30/2000	
	20-7	Smart Cards on Line (http://www.smartex.com), dated Aug. 18, 2000	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
FORM PTO-1449 (modified)

Sheet 21 of 28

Application No.	13/438,754
Filing Date	April 3, 2012
First Named Inventor	Patrick Sandor Racz
Group Art Unit	2887
Examiner Name	Le, Thien Minh
Attorney Docket No.	4037-0003
Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	21-1	~~Smartflash LLC et. al. v. Apple Inc., Case No. 6:13cv00447-MHS-KNM, Exhibit B to Defendants' Preliminary Invalidity Contentions~~	
	21-2	~~Smartflash LLC et. al. v. Apple Inc., Case No. 6:13cv00447-MHS-KNM, Exhibit E to Defendants' Preliminary Invalidity Contentions~~	
	21-3	~~Smartflash LLC et. al. v. Apple Inc., Case No. 6:13cv00447-MHS-KNM, Exhibit F to Defendants' Preliminary Invalidity Contentions~~	
	21-4	~~Smartflash LLC et. al. v. Apple Inc., Case No. 6:13cv00447-MHS-KNM, Exhibit H to Defendants' Preliminary Invalidity Contentions~~ not dated	
	21-5	Sonicnet Music News, (http://partners.sonicnet.com/ap/rioport/newstory.jhtml?id=620291), pp.1-2, Jan 7. 2000	
	21-6	Standard Microsystems and Wave Systems Agree to Embed Wave's "WaveMeter" Technology in Standard Microsystems Integrated Circuits, May 28, 1998	
	21-7	T3 News, July, 1999, pgs. 29-30	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 22 of 28	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	22-1	Technology Standards Developed for Music Delivery System over Mobile Phones - December 9, 1999	
	22-2	The MP3 Place (http://www.mp3hardware.com/clickman.shtml), printed on 11/22/99	
	22-3	The ROS-Card: the audio storage media of the future (http://www.mplayer3.com/site_e/ho_ros_e.htm), printed Nov. 22, 1999	
	22-4	The storage medium: MultiMediaCard (http://www.mplayer3.com/site_e/ho_mmc_e.htm), printed on 11/22/99	
	22-5	The technology behind the world's first consumer trial of mobile electronic cash, printed 9/25/1999	
	22-6	The top 50 bootlegged artists in the UK, printed 11/25/1999	
	22-7	The Trusted Client from Wave Systems Corp. Web Site (http://www.wavesys.com/news/featuredstory.html), printed May 15, 2000	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 23 of 28	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	23-1	The Trusted Client from Wave Systems Corp. Web Site (http://www.wavesys.com/technology/embassytc1.html), archived May 20, 2000	
	23-2	The World Market in Review, archived Mar. 3, 2000	
	23-3	The World of Multos (http://www.multos.com/multoglobe.ihtml), printed Jan. 5, 2000	
	23-4	THEGLOBE.COM and Wave Systems Offer Internet Industry's First e-Commerce Service for the Sale of Personal Digital Content, Apr. 22, 1999	
	23-5	There is a DIVA in my pocket - Printed on 11/22/99	
	23-6	Thomson Lrya (http://www.vitaminic.co.uk/hardware/lyra.shtml), printed on 11/22/99	
	23-7	Uskela, S., Services in Cellular Packet Data Networks, Masters Thesis, Helsinki University of Technology (Sept. 1999)	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 24 of 28	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	24-1	~~Verifides Technology~~ - Secure Information Sharing via Technology Enforced Originator Control ~~(date unknown)~~ not dated	
	24-2	Wave - Wave Digital Network (dated 2000)	
	24-3	Wave Systems and IBM to Collaborate on Technology for Accessing Digital Content, Dec. 18, 1997	
	24-4	Wave Systems and Sarnoff Corporation Announce Board of Directors for WaveXpress Joint Venture, Sep. 9, 1999	
	24-5	Wave Systems Announces Agreement with KISS Nordic A/S to Bundle Wave's E-Commerce System with KISS Nordic Multimedia Products, Aug. 28, 1999	
	24-6	Wave Systems Announces New Technology Enabling Secure Electronic Commerce Transactions in the PC, Oct. 27, 1998	
	24-7	Wave Systems Announces Support of SDMI Specification, July 13, 1999	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 25 of 28	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	25-1	Wave Systems Appoints Alan Chaplin Vice President of Entertainment Service, May 11, 1999	
	25-2	Wave Systems Continues US Momentum In Europe by Showcasing Digital Content Distribution and E-Commerce Technology with GlobalWave at Milia 2000, Feb. 16, 2000	
	25-3	Wave Systems Corp. Announces Three Pay-Per-Use CD-ROM Software Publishing Partners, Feb. 10, 1997	
	25-4	Wave Systems Corp. Debuts Micro-Transaction System for Purchasing Digital Content, Feb. 10, 1997	
	25-5	Wave Systems Corp. Enhances Consumer-Publisher Relationships with Upgraded Great Stuff Network Technology, Mar. 4, 1998	
	25-6	Wave Systems Corp. Forges Strategic Relationship With Leading Broadcast Systems Solution Provider, Apr. 7, 1998	
	25-7	Wave Systems Demonstrates Trusted Client Services and Applications at Comdex, Nov. 15, 1999	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 26 of 28	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	26-1	Wave Systems Introduces MyPublish Community E-Commerce Service, Apr. 22, 1999	
	26-2	Wave Systems Introduces WaveDirect E-Commerce Service for PC Users, June 1, 1999	
	26-3	Wave Systems Plans to Acquire N*Able Technologies, June 14, 1999	
	26-4	Wave Systems Reports 1998 Results, Apr. 1, 1999	
	26-5	Wave Systems Signs Letter of Intent with Lego Media International to Deliver Software by Satellite for Europe Online Networks, Aug. 28, 1999	
	26-6	Wave Systems To Integrate Sun Microsystem's Java Card Technology Into Consumer Devices to Extend EMBASSY E-Commerce Solutions, May 10, 1999	
	26-7	Wave Systems to Offer Interactive Magic Software Titles Through Innovative WaveMeter Distribution System, June 15, 1998	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 27 of 28	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	27-1	Wave Systems to Participate in Trusted Computing Platform, Oct. 11, 1999	
	27-2	WaveEnabled Trusted Client Applications, from Wave Systems Corp. Web Site (http://www.wavesys.com/technology/embassytc2.html), archived May 26, 2000	
	27-3	Wave's Embassy Technology Demonstrated in Compaq Smart Card Readers at Windows 2000 Launch, Feb. 17, 2000	
	27-4	WaveSystems Supports IBM Initiative to Embed Security in the PC, Sep. 28, 1999	
	27-5	Which Net music technology will win, May 7, 1999	
	27-6	Who's Gonna Own the Music (http://www.wired.com/news/culture/0,1284,31682,00.html), October 18, 1999	
	27-7	Worldwide music industry coordinates its strategy againts piracy, RIAA Press Release, 10/28/1999	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 28 of 28	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	28-1	~~www.thenetnow.co.uk - MP3 Players - p.144 - 145 (date unknown)~~	
	28-2	~~X-Card - The Pocket smart card reader/Euro converter (date unknown)~~	
	28-3	~~X-Collection - Handy-Sized Smart Card Reader (date unknown)~~	
	28-4	~~X-Sign (date unknown)~~	
	28-5	~~X-Smart - Smart Card Reader/Writer (date unknown)~~ not dated	
	28-6	YP-E32 (http://yepp.co.kr/eng/pd.html) - Printed on 11/22/99	
	28-7	Zappee ZMP 3000 (http://www.zappee.com/html/body_mp3_hardware.html), printed on 11/22/99	

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	17	(US-20120217302-$ or US-20090122565-$ or US-20100066765-$ or US-20090009506-$ or US-20080068207-$ or US-20120098451-$ or US-20110199010-$ or US-20120181338-$).did. or (US-4590365-$ or US-5682027-$ or US-6981179-$ or US-6025973-$ or US-8179231-$ or US-6519241-$ or US-7043456-$ or US-6892941-$ or US-6476306-$).did.	US-PGPUB; USPAT	OR	OFF	2014/03/31 12:09
L2	5595	((G06Q20/341).CPC.)	US-PGPUB; USPAT	OR	OFF	2014/03/31 12:09
L3	6771	((G07F7/1008).CPC.)	US-PGPUB; USPAT	OR	OFF	2014/03/31 12:09
L4	7306	((H04L67/04).CPC.)	US-PGPUB; USPAT	OR	OFF	2014/03/31 12:10
L5	1028	((H04N5/63).CPC.)	US-PGPUB; USPAT	OR	OFF	2014/03/31 12:10
L6	1186	((A61B5/00).CPC.)	US-PGPUB; USPAT	OR	OFF	2014/03/31 12:11
L7	4555	((G06Q20/3415 OR G06Q20/3552 OR G06Q20/3576 OR G06Q20/3821 OR A61B5/00).CPC.)	US-PGPUB; USPAT	OR	OFF	2014/03/31 12:11
L8	22911	((H04L2209/56 OR H04L2209/80 OR H04L63/0281 OR H04L63/08 OR H04L63/12 OR H04L9/3247 OR A61B5/00).CPC.)	US-PGPUB; USPAT	OR	OFF	2014/03/31 12:12
L9	106	memory same storing same card same ((payment or transaction) near10 data) same instructions	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2014/03/31 12:13
L10	35669	code.clm. same request.clm.	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2014/03/31 12:42
L11	38629	l2 or l3 or l4 or l5 or l6 or l7 or l8	US-PGPUB; USPAT	OR	OFF	2014/03/31 12:43
L12	21	l11 and l9	US-PGPUB; USPAT	OR	OFF	2014/03/31 12:43

L13	0	l12 and l10	US-PGPUB; USPAT	OR	OFF	2014/03/31 12:43
L14	1889	l11 and l10	US-PGPUB; USPAT	OR	OFF	2014/03/31 12:43
L15	106	memory same storing same card same ((payment or transaction) near10 data) same instructions	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2014/03/31 12:43
L16	0	l14 and l15	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2014/03/31 12:43
L17	21	l11 and l15	US-PGPUB; USPAT	OR	OFF	2014/03/31 12:44

EAST Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L18	17	"Term Removed" or "Term Removed"	US-PGPUB; USPAT; UPAD	OR	OFF	2014/03/31 12:45
L20	5595	((G06Q20/341).CPC.)	US-PGPUB; USPAT; UPAD	OR	OFF	2014/03/31 12:45
L21	6771	((G07F7/1008).CPC.)	US-PGPUB; USPAT; UPAD	OR	OFF	2014/03/31 12:45
L22	7306	((H04L67/04).CPC.)	US-PGPUB; USPAT; UPAD	OR	OFF	2014/03/31 12:45
L23	1028	((H04N5/63).CPC.)	US-PGPUB; USPAT; UPAD	OR	OFF	2014/03/31 12:45
L24	1186	((A61B5/00).CPC.)	US-PGPUB; USPAT; UPAD	OR	OFF	2014/03/31 12:45
L25	4555	((G06Q20/3415 OR G06Q20/3552 OR G06Q20/3576 OR G06Q20/3821 OR A61B5/00).CPC.)	US-PGPUB; USPAT; UPAD	OR	OFF	2014/03/31 12:45
L26	22911	((H04L2209/56 OR H04L2209/80 OR H04L63/0281 OR H04L63/08 OR H04L63/12 OR H04L9/3247 OR A61B5/00).CPC.)	US-PGPUB; USPAT; UPAD	OR	OFF	2014/03/31 12:46
L28	110	memory same storing same card same ((payment or transaction) near10 data) same instructions	US-PGPUB; USPAT;	OR	OFF	2014/03/31 12:46

			UPAD			

L29	38629	l20 or l21 or l22 or l23 or l24 or l25 or l26	US-PGPUB;
			USPAT;
			UPAD
				OR	OFF	2014/03/31 12:46

L30	21	l28 and l29	US-PGPUB;
			USPAT;
			UPAD
				OR	OFF	2014/03/31 12:46

3/ 31/ 2014 12:47:14 PM
C:\ Users\ tle4\ Documents\ EAST\ Workspaces\ 13438754.wsp

Search Notes	**Application/Control No.** 13438754	**Applicant(s)/Patent Under Reexamination** HULST ET AL.
	Examiner THIEN M LE	**Art Unit** 2887

CPC- SEARCHED

Symbol	Date	Examiner

CPC COMBINATION SETS - SEARCHED

Symbol	Date	Examiner

US CLASSIFICATION SEARCHED

Class	Subclass	Date	Examiner
235	380, 382, 492, 451, 486, 487	12/20/2012	LTM
updated	same as above	4/1/2013	LTM
updated	same as above	9/4/2013	LTM
updated	same as above	1/15/2014	LTM
updated	same as above	3/31/2014	LTM

SEARCH NOTES

Search Notes	Date	Examiner
EAST, review parent applications for double patenting	12/20/2012	LTM
EAST	4/1/2013	LTM
EAST	9/4/2013	LTM
EAST	1/15/2014	LTM
EAST	3/31/2014	LTM

INTERFERENCE SEARCH

US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner
235	380,382	4/1/2013	LTM
updated	same as above	9/4/2013	LTM
updated	same as above	1/15/2014	LTM
updated	same as above	3/31/2014	LTM

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 1 of 1	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	1-1	Smartflash LLC et. al. v. Apple Inc., Case No. 6:13cv00447-MHS-KNM, Exhibit B to Defendants' Preliminary Invalidity Contentions, January 27, 2014	
	1-2	Smartflash LLC et. al. v. Apple Inc., Case No. 6:13cv00447-MHS-KNM, Exhibit E to Defendants' Preliminary Invalidity Contentions, January 27, 2014	
	1-3	Smartflash LLC et. al. v. Apple Inc., Case No. 6:13cv00447-MHS-KNM, Exhibit F to Defendants' Preliminary Invalidity Contentions, January 27, 2014	
	1-4	Smartflash LLC et. al. v. Apple Inc., Case No. 6:13cv00447-MHS-KNM, Exhibit H to Defendants' Preliminary Invalidity Contentions, January 27, 2014	
	1-5		
	1-6		
	1-7		

Examiner Signature		Date Considered	

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re PATENT Application of:		Confirmation No.:	3525
Patrick Sandor Racz		Attorney Docket:	4037-0003
Appl. S.N.:	13/438,754	Group Art Unit:	2887
Filing Date:	April 3, 2012	Examiner:	Le, Thien Minh
Title:	DATA STORAGE AND ACCESS SYSTEMS	Date:	04/21/2014

INFORMATION DISCLOSURE STATEMENT

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

Commissioner:

Pursuant to 37 C.F.R. § 1.56, the attention of the Patent and Trademark Office is hereby directed to the reference(s) listed on the attached PTO-1449. One copy of each non-U.S. Patent reference is attached. It is respectfully requested that the information be expressly considered during the prosecution of this application, and that the reference(s) be made of record therein and appear among the "References Cited" on any patent to issue therefrom.

The submission of any document herewith, which is not a statutory bar, is not intended that any such document constitutes prior art against any of the claims of the present application or is considered to be material to patentability as defined in 37 C.F.R. § 1.56(b). Applicants do not waive any rights to take any action which would be appropriate to antedate or otherwise remove as a competent reference against the claims of the present application, including, but not limited to, showing that the references cited in Defendants' Claim Constructions are or were not available as asserted.

This Information Disclosure Statement (IDS) lists the date of four references previously submitted but crossed out for not having a date associated with the entry. An additional IDS filed before the mailing of the Notice of Allowance has yet to be considered by the examiner, and the references cited in this IDS were served less than three months before the filing of this IDS. The Patent Office is hereby authorized to charge any fee necessary for consideration of this IDS to the deposit account below.

CHARGE STATEMENT: Deposit Account No. 501860, order no. **4037-0003**.

The Commissioner is hereby authorized to charge any fee specifically authorized hereafter, or any missing or insufficient fee(s) filed, or asserted to be filed, or which should have been filed herewith or concerning any paper filed hereafter, and which may be required under Rules 16-18 (missing or insufficiencies only) now or hereafter relative to this application and the resulting Official Document under Rule 20, or credit any overpayment, to our Accounting/Order Nos. shown above, for which purpose a duplicate copy of this sheet is attached

This CHARGE STATEMENT does not authorize charge of the issue fee until/unless an issue fee transmittal sheet is filed.

CUSTOMER NUMBER
42624

Davidson Berquist Jackson & Gowdey LLP
4300 Wilson Blvd., 7th Floor,
Arlington Virginia 22203
Main: (703) 894-6400 ● FAX: (703) 894-6430

Respectfully submitted,

By: / Michael R. Casey /

Michael R. Casey
Registration No.: 40,294

Electronic Acknowledgement Receipt

EFS ID:	18811333
Application Number:	13438754
International Application Number:	
Confirmation Number:	3525
Title of Invention:	DATA STORAGE AND ACCESS SYSTEMS
First Named Inventor/Applicant Name:	Patrick Sandor Racz
Customer Number:	42624
Filer:	Michael R. Casey
Filer Authorized By:	
Attorney Docket Number:	4037-0003
Receipt Date:	21-APR-2014
Filing Date:	03-APR-2012
Time Stamp:	11:56:36
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	Information Disclosure Statement (IDS) Form (SB08)	20140421-1449.pdf	55768 335f53086c03001cd9a40925f24ae9a3812d922c	no	1

Warnings:

Information:

This is not an USPTO supplied IDS fillable form

2	Non Patent Literature	NP0000.pdf	6914861 701f1b0470b68e062e63f556e350516eecdf0dbe	no	112
Warnings:					
Information:					
3	Non Patent Literature	NP0001.pdf	6161388 270f67539b7ab659a64e50bf95a2f9eefbad9b51	no	308
Warnings:					
Information:					
4	Non Patent Literature	NP0002.pdf	4772629 15fabae3ede41a5630b96470b89d718f6bfa7f20	no	237
Warnings:					
Information:					
5	Non Patent Literature	NP0003.pdf	6707477 459e1fb9f204f5a3db9a360225b3c6cbbbfbf239	no	316
Warnings:					
Information:					
6	Transmittal Letter	20140421_IDS_Transmittal.pdf	135274 f6fe438c5dc47fa7579b2c825126574dbe1ba70c	no	2
Warnings:					
Information:					
			Total Files Size (in bytes):	24747397	

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.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/438,754	04/03/2012	Patrick Sandor Racz	4037-0003	3525

42624 7590 05/13/2014
DAVIDSON BERQUIST JACKSON & GOWDEY LLP
4300 WILSON BLVD., 7TH FLOOR
ARLINGTON, VA 22203

EXAMINER	
LE, THIEN MINH	

ART UNIT	PAPER NUMBER
2887	

MAIL DATE	DELIVERY MODE
05/13/2014	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES DEPARTMENT OF COMMERCE
U.S. Patent and Trademark Office
Address : COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPLICATION NO./ CONTROL NO.	FILING DATE	FIRST NAMED INVENTOR / PATENT IN REEXAMINATION	ATTORNEY DOCKET NO.
13/438,754	03 April, 2012	RACZ ET AL.	4037-0003

		EXAMINER	
DAVIDSON BERQUIST JACKSON & GOWDEY LLP 4300 WILSON BLVD., 7TH FLOOR ARLINGTON, VA 22203		Thien M. Le	
		ART UNIT	PAPER
		2887	20140508

DATE MAILED:

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner for Patents

The information disclosure statements filed on 4/10/2014 and 4/21/2014 have been entered.

/Thien M. Le/
Primary Examiner, Art Unit 2887

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 1 of 6	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	1-1	"Delkin Breaks 400MB Flash Memory Barrier for MP3 Players", Richard Menta, MP3newswire.net, October 2, 2000, available at http://www.mp3newswire.net/stories/2000/delkin2.html	
	1-2	"First MP3 Portable with 128MB Built-in Flash Review: The Soul", Richard Menta, MP3newswire.net, December 7, 1999, available at http://www.mp3newswire.net/stories/2000/soul.html	
	1-3	"New_RaveMP_MP3_Players Debut", Richard Menta, MP3Newswire.net, June 30, 2000, available at http://www.mp3newswire.net/stories/2000/drive.html	
	1-4	"Pirates Beware: We're Watching", Wired.com, January 3, 2001, available at http://archive.wired.com/science/discoveries/news/2001/01/40866	
	1-5	"SDMI Executive Director Challenges MP3.com Editorial", Rich Menta, November 4, 1999, available at http://www.mp3newswire.net/stories/sdmi.html	
	1-6	"SDMI: Divide or Conquer?", Wired.com, November 18, 1999, available at http://archive.wired.com/science/discoveries/news/1999/11/32513	
	1-7	"Smart Cards: A Case Study", IBM International Technical Support Organization, October 1998, available at http://www.redbooks.ibm.com/redbooks/pdfs/sg245239.pdf	

Examiner Signature	/Thien Le/	Date Considered	05/08/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified) Sheet 2 of 6	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	2-1	"Smart cards: A primer", JavaWorld, December 1997, available at http://www.javaworld.com/article/2077101/learnjava/smart-cards--a-primer.html	
	2-2	"The End of SDMI", Eric Scheirer, Technology Correspondent, MP3.com, October 15, 1999, available at ftp://ftp.gwdg.de/pub/eff/cafe/scheirer1.html	
	2-3	"Web Sites and Recording Labels at Impasse on Fees", Richtel, Matt, The New York Times, November 29, 1999, available at http://www.nytimes.com/library/tech/99/11/biztech/articles/29tune.html	
	2-4	American Heritage College Dictionary (3rd Edition 1997): (definition of "payment" and "pay")	
	2-5	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00110 (U.S. Pat. No. 8,336,772), dated April 3, 2014 (including Declarations)	
	2-6	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00111 (U.S. Pat. No. 8,336,772), dated April 3, 2014 (including Declarations)	
	2-7	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00112 (U.S. Pat. No. 7,942,317), dated April 3, 2014 (including Declarations)	

Examiner Signature	/Thien Le/	Date Considered	05/08/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 3 of 6	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	3-1	Apple Inc.'s Petition for Covered Business Method Patent Review in CBM2014-00113 (U.S. Pat. No. 7,942,317), dated April 3, 2014 (including Declarations)	
	3-2	Freedman & Glossbrenner, The Internet Glossary and Quick Reference Guide, 1998, pgs. 79, 246.	
	3-3	Hartel, "Formalizing the Safety of Java, the Java Virtual Machine". ACM Comp. Surv. Vol.33 No.4, Dec. 2001 p517-558.	
	3-4	Herreweghen and Wille, Risks and Potentials of using EMV for Internet Payments, USENIX Workshop on Smartcard Technology, May 10-11, 1999, pp.163-173	
	3-5	IBM Dictionary of Computing, 10th Ed. 1994, pgs. 297, 533 and 637.	
	3-6	Kyu Ha Lee' et al., "AN ARCHITECTURE AND IMPLEMENTATION OF MPEG AUDIO LAYER III DECODER USING DUAL-CORE DSP." IEEE Transactions on Consumer Electronics, Vol. 47, No. 4, NOVEMBER 2001.	
	3-7	Lawrence Haynes, "Theatre Medical Data Store." IEEE (pub), 1998.	

Examiner Signature	/Thien Le/	Date Considered	05/08/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 4 of 6	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	4-1	McGraw-Hill Dictionary of Scientific and Technical Terms (4th Edition 1989) (definition of “data carrier”)	
	4-2	Merriam-Webster Collegiage Dictionary (10th Edition 1997) : (definition of “pay” and “payment”)	
	4-3	Microsoft Press Computer User’s Dictionary, 1998. P. 157, 227, 367	
	4-4	Scheuermann, D., "The Smart Card as a mobile security device." Ch 4, Chris Mitchell (ed.), Security for Mobility. Institution of Engineering and Technology (pub), 2004	
	4-5	Scheuermann, D., "The Smart Card as a mobile security device." Security for Mobility. Electronics and Communications Engineering Journal, Vol.14 No. 5, Oct 2002	
	4-6	Smart Cards: Seizing Strategic Business Opportunities Smart Card Forum; Hardcover (including but not limited to definitions in “Glossary of Terms”)	
	4-7	Smartflash LLC et al. v. Apple Inc. et al., Civil Action 6:13-CV-00447-MHS-KNM, Defendants' Preliminary Claim Constructions and Extrinsic Evidence, dated April 1, 2014	

Examiner Signature	/Thien Le/	Date Considered	05/08/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 5 of 6	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	5-1	Smartflash LLC et al. v. Apple Inc. et al., Civil Action 6:13-CV-00447-MHS-KNM, Plaintiffs' Preliminary Claim Constructions and Extrinsic Evidence, dated April 1, 2014	
	5-2	Smartflash LLC et al. v. Samsung Electronics et al., Civil Action 6:13-CV-00448-MHS-KNM, Defendants' Preliminary Claim Constructions and Extrinsic Evidence, dated April 1, 2014	
	5-3	Smith,M., "Smart cards: Integrating for portable complexity", IEEE Computer, 1998	
	5-4	Sony Press Release, “Sony Announces ‘Memory Stick’ Recordable IC Memory Card Products, July 30, 1998, available at http://www.sony.net/SonyInfo/News/Press_Archive/199807/98-067/	
	5-5	The IEEE Standard Dictionary of Electrical and Electronics Terms (6th Edition 1996): (definitions of “data carrier” and “data medium”)	
	5-6	THE JAVA CARD 3 PLATFORM, White Paper, August 2008, Oracle Corp.	
	5-7	The New IEEE Standard Dictionary of Electrical and Electronic Terms (5th Edition 1993), pgs. 305, 533, 1011, 1252	

Examiner Signature	/Thien Le/	Date Considered	05/08/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 6 of 6	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	6-1	Webster's New World Dictionary, Third College Edition, 1991, pg. 1173	
	6-2		
	6-3		
	6-4		
	6-5		
	6-6		
	6-7		

Examiner Signature	/Thien Le/	Date Considered	05/08/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 1 of 1	Confirmation No.	3525

NON-PATENT REFERENCES

Examiner Initials*	Cite No.	Non-patent Reference bibliographic information, where available	Notes
	1-1	Smartflash LLC et. al. v. Apple Inc., Case No. 6:13cv00447-MHS-KNM, Exhibit B to Defendants' Preliminary Invalidity Contentions, January 27, 2014	
	1-2	Smartflash LLC et. al. v. Apple Inc., Case No. 6:13cv00447-MHS-KNM, Exhibit E to Defendants' Preliminary Invalidity Contentions, January 27, 2014	
	1-3	Smartflash LLC et. al. v. Apple Inc., Case No. 6:13cv00447-MHS-KNM, Exhibit F to Defendants' Preliminary Invalidity Contentions, January 27, 2014	
	1-4	Smartflash LLC et. al. v. Apple Inc., Case No. 6:13cv00447-MHS-KNM, Exhibit H to Defendants' Preliminary Invalidity Contentions, January 27, 2014	
	1-5		
	1-6		
	1-7		

Examiner Signature	/Thien Le/	Date Considered	05/08/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant. Notes: If identified, the following is provided: EA = English Abstract, T = Translation, PT = Partial Translation, SOR = Statement of Relevancy, PF = Patent Family.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

42624 7590 04/11/2014
DAVIDSON BERQUIST JACKSON & GOWDEY LLP
4300 WILSON BLVD., 7TH FLOOR
ARLINGTON, VA 22203

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/438,754	04/03/2012	Patrick Sandor Racz	4037-0003	3525

TITLE OF INVENTION: DATA STORAGE AND ACCESS SYSTEMS

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	$0	$0	$1780	$0	07/11/2014

EXAMINER	ART UNIT	CLASS-SUBCLASS
LE, THIEN MINH	2887	235-380000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

□ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

□ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) The names of up to 3 registered patent attorneys or agents OR, alternatively,

(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 Davidson Berquist
2 Jackson & Gowdey, LLP
3 ______

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE
Smartflash, LLC

(B) RESIDENCE: (CITY and STATE OR COUNTRY)
Tyler, Texas

Please check the appropriate assignee category or categories (will not be printed on the patent) : □ Individual ☒ Corporation or other private group entity □ Government

4a. The following fee(s) are submitted:

□ Issue Fee
□ Publication Fee (No small entity discount permitted)
□ Advance Order - # of Copies ______

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**

□ A check is enclosed.
□ Payment by credit card. Form PTO-2038 is attached.
☒ The Director is hereby authorized to charge the required fee(s), any deficiency, or credits any overpayment, to Deposit Account Number 501860 (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)

□ Applicant certifying micro entity status. See 37 CFR 1.29

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

□ Applicant asserting small entity status. See 37 CFR 1.27

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

□ Applicant changing to regular undiscounted fee status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature / Michael R. Casey /
Date May 23, 2014

Typed or printed name Michael R. Casey
Registration No. 40,294

Electronic Acknowledgement Receipt

EFS ID:	19113086
Application Number:	13438754
International Application Number:	
Confirmation Number:	3525
Title of Invention:	DATA STORAGE AND ACCESS SYSTEMS
First Named Inventor/Applicant Name:	Patrick Sandor Racz
Customer Number:	42624
Filer:	Michael R. Casey
Filer Authorized By:	
Attorney Docket Number:	4037-0003
Receipt Date:	23-MAY-2014
Filing Date:	03-APR-2012
Time Stamp:	11:51:07
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	Issue Fee Payment (PTO-85B)	2014023_IssueFee.pdf	100516 04990c81ef6a75dfc068e479bcb8ded7739d6b93	no	1

Warnings:

Information:

Total Files Size (in bytes):	100516

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.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

42624 7590 04/11/2014
DAVIDSON BERQUIST JACKSON & GOWDEY LLP
4300 WILSON BLVD., 7TH FLOOR
ARLINGTON, VA 22203

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/438,754	04/03/2012	Patrick Sandor Racz	4037-0003	3525

TITLE OF INVENTION: DATA STORAGE AND ACCESS SYSTEMS

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	$0	$0	$1780	$0	07/11/2014

EXAMINER	ART UNIT	CLASS-SUBCLASS
LE, THIEN MINH	2887	235-380000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

□ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

□ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) The names of up to 3 registered patent attorneys or agents OR, alternatively,

(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 Davidson Berquist
2 Jackson & Gowdey, LLP
3 ______

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE
Smartflash, LLC

(B) RESIDENCE: (CITY and STATE OR COUNTRY)
Tyler, Texas

Please check the appropriate assignee category or categories (will not be printed on the patent) : □ Individual ☒ Corporation or other private group entity □ Government

4a. The following fee(s) are submitted:
□ Issue Fee
□ Publication Fee (No small entity discount permitted)
□ Advance Order - # of Copies ______

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
□ A check is enclosed.
□ Payment by credit card. Form PTO-2038 is attached.
☒ The Director is hereby authorized to charge the required fee(s), any deficiency, or credits any overpayment, to Deposit Account Number 501860 (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)

□ Applicant certifying micro entity status. See 37 CFR 1.29

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

□ Applicant asserting small entity status. See 37 CFR 1.27

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

□ Applicant changing to regular undiscounted fee status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature / Michael R. Casey /

Typed or printed name Michael R. Casey

Date May 23, 2014

Registration No. 40,294

Page 2 of 3

```
Adjustment date: 05/27/2014  ZJUHAR2
12/12/2013 SDIRETA2 00000002 501860   13438754
01 FC:1501          890.00 CR      -890.00 OP
```

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 3 of 28	Confirmation No.	3525

Change(s) applied to document, /T.W./ 7/7/2014

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.	Document No.	Publication/ Issue Date	Name of Patentee or Applicant of Cited Document
	3-1	US-5960082	1999/09/28	Haenel
	3-2	US-5963980	1999/10/05	Coulier et al.
	3-3	US-~~5972738~~	1999/08/24	~~Cesarie~~ et al. 5942738 Cesaire
	3-4	US-5995965	1999/11/30	Bettina Experton
	3-5	US-6003113	1999/12/14	Hoshino
	3-6	US-6005942	1999/12/21	Chan et al.
	3-7	US-6032857	2000/03/07	Kitagawa et al.
	3-8	US-~~6289711~~ B1	2004/12/07	Kwok et al 6829711
	3-9	US-6314409 B2	2001/11/06	Schneck, et al.
	3-10	US-6449684	2002/09/10	MacSmith et al.
	3-11	US-6532518	2003/03/11	MacSmith et al.
	3-12	US-6697944 B1	2004/02/24	Jones et al.
	3-13	US-6880761 B1	2005/04/19	Ritter et al.
	3-14	US-8033458	2011/10/11	Racz
	3-15	US-8061598	2011/11/22	Racz
	3-16	US-8118221	2012/02/21	Racz
	3-17	US-8336772	2012/12/25	Racz
	3-18			
	3-19			
	3-20			
	3-21			
	3-22			
	3-23			
	3-24			
	3-25			
	3-26			

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
FORM PTO-1449 (modified)

Sheet 2 of 28

Application No.	13/438,754
Filing Date	April 3, 2012
First Named Inventor	Patrick Sandor Racz
Group Art Unit	2887
Examiner Name	Le, Thien Minh
Attorney Docket No.	4037-0003
Confirmation No.	3525

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.	Document No.	Publication/ Issue Date	Name of Patentee or Applicant of Cited Document
	2-1	US-5401945	1995/03/28	Buschmann et al.
	2-2	US-5420912	1995/05/30	Kopp et al.
	2-3	US-5426432	1995/06/20	Sanemitsu
	2-4	US-5442704	1995/08/15	Holtey
	2-5	US-5511000	1996/04/23	Kaloi, et al.
	2-6	US-5511023	1996/04/23	Schrenk
	2-7	US-5523794	1996/06/04	Mankovitz et al.
	2-8	US-5557679	1996/09/17	Julin et al.
	2-9	US-5606143	1997/02/25	Young
	2-10	US-5610774	1997/03/11	Hayashi et al.
	2-11	US-5636276	1997/06/03	Brugger
	2-12	US-5664228	1997/09/02	Mital
	2-13	US-5686714	1997/11/11	Abe et al.
	2-14	US-5687398	1997/11/11	Martineau
	2-15	US-5737571	1998 ~~1997~~/04/07	Fukuzumi
	2-16	US-5763869	1998/06/09	Moll et al.
	2-17	US-5802325	1998/09/01	Le Roux
	2-18	US-5825875	1998/10/20	Ugon
	2-19	US-5825882	1998/10/20	Kowalski et al.
	2-20	US-5841979	1998/11/24	Schulhof, et al.
	2-21	US-5844281	1998/12/01	Kawan et al.
	2-22	US-5856699	1999/01/05	Drupsteen et al.
	2-23	US-5892975	1999/04/06	Barnes
	2-24	US-5896507	1999/04/20	Martineau
	2-25	US-5911031	1999/06/08	Young-Man Lee
	2-26	US-5943423	1999/08/24	Sead Muftic

Change(s) applied
to document,
/T.W./
7/7/2014

Examiner Signature	/Thien Le/	Date Considered	03/31/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT FORM PTO-1449 (modified)	Application No.	13/438,754
	Filing Date	April 3, 2012
	First Named Inventor	Patrick Sandor Racz
	Group Art Unit	2887
	Examiner Name	Le, Thien Minh
	Attorney Docket No.	4037-0003
Sheet 3 of 4	Confirmation No.	3525

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.	Document No.	Publication/ Issue Date	Name of Patentee or Applicant of Cited Document
	3-1	US-5636139	Jun-97	McLaughlin et al.
	3-2	US-5646992	Jul-97	Subler et al.
	3-3	US-5646998	Jul-97	Stambler
	3-4	US-5649187	Jul-97	Hornbuckle
	3-5	US-5666420	Sep-97	Micali
	3-6	US-5673316	Sep-97	Auerbach et al.
	3-7	US-5675734	Oct-97	Hair
	3-8	US-5706347	Jan-98	Burke et al.
	3-9	US-5710887	Jan-98	Chelliah et al.
	3-10	US-5745574	Apr-98	Muftic
	3-11	US-5765152	Jun-98	Erickson
	3-12	US-5796841	Aug-98	Cordery et al.
	3-13	US-5864620	Jan-99	Pettitt
	3-14	US-5892900	Apr-99	Ginter et al.
	3-15	US-5915025	~~Dec~~-99 Jun	Taguchi et al.
	3-16	US-5925127	Jul-99	Ahmad
	3-17	US-5982892	Nov-99	Hicks et al.
	3-18	US-5991399	Nov-99	Graunke et al.
	3-19	US-5999629	Dec-99	Heer et al.
	3-20	US-6064739	May-00	Davis
	3-21	US-6098056	Aug-00	Rusnak et al.
	3-22	US-6275936	Aug-01	Kyojima et al.
	3-23			
	3-24			
	3-25			
	3-26			

Change(s) applied
to document,
/T.W./
7/7/2014

Examiner Signature	/Thien Le/	Date Considered	04/08/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
FORM PTO-1449 (modified)

Sheet 3 of 4

Application No.	13/438,754
Filing Date	April 3, 2012
First Named Inventor	Patrick Sandor Racz
Group Art Unit	2887
Examiner Name	Le, Thien Minh
Attorney Docket No.	4037-0003
Confirmation No.	3525

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.	Document No.	Publication/ Issue Date	Name of Patentee or Applicant of Cited Document
	3-1	US-5636139	Jun-97	McLaughlin et al.
	3-2	US-5646992	Jul-97	Subler et al.
	3-3	US-5646998	Jul-97	Stambler
	3-4	US-5649187	Jul-97	Hornbuckle
	3-5	US-5666420	Sep-97	Micali
	3-6	US-5673316	Sep-97	Auerbach et al.
	3-7	US-5675734	Oct-97	Hair
	3-8	US-5706347	Jan-98	Burke et al.
	3-9	US-5710887	Jan-98	Chelliah et al.
	3-10	US-5745574	Apr-98	Muftic
	3-11	US-5765152	Jun-98	Erickson
	3-12	US-5796841	Aug-98	Cordery et al.
	3-13	US-5864620	Jan-99	Pettitt
	3-14	US-5892900	Apr-99	Ginter et al.
	3-15	US-5915025	~~Dec~~-99 Jun	Taguchi et al.
	3-16	US-5925127	Jul-99	Ahmad
	3-17	US-5982892	Nov-99	Hicks et al.
	3-18	US-5991399	Nov-99	Graunke et al.
	3-19	US-5999629	Dec-99	Heer et al.
	3-20	US-6064739	May-00	Davis
	3-21	US-6098056	Aug-00	Rusnak et al.
	3-22	US-6275936	Aug-01	Kyojima et al.
	3-23			
	3-24			
	3-25			
	3-26			

Change(s) applied
to document,
/T.W./
7/7/2014

Examiner Signature	/Thien Le/	Date Considered	01/14/2014

*Examiner: Initial if reference was considered, whether or not citation is in conformance with MPEP 609. Draw a line through citation if not in conformance and not considered. Include a copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /T.L./

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/438,754	08/05/2014	8794516	4037-0003	3525

42624 7590 07/16/2014

DAVIDSON BERQUIST JACKSON & GOWDEY LLP
4300 WILSON BLVD., 7TH FLOOR
ARLINGTON, VA 22203

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment is 0 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Data Management (ODM) at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site http://pair.uspto.gov for additional applicants):

Patrick Sandor Racz, Saint Heller, UNITED KINGDOM;
Hermen-ard Hulst, Amsterdam, NETHERLANDS;

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re PATENT Application of:		Confirmation No.:	3525
Patrick Sandor Racz		Attorney Docket:	4037-0003
Appl. S.N.:	13/438,754	Patent No.:	8,794,516
Filing Date:	April 3, 2012	Issue Date:	8/5//2014
Title:	DATA STORAGE AND ACCESS SYSTEMS	Date:	12/19/2014

REQUEST FOR CERTIFICATE OF CORRECTION

Certificate of Corrections Branch
Hon. Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Commissioner:

Pursuant to 37 C.F.R. § 1.323, the Assignee of U.S. Patent No. 8,794,516 hereby requests that the patent be corrected as shown on the attached. The mistake for which correction is sought is of a clerical nature (as the patent number and issue date are correct) and the correction does not involve changes which would (1) constitute new matter or (2) require reexamination. The fee may be charged as shown below.

CHARGE STATEMENT: Deposit Account No. 501860, order no. **4037-0003**.

The Commissioner is hereby authorized to charge any fee specifically authorized hereafter, or any missing or insufficient fee(s) filed, or asserted to be filed, or which should have been filed herewith or concerning any paper filed hereafter, and which may be required under Rules 16-18 (missing or insufficiencies only) now or hereafter relative to this application and the resulting Official Document under Rule 20, or credit any overpayment, to our Accounting/Order Nos. shown above.

CUSTOMER NUMBER

42624

Davidson Berquist Jackson & Gowdey LLP
4300 Wilson Blvd., 7th Floor,
Arlington Virginia 22203
Main: (703) 894-6400 • FAX: (703) 894-6430

Respectfully submitted,

By: / Michael R. Casey /

Michael R. Casey
Registration No.: 40,294

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

Page 1 of 1

PATENT NO. : 8,794,516
APPLICATION NO.: 13/438,754
ISSUE DATE : 08/05/2014
INVENTOR(S) : Patrick Sandor Racz and Hermen-ard Hulst

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On page 3, in section (56), left-hand column, line 33,

change "6389538 May 2002 Downs et al." to

-- 6389538 May 2002 Gruse et al. --.

MAILING ADDRESS OF SENDER (Please do not use customer number below):

Michael R. Casey / Davidson Berqist Jackson & Gowdey, LLP
4300 Wilson Blvd., Suite 700, Arlington, VA 22203

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 disclosure of these records is required by the Freedom of Information Act.
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 inspection 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.

Electronic Patent Application Fee Transmittal

Application Number:	13438754
Filing Date:	03-Apr-2012
Title of Invention:	DATA STORAGE AND ACCESS SYSTEMS
First Named Inventor/Applicant Name:	Patrick Sandor Racz
Filer:	Michael R. Casey
Attorney Docket Number:	4037-0003
Filed as Large Entity	
Filing Fees for **Utility under 35 USC 111(a)**	

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Certificate of Correction	1811	1	100	100

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension-of-Time:				
Miscellaneous:				
	Total in USD ($)			**100**

Electronic Acknowledgement Receipt

EFS ID:	21020317
Application Number:	13438754
International Application Number:	
Confirmation Number:	3525
Title of Invention:	DATA STORAGE AND ACCESS SYSTEMS
First Named Inventor/Applicant Name:	Patrick Sandor Racz
Customer Number:	42624
Filer:	Michael R. Casey
Filer Authorized By:	
Attorney Docket Number:	4037-0003
Receipt Date:	19-DEC-2014
Filing Date:	03-APR-2012
Time Stamp:	15:20:51
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$100
RAM confirmation Number	2032
Deposit Account	501860
Authorized User	

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 1.20 (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	Request for Certificate of Correction	201412219_COCRequest.pdf	123956 f48f39db00d0c75ebce6052eecc815706677f837	no	1
Warnings:					
Information:					
2	Request for Certificate of Correction	20141219_RequestforCOC.pdf	164536 f5ec0a80cacb64626f6d89022dd2a963c2184647	no	2
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	29970 b193d040be75b05751b4d4a2beb8af144afe5d57	no	2
Warnings:					
Information:					
			Total Files Size (in bytes):	318462	

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.

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO.	: 8,794,516 B2	Page 1 of 1
APPLICATION NO.	: 13/438754	
DATED	: August 5, 2014	
INVENTOR(S)	: Patrick Sandor Racz and Hermen-ard Hulst	

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On the Title Page

On page 3, item (56), left-hand column, line 33,

change "6389538 May 2002 Downs et al." to

-- 6389538 May 2002 Gruse et al. --.

Signed and Sealed this
Seventeenth Day of March, 2015

Michelle K. Lee

Michelle K. Lee
Director of the United States Patent and Trademark Office

AO 120 (Rev. 08/10)

TO: Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas Tyler Division on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO. 6:15-cv-145	DATE FILED 2/25/2015	U.S. DISTRICT COURT Eastern District of Texas Tyler Division

PLAINTIFF Smartflash LLC and Smartflash Technologies Limited	DEFENDANT Apple Inc.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 7,334,720	2/26/2008	Smartflash LLC
2 7,334,720	5/17/2011	Smartflash LLC
3 8,033,458	10/11/2011	Smartflash LLC
4 8,061,598	11/22/2011	Smartflash LLC
5 8,118,221	2/21/2012	Smartflash LLC

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED 2/25/2015	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 8,336,772	12/25/2012	Smartflash LLC
2 8,794,516	8/5/2014	Smartflash LLC
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy